Tasks:

1. Show that columns of the rotation matrix R_0^1 are orthogonal.

Assignment 2

Task 1 :-

To prove colums of rotation matrix are ~~otg~~ orthogonal, we start with a simple basic rotation matrix $R_{x,\theta}$.

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Here, column vectors are

$$r_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} 0 \\ \cos\theta \\ \sin\theta \end{bmatrix}, \quad r_3 = \begin{bmatrix} 0 \\ -\sin\theta \\ \cos\theta \end{bmatrix}$$

$$r_1 \cdot r_2 = (1\times 0) + (0\times\cos\theta) + (0\times\sin\theta)$$
$$= 0$$

$$r_2 \cdot r_3 = (0\times 0) + (\cos\theta\times -\sin\theta) + (\sin\theta\times\cos\theta)$$
$$= 0$$

$$r_1 \cdot r_3 = (1\times 0) + (0\times -\sin\theta) + (0\times\cos\theta)$$
$$= 0$$

Also, we know product of two orthogonal matrices is orthogonal.
We can prove similarly for $R_{y,\theta}$ & $R_{z,\theta}$.
Since all rotation matrices can be written as a product of $R_{x,\theta}$, $R_{y,\theta}$ & $R_{z,\theta}$, we can say that ~~th~~ columns of rotation matrix are orthogonal.

2. Show that $\det(R_0^1) = 1$.

Task 2

Any rotation matrix R_0^1 can be written as

$$R_0^1 = R_{x,\theta} \, R_{y,\phi} \, R_{z,\psi}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$|R_0^1| = (\cos^2\theta + \sin^2\theta)(\cos^2\phi + \sin^2\phi)(\cos^2\psi + \sin^2\psi)$$

$$\boxed{|R_0^1| = 1}$$

3. Read about the order of rotations and sample examples in the textbook.

4. Review the textbook explanation and example related to a rotation matrix for rotation about an arbitrary axis k.

5. Show that $RS(a)R^T = S(Ra)$, where is a rotation matrix.

Task 5.

We have

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

$$RS(a)R^T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} C_\theta & S_\theta & 0 \\ -S_\theta & C_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -a_z S_\theta & -a_z C_\theta & a_y C_\theta + a_x S_\theta \\ a_z C_\theta & -a_z S_\theta & -a_y S_\theta - a_x C_\theta \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} C_\theta & S_\theta & 0 \\ -S_\theta & C_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -a_z S_\theta C_\theta + a_z S_\theta C_\theta & -a_z \sin^2\theta - a_z C_\theta^2 & a_y C_\theta + a_x S_\theta \\ a_z C_\theta^2 + a_z S_\theta^2 & a_z C_\theta S_\theta - a_z C_\theta S_\theta & -a_y S_\theta - a_x C_\theta \\ -a_y C_\theta - a_x S_\theta & a_y S_\theta + a_x C_\theta & 0 \end{bmatrix}$$

$$RS(a)R^T = \begin{bmatrix} 0 & -a_z & a_y C_\theta + a_x S_\theta \\ a_z & 0 & -a_y S_\theta - a_x C_\theta \\ -a_y C_\theta - a_x S_\theta & -a_y S_\theta + a_x C_\theta & 0 \end{bmatrix}$$

Now,

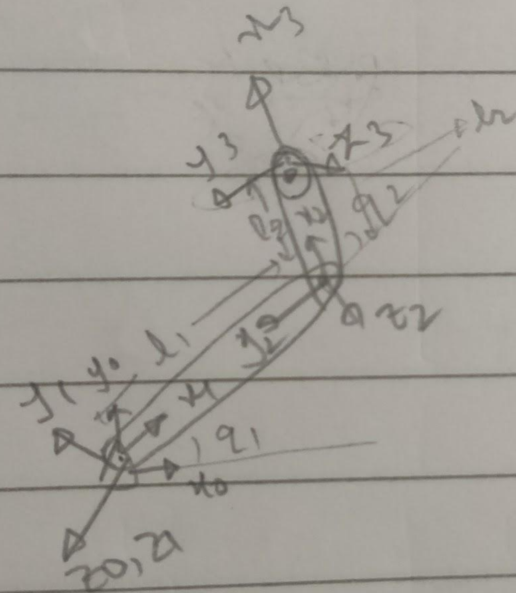
$$Ra = \begin{bmatrix} C_\theta & -S_\theta & 0 \\ S_\theta & C_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} a_x C_\theta - a_y S_\theta \\ a_x S_\theta + a_y C_\theta \\ a_z \end{bmatrix}$$

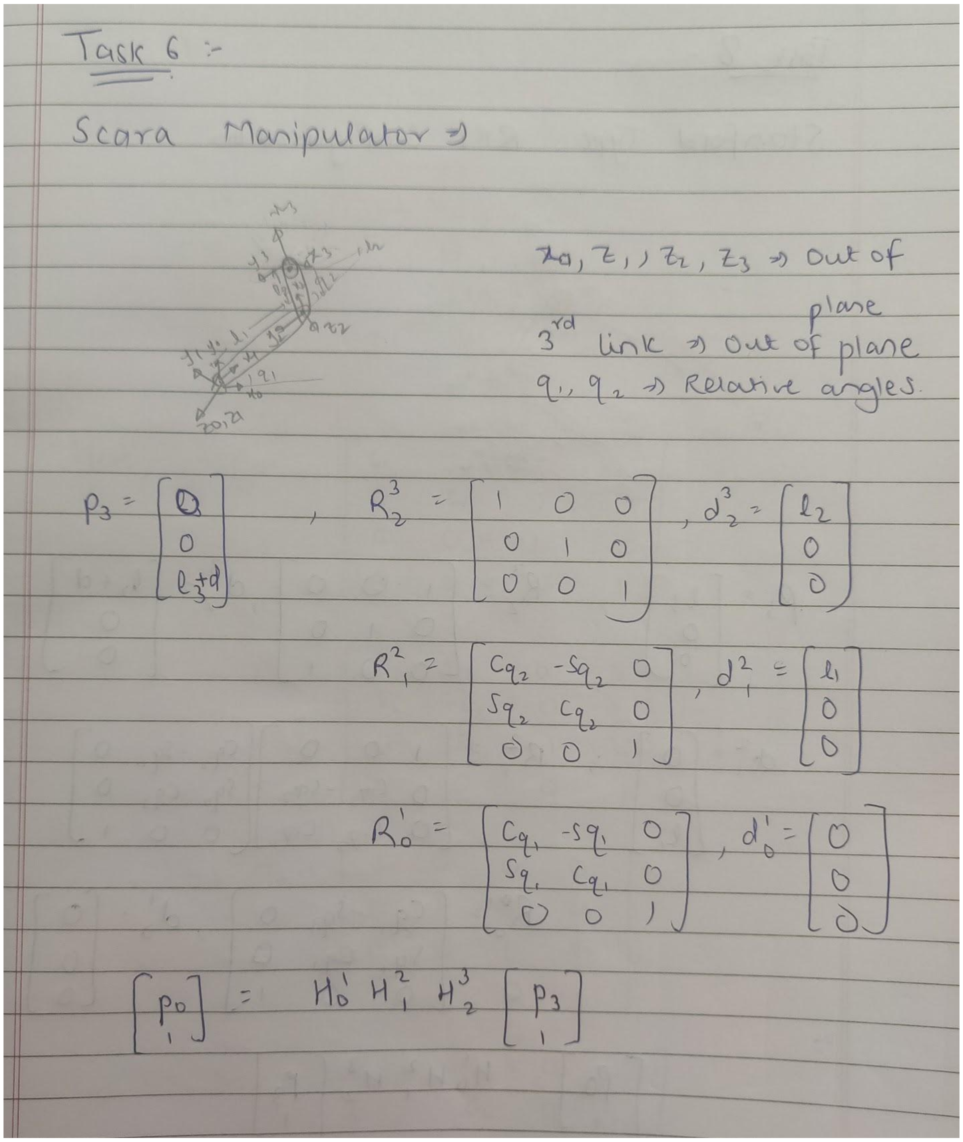
$$S(Ra) = \begin{bmatrix} 0 & -a_z & a_y C_\theta + a_x S_\theta \\ a_z & 0 & a_y S_\theta - a_x C_\theta \\ -a_y C_\theta - a_x S_\theta & -a_y S_\theta + a_x C_\theta & 0 \end{bmatrix}$$

$$\boxed{S(Ra) = RS(a)R^T}$$

6. Work out the various coordinate frames (show them on a clearly marked figure) and work out p using a composition of homogeneous transformations for the RRP SCARA configuration.

Scara Manipulator ⇒




Task 6 :-

Scara Manipulator ⇒

z_0, z_1, z_2, z_3 ⇒ Out of plane

3rd link ⇒ Out of plane

q_1, q_2 ⇒ Relative angles.

$$p_3 = \begin{bmatrix} 0 \\ 0 \\ l_3+d \end{bmatrix}, \quad R_2^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d_2^3 = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix}$$

$$R_1^2 = \begin{bmatrix} Cq_2 & -Sq_2 & 0 \\ Sq_2 & Cq_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d_1^2 = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}$$

$$R_0^1 = \begin{bmatrix} Cq_1 & -Sq_1 & 0 \\ Sq_1 & Cq_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d_0^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p_0 \\ 1 \end{bmatrix} = H_0^1 H_1^2 H_2^3 \begin{bmatrix} p_3 \\ 1 \end{bmatrix}$$

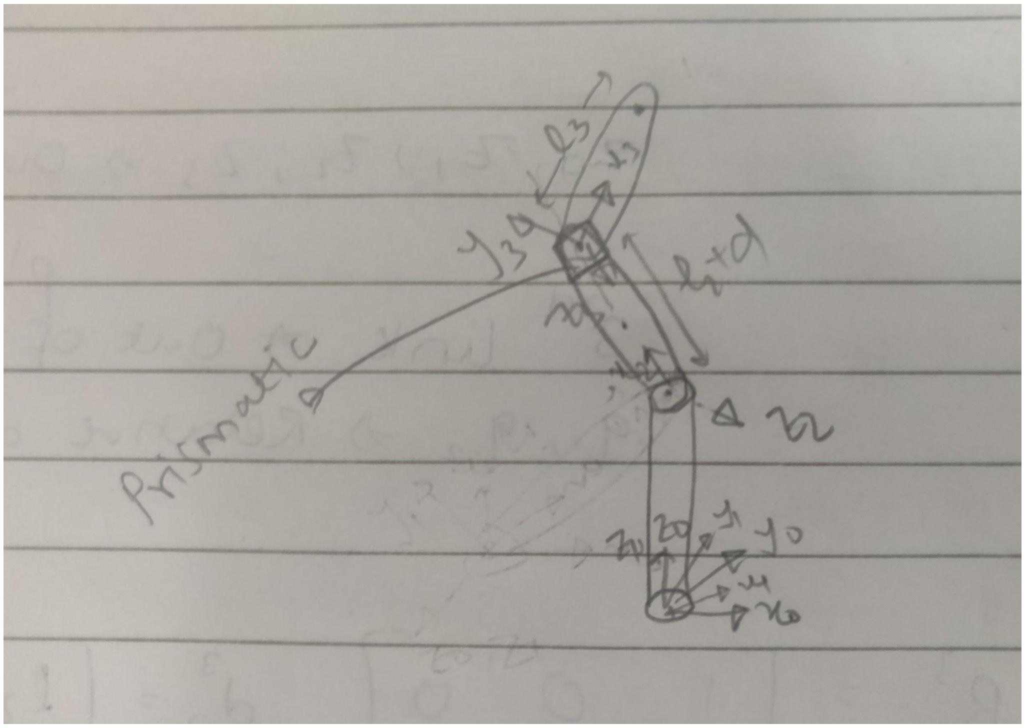
7. Write a python code incorporating the above calculation that can return the position vector of the end effector for any given configuration of joint variables (angles and extension).

Ans:

https://colab.research.google.com/drive/164J_rvRHLBn25TlBmhFr3_mnCPCa6sGP?usp=sharing

8. Repeat the above exercise for the Stanford-type RRP configuration, again write a python code that can return the position vector of the end effector for any given configuration of joint variables (angles and extension).

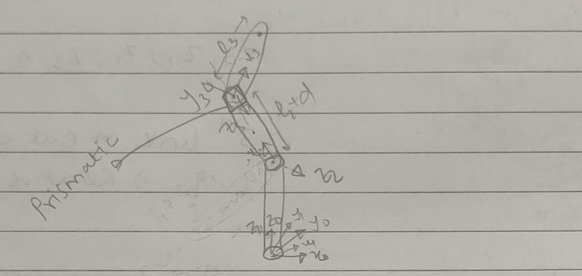
Ans:



l_3
x_3
y_3
l_2+d
Prismatic
x_2
z_1
z_0
y_1
y_0
x_1
x_0

Task 8 :-

Stanford Type RRP Configuration



$$p_3 = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}, \quad R^3_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d^3_2 = \begin{bmatrix} l_2 + d \\ 0 \\ 0 \end{bmatrix}$$

$$d^2_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix}, \quad R^2_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\pi/_2 & -S\pi/_2 \\ 0 & S\pi/_2 & C\pi/_2 \end{bmatrix} \begin{bmatrix} Cq_2 & -Sq_2 & 0 \\ Sq_2 & Cq_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R^1_0 = \begin{bmatrix} Cq_1 & -Sq_1 & 0 \\ Sq_1 & Cq_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d^1_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

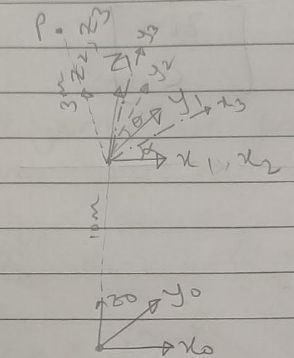
$$\begin{bmatrix} p_0 \\ 1 \end{bmatrix} = H^1_0 H^2_1 H^3_2 \begin{bmatrix} p_3 \\ 1 \end{bmatrix}$$

https://colab.research.google.com/drive/1G0-y__TD6JIhSwyFHlGV8btHq84HbX_y?usp=sharing

9. A drone took off from a base station and travelled 10m straight up. If you consider an inertial frame attached at the base station with the z-axis pointing straight up and x and y axes along the ground forming a right-hand system, then this would be 10m in the z-direction. At this hover point, the drone orientation is as if it completed a 30-degree rotation about
the x-axis followed by a 60-degree rotation about the resulting new (current) z-axis. Further, it is then observed using a lidar installed on the drone that an obstacle is 3m exactly above the drone (in the drone frame). Find the position vector of the obstacle with respect to the base coordinate frame using a composition of homogeneous transformations. Also, show the choice of

coordinate frames using a neat sketch.

Task 9 :-



Here, P is the position of the obstacle

$\alpha = 60^\circ$, $\theta = 30^\circ$

Here, we have $P_3 = \begin{bmatrix} 0 \\ 0 \\ 3 \end{bmatrix}$

Now, $R_2^3 = \begin{bmatrix} \cos 60 & -\sin 60 & 0 \\ \sin 60 & \cos 60 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $d_2^3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$R_1^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 30 & -\sin 30 \\ 0 & \sin 30 & \cos 30 \end{bmatrix}$, $d_1^2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$$R_0^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad d_0^1 = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}$$

Now, $$H_2^3 = \begin{bmatrix} \cancel{\cos}\ 1/2 & -\sqrt{3}/2 & 0 & 0 \\ \sqrt{3}/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \sqrt{3}/2 & -1/2 & 0 \\ 0 & 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_0^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} p_0 \\ 1 \end{bmatrix} = H_0^1 H_1^2 H_2^3 \begin{bmatrix} p_3 \\ 1 \end{bmatrix} = H_0^1 H_1^2 H_2^3 [0 \ 0 \ 3 \ 1]^T$$

$$= H_0^1 H_1^2 \begin{bmatrix} 0 \\ 0 \\ 3 \\ 1 \end{bmatrix} = H_0^1 \begin{bmatrix} 0 \\ -1.5 \\ 2.595 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.5 \\ 12.595 \\ 1 \end{bmatrix}$$

So, $p_0 = [0 \ \ -1.5 \ \ 12.595]^T$

10. Read about a few different types of gearboxes typically used with motors in a robotic application and explain in 2-3 sentences in your own words some key pros and cons of each gearbox type and where it is typically used. Further, explain if you would typically see a gearbox used along with a motor in a drone application. Explain the reasons.

Ans: The different types of gearboxes used in industrial and robotic applications are as follows:

- Helical Gearbox: It is a gearbox wherein it is fixed at an angle that enables more teeth to interact in the same direction when in motion. This ensures constant contact for a certain period. It has applications in heavy-duty operations and low-power applications.

- Coaxial helical inline gearbox: Here, the drive and the output shaft are on the same rotation axis. These types of gearboxes have extremely high efficiency and high power density.

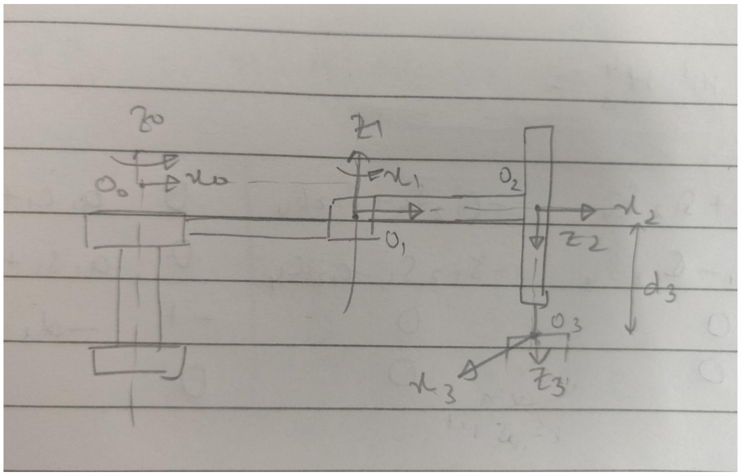
- Bevel helical gearbox: It has a curved set of teeth on a cone-shaped surface that are used to provide rotary motion in non-parallel shafts.

- Skew bevel helical gearbox: These are customizable gearboxes with a rigid and monolithic structure. These can be customized on the basis of number of teeth and gears and so can be used in a variety of applications to increase the mechanical advantage of a transmission.

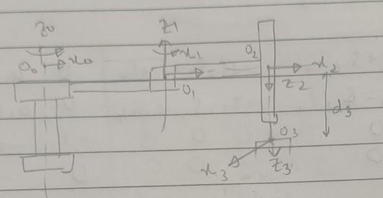
- Worm reduction gearboxes: These consist of a worm wheel with a large diameter along with a worm that meshes with the teeth on the periphery of the gearbox. This gives the structure a screw-like movement which has applications in speed reduction in non-intersecting crossed axis shafts.

- Planetary gearbox: It has a central Sun gear with 3-4 planetary gears that are held together through an outer ting internal gear. This gear system can achieve a high torque within a small space and has many applications in robotics.

11. Derive the Manipulator Jacobian for the RRP SCARA configuration.



Task 11



Here, $H_0^1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$H_1^2 = \begin{bmatrix} C_2 & S_2 & 0 & a_2C_2 \\ S_2 & -C_2 & 0 & a_2S_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^4 = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, we have

$$T_0^3 = H_0^1 H_1^2 H_2^3$$

$$= \begin{bmatrix} c_{12} & s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & -c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, joints 1&2 ⇒ revolute, 3 ⇒ prismatic

$$J = \begin{bmatrix} z_0 \times (O_3 - O_0) & z_1 \times (O_3 - O_1) & z_2 \\ z_0 & z_1 & 0 \end{bmatrix}$$

$$O_1 = \begin{bmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{bmatrix}, \quad O_2 = \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ 0 \end{bmatrix}, \quad O_3 = \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ -d_3 \end{bmatrix}$$

$$z_0 = z_1 = k \quad \& \quad z_2 = -k$$

$$J = \begin{bmatrix} -a_1s_1 - a_2s_{12} & -a_2s_{12} & 0 \\ a_1c_1 + a_2c_{12} & a_2c_{12} & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

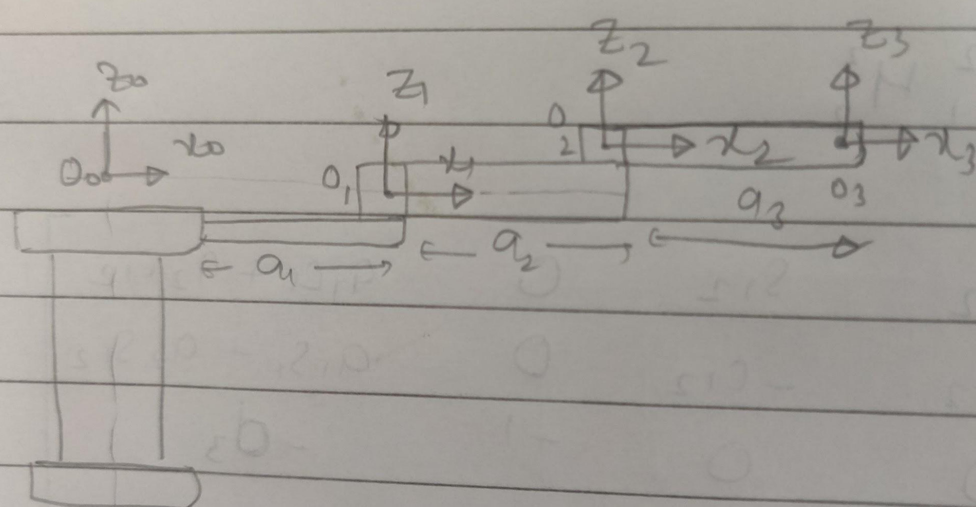
12. Write a python code implementing the above Jacobian such that the entire Jacobian matrix is output for any given values of joint variables.

Ans:

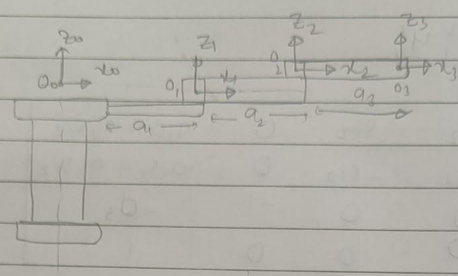
https://colab.research.google.com/drive/1LTGfst7aSIgtomXWVu9t13COvxQoN7ya?usp=sharing

13. Derive the Manipulator Jacobian for the RRR configuration with all rotation axis parallel to each other (the entire robot is planar like the elbow manipulator).

Task 13



Task 13



For the above RRR configuration, all rotation axes are parallel, i.e., z_0, z_1 & z_2

Here, $H_0^1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$H_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$H_2^3 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

So, we have

$$T_0^3 = H_0^1 H_1^2 H_2^3$$

$$T_0^3 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_3c_{123}+a_2c_{12}+a_1c_1 \\ s_{123} & c_{123} & 0 & a_3s_{123}+a_2s_{12}+a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

All joints are revolute, so,

$$J = \begin{bmatrix} z_0 \times (O_3 - O_0) & z_1 \times (O_3 - O_1) & z_2 \times (O_3 - O_2) \\ z_0 & z_1 & z_2 \end{bmatrix}$$

$$O_1 = \begin{bmatrix} a_1c_1 \\ a_1s_1 \\ 0 \end{bmatrix}, \quad O_2 = \begin{bmatrix} a_1c_1 + a_2c_{12} \\ a_1s_1 + a_2s_{12} \\ 0 \end{bmatrix}, \quad O_3 = \begin{bmatrix} a_3c_{123} + a_2c_{12} + a_1c_1 \\ a_3s_{123} + a_2s_{12} + a_1s_1 \\ 0 \end{bmatrix}$$

Here, $z_0 = z_1 = z_2 = k$

$$J = \begin{bmatrix} -(a_3s_{123}+a_2s_{12}+a_1s_1) & -(a_3s_{123}+a_2s_{12}) & -a_3s_{123} \\ a_3c_{123}+a_2c_{12}+a_1c_1 & a_3c_{123}+a_2c_{12} & a_3c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

14. Write a python code implementing the above Jacobian such that the entire Jacobian matrix is output for any given values of joint variables.

Ans:

https://colab.research.google.com/drive/1kTNhVptt5LEoaR7noC7VlTr4XgXKxG1I?usp=sharing