Robotics Assignment 1

15 November 2020

Group: 1_Tue_I

Author	**Student ID**
Prathamesh	408508
Vasilije Rakcevic	414014
Jingsheng Lyu	398756

Implementation Table

Student Name	A1	B1	B2	B3	B4	B5	B6	B7
Prathamesh More	x	x	x	x	x	x	x	x
Vasilije Rakcevic	x	x	x	x	x	x	x	x
Jingsheng Lyu	x	x	x	x	x	x	x	x

Instructor: Prof. Dr. Oliver Brock

Teaching assistant: Aditya Bhatt

A Calculation

1. Compute the gravity vector $G(q1,\ q2,\ q3) =\ [?\ ?\ ?]^T$ which estimates the torque ($\frac{kg\cdot m^2}{s^2}$) caused by gravity at each joint.

$$\tau_1 = -\begin{pmatrix} m_1 g r_1 c_1 \\ 0 \\ 0 \end{pmatrix}$$

$$\tau_2 = -\begin{pmatrix} m_2 g\left(L_1 c_1 + r_2 c_{12}\right) \\ m_2 g r_2 c_{12} \\ 0 \end{pmatrix}$$

$$\tau_3 = -\begin{pmatrix} m_3 g\left(L_1 c_1 + L_2 c_{12} + r_3 c_{123}\right) \\ m_3 g\left(L_2 c_{12} + r_3 c_{123}\right) \\ m_3 g r_3 c_{123} \end{pmatrix}$$

With

$$c_1 = cos\theta_1$$

$$c_{12} =- cos\,(\theta_1 + \theta_2 + 90)$$

$$c_{123} =- cos\,(\theta_1 + \theta_2 + \theta_3 + 90)$$

Finally

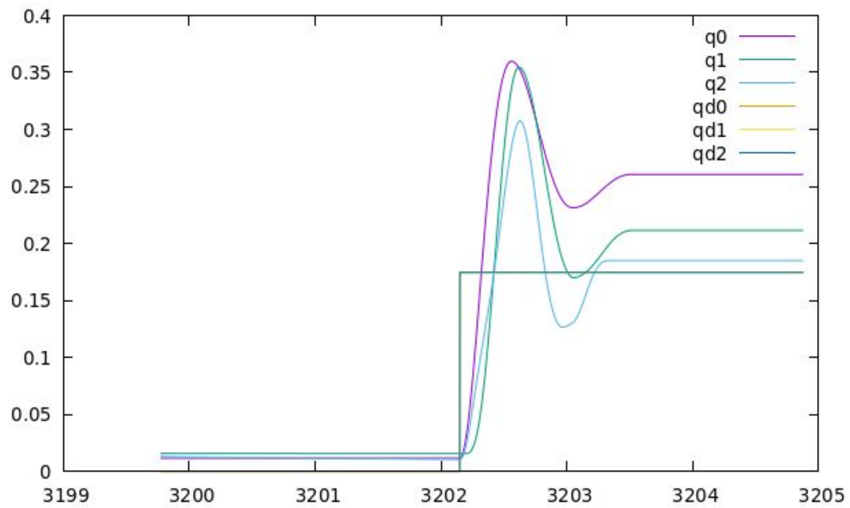
$$G(q_1, q_2, q_3) = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \left(\tau_1 + \tau_2 + \tau_3 \right) = -\begin{pmatrix} m_1 c_1 r_1 + m_2(L_1 c_1 + r_1 c_{12}) + m_3(L_1 c_1 + L_2 c_{12} + r_3 c_{123}) \\ m_2 r_2 c_{12} + m_3(L_2 c_{12} + r_3 c_{123}) \\ m_3 r_3 c_{123} \end{pmatrix} g$$

B Implementations

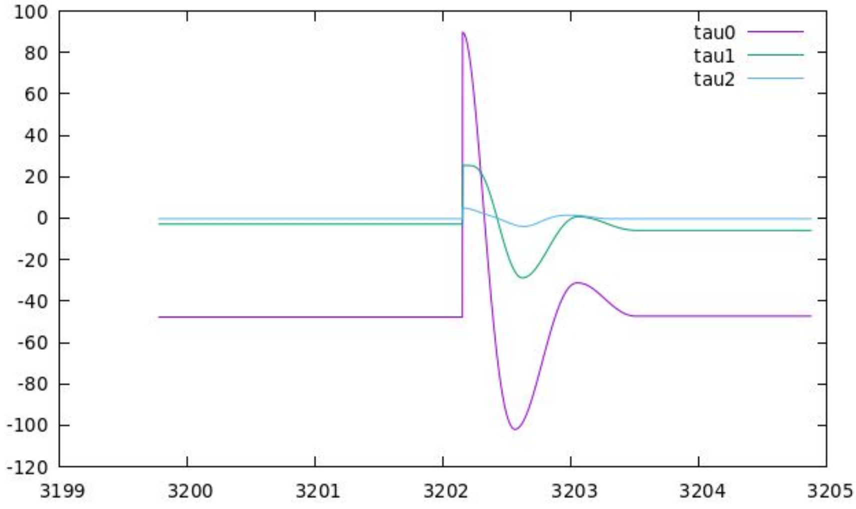
1. P-controller - njmoveControl()

   Implement a P-controller for the joints 2, 3 and 5 of the Puma in njmoveControl().

	$q1$	$q2$	$q3$
Kp	550	160	30



Y axes - Angle [rad]; X axes - Time
q0,q1,q2 - Angles of joints 1,2,3 respectively
qd0,qd1,qd2 - Desired angles of joints respectively



Y axis - Torque [Nm]; X axes - Time
tau0,tau1,tau2 - torques produced by the joints 1,2,3 respectively

2. Tune the controllers,

   2.1. What kind of behaviors do you observe with different gains?

   With different gains we observe the change in the nature of controllers. With Higher Kp we found that the oscillations increased but with the lower kp, the q did not reach qd. Hence an optimised gain was selected so that we get the overshoot and oscillations in acceptable limits.

   2.2. Why are well tuned gains different for each joint?

   Each joint is associated with different links which have different mass and dimensional properties. Hence reaction to the controller will vary based on its location on the Robotic Arm. Therefore, all have to be tuned individually.

3. Plot the desired angles qd, the actual joint angles q, and the applied torques

   The Plots are in B.1

4. Calculate the gravity vector in PreprocessControl()

```
//Compute g123 here!
 double r1, r2, r3, l1, l2, l3,
          m1, m2, m3, g, c1, c12, c123;

 r1 = R2;     r2 = 0.189738;     r3 = R6;
 l1 = L2;     l2 = L3;           l3 = L6;
 m1 = M2;     m2 = M3 + M4 + M5;m3 = M6;
 g = -9.81;
 c1 = cos(q1);   c12 = -cos(q1 + q2 + 3.14/2);     c123 = -cos(q1 + q2 + 3.14/2 + q3);
 // printV("gv.q",gv.q);
 // printV("c1",c1);

 g123[0] = -(r1*c1*m1 + m2*(l1*c1 + r2*c12) + m3*(l1*c1 + l2*c12 + r3*c123))*g;
 g123[1] = -(r2*c12*m2 + m3*(l2*c12 + r3*c123))*g;
 g123[2] = -r3*c123*m3*g;

if (gv.dof == 3) {
    gv.G[0] = g123[0];
    gv.G[1] = g123[1];
    gv.G[2] = g123[2];
} else if (gv.dof == 6) {
    gv.G[1] = g123[0];
    gv.G[2] = g123[1];
    gv.G[4] = g123[2];
}
```

5. floatControl()

5.1. the robot should keep its current pose and only move when an external force is applied. So the external force is the same with the opposite value of gravity.

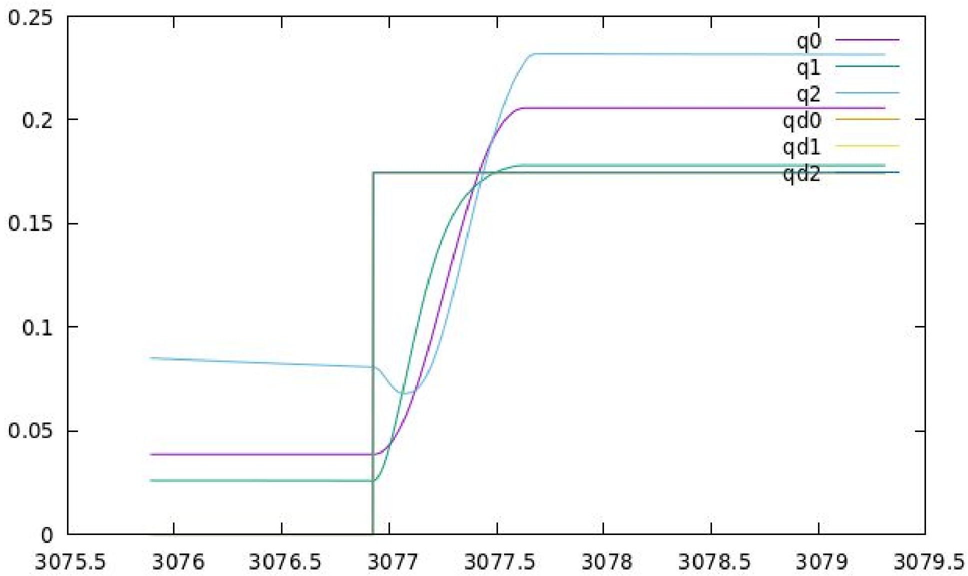
$$\tau = -G$$

7. njgotoControl()

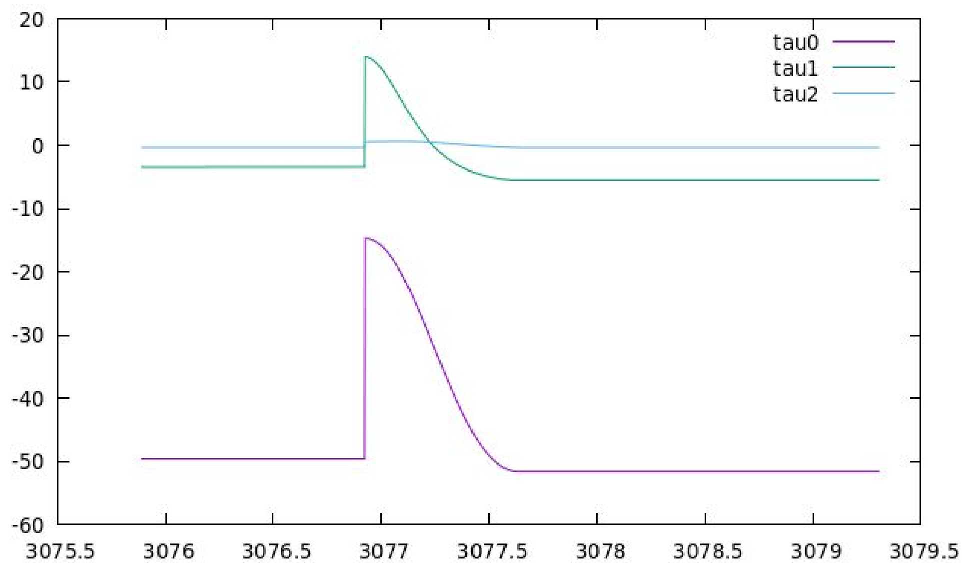
7.1. P Controller

$$\tau = -K_p\left(q - q_d\right) - G$$

	P_1	P_2	P_3
K_p	200	100	6



Y axes - Angle [rad]; X axes - Time
q0,q1,q2 - Angles of joints 1,2,3 respectively
qd0,qd1,qd2 - Desired angles of joints respectively



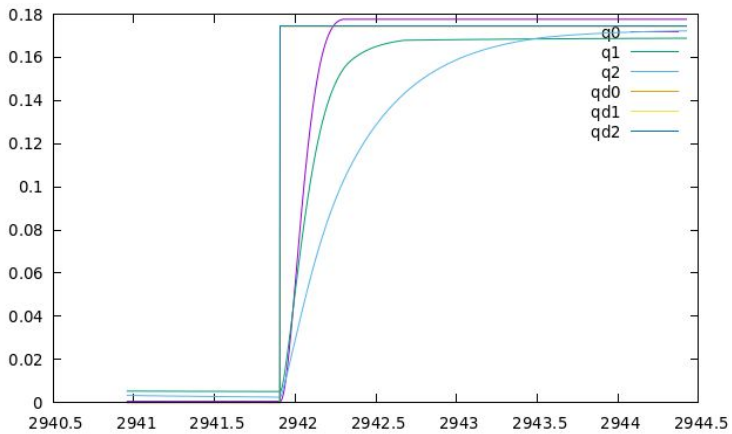
Y axis - Torque [Nm]; X axes - Time
tau0,tau1,tau2 - torques produced by the joints 1,2,3 respectively

8. jgotoControl()

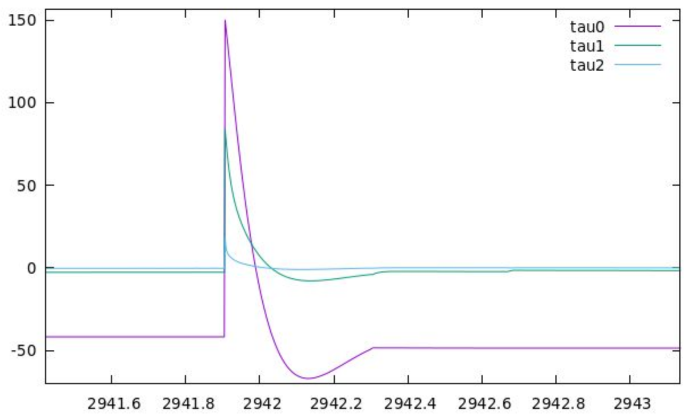
8.1. PD Controller

$$\tau = -K_p\left(q - q_d\right) - K_v \cdot \left(\dot{q}_d - \dot{q}\right) - G$$

	K_1	K_1	K_1
K_p	1100	500	110
K_d	120	90	50



Y axes - Angle [rad]; X axes - Time
q0,q1,q2 - Angles of joints 1,2,3 respectively
qd0,qd1,qd2 - Desired angles of joints respectively



Y axis - Torque [Nm]; X axes - Time
tau0,tau1,tau2 - torques produced by the joints 1,2,3 respectively

Why can the gains kp now be higher compared to the P-controller?

Ans: The Kp is a proportional gain, which increases the speed of joint reaction. But because of this reason, the joints overshoot and oscillate unless brought to steady state error. When we implement the D-Controller, this high speed reaction is detected and controlled. The D-controller in short detects the high slope from the P-controller and tries to control it. Hence we could use higher Kp as Kv will be able to compensate for it.

Note

Torque limits for 3 Joints configuration.

Joint:	1	2	3
τ_{max}	156.4Nm	89.4Nm	21.2Nm