# Homework - 05

Trajectory Generation + Torque Calculation

Task to draw a circle of radius 10 cm within 200 seconds

using the inverse velocity kinematics approach

(Inverse Jacobians)

Consider Joint 3 to be locked $\theta_3 = 0$

$$q(\text{at } t=0) = [q_1, q_2, q_4, q_5, q_6, q_7]^T$$

$$= [0.0, 0.0, pi/2, 0.0, pi, 0.0]^T$$

**Step 1** setting up the frames

# HOMEWORK - 4

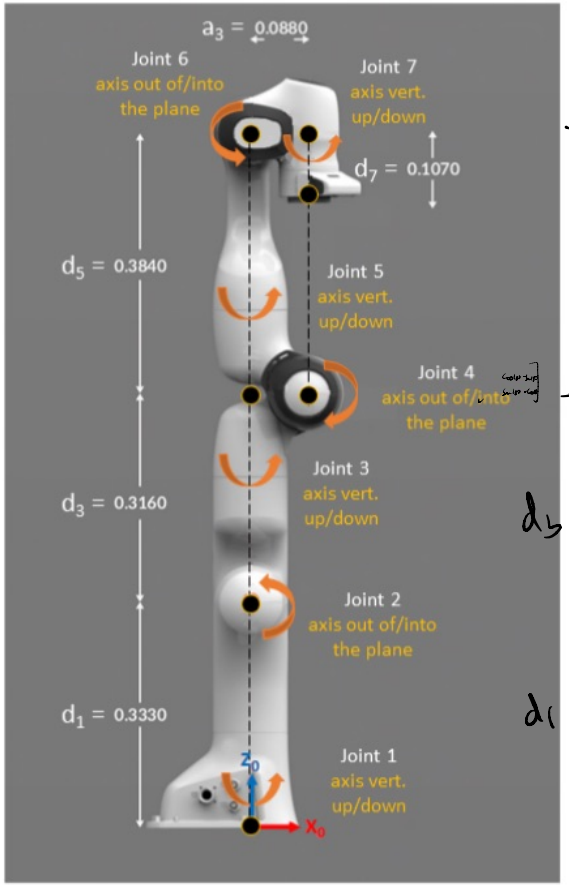
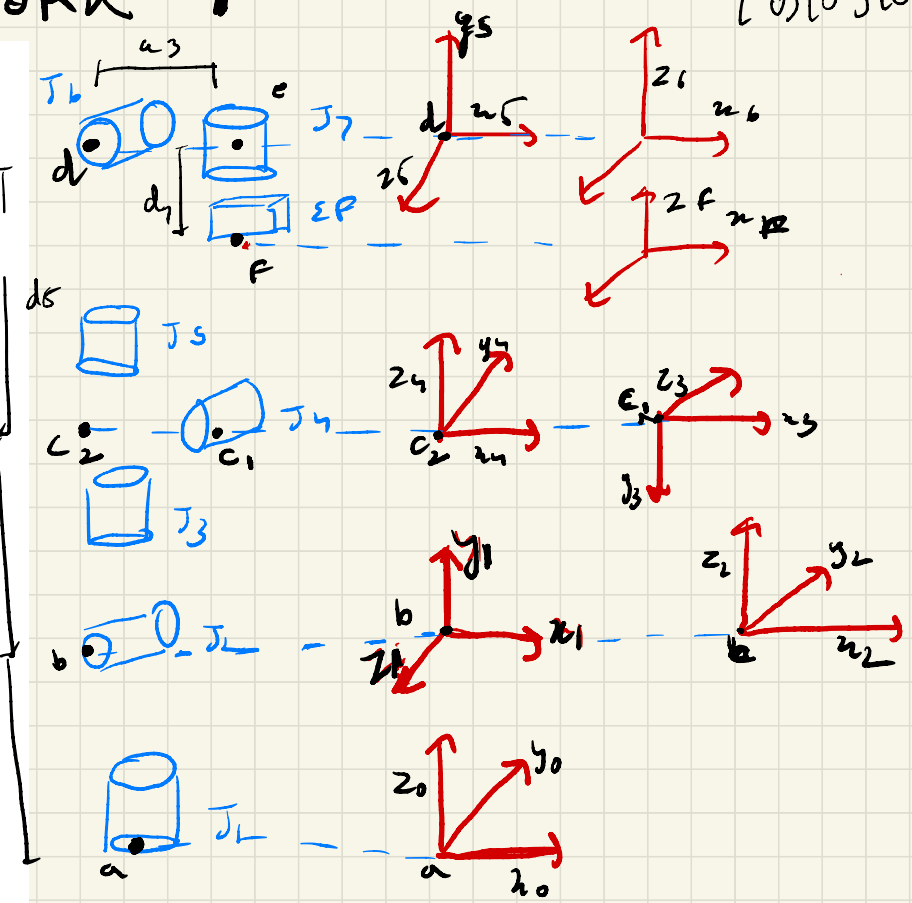



Fig. 2: Panda Robot - Home configuration

(Orange arrows indicate the positive sense of joint rotation)



* $\theta_3 = 0$ Rest all is same

DH Table from Selected frame Spongs

| . | $\theta$ | d | a | $\alpha$ |
|---|---|---|---|---|
| 1 | $\theta_1$ | $d_1$ | 0 | $\pi/2$ |
| 2 | $\theta_2$ | 0 | 0 | $-\pi/2$ |
| 3 | 0 | $d_3$ | $+a_3$ | $-\pi/2$ |
| 4 | $\theta_4$ | 0 | $-a_3$ | $\pi/2$ |
| 5 | $\theta_5$ | $d_5$ | 0 | $\pi/2$ |
| 6 | $\theta_6$ | 0 | $a_3$ | $-\pi/2$ |
| 7 | $\theta_7$ | $-d_7$ | 0 | 0 |

$z_{i\frac{1}{n}} z_{i-1}$

D-H parameters

$d_i$ : $x_i$ distance from $x_{i-1}$ along $z_{i-1}$

$\theta_i$ : $x_i$ angle from $x_{i-1}$ around $z_{i-1}$

$a_i$ : $z_i$ distance from $z_{i-1}$ along $x_i$

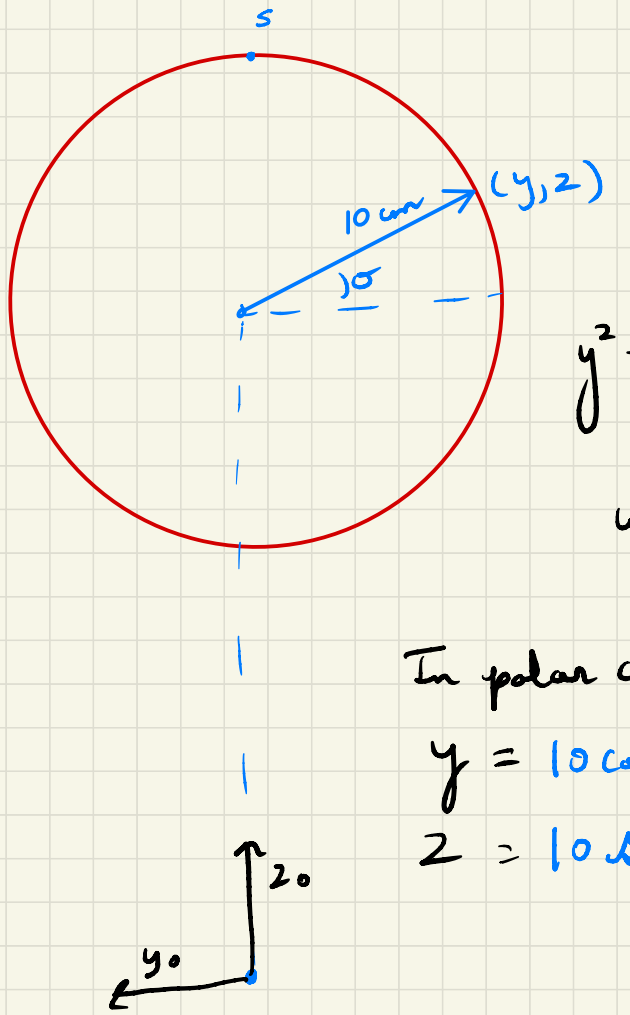
$\alpha_i$ : $z_i$ angle from $z_{i-1}$ around $x_i$

=) This is computed in Jupyter notebook

=) The 5 geometric configurations are also made in the code

=) Even $T_i$ is also been printed in it.

To add the end effector pen length add 10cm to $d_7$




plane x = 67.9cm
wall

eqn of circle

$y^2 + (z - 72.5)^2 = 10^2$

in cartesian form
with respect to origin
frame

In polar coordinates

$y = 10\cos(\theta)$

$z = 10\sin(\theta)$

$$y = 10\cos(\theta)$$

$$z = 10\sin(\theta) + 72.5$$

$$V_y = \dot{y} = -10\sin(\theta)\dot{\theta}$$

$$V_z = \dot{z} = 10\cos(\theta)\dot{\theta}$$

$$\dot{\theta} = \frac{2\pi}{200} \quad \text{as time Secs}$$

$$\theta = \dot{\theta}(t) = \frac{2\pi}{200}t$$

As our starting point is at $\frac{\pi}{2}$

$$\theta = \frac{\pi}{2} + \theta$$

$$\dot{x} = \begin{bmatrix} 0 \\ -10\sin\left(\frac{2\pi}{200}t\right)\frac{2\pi}{200} \\ 10\cos\left(\frac{2\pi}{200}t\right)\frac{2\pi}{200} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} 0 \\ -10\cos\left(\frac{2\pi}{200}t\right)\frac{2\pi}{200} \\ -10\sin\left(\frac{2\pi}{200}t\right)\frac{2\pi}{200} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

To find the jacobian :- write J

Step 1 Calculate ${}^{0}_{i}T$

Step 2 Calculate $Z_i$

Step 3 Calculate $h(q_1, q_2, \ldots q_n)$

Step 4 Calculate $\partial h / \partial q_i$

$$A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$T_0^1 = A_1$

$T_0^2 = A_1 A_2$

$T_0^3 = A_1 A_2 A_3$

$T_0^4 = A_1 A_2 A_3 A_4$

$T_0^5 = A_1 A_2 A_3 A_4 A_5$

$T_0^6 = A_1 A_2 A_3 A_4 A_5 A_6$

$T_0^7 = A_1 A_2 A_3 A_4 A_5 A_6 A_7$

$$h(q_1, q_2, \cdots q_7) = x_p(q_1, q_2 \cdots q_7) = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}$$

$$^0_7T = \left[\begin{array}{ccc|c} & {}^0R_n & & {}^0\vec{p} \\ & & & {}^n \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

P: origin of the end effector coordinates

$x_p, y_p, z_p$ are end effector coordinates

${}^0Z_i$ is the 3rd column of the ${}^0T_i$ matrix for $i = 0, 1, 2, 4, 5, 6, 7$

$$ {}^0_7J = \begin{bmatrix} \frac{\partial {}^0x_p}{\partial q_1} & \frac{\partial {}^0x_p}{\partial q_2} & \frac{\partial {}^0x_p}{\partial q_4} & \frac{\partial {}^0x_p}{\partial q_5} & \frac{\partial {}^0x_p}{\partial q_6} & \frac{\partial {}^0x_p}{\partial q_7} \\ {}^0Z_1 & {}^0Z_2 & {}^0Z_4 & {}^0Z_5 & {}^0Z_6 & {}^0Z_7 \end{bmatrix} $$

At $t = 0$

$$q\text{-current} = \begin{bmatrix} 0 \\ 0 \\ \pi/2 \\ 0 \\ \pi \\ 0 \end{bmatrix}$$

$$q\text{-dot} = J^{-1} \times \dot{X}$$

for integration $\quad \Delta t = \frac{t}{N}$ → total time
N → points to be plotted

$$q\text{-current} = q\text{-current} + q\text{-dot}\,\Delta t$$

Use this q_current to get the end effector position

$x_p$.

This process will be continuate (N) number of times

& then $x_p$ can be used to plot the x, y, z.

* Please refer the codes Html with the explanation for better understanding of the process

# JACOBIAN MATRIX

In [67]:
```
q_current=Matrix(([0],
                  [0],
                  [pi/2],
                  [0],
                  [pi],
                  [0]))
J_initial=J.subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]})
J_initial
```

Out[67]:

$$\begin{bmatrix} 0 & -49.2 & 17.6 & 0 & -8.8 & 0 \\ 67.9 & 0 & 0 & -8.8 & 0 & 0 \\ 0 & 67.9 & -59.1 & 0 & 20.7 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Calculations for TORQUE →

For Each Joint

* The masses for each link were taken from URDF file

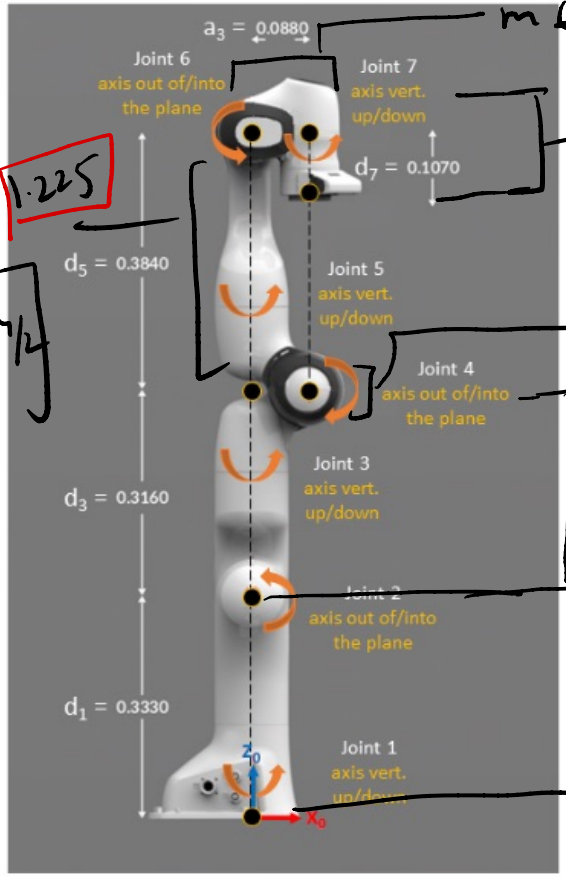



Fig. 2: Panda Robot - Home configuration

(Lengths shown are in **meters**)

$m_6 = 1.666$ $cog_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

$m_7 = 0.73 + 0.735$ $cog_7 = \begin{bmatrix} 0 \\ 0 \\ -0.107/2 \\ 1 \end{bmatrix}$

$m_5 = 1.225$

$cog_5 = \begin{bmatrix} 0 \\ 38.4/2 \\ \\ \end{bmatrix}$

the cog will be almost at the motor

$m_4 = 3.58$

$cog_4 = \begin{bmatrix} 0 \\ 0 \\ 5 \\ 1 \end{bmatrix}$

$m_2 = 3.228 + 0.626$ $cog_2 = \begin{bmatrix} 0 \\ 0 \\ 31.6/2 \\ 1 \end{bmatrix}$

$m_1 = 3.06 + 4.97$ $cog_1 = \begin{bmatrix} 0 \\ 0 \\ 33.30/2 \\ 1 \end{bmatrix}$

The CoG values have been assumed and the assumptions are as follows

① CoG for links are at the mid point of the link

② CoG for motors are $\approx 0$

Robot Dynamic Eq:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = T + J^{T}(q)F$$

but we know that for this problem

$$\ddot{q}, \dot{q} \equiv 0$$

$$\therefore \quad M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau + J(q)^T f$$

$$g(q) = \tau + J(q)^T f$$

$$\tau = g(q) - J(q)^T f$$

We have already computed $J(q)$

$f = [-5, 0, 0, 0, 0, 0]^T$ as given in the question

∴ we just need to compute $g(q)$

$$g(q) = \begin{bmatrix} \frac{\partial P}{\partial q_1} & \frac{\partial P}{\partial q_2} & \frac{\partial P}{\partial q_4} & \frac{\partial P}{\partial q_5} & \frac{\partial P}{\partial q_6} & \frac{\partial P}{\partial q_7} \end{bmatrix}$$

### 6.2.3 Potential Energy for an $n$-Link Robot

Now consider the potential energy term. In the case of rigid dynamics, the only source of potential energy is gravity. The potential energy of the $i$-th link can be computed by assuming that the mass of the entire object is concentrated at its center of mass and is given by

$$P_i = g^T r_{ci} m_i \tag{6.48}$$

where $g$ is vector giving the direction of gravity in the inertial frame and the vector $r_{ci}$ gives the coordinates of the center of mass of link $i$. The total potential energy of the $n$-link robot is therefore

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} g^T r_{ci} m_i \tag{6.49}$$

In the case that the robot contains elasticity, for example, flexible joints, then the potential energy will include terms containing the energy stored in the elastic elements.

Note that the potential energy is a function only of the generalized coordinates and not their derivatives, i.e. the potential energy depends on the configuration of the robot but not on its velocity.

where P is the potential Energy

← Given by

$$P = \sum_{i=1}^{n} \vec{g}\, r_{ci}\, m_i$$

where $m_i$ is the mass of link

$r_{ci}$ is the COG

and $G_1 = 9.8\ m/s^2$

For our Robot

The COG we have taken are in the local frames so we need to convert them via the global frame

$COG_1 = cog_1$ ← $r_1$

$COG_2 = T_0^1 cog_2$ ← $r_2$

$COG_4 = T_0^4 cog_4$ ← $r_3$

$COG_5 = T_0^5 cog_5$ ← $r_4$

$COG_6 = T_0^6 cog_6$ ← $r_5$

$COG_7 = T_0^7 cog_7$ ← $r_6$

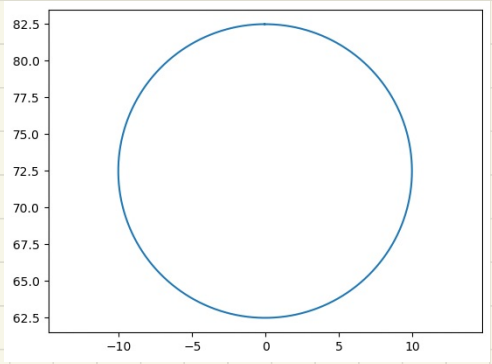
$$P = [m_1 \times COG1 + m_2 \times COG2 + m_4 * COG4 + m_5 * COG5 + m_6 * COG6 + m_7 \times COG7] \times G$$

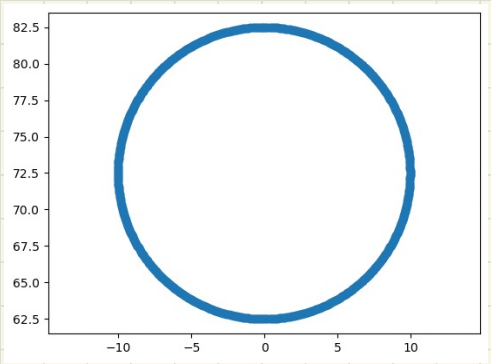
$$g(q) = \begin{bmatrix} \frac{d}{dq_1} P & \frac{d}{dq_2} P & \frac{d}{dq_4} P & \frac{dP}{dq_5} & \frac{dP}{q_6} & \frac{dP}{dq_7} \end{bmatrix}$$

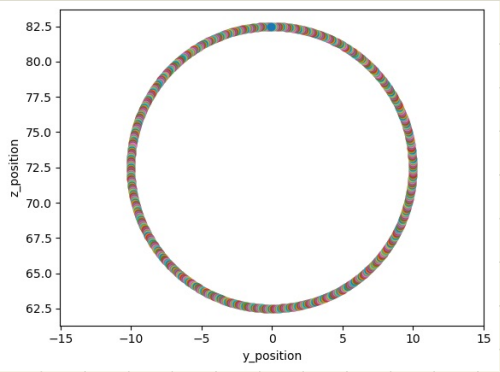
$$\tau = g(q) - J^T(q) f$$

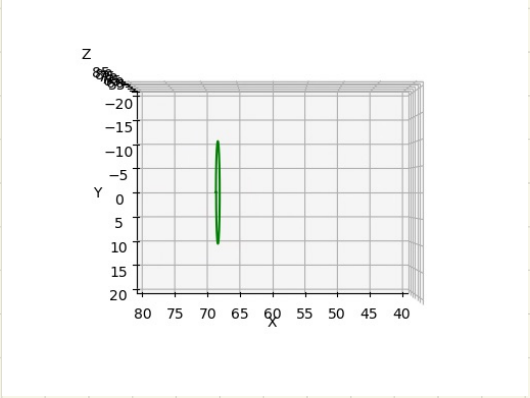
Replacing the values of $g(q)$, $J^T(q)$, $F$ and iterating this we will get the torque values at every second

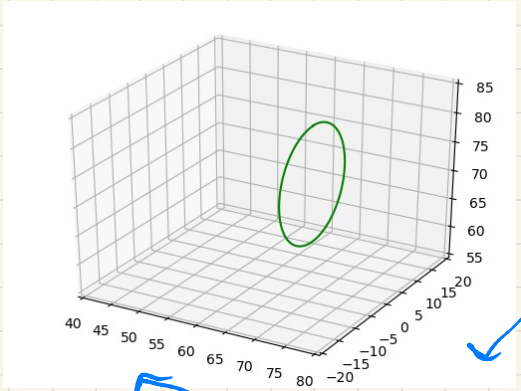
# PLOTS


z
y
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
62.5
−10
−5
0
5
10


z
y
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
62.5
−10
−5
0
5
10


z_position
y_position
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
62.5
−15
−10
−5
0
5
10
15

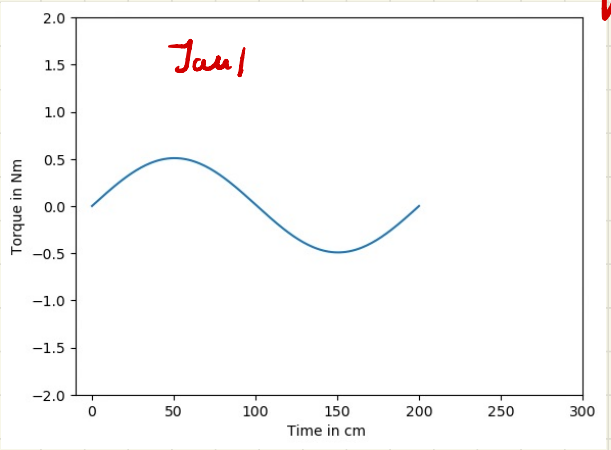

Z
Y
X
−20
−15
−10
−5
0
5
10
15
20
80
75
70
65
60
55
50
45
40

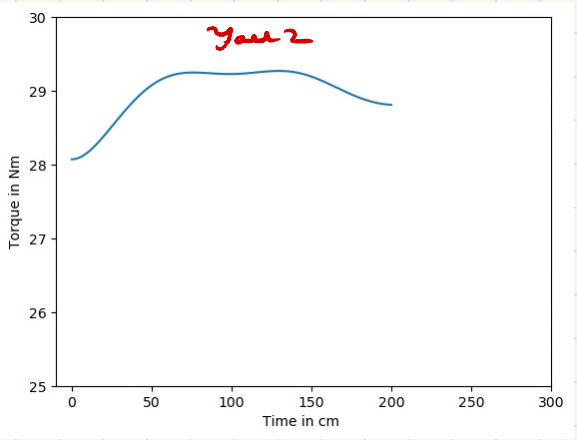

z
y
x
85
80
75
70
65
60
55
40
45
50
55
60
65
70
75
80
−20
−15
−10
−5
0
5
10
15
20

Joint 1

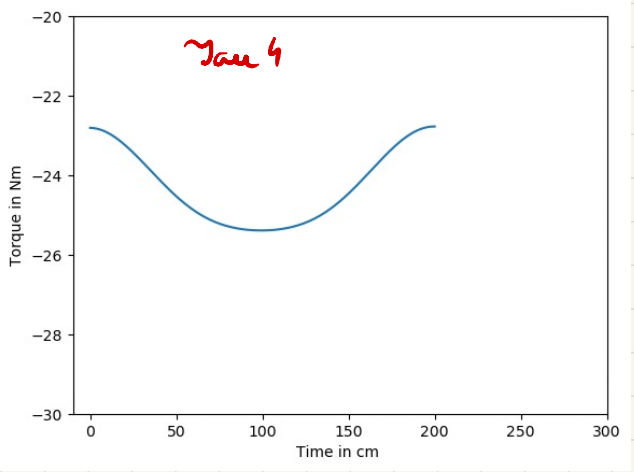
# Torque Plots

Joint 2



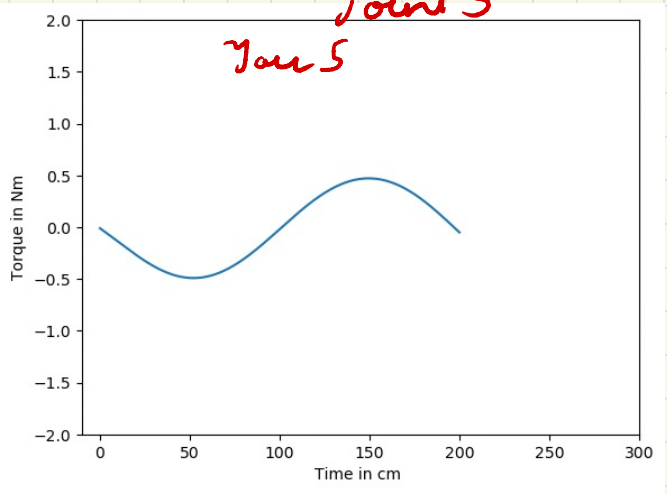




Joint 4



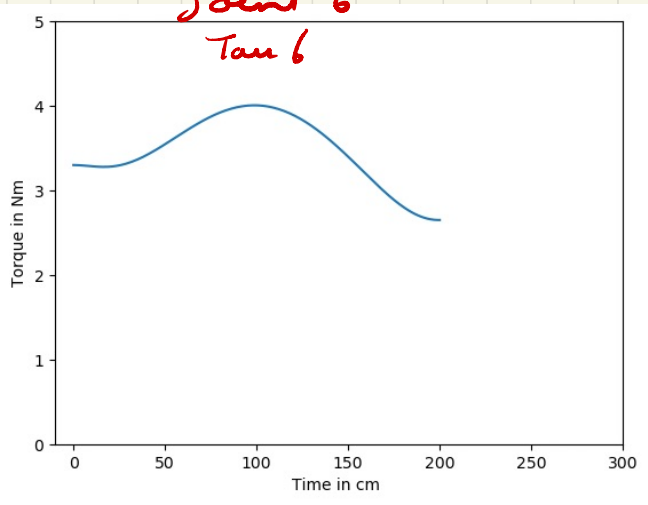
Joint 5

Tau 5



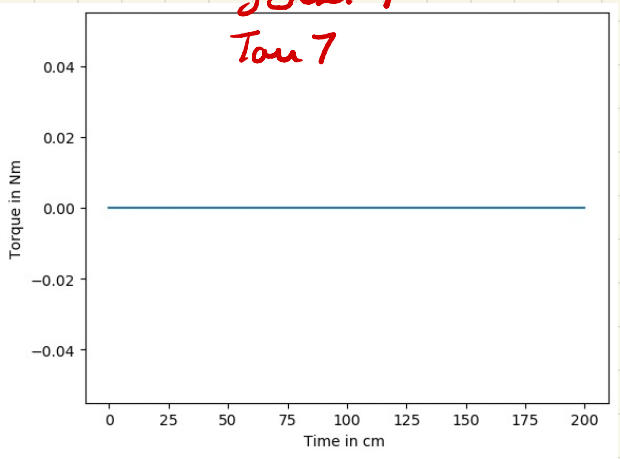

Joint 6

Tau 6




Joint 7

Tau 7



# HOMEWORK-5

In [1]:
```
#sympy library used to express matrices in terms of symbols
# matplot lib used to plot the graph

from sympy import *
from IPython.display import Image,display,HTML
import matplotlib.pyplot as plt
from matplotlib import cm
import numpy as np

from mpl_toolkits.mplot3d.axes3d import get_test_data
init_printing()

thetadot,t=symbols('thetadot,t')
theta,d,a,alpha=symbols('theta,d,a,alpha')
d_1,d_3,d_5,d_7,a_3=symbols('d_1,d_3,d_5,d_7,a_3')
theta_1,theta_2,theta_3,theta_4,theta_5,theta_6,theta_7=symbols('theta_1,theta_2,theta_3
Rot_z_theta=Matrix([[cos(theta),-sin(theta),0,0],
                    [sin(theta),cos(theta),0,0],
                    [0,0,1,0],
                    [0,0,0,1]])
Trans_z_d=Matrix([[1,0,0,0],
                  [0,1,0,0],
                  [0,0,1,d],
                  [0,0,0,1]])
Trans_x_a=Matrix([[1,0,0,a],
                  [0,1,0,0],
                  [0,0,1,0],
                  [0,0,0,1]])
Rot_x_alpha=Matrix([[1,0,0,0],
                    [0,cos(alpha),-sin(alpha),0],
                    [0,sin(alpha),cos(alpha),0],
                    [0,0,0,1]])
```

In [2]:
```
# Transfromation matrix for each frame
A=Rot_z_theta*Trans_z_d*Trans_x_a*Rot_x_alpha
```

In [3]:
```
# Substituting values form DH table
A_1=A.subs({theta:theta_1,d:d_1,a:0,alpha:pi/2})
A_2=A.subs({theta:theta_2,d:0,a:0,alpha:-pi/2})
A_3=A.subs({theta:0,d:d_3,a:a_3,alpha:-pi/2})
A_4=A.subs({theta:theta_4,d:0,a:-a_3,alpha:pi/2})
A_5=A.subs({theta:theta_5,d:d_5,a:0,alpha:pi/2})
A_6=A.subs({theta:theta_6,d:0,a:a_3,alpha:-pi/2})
A_7=A.subs({theta:theta_7,d:-d_7,a:0,alpha:0})
```

In [4]:
```
# Transformation matrix for each frame with respect to origin

Transformation=A_1*A_2*A_3*A_4*A_5*A_6*A_7
Transformation
T_1=A_1
T_2=A_1*A_2
T_4=A_1*A_2*A_3*A_4
T_5=A_1*A_2*A_3*A_4*A_5
T_6=A_1*A_2*A_3*A_4*A_5*A_6
T_7=A_1*A_2*A_3*A_4*A_5*A_6*A_7
```

In [5]:
```
# 1st frame wrt origin
T_1.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

Out[5]:
$$\begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & 33.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In [6]:
```
# 2nd frame wrt origin
T_2.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

Out[6]:
$$\begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & -\sin(\theta_1) & -\sin(\theta_2)\cos(\theta_1) & 0 \\ \sin(\theta_1)\cos(\theta_2) & \cos(\theta_1) & -\sin(\theta_1)\sin(\theta_2) & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) & 33.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In [7]:
```
# 4th framw frame wrt origin
T_4.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

Out[7]:
$$\begin{bmatrix} \sin(\theta_2)\sin(\theta_4)\cos(\theta_1) + \cos(\theta_1)\cos(\theta_2)\cos(\theta_4) & -\sin(\theta_1) & -\sin(\theta_2)\cos(\theta_1)\cos(\theta_4) + \sin(\theta \\ \sin(\theta_1)\sin(\theta_2)\sin(\theta_4) + \sin(\theta_1)\cos(\theta_2)\cos(\theta_4) & \cos(\theta_1) & -\sin(\theta_1)\sin(\theta_2)\cos(\theta_4) + \sin(\theta \\ \sin(\theta_2)\cos(\theta_4) - \sin(\theta_4)\cos(\theta_2) & 0 & \sin(\theta_2)\sin(\theta_4) + \cos(\theta_2) \\ 0 & 0 & 0 \end{bmatrix}$$

In [8]:
```
# 5th frame wrt origin
T_5.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

Out[8]:
$$\begin{bmatrix} (\sin(\theta_2)\sin(\theta_4)\cos(\theta_1) + \cos(\theta_1)\cos(\theta_2)\cos(\theta_4))\cos(\theta_5) - \sin(\theta_1)\sin(\theta_5) & -\sin(\theta_2)\cos(\theta_1)\c \\ (\sin(\theta_1)\sin(\theta_2)\sin(\theta_4) + \sin(\theta_1)\cos(\theta_2)\cos(\theta_4))\cos(\theta_5) + \sin(\theta_5)\cos(\theta_1) & -\sin(\theta_1)\sin(\theta_2)\c \\ (\sin(\theta_2)\cos(\theta_4) - \sin(\theta_4)\cos(\theta_2))\cos(\theta_5) & \sin(\theta_2)\si \\ 0 \end{bmatrix}$$

In [9]:
```
# 6th frame wrt origin
T_6.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

Out[9]:

$$\begin{bmatrix} ((\sin(\theta_2)\sin(\theta_4)\cos(\theta_1) + \cos(\theta_1)\cos(\theta_2)\cos(\theta_4))\cos(\theta_5) - \sin(\theta_1)\sin(\theta_5))\cos(\theta_6) \\ + (-\sin(\theta_2)\cos(\theta_1)\cos(\theta_4) + \sin(\theta_4)\cos(\theta_1)\cos(\theta_2))\sin(\theta_6) \\ \\ ((\sin(\theta_1)\sin(\theta_2)\sin(\theta_4) + \sin(\theta_1)\cos(\theta_2)\cos(\theta_4))\cos(\theta_5) + \sin(\theta_5)\cos(\theta_1))\cos(\theta_6) \\ + (-\sin(\theta_1)\sin(\theta_2)\cos(\theta_4) + \sin(\theta_1)\sin(\theta_4)\cos(\theta_2))\sin(\theta_6) \\ \\ (\sin(\theta_2)\sin(\theta_4) + \cos(\theta_2)\cos(\theta_4))\sin(\theta_6) + (\sin(\theta_2)\cos(\theta_4) - \sin(\theta_4)\cos(\theta_2))\cos(\theta_5)\cos(\theta_6) \\ \\ 0 \end{bmatrix}$$

In [10]:
```
# 7th frame wrt orign
T_7.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

Out[10]:

$$\begin{bmatrix} (((\sin(\theta_2)\sin(\theta_4)\cos(\theta_1) + \cos(\theta_1)\cos(\theta_2)\cos(\theta_4))\cos(\theta_5) - \sin(\theta_1)\sin(\theta_5))\cos(\theta_6) \\ + (-\sin(\theta_2)\cos(\theta_1)\cos(\theta_4) + \sin(\theta_4)\cos(\theta_1)\cos(\theta_2))\sin(\theta_6))\cos(\theta_7) \\ + (-(\sin(\theta_2)\sin(\theta_4)\cos(\theta_1) + \cos(\theta_1)\cos(\theta_2)\cos(\theta_4))\sin(\theta_5) - \sin(\theta_1)\cos(\theta_5))\sin(\theta_7) \\ \\ (((\sin(\theta_1)\sin(\theta_2)\sin(\theta_4) + \sin(\theta_1)\cos(\theta_2)\cos(\theta_4))\cos(\theta_5) + \sin(\theta_5)\cos(\theta_1))\cos(\theta_6) \\ + (-\sin(\theta_1)\sin(\theta_2)\cos(\theta_4) + \sin(\theta_1)\sin(\theta_4)\cos(\theta_2))\sin(\theta_6))\cos(\theta_7) \\ + (-(\sin(\theta_1)\sin(\theta_2)\sin(\theta_4) + \sin(\theta_1)\cos(\theta_2)\cos(\theta_4))\sin(\theta_5) + \cos(\theta_1)\cos(\theta_5))\sin(\theta_7) \\ \\ ((\sin(\theta_2)\sin(\theta_4) + \cos(\theta_2)\cos(\theta_4))\sin(\theta_6) + (\sin(\theta_2)\cos(\theta_4) - \sin(\theta_4)\cos(\theta_2))\cos(\theta_5)\cos(\theta_6))\cos(\theta_7) \\ - (\sin(\theta_2)\cos(\theta_4) - \sin(\theta_4)\cos(\theta_2))\sin(\theta_5)\sin(\theta_7) \\ \\ 0 \end{bmatrix}$$

In [11]:
```
# End effector position
Xp=Transformation.col(3)
Xp.row_del(3)
```

In [12]:
```
# substituting values to end effector positions
Xp_copy=Xp.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

In [13]:
```
# Taking individual x,y,z for the end effector position
x=Xp[0]
y=Xp[1]
z=Xp[2]
```

In [14]:
```
# Setting up the Jacobian Matrix

J=Matrix(([diff(x,theta_1),diff(x,theta_2),diff(x,theta_4),diff(x,theta_5),diff(x,theta_
        [diff(y,theta_1),diff(y,theta_2),diff(y,theta_4),diff(y,theta_5),diff(y,theta_6)
        [diff(z,theta_1),diff(z,theta_2),diff(z,theta_4),diff(z,theta_5),diff(z,theta_6)
        [T_1[0,2],T_2[0,2],T_4[0,2],T_5[0,2],T_6[0,2],T_7[0,2]],
        [T_1[1,2],T_2[1,2],T_4[1,2],T_5[1,2],T_6[1,2],T_7[1,2]],
        [T_1[2,2],T_2[2,2],T_4[2,2],T_5[2,2],T_6[2,2],T_7[2,2]]))
```

In [15]:
```
# Substituting values to Jacobian Matrix
J=J.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
```

In [16]:
```
J
```

Out[16]:

$$\begin{bmatrix} 20.7\left(\left(-\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)\sin\left(\theta_{4}\right)-\sin\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\cos\left(\theta_{4}\right)\right)\cos\left(\theta_{5}\right)-\sin\left(\theta_{5}\right)\cos\left(\theta_{1}\right)\right)\sin\left(\theta_{6}\right) \\ +8.8\left(\left(-\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)\sin\left(\theta_{4}\right)-\sin\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\cos\left(\theta_{4}\right)\right)\cos\left(\theta_{5}\right)-\sin\left(\theta_{5}\right)\cos\left(\theta_{1}\right)\right)\cos\left(\theta_{6}\right) \\ +8.8\left(\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)\cos\left(\theta_{4}\right)-\sin\left(\theta_{1}\right)\sin\left(\theta_{4}\right)\cos\left(\theta_{2}\right)\right)\sin\left(\theta_{6}\right) \\ -20.7\left(\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)\cos\left(\theta_{4}\right)-\sin\left(\theta_{1}\right)\sin\left(\theta_{4}\right)\cos\left(\theta_{2}\right)\right)\cos\left(\theta_{6}\right)+8.8\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)\sin\left(\theta_{4}\right) \\ +38.4\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)\cos\left(\theta_{4}\right)+31.6\sin\left(\theta_{1}\right)\sin\left(\theta_{2}\right)-38.4\sin\left(\theta_{1}\right)\sin\left(\theta_{4}\right)\cos\left(\theta_{2}\right)+8.8\sin \\ \left(\theta_{1}\right)\cos\left(\theta_{2}\right)\cos\left(\theta_{4}\right)-8.8\sin\left(\theta_{1}\right)\cos\left(\theta_{2}\right) \\ 20.7\left(\left(\sin\left(\theta_{2}\right)\sin\left(\theta_{4}\right)\cos\left(\theta_{1}\right)+\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\cos\left(\theta_{4}\right)\right)\cos\left(\theta_{5}\right)-\sin\left(\theta_{1}\right)\sin\left(\theta_{5}\right)\right)\sin\left(\theta_{6}\right) \\ +8.8\left(\left(\sin\left(\theta_{2}\right)\sin\left(\theta_{4}\right)\cos\left(\theta_{1}\right)+\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\cos\left(\theta_{4}\right)\right)\cos\left(\theta_{5}\right)-\sin\left(\theta_{1}\right)\sin\left(\theta_{5}\right)\right)\cos\left(\theta_{6}\right) \\ +8.8\left(-\sin\left(\theta_{2}\right)\cos\left(\theta_{1}\right)\cos\left(\theta_{4}\right)+\sin\left(\theta_{4}\right)\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\right)\sin\left(\theta_{6}\right) \\ -20.7\left(-\sin\left(\theta_{2}\right)\cos\left(\theta_{1}\right)\cos\left(\theta_{4}\right)+\sin\left(\theta_{4}\right)\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\right)\cos\left(\theta_{6}\right)-8.8\sin\left(\theta_{2}\right)\sin\left(\theta_{4}\right)\cos \\ \left(\theta_{1}\right)-38.4\sin\left(\theta_{2}\right)\cos\left(\theta_{1}\right)\cos\left(\theta_{4}\right)-31.6\sin\left(\theta_{2}\right)\cos\left(\theta_{1}\right)+38.4\sin\left(\theta_{4}\right)\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right) \\ -8.8\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right)\cos\left(\theta_{4}\right)+8.8\cos\left(\theta_{1}\right)\cos\left(\theta_{2}\right) \\ 0 \\ \\ \\ \\ \\ \sin\left(\theta_{1}\right) \\ \\ -\cos\left(\theta_{1}\right) \\ \\ 0 \end{bmatrix}$$

In [17]:
```
q_current=Matrix(([0],
          [0],
          [pi/2],
          [0],
          [pi],
          [0]))
J_initial=J.subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]})
J_initial
```

Out[17]:

$$\begin{bmatrix} 0 & -49.2 & 17.6 & 0 & -8.8 & 0 \\ 67.9 & 0 & 0 & -8.8 & 0 & 0 \\ 0 & 67.9 & -59.1 & 0 & 20.7 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In [18]:
```
# The velocity matrix for end effector velocity
X_dot=Matrix(([0],
             [-10*sin(pi/2+ thetadot*t)*thetadot],
             [10*cos(pi/2 + thetadot*t)*thetadot],
```

```
              [0],
              [0],
              [0]))
X_dot=X_dot.subs({thetadot:(2*pi)/200})
X_dot
```

Out[18]:

$$\begin{bmatrix} 0 \\ -\frac{\pi \cos\left(\frac{\pi t}{100}\right)}{10} \\ -\frac{\pi \sin\left(\frac{\pi t}{100}\right)}{10} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

In [19]:

```
# mass for each link
m1=3.06+4.97
m2=3.228+0.646
m4=3.58
m5=1.225
m6=1.666
m7=0.73+0.735



# center of gravity for each link
cog1=Matrix([[0],[0],[33.30/2],[1]])
cog2=Matrix([[0],[0],[31.6/2],[1]])
cog4=Matrix([[0],[0],[5],[1]])
cog5=Matrix([[0],[38.4/2],[0],[1]])
cog6=Matrix([[0],[0],[0],[1]])
cog7=Matrix([[0],[0],[-10.7/2],[1]])

# gravitational accelaretion
G=9.8

CoG1 = cog1
CoG2 = T_1*cog2
CoG4 = T_4*cog4
CoG5 = T_5*cog5
CoG6 = T_6*cog6
CoG7 = T_7*cog7

# Potential Energy
P = m1*CoG1[2]*G + m2*CoG2[2]*G + m4*CoG4[2]*G + m5*CoG5[2]*G + m6*CoG6[2]*G + m7*CoG7[2

# g(theta) matrix
g = Matrix([[diff(P,theta_1)],[diff(P,theta_2)],[diff(P,theta_4)],[diff(P,theta_5)],[dif

# force matrix
F=Matrix([[-5],[0],[0],[0],[0],[0]])

# substituting values in g
g=g.subs({d_1:33.30,
    d_3:31.60,
    d_5:38.40,
    a_3:8.80,
    d_7:20.7})
g
```

Out[19]:

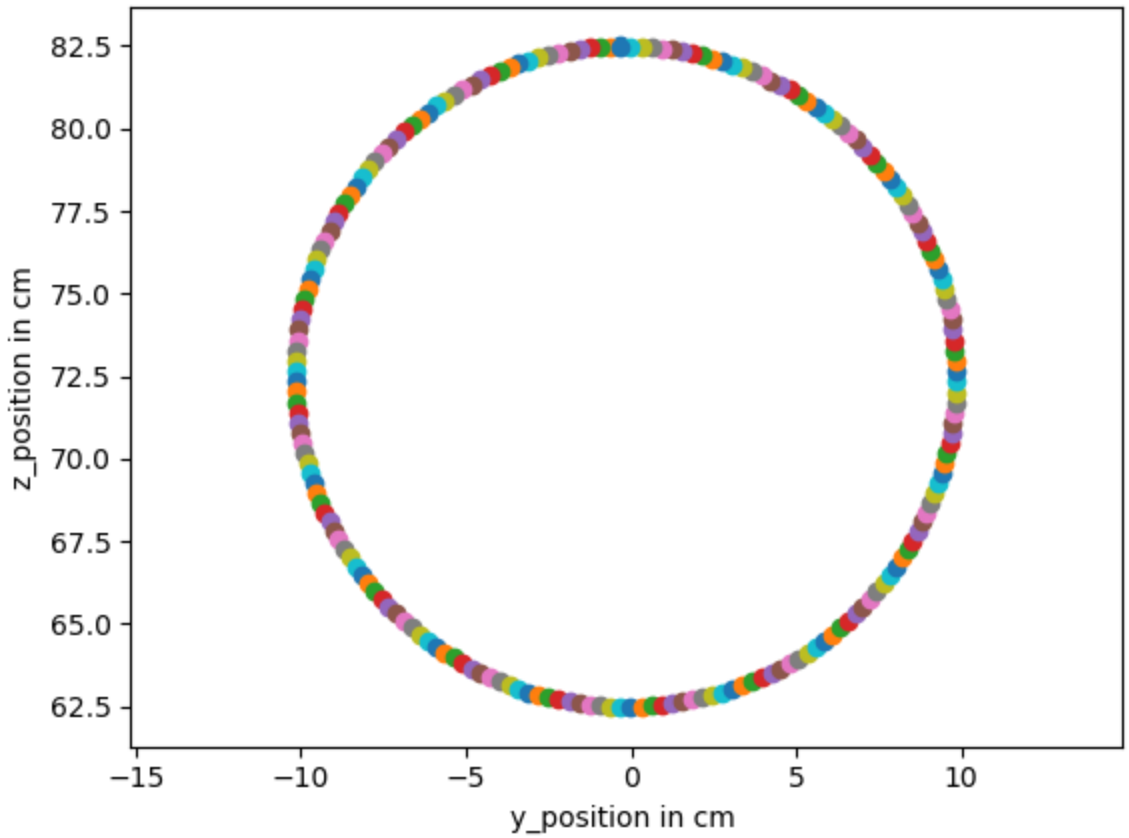
$$\begin{bmatrix} 0 \\ 373.99985\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_6\right)\cos\left(\theta_5\right) \\ +\,270.01744\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_5\right)\cos\left(\theta_6\right) \\ +\,270.01744\left(-\sin\left(\theta_2\right)\cos\left(\theta_4\right)+\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_6\right) \\ -\,373.99985\left(-\sin\left(\theta_2\right)\cos\left(\theta_4\right)+\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_6\right)-684.40064\sin\left(\theta_2\right)\sin\left(\theta_4\right) \\ -\,2045.16592\sin\left(\theta_2\right)\cos\left(\theta_4\right)-2457.62048\sin\left(\theta_2\right)+2045.16592\sin\left(\theta_4\right)\cos\left(\theta_2\right) \\ -\,684.40064\cos\left(\theta_2\right)\cos\left(\theta_4\right)+684.40064\cos\left(\theta_2\right) \\ 373.99985\left(-\sin\left(\theta_2\right)\sin\left(\theta_4\right)-\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_6\right)\cos\left(\theta_5\right) \\ +\,270.01744\left(-\sin\left(\theta_2\right)\sin\left(\theta_4\right)-\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_5\right)\cos\left(\theta_6\right) \\ +\,270.01744\left(\sin\left(\theta_2\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_6\right) \\ -\,373.99985\left(\sin\left(\theta_2\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_6\right)+684.40064\sin\left(\theta_2\right)\sin\left(\theta_4\right) \\ +\,2045.16592\sin\left(\theta_2\right)\cos\left(\theta_4\right)-2045.16592\sin\left(\theta_4\right)\cos\left(\theta_2\right)+684.40064\cos\left(\theta_2\right)\cos\left(\theta_4\right) \\ -373.99985\left(\sin\left(\theta_2\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_5\right)\sin\left(\theta_6\right) \\ -\,270.01744\left(\sin\left(\theta_2\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_5\right)\cos\left(\theta_6\right) \\ 373.99985\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\sin\left(\theta_6\right) \\ +\,270.01744\left(\sin\left(\theta_2\right)\sin\left(\theta_4\right)+\cos\left(\theta_2\right)\cos\left(\theta_4\right)\right)\cos\left(\theta_6\right) \\ -\,270.01744\left(\sin\left(\theta_2\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\sin\left(\theta_6\right)\cos\left(\theta_5\right) \\ +\,373.99985\left(\sin\left(\theta_2\right)\cos\left(\theta_4\right)-\sin\left(\theta_4\right)\cos\left(\theta_2\right)\right)\cos\left(\theta_5\right)\cos\left(\theta_6\right) \\ 0 \end{bmatrix}$$

In [20]:

```
list1=[]
x_plt=[]
y_plt=[]
z_plt=[]
i=0
j=0
x=[]
# The initial q_current values are given to us
q_current=Matrix(([0],
          [0],
          [pi/2],
          [0],
          [pi],
          [0]))
tau1=[]
tau2=[]
tau4=[]
tau5=[]
tau6=[]
tau7=[]

# Using while loop for
while(i<=200):
    Orientation=J
    print(1)
```

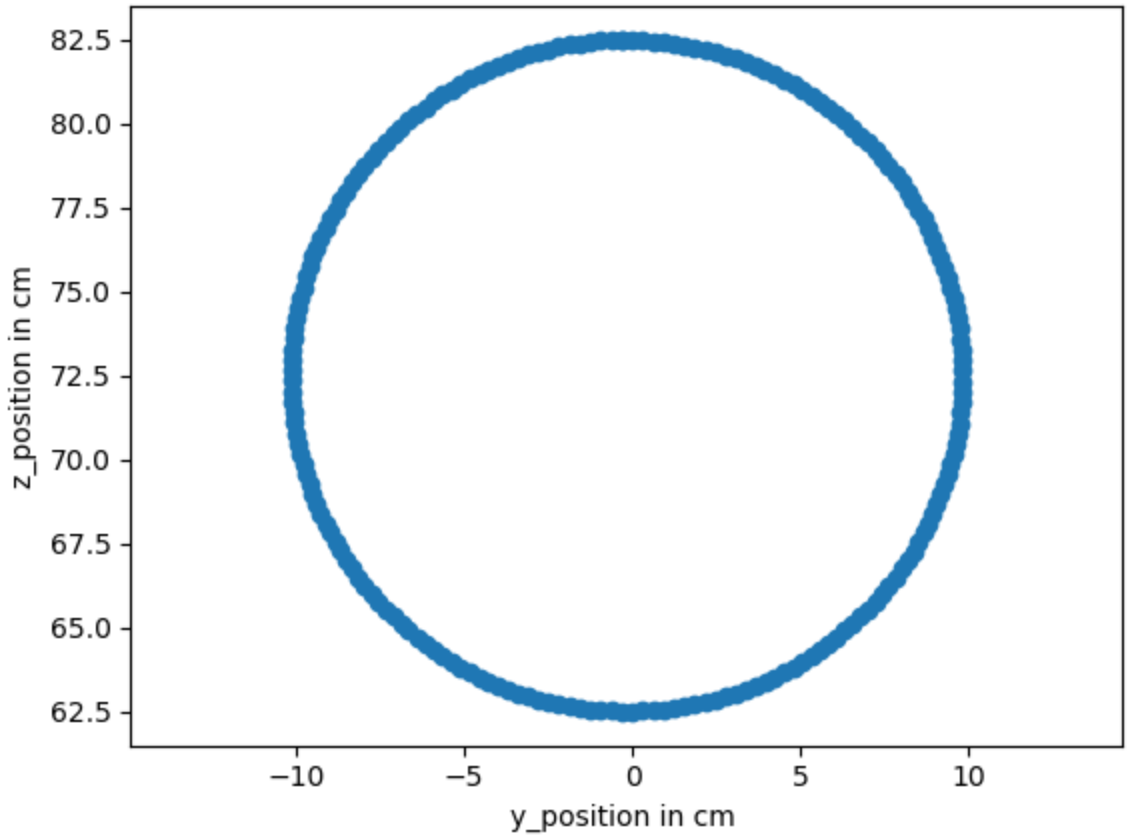
```
    Orientation=Orientation.subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]
    })

    Orientation_inverse=Orientation.evalf().inv()

    q_dot=Orientation_inverse*X_dot
    Z=q_dot
    Z=Z.subs({t:i}).evalf()
    q_current=q_current+Z*1

    list1.append(q_current)

    x_plt.append(Xp_copy[0].subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]}).evalf())

    y_plt.append(Xp_copy[1].subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]}).evalf())

    z_plt.append(Xp_copy[2].subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]}).evalf())

    i=i+1
    plt.axis("equal")
    plt.scatter(y_plt[j],z_plt[j])
    j+=1
    plt.xlabel("y_position in cm ")
    plt.ylabel("z_position in cm")
    x.append(67.9)

    tau=g.subs({theta_1:q_current[0],
    theta_2:q_current[1],
    theta_4:q_current[2],
    theta_5:q_current[3],
    theta_6:q_current[4],
    theta_7:q_current[5]}).evalf()-Orientation.T*F

    tau1.append(tau[0]/100)
    tau2.append(tau[1]/100)
    tau4.append(tau[2]/100)
    tau5.append(tau[3]/100)
    tau6.append(tau[4]/100)
    tau7.append(tau[5]/100)
```

```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```



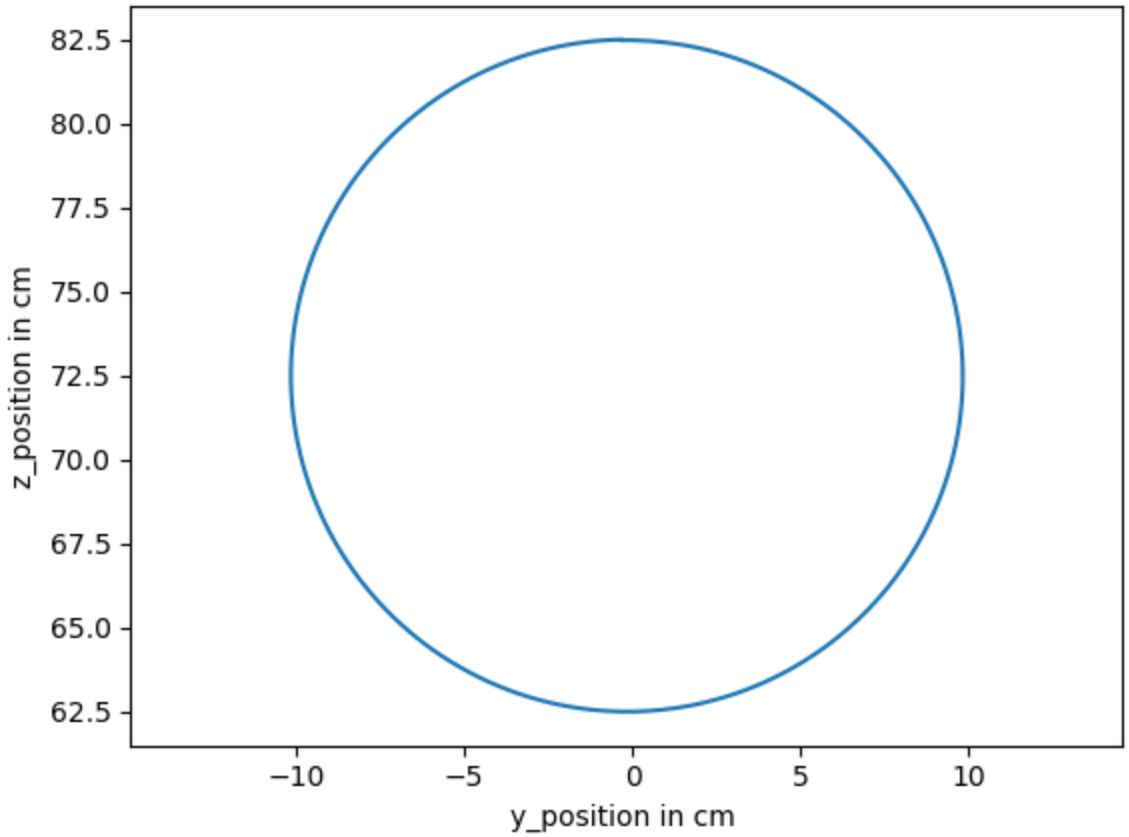

In [21]:
```
# Plot of the y and z scatter
plt.axis("equal")
plt.xlabel("y_position in cm ")
plt.ylabel("z_position in cm")
plt.scatter(y_plt, z_plt)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x7f9280c5b0a0>
```



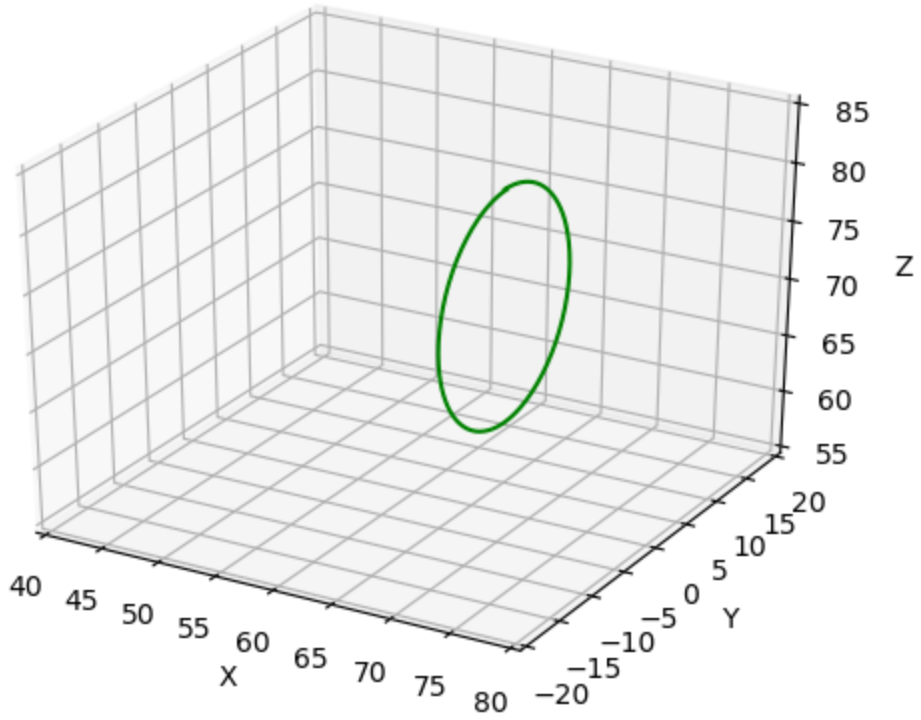

In [22]:
```
# normal plot for y and z
plt.axis("equal")
plt.xlabel("y_position in cm ")
plt.ylabel("z_position in cm")
plt.plot(y_plt,z_plt)
```

Out[22]:
```
[<matplotlib.lines.Line2D at 0x7f9280ea47c0>]
```




In [23]:
```
from mpl_toolkits.mplot3d import Axes3D
```

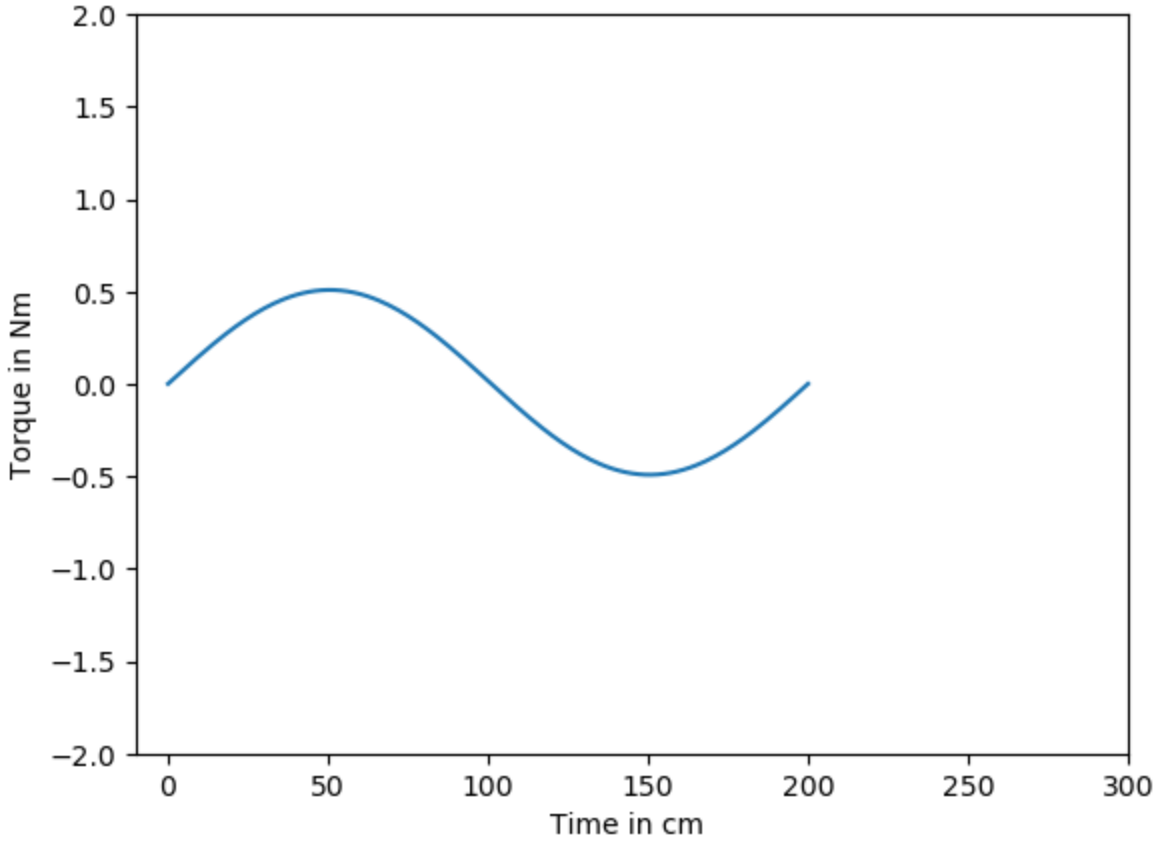
```
from matplotlib import pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(111,projection='3d')
ax.axes.set_xlim3d(left=40, right=80)
ax.axes.set_ylim3d(bottom=-20, top=20)
ax.axes.set_zlim3d(bottom=55, top=85)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.plot3D(x_plt,y_plt,z_plt, color="green")
plt.show()
```




In [24]:
```
time=[]
for i in range(0,201):
    time.append(i)
```

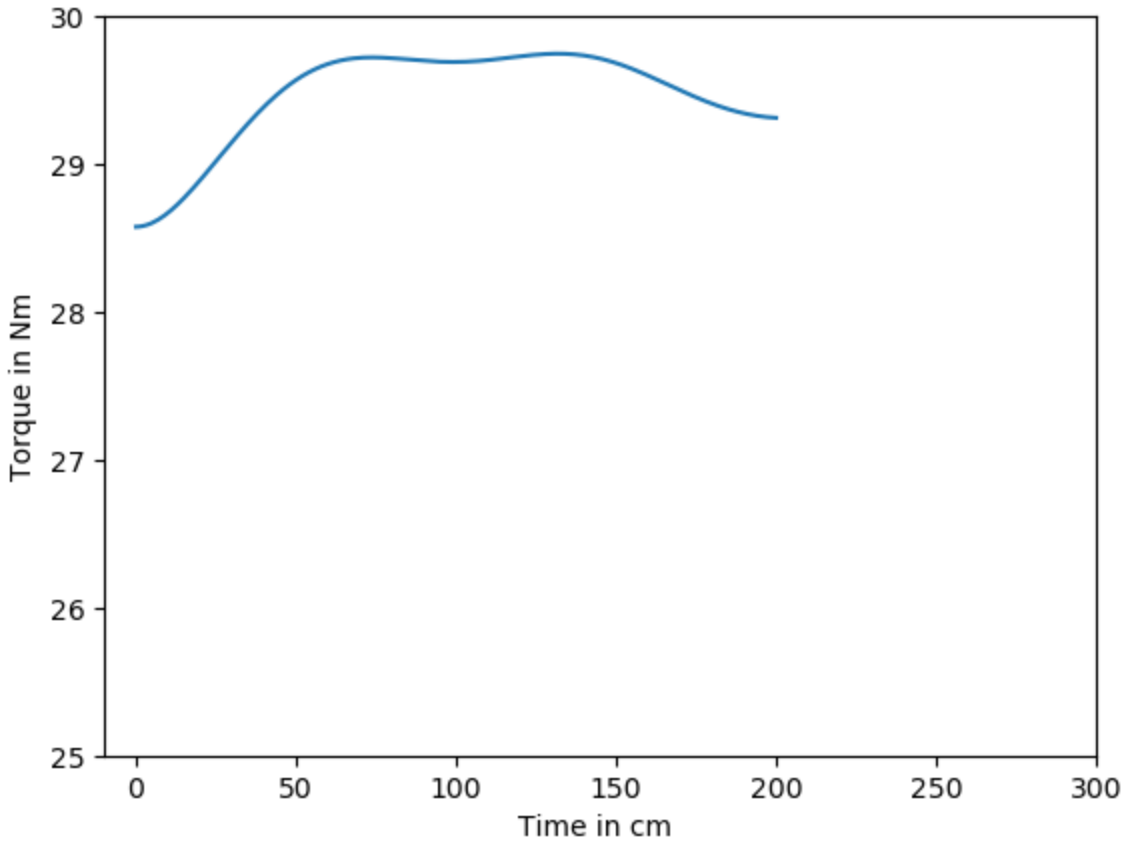
In [25]:
```
plt.xlabel("Time in cm ")
plt.ylabel("Torque in Nm")
plt.xlim(-10, 300)
plt.ylim(-2, 2)
plt.plot(time,tau1)
```

Out[25]:
```
[<matplotlib.lines.Line2D at 0x7f92809aa250>]
```



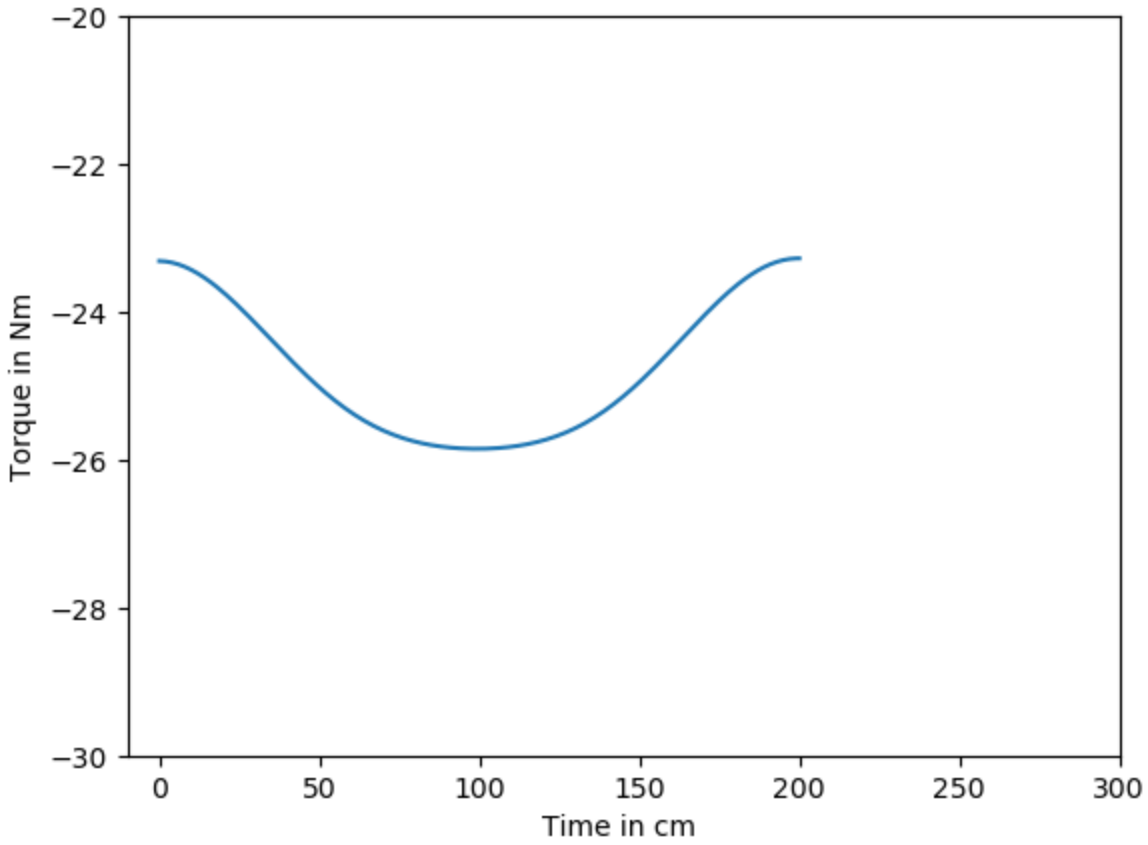

In [26]:
```
plt.xlabel("Time in cm ")
plt.ylabel("Torque in Nm")
plt.xlim(-10, 300)
plt.ylim(25,30)
plt.plot(time,tau2)
```

Out[26]: [<matplotlib.lines.Line2D at 0x7f9280976ee0>]



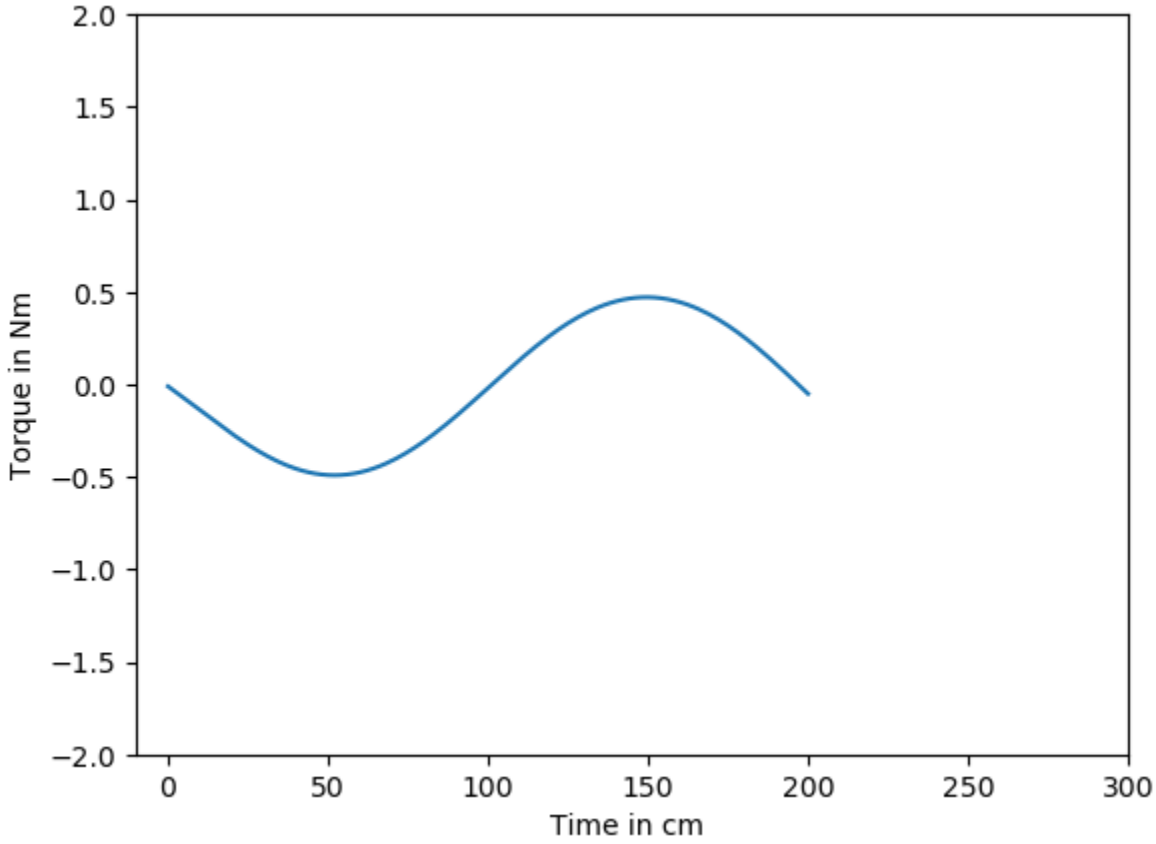
In [27]:
```
plt.xlabel("Time in cm ")
plt.ylabel("Torque in Nm")
plt.xlim(-10, 300)
plt.ylim(-30,-20)
plt.plot(time,tau4)
```

Out[27]:
```
[<matplotlib.lines.Line2D at 0x7f928093ca00>]
```



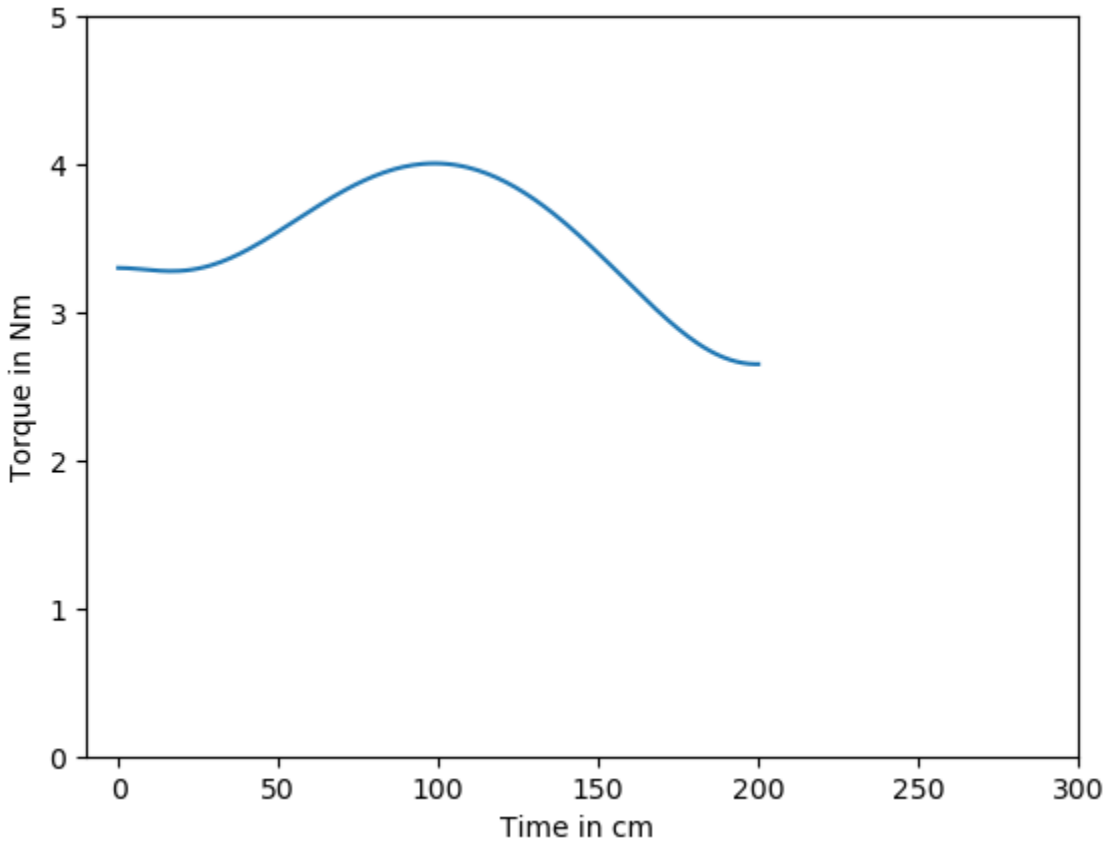

In [28]:
```
plt.xlabel("Time in cm ")
plt.ylabel("Torque in Nm")
plt.xlim(-10, 300)
plt.ylim(-2,2)
plt.plot(time,tau5)
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x7f92808efee0>]
```



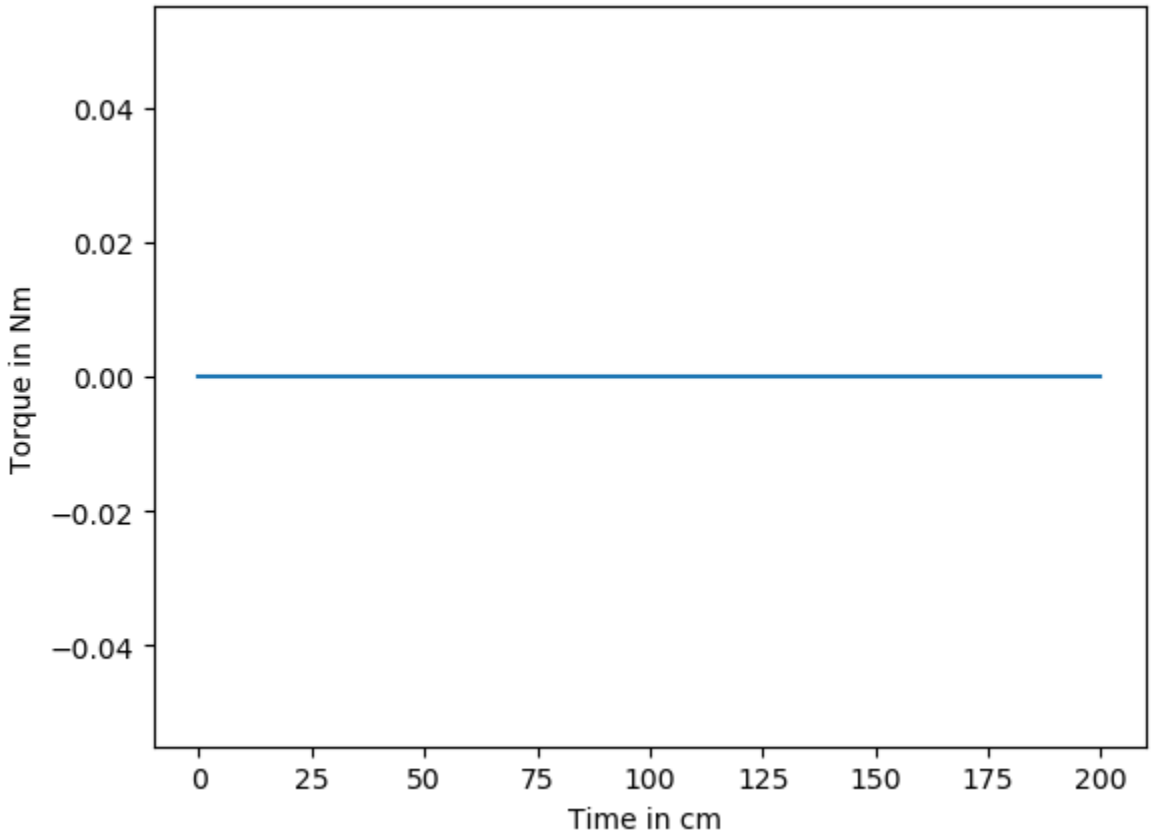

In [29]:
```
plt.xlabel("Time in cm ")
plt.ylabel("Torque in Nm")
plt.xlim(-10, 300)
plt.ylim(0,5)
plt.plot(time,tau6)
```

Out[29]:
```
[<matplotlib.lines.Line2D at 0x7f9288140ac0>]
```



In [30]:
```
plt.xlabel("Time in cm ")
plt.ylabel("Torque in Nm")
plt.plot(time,tau7)
```

Out[30]:
```
[<matplotlib.lines.Line2D at 0x7f9288f54820>]
```




In [ ]: