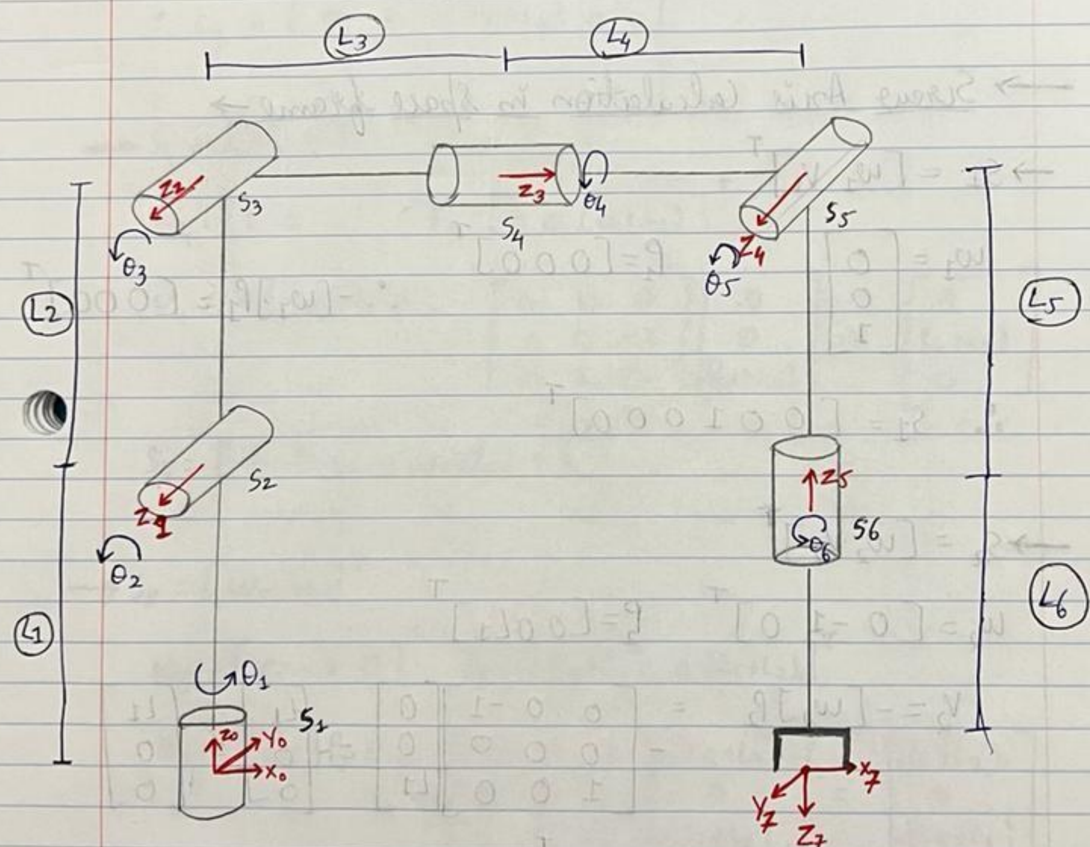
1. Screw Axis Calculation –

RBE 501 - Robot Dynamics - Final Exam

1 Second digit of WPI-ID → $\boxed{4}$



$$\rightarrow M = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$

$$\therefore R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad P = \begin{bmatrix} L_3+L_4 \\ 0 \\ (L_1+L_2)-(L_5+L_6) \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & L_3+L_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & (L_1+L_2)-(L_5+L_6) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\rightarrow$ Screw Axis Calculation in Space frame $\rightarrow$

$\rightarrow S_1 = [w_1 \ V_1]^T$

$$w_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad P_1 = [0\ 0\ 0]^T \qquad \therefore -[w_1]P_1 = [0\ 0\ 0]^T$$

$\therefore \ S_1 = [0\ 0\ 1\ 0\ 0\ 0]^T$

$\rightarrow S_2 = [w_2 \ V_2]^T$

$w_2 = [0\ -1\ 0]^T \qquad P_2 = [0\ 0\ L_1]^T$

$$V_2 = -[w_2]P_2 = -\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix} = -\begin{bmatrix} -L_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix}$$

$\therefore S_2 = [0\ -1\ 0\ L_1\ 0\ 0]^T$

$\rightarrow S_3 = [w_3 \ V_3]^T$

$w_3 = [0\ -1\ 0]^T \qquad P_3 = [0\ 0\ (L_1+L_2)]^T$

Fenil Desai (WPI ID: 249812226)
RBE 501 - Robot Dynamics (Fall-2021)

$$V_3 = -[\omega_3]P_3 = -\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ L_1+L_2 \end{bmatrix} = \begin{bmatrix} (L_1+L_2) \\ 0 \\ 0 \end{bmatrix}$$

$$\therefore S_3 = [0 \ -1 \ 0 \ (L_1+L_2) \ 0 \ 0]^T$$

$\rightarrow S_4 = [\omega_4 \ V_4]^T$

$\omega_4 = [1 \ 0 \ 0]^T \quad P_4 = [0 \ 0 \ (L_1+L_2)]^T$

$$V_4 = -[\omega_4]P_4 = -\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ (L_1+L_2) \end{bmatrix} = \begin{bmatrix} 0 \\ (L_1+L_2) \\ 0 \end{bmatrix}$$

$$S_4 = [1 \ 0 \ 0 \ 0 \ (L_1+L_2) \ 0]^T$$

$\rightarrow S_5 = [\omega_5 \ V_5]^T$

$\omega_5 = [0 \ -1 \ 0]^T \quad P_5 = [(L_3+L_4) \ 0 \ (L_1+L_2)]^T$

$$V_5 = -[\omega_5]P_5 = -\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} (L_3+L_4) \\ 0 \\ (L_1+L_2) \end{bmatrix} = \begin{bmatrix} (L_1+L_2) \\ 0 \\ -(L_3+L_4) \end{bmatrix}$$

$$S_5 = [0 \ -1 \ 0 \ (L_1+L_2) \ 0 \ -(L_3+L_4)]^T$$

$$\rightarrow S_6 = [\omega_6 \; v_6]^T$$

$$\omega_6 = [0 \; 0 \; 1]^T \quad p_6 = [(L_3+L_4) \; 0 \; 0]^T$$

$$v_6 = -[\omega_6] p_6 = -\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} L_3+L_4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -(L_3+L_4) \\ 0 \end{bmatrix}$$

$$S_6 = [0 \; 0 \; 1 \; 0 \; -(L_3+L_4) \; 0]^T$$

$$S = \begin{array}{cccccc} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array}$$
$$S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & L_1 & (L_1+L_2) & 0 & (L_1+L_2) & 0 \\ 0 & 0 & 0 & (L_1+L_2) & 0 & -(L_3+L_4) \\ 0 & 0 & 0 & 0 & -(L_3+L_4) & 0 \end{bmatrix}$$

2. **Manipulator Forward Kinematics –**

I am implemented Forward Kinematics of the given robot using Product of Exponentials Methods and also made variables S and M as per the given directions.

I have also implemented 20 random configurations to check the correctness of my code.

3. Inverse Kinematics –

First, I implemented NEWTON-RAPHSON numerical method to find the inverse kinematic solution for the given robotic manipulator.

It iteratively calculates detlaQ using the following equation -

deltaQ = pinv(J_a)*(target_twist – current_twist) here,

- deltaQ is a vector whose elements represent the change in joint values from the previous joint values.
- pinv(J_a) is Moore-Penrose pseudo-inverse of the analytical jacobian matrix. Pseudoinverse is used to check abruptly high joint values.
- target_twist is twist of the final desire pose of the end effector.
- Current_twist is the twist of the current pose (at current iteration) of the end effector.

This method manages to converge to the solution and finds all the joint variables for all 100 points given in the question.

However, as a better alternative method I implemented DAMPED-LEAST SQUARE method with very small damping coefficient (lambda = 0.05) as it does not diverge as much as the NewtonRaphson method. The damped least square method is implemented as follows - J* = transpose(J_a)*pinv(J_a*transpose(J_a) + (lamda)^2*I) deltaQ = (J*)*(target_twist – current_twist)

here,

- deltaQ is a vector whose elements represent the change in joint values from the previous joint values.
- pinv(k) is Moore-Penrose pseudo-inverse of the matrix k.
- target_twist is twist of the final desire pose of the end effector.
- Current_twist is the twist of the current pose (at current iteration) of the end effector.
- lambda is the damping factor.
- I is 3*3 Identity matrix
-

**You can toggle the plotOn variable to false to not see the robot moving while it is searching for the Inverse Kinematics solution.

**The output of the code is the simulation of the robot traversing the curvilinear path as per directions.

4. Dynamic Modelling –

The Mlist matrix and Glist matrix is filled as per directions in the given starter code.

Fenil Desai (WPI ID: 249812226)
RBE 501 - Robot Dynamics (Fall-2021)

$\boxed{4}$ → By inspection we can do this step.

→ $M_{01} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$ $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $P = \begin{bmatrix} 0 \\ 0 \\ L1/2 \end{bmatrix}$

$$M_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ $M_{12} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$ $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $P = \begin{bmatrix} 0 \\ 0 \\ (L_1+L_2)2 \end{bmatrix}$

$$M_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{(L_1+L_2)}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ $M_{23} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$ $R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$ $P = \begin{bmatrix} L_3/2 \\ 0 \\ L2/2 \end{bmatrix}$

$$M_{23} = \begin{bmatrix} 0 & 0 & 1 & L3/2 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & L2/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow M_{34} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 \\ 0 \\ (L_3+L_4)_2 \end{bmatrix}$$

$$M_{34} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & (L_3+L_4)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow M_{45} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad P = \begin{bmatrix} L_5/2 \\ 0 \\ L_4/2 \end{bmatrix}$$

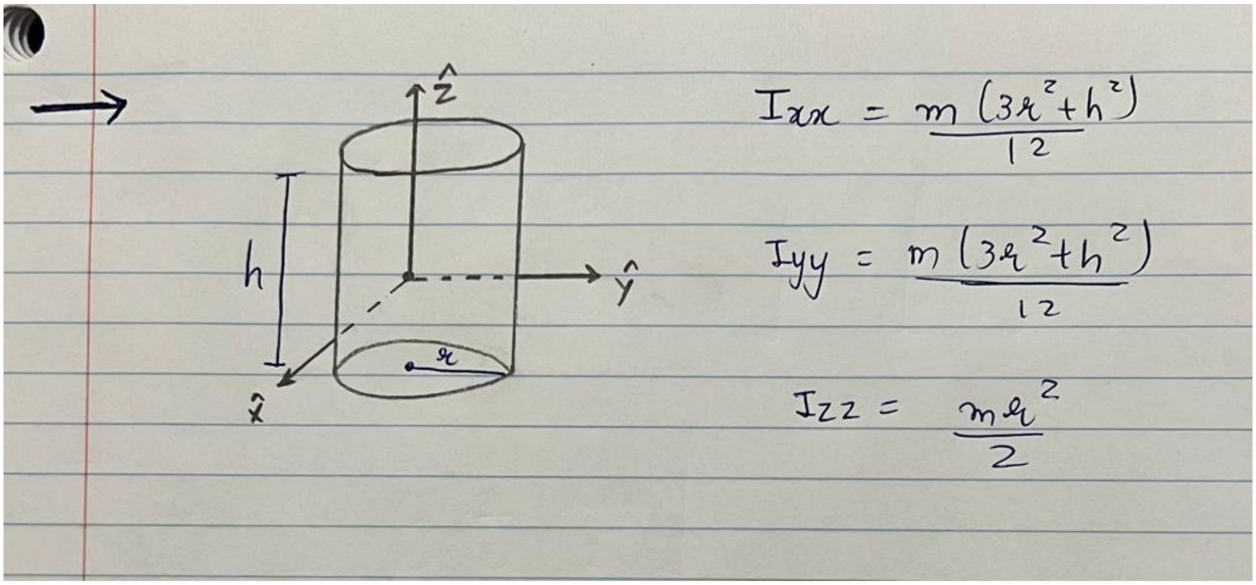
$$M_{45} = \begin{bmatrix} 0 & 0 & 1 & L_5/2 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & L_4/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow M_{56} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 \\ 0 \\ (L_5+L_6)/2 \end{bmatrix}$$

$$M_{56} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & (L_5+L_6)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow M_{67} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 \\ 0 \\ L_6/2 \end{bmatrix}$$

$$M_{67} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_6/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



5. Gravity Compensation –

The output of this script is –

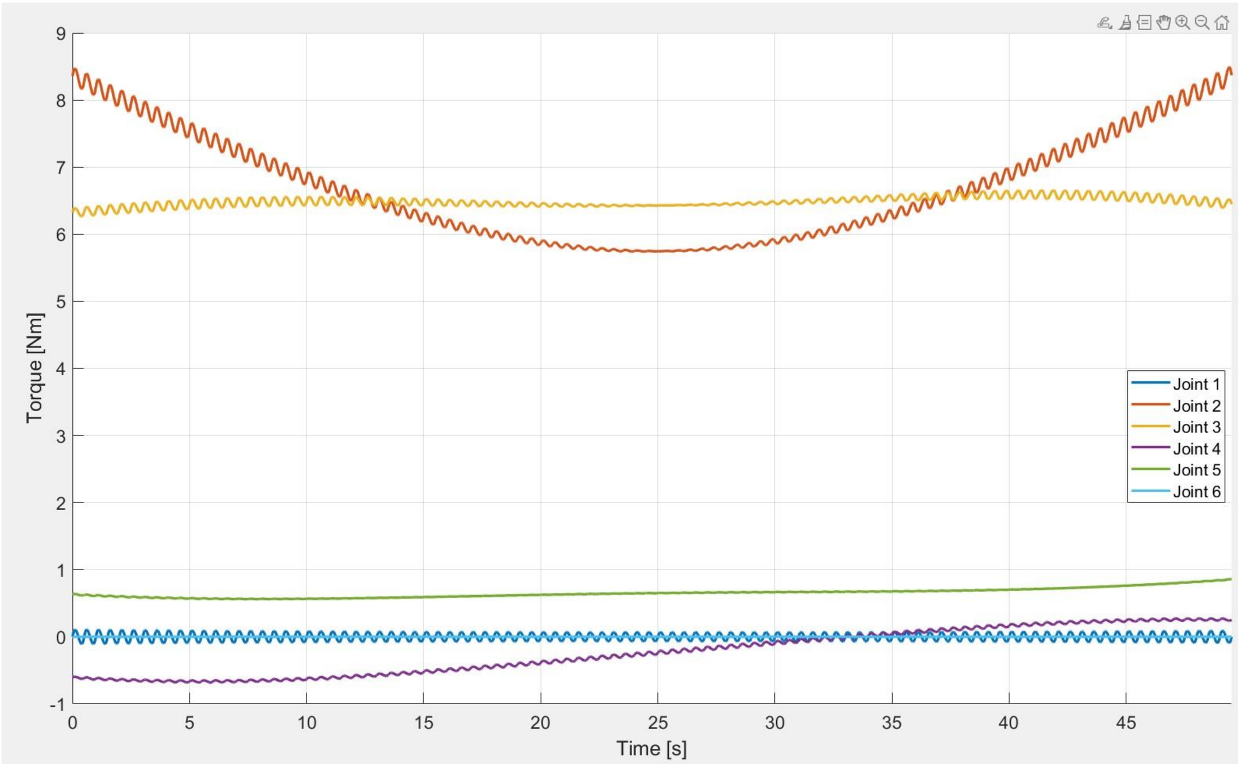
Joint Torques: [0.000000 5.886000 5.886000 0.000000 0.000000 0.000000] Nm

This implies that the **joint-2** and **joint-3** both will require 5.886000 Nm torque to save robot from falling under its own weight under the influence of gravity.

6. Inverse Dynamics –

The output of this script gives the simulation of the robot after calculating the joint torques using inverse dynamics.

The following figure represents the joint torque profile obtained for the given trajectory.


Torque [Nm]
Time [s]
Joint 1
Joint 2
Joint 3
Joint 4
Joint 5
Joint 6
9
8
7
6
5
4
3
2
1
0
-1
0
5
10
15
20
25
30
35
40
45