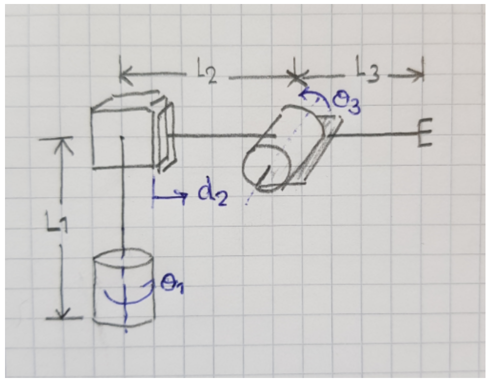
PART II.1.a World Building

The three-link manipulator of Origin of coordinate frame B relative World coordinate frame W is located at position $O_B = (2.5m, 2.5m, 0m)$.

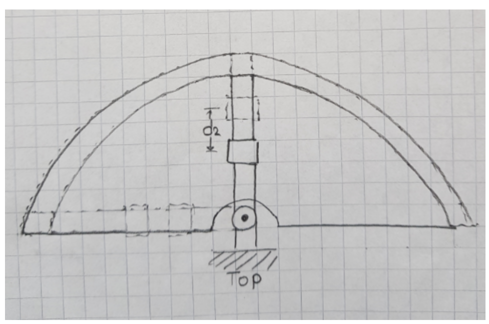
The choice of three-link manipulator is shown below:



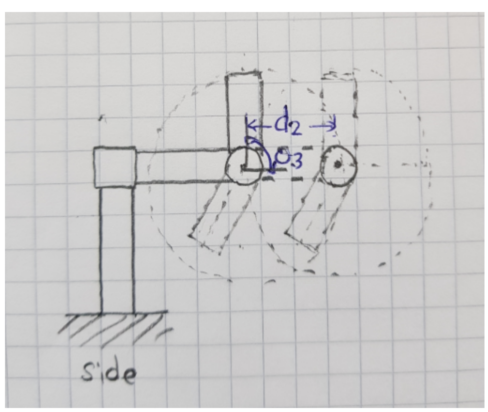
Joint1 is given as revolute, joint2 as prismative and joint3 as revolute respectively. The link-length are: $l_1 = 1.0m$ $l_2 = 0.3m$ $l_3 = 0.2m$

PART II.1.b World Building

A sketch about the workplace of a robot. A 2D drawing from top view is shown below:

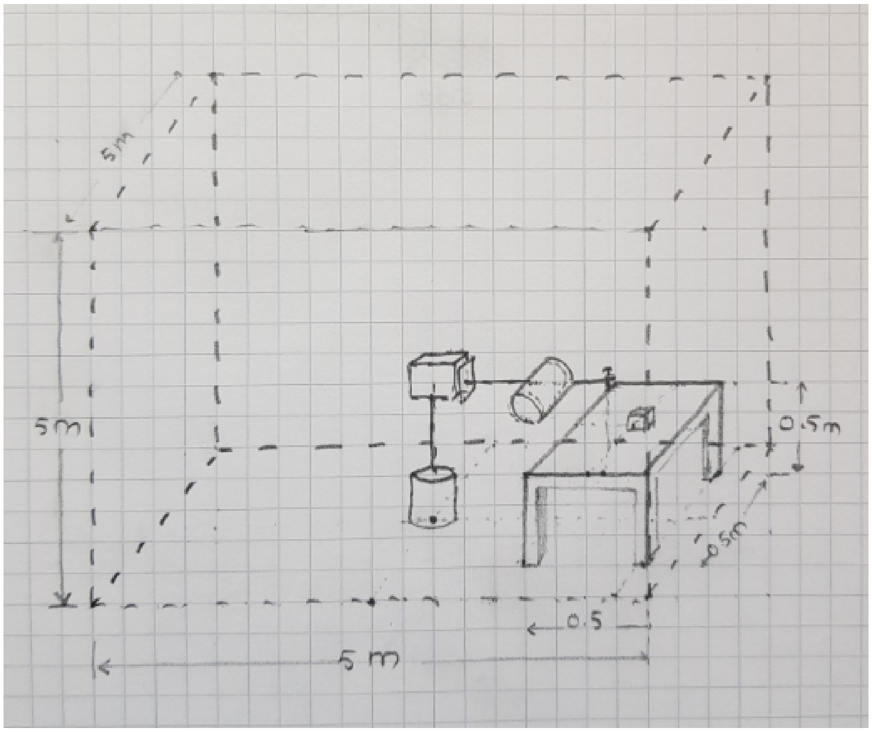


A 2D drawing from side view is shown below:

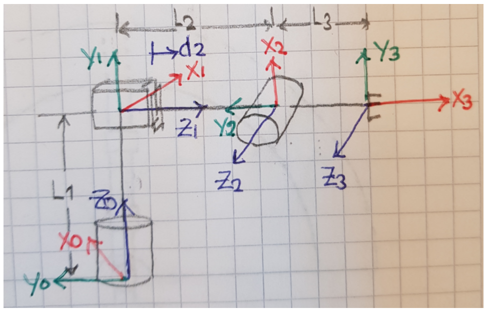


PART II.1.c World Building

The manipulator is located 0.5m from a table. A table is 0.5m high and 0.5m square. A cube with 10cm on a sides is placed at the center top of the table.



PART II.2.a Kinematics and Transformation



A DH table is showing

No	Rz	Tz	Tx	Rx
1	theta1	L1	0	90
2	90	L2 + d2	0	-90
3	theta3	0	L3	0

Table 1: DH table

PART II.2.b Kinematics and Transformation

Calculating the forward kinematics for the manipulator.

$$H_3^0 = H_1^0 H_2^1 H_3^2 \tag{1}$$

$$H_1^0 = R_z T_z T_x R_x \tag{2}$$

$$H_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = R_z T_z T_x R_x \tag{3}$$

$$H_2^1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & l_2 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & l_2 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = R_z T_z T_x R_x \tag{4}$$

$$H_3^2 = \begin{bmatrix} s_3 & c_3 & 0 & 0 \\ -c_3 & s_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_3 & c_3 & 0 & s_3 l_3 \\ -c_3 & s_3 & 0 & -c_3 l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From equation (1) we get

$$H_3^0 = H_1^0 H_2^1 H_3^2$$

$$H_3^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & l_2 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_3 & c_3 & 0 & s_3 l_3 \\ -c_3 & s_3 & 0 & -c_3 l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

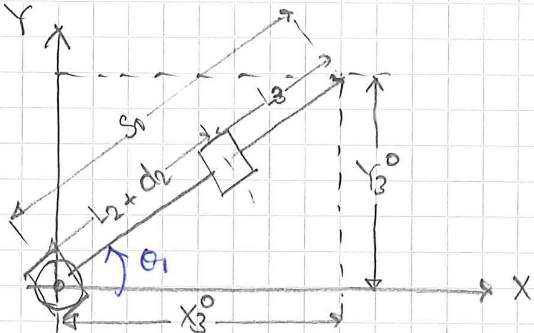
$$H_3^0 = \begin{bmatrix} 0 & -s_1 & -c_1 & s_1(l_2 + d_2) \\ 0 & c_1 & -s_1 & -c_1(l_2 + d_2) \\ 1 & 0 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_3 & c_3 & 0 & s_3 l_3 \\ -c_3 & s_3 & 0 & -c_3 l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^0 = \begin{bmatrix} s_1 c_3 & -s_1 s_3 & -c_1 & s_1 c_3 l_3 + s_1(l_2 + d_2) \\ -c_1 c_3 & c_1 s_3 & -s_1 & -c_1 c_3 l_3 - c_1(l_2 + d_2) \\ s_3 & c_3 & 0 & s_3 l_3 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PART II.2.c Kinematics and Transformation

The inverse kinematics of the manipulator.

TopView



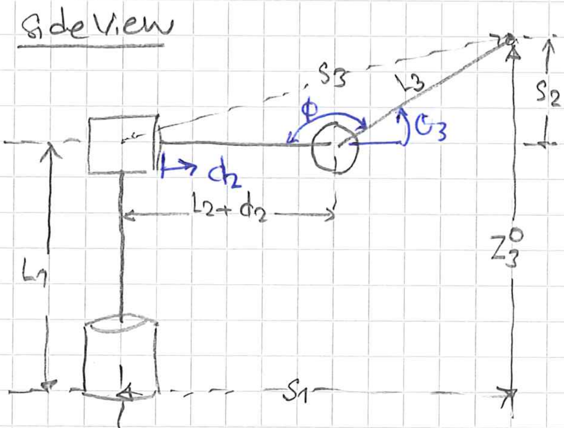
$$\theta_1 = \tan^{-1}\left(\frac{Y_3^0}{X_3^0}\right)$$

$$(X_3^0)^2 + (Y_3^0)^2 = (L_2 + d_2 + L_3)^2$$

$$L_2 + d_2 + L_3 = \sqrt{(X_3^0)^2 + (Y_3^0)^2}$$

$$d_2 = \sqrt{(X_3^0)^2 + (Y_3^0)^2} - L_2 - L_3$$

SideView



$$S_2 = Z_3^0 - L_1$$

$$S_1 = \sqrt{(X_3^0)^2 + (Y_3^0)^2}$$

$$S_3 = \sqrt{S_1^2 + S_2^2}$$

$$S_3^2 = S_1^2 + S_2^2 - 2 S_1 S_2 \cos\phi$$

$$\phi = \cos^{-1}\left(\frac{S_1^2 + S_2^2 - S_3^2}{2 S_1 S_2}\right)$$

$$\theta_3 = 180 - \phi$$

PART II.2.d Kinematics and Transformation

The manipulator is set up 0.5m away from the table. The table is 0.5m height and 0.5 meter square. A cube is 10cm on each sides and placed at the center of the table with frame $O_T(x, y, z)$.
The origin of coordinate frame B, relative to the world frame W, is located at position $O_B = (2.5, 2.5, 0)$
The origin of coordinate point frame T, relative to the world frame W, is located at position $O_T = (2.5, 3.25, 0.5)$

PART II.2.e Kinematics and Transformation

The Jacobian of the manipulator can be derive as below.

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

$$J_v = \begin{bmatrix} Z_0^0 \times (O_3^0 - O_0^0) & Z_1^0 & Z_2^0 \times (O_3^0 - O_2^0) \end{bmatrix}$$
$$J_\omega = \begin{bmatrix} Z_0^0 & 0 & Z_2^0 \end{bmatrix}$$

$$J_{v1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} s_1 c_3 l_3 + s_1(l_2 + d_2) \\ -c_1 c_3 l_3 - c_1(l_2 + d_2) \\ s_3 l_3 + l_1 \end{bmatrix}$$

$$J_{v1} = \begin{bmatrix} c_1 c_3 l_3 + c_1(l_2 + d_2) \\ s_1 c_3 l_3 + s_1(l_2 + d_2) \\ 0 \end{bmatrix}$$

$$J_{v2} = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}$$

$$J_{v3} = \begin{bmatrix} -c_1 \\ -s_1 \\ 0 \end{bmatrix} \times \begin{bmatrix} s_1 c_3 l_3 \\ -c_1 c_3 l_3 \\ s_3 l_3 \end{bmatrix}$$

$$J_{v3} = \begin{bmatrix} -s_1 s_3 l_3 \\ c_1 s_3 l_3 \\ c_3 l_3 \end{bmatrix}$$

$$J_\omega = \begin{bmatrix} Z_0^0 & Z_1^0 & Z_2^0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -c_1 \\ 0 & 0 & -s_1 \\ 1 & 0 & 0 \end{bmatrix}$$

We can now describe Jacobian matrix as we plug in J_v and J_ω.

$$J = \begin{bmatrix} c_1c_3l_3 + c_1(l_2 + d_2) & s_1 & -s_1s_3l_3 \\ s_1c_3l_3 + s_1(l_2 + d_2) & -c_1 & c_1s_3l_3 \\ 0 & 0 & c_3l_3 \\ 0 & 0 & -c_1 \\ 0 & 0 & -s_1 \\ 1 & 0 & 0 \end{bmatrix}$$

PART II.2.f Kinematics and Transformation

Finding all possible singularities of the manipulator.

$$J_v = \begin{bmatrix} c_1c_3l_3 + c_1(l_2 + d_2) & s_1 & -s_1s_3l_3 \\ s_1c_3l_3 + s_1(l_2 + d_2) & -c_1 & c_1s_3l_3 \\ 0 & 0 & c_3l_3 \end{bmatrix}$$

For simplicity, we can make the equation as

$A = c_3l_3$

$B = l_2 + d_2$

$A = s_3l_3$

$det(J_v) = 0$

$det(J_v) = (c_1A + c_1B)(-c_1A) - s_1(A(s_1A + s_1B)) - s(0)$

$det(J_v) = -c_1^2A^2 - c_1^2AB - A^2s_1^2 - ABs_1^2$

$det(J_v) = -AB(s_1^2 + c_1^2) - A^2(s_1^2 + c_1^2)$

$det(J_v) = -AB - A^2$

$det(J_v) = -(c_3l_3)(l_2 + d_2) - (c_3l_3)^2$

$det(J_v) = -c_3l_3l_2 - c_3l_3d_2 - c_3^2l_3^2$

We have singularities configuration for $c_3 = \frac{\pi}{2}$

PART II.3.a Forward/Inverse Kinametics Programming

Listing 1: Forward Kinametics Python code

```
import numpy as np
import math

l1 = 1.0   # m
l2 = 0.3   # m
l3 = 0.2   # m

def forward(theta_1,d_2,theta_3):
        theta_1 = (theta_1/180.0)*np.pi
        theta_3 = (theta_3/180.0)*np.pi

        RTz_1 = [[np.cos(theta_1),-np.sin(theta_1),0,0],
                 [np.sin(theta_1),np.cos(theta_1),0,0],
                 [0,0,1,l1],
                 [0,0,0,1]]
        RTx_1 = [[1,0,0,0],
                 [0,0,-1,0],
                 [0,1,0,0],
                 [0,0,0,1]]
        RTz_2 = [[0,-1,0,0],
                 [1,0,0,0],
                 [0,0,1,l2 + d2],
                 [0,0,0,1]]
        RTx_2 = [[1,0,0,0],
                 [0,0,1,0],
                 [0,-1,0,0],
                 [0,0,0,1]]
        RTz_3 = [[np.cos(theta_3),-np.sin(theta_3),0,0],
                 [np.sin(theta_3),np.cos(theta_3),0,0],
                 [0,0,1,0],
                 [0,0,0,1]]
        RTx_3 = [[1,0,0,.3],
```

```
            [0,1,0,0],
            [0,0,1,0],
            [0,0,0,1]]


    H0_1 = np.dot(RTz_1,RTx_1)
    H1_2 = np.dot(RTz_2,RTx_2)
    H2_3 = np.dot(RTz_3,RTx_3)
    cart_cord = np.dot(np.dot(H0_1,H1_2),H2_3)

    return cart_cord
```

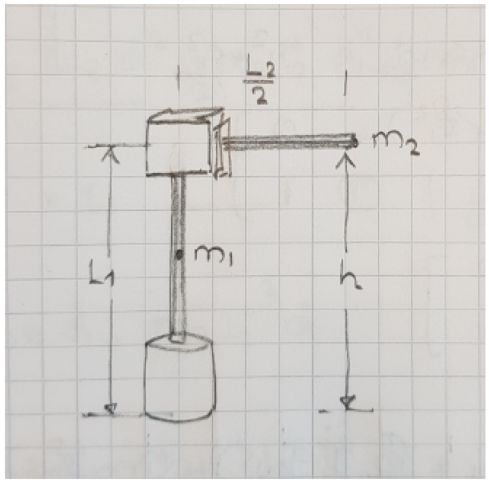
PART II.3.b Forward/Inverse Kinematics Programming

Listing 2: Inverse Kinematics Python code

```
import numpy as np
import math

l1 = 1.0  # m
l2 = 0.3  # m
l3 = 0.2  # m



def inverse(X, Y, Z):
    theta_1 = np.arctan2(Y, X)

    d_2 = math.sqrt(X * X + Y * Y) - l2 - l3

    S2 = Z - l1
    S1 = math.sqrt(X * X + Y * Y)
    S3 = math.sqrt(S1 * S1 + S2 * S2)
    phi = np.arccos((S1*S1 + S2*S2 - S3*S3)/(2*S1*S2))

    theta_3 = 180 - phi

    joint_angles = [theta_1 * (180 / np.pi), d_2, theta_3 * (180 / np.pi)]

    return joint_angles
```

PART II.4.a Dynamics

The last link of the manipulator is removed so that only two links remain. The choice of the manipulator has rectangular solid as a link and mass are uniformly distributed.



Potential energy for each mass and summing the together:

$$\mathcal{P} = P_1 + P_2 \tag{5}$$

$$P_1 = m_1 g \frac{l_1}{2}; P_2 = m_2 g l_1$$

From the equation (5) we get

$$P = m_1 g \frac{l_1}{2} + m_2 g l_1 \tag{6}$$

PART II.4.b Dynamics

$\mathcal{K}$ is kinetic energy and we have the following equation.

$$\mathcal{K} = \frac{1}{2}\dot{q}^T[\sum_{i=1}^{n}(m_i J_{vi}^T J_{vi} + J_{\omega i}^T R_i I_i R_i^T J_{\omega i})]\dot{q} \tag{7}$$

We have the Jacobian from the previous task for calculating velocities.

$$J = \begin{bmatrix} c_1c_3l_3 + c_1(l_2+d_2) & s_1 & -s_1s_3l_3 \\ s_1c_3l_3 + s_1(l_2+d_2) & -c_1 & c_1s_3l_3 \\ 0 & 0 & c_3l_3 \\ 0 & 0 & -c_1 \\ 0 & 0 & -s_1 \\ 1 & 0 & 0 \end{bmatrix}$$

Since it is the third joint that has been removed, the third column and all instances of l_3 in J should be set to zero. Therefore we can derive the Jacobian as below:

Let $l_2^* = l_2 + d_2$

$$J = \begin{bmatrix} c_1l_2^* & s_1 & 0 \\ s_1l_2^* & -c_1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

For i = 1;

$$J_{v1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; J_{v1}^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$J_{\omega 1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}; J_{\omega 1}^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$J_{v1}^T J_{v1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$J_{\omega 1}^T I J_{\omega 1} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_{1,x} & 0 & 0 \\ 0 & I_{1,y} & 0 \\ 0 & 0 & I_{1,z} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & I_{1,z} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} I_{1,z} & 0 \\ 0 & 0 \end{bmatrix}$$

For i = 2;

$$J_{v2} = \begin{bmatrix} c_1 l_2^* & s_1 \\ s_1 l_2^* & -c_1 \\ 0 & 0 \end{bmatrix} ; J_{v2}^T = \begin{bmatrix} c_1 l_2^* & s_1 l_2^* & 0 \\ s_1 & -c_1 & 0 \end{bmatrix}$$

$$J_{v2}^T J_{v2} = \begin{bmatrix} l_2^{*2} & 0 \\ 0 & 1 \end{bmatrix}$$

There would not be rotational part for joint 2 as it is given as prismatic joint.

Then from equation (6) we get:

$$\mathcal{K} = \frac{1}{2}\dot{q}^T[(m_1 J_{v1}^T J_{v1} + J_{\omega 1}^T R_1 I_1 R_1^T J_{\omega 1}) + (m_2 J_{v2}^T J_{v2})]\dot{q} \tag{8}$$

$$\mathcal{K} = \frac{1}{2}\dot{q}^T[(m_1 \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} I_{1,z} & 0 \\ 0 & 0 \end{bmatrix}) + (m_2 \begin{bmatrix} l_2^{*2} & 0 \\ 0 & 1 \end{bmatrix})]\dot{q} \tag{9}$$

$$\mathcal{K} = \frac{1}{2}\dot{q}^T[\begin{bmatrix} I_{1,z} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} m_2 l_2^{*2} & 0 \\ 0 & m_2 \end{bmatrix}]\dot{q} \tag{10}$$

$$\mathcal{K} = \frac{1}{2}\begin{bmatrix} \dot{\theta}_1 & \dot{l}_2^* \end{bmatrix} \begin{bmatrix} I_{1,z} + m_2 l_2^{*2} & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{l}_2^* \end{bmatrix} \tag{11}$$

$$\mathcal{K} = \frac{1}{2}\begin{bmatrix} (I_{1,z} + m_2 l_2^{*2})\dot{\theta}_1 & m_2 \dot{l}_2^* \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{l}_2^* \end{bmatrix} \tag{12}$$

$$\mathcal{K} = \frac{1}{2}[(I_{1,z} + m_2 l_2^{*2})\dot{\theta}_1^2 + m_2 \dot{l}_2^{*2}] \tag{13}$$

Then from equation (6) and (13) we get:

$$\mathcal{L} = \frac{1}{2}[(I_{1,z} + m_2 l_2^{*2})\dot{\theta}_1^2 + m_2 \dot{l}_2^{*2}] + m_1 g \frac{l_1}{2} + m_2 g l_1 \tag{14}$$

PART II.4.c Dynamics

Deriving the dynamical model of the manipulator using the Euler-Lagrange method

The dynamical model of robot can be describe:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau \tag{15}$$

For the manipulatior, the equation should be:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} - \frac{\partial \mathcal{L}}{\partial \theta_1} = \tau_1 \tag{16}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{l}_2^*} - \frac{\partial \mathcal{L}}{\partial l_2^*} = f_2 \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} = I_{1,z}\dot{\theta}_1 + m_2 l_2^{*2}\dot{\theta}_1$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} = I_{1,z}\ddot{\theta}_1 + 2m_2 l_2^* \dot{l}_2^* \dot{\theta}_1 + m_2 l_2^{*2}\ddot{\theta}_1$$

$$\frac{\partial \mathcal{L}}{\partial \theta_1} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \dot{l}_2^*} = m_2 \dot{l}_2^*$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{l}_2^*} = m_2 \ddot{l}_2^*$$

$$\frac{\partial \mathcal{L}}{\partial l_2^*} = m_2 l_2^* \dot{\theta}_1^2$$

$$\tau_1 = I_{1,z}\ddot{\theta}_1 + 2m_2 l_2^* \dot{l}_2^* \dot{\theta}_1 + m_2 l_2^{*2}\ddot{\theta}_1$$

$$f_2 = m_2 \ddot{l}_2^* - m_2 l_2^* \dot{\theta}_1^2$$

$$\begin{bmatrix} \tau_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} I_{1,z} & m_2 \\ m_2 l_2^{*2} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{l}_2^2 \end{bmatrix} + \begin{bmatrix} 2m_2 l_2^* \dot{l}_2^* & 0 \\ -m_2 l_2^* \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{l}_1^* \end{bmatrix} \tag{18}$$

PART II.5.a Control Theory

The resulting dynamic equation from the 2 DOF system in the previous task is shown below.

$$\begin{bmatrix} \tau_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} I_{1,z} & m_2 \\ m_2 l_2^{*2} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{l}_2^2 \end{bmatrix} + \begin{bmatrix} 2m_2 l_2^* \dot{l}_2^* & 0 \\ -m_2 l_2^* \dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{l}_1^* \end{bmatrix} \tag{19}$$

By locking the last joint of the system and the system left with just 1 DOF manipulator.

$$\tau_1 = (I_{1,z} + m_2 l_2^{*2})\ddot{\theta}_1 + 2m_2 l_2^* \dot{l}_2^* \dot{\theta}_1$$

For simplicity, let $I_{1,z} = 0$. The derivative of joint2 is constant and $l_2^{*2} = l_2^2$ since joint2 was locked.

$$\tau_1 = m_2 l_2^2 \ddot{\theta}_1$$

By comparing with the equation which was given

$$u(t) = \mathbf{J}\ddot{\mathbf{q}} + \mathbf{B}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q}$$

We can simply get

$J = m_2 l_2^2$

$B = 0$

$K = 0$

PART II.5.b Control Theory

To transform the dynamic equation into the Laplace domain, we get

$$u(t) = J\ddot{\theta}(t) + B\dot{\theta}(t) + K(t) \tag{20}$$

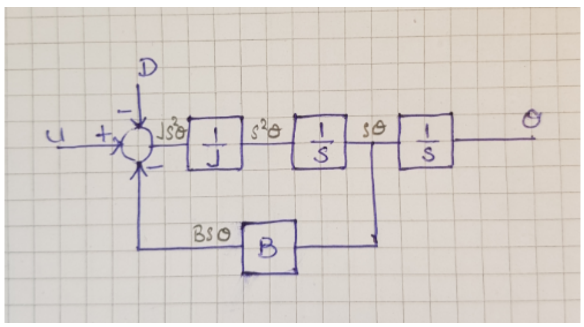
$$U = Js^2\theta + Bs\theta + K$$

$$U - K = Js^2\theta + Bs\theta$$

$$U - K = \theta(Js^2 + Bs)$$

$$\theta = \frac{U - K}{Js^2 + Bs} \tag{21}$$

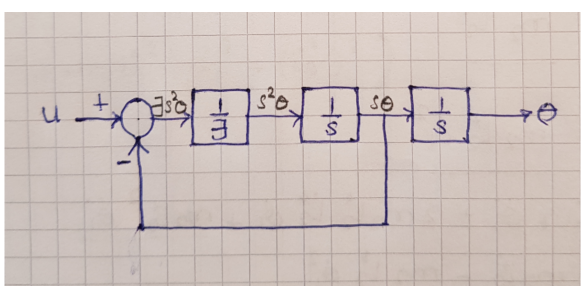
So the transfer function block diagram can be describe as follows:



Since we have the value of B and K equal zero, we could modify the Laplace and block diagram as below;

$$U = Js^2\theta$$

$$\theta = \frac{U}{Js^2} \tag{22}$$



Add a PD-controller to the block diagram and derive the transfer function between the input $\theta_d(s)$ and the output $\theta(s)$.

The error is defined as $e(t) = \theta_d(t) - \theta(t)$.

Firstly, we would like to drive P-controller (K_p).

(The controller (K) use the error e(t) to calculate its output, called control effort, denote as u (U in block diagram)).

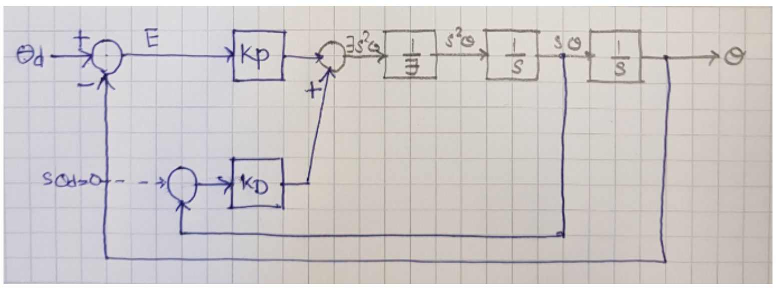
$$U(t) = K_p e(t)$$

where $e(t) = \theta_d(t) - \theta(t)$, and make Laplace transform.

$$U(s) = K_p E(s)$$

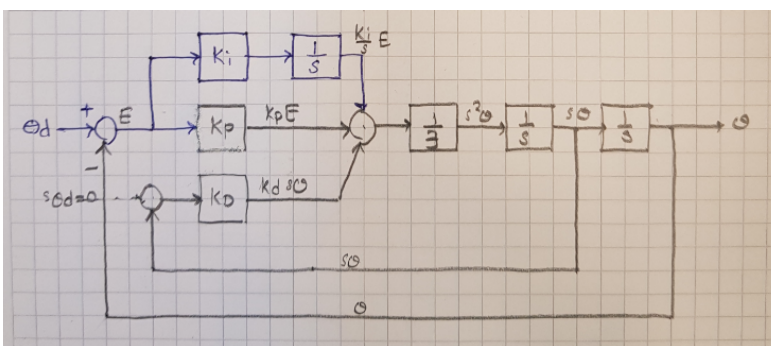
The transfer function block diagram with Proportional Derivative (PD) Controller can be describe as follows:

$$U(s) = (\theta_d - \theta)(K_p + K_d s)$$



The transfer function block diagram with Proportional Derivative (PID) Controller can be describe as follows:

$$U(s) = (\theta_d - \theta)(K_p + K_d s + \frac{K_i}{s})$$



PART II.5.c Control Theory

The PD-controller transfer function between the desired angle input and the resulting angle of the close loop system PD-controller (K_p and K_d).

$$u(t) = K_p e(t) + K_d \dot{e}(t)$$

$$u(t) = K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta})$$

Laplace transform:

$$U(s) = K_p(\theta_d - \theta) + K_d(s\theta_d - s\theta)$$

$$U(s) = K_p(\theta_d - \theta) + K_d s(\theta_d - \theta)$$

$$U(s) = (\theta_d - \theta)(K_p + K_d s)$$

From equation (22) we can make PD-transfer function as below:

$$\theta = \frac{(\theta_d - \theta)(K_p + K_d s)}{Js^2} \tag{23}$$

The PID-controller transfer function between the desired angle input and the resulting angle of the close loop system of the PID-controller (K_p, K_d and K_i).

$$u(t) = K_p e(t) + K_d \dot{e}(t) + K_i \int e(t)dt$$

Laplace transform:

$$U(s) = (\theta_d - \theta)(K_p + K_d s + \frac{K_i}{s})$$

From equation (22) we can make PD-transfer function as below:

$$\theta = \frac{(\theta_d - \theta)(K_p + K_d s + \frac{K_i}{s})}{Js^2} \tag{24}$$