ME462 DYNAMICAL AND CONTROL OF ROBOTICS

Project 2

Control Design Using Computed Torque Technique for RRR Manipulator

Evrim Selin ALTINKAYNAK

10.06.2015

The task of this project is to design the control system using computed torque technique for the robot's 2nd and 3rd joints so that the robot will be able to complete the task of drawing a 10 cm straight line that is parallel to the x-axis while keeping the orientation of the end-effector at 90° in 2 seconds.

For this project, the RRR manipulator designed using solidworks was exported to matlab. The created SimMechanics model is given in Fig. 1.

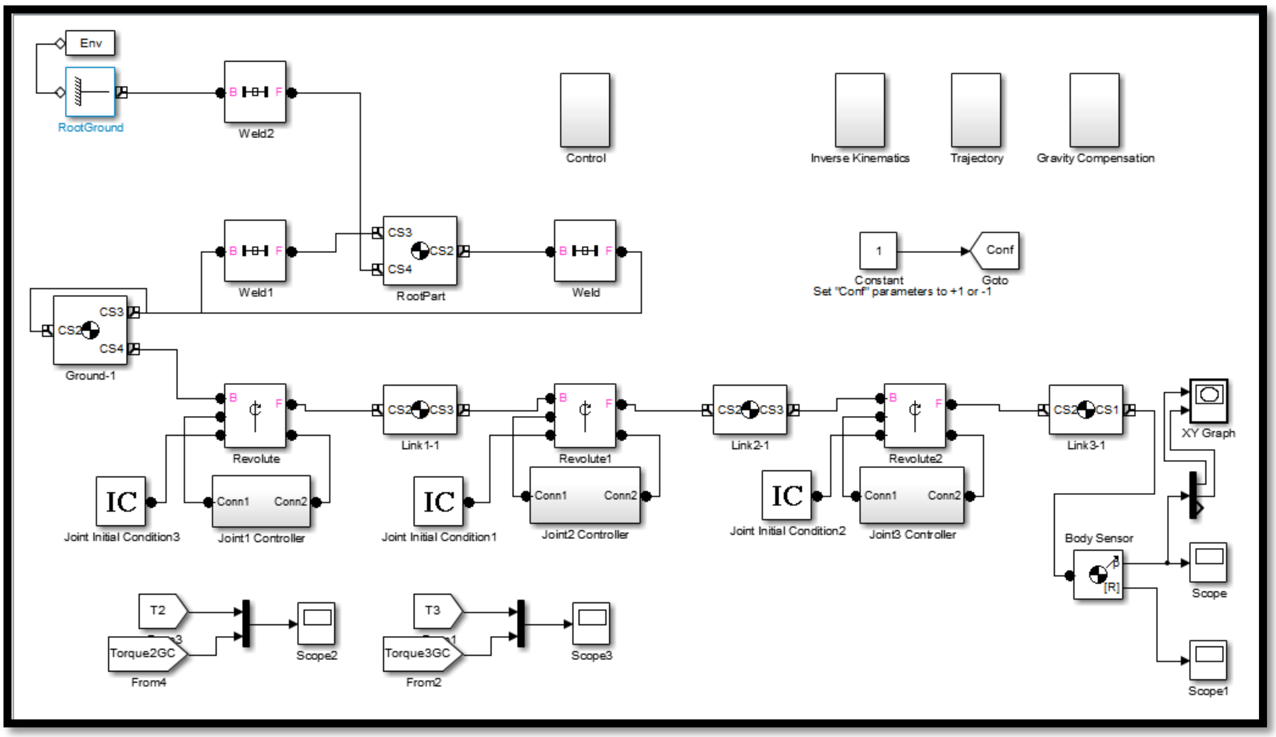

Figure 1: RRR SimMechanics model with its controllers, inverse kinematics, trajectory, and control

The 3 revolute joints can be easily seen in Fig.1. While exporting the manipulator, the mass and inertia information for each link between the joints is also exported to the SimMechanics model. The total torque which is the sum of the computed torque and gravity compensation torque is compared with the gravity compensation torque. This comparison is performed for the second and third joints of the manipulator. This is observed using a scope in Simulink. An initial condition was given looking at the joint angles from the excel file.

For the first joint of the manipulator, motion mode was used as the joint actuator. For the other two joints, the generalized forces were used as the joint actuators. These controllers can be seen in the following 3 figures.

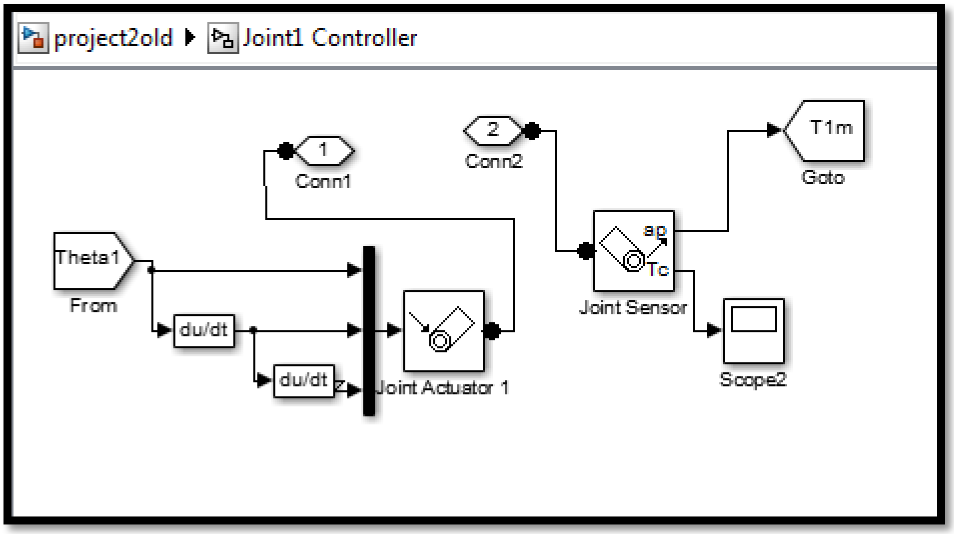

Figure 2: Joint 1 controller with motion mode

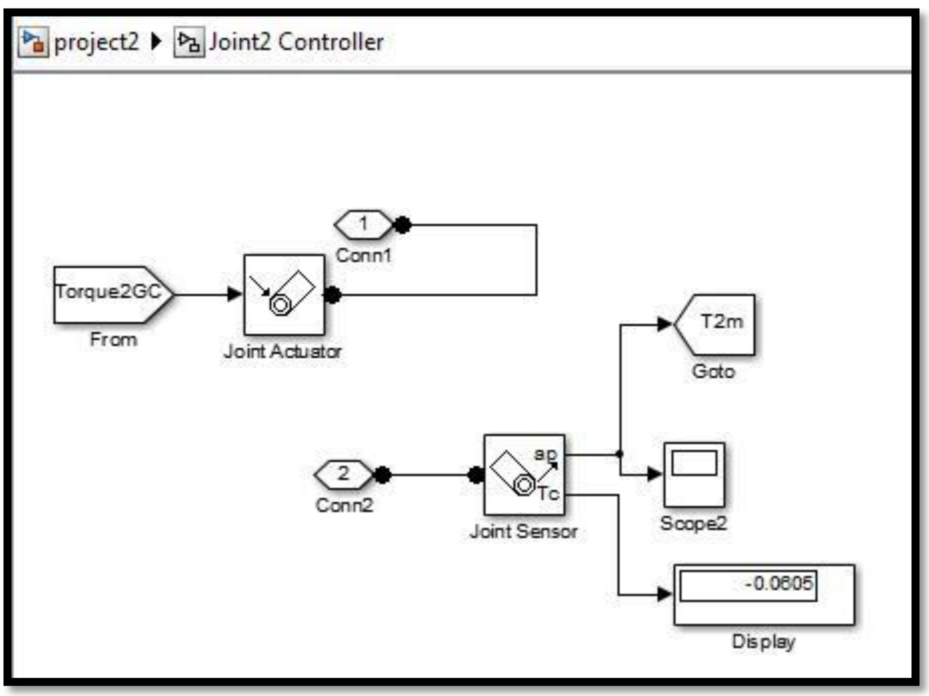

Figure 3: Joint 2 controller with generalized forces

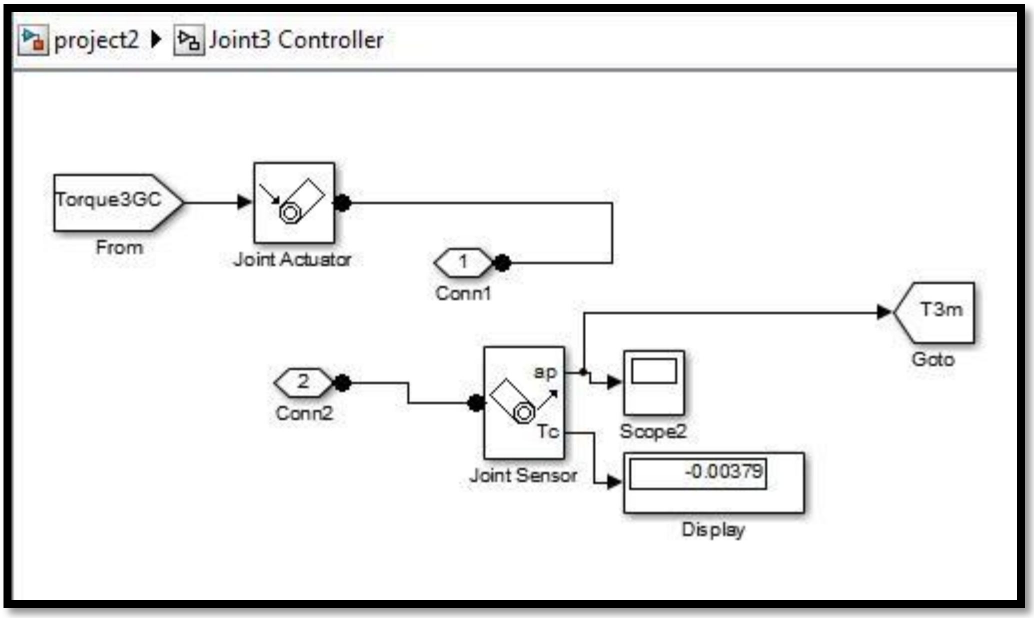

Figure 4: Joint 3 controller with generalized forces

The robot was required to go a straight line of 10cm on the x-axis in just 2 seconds. The end-effector orientation was kept to 90^o. The starting point for the robot was chosen to be (200,200). The trajectory assigned in Matlab is shown as in below.

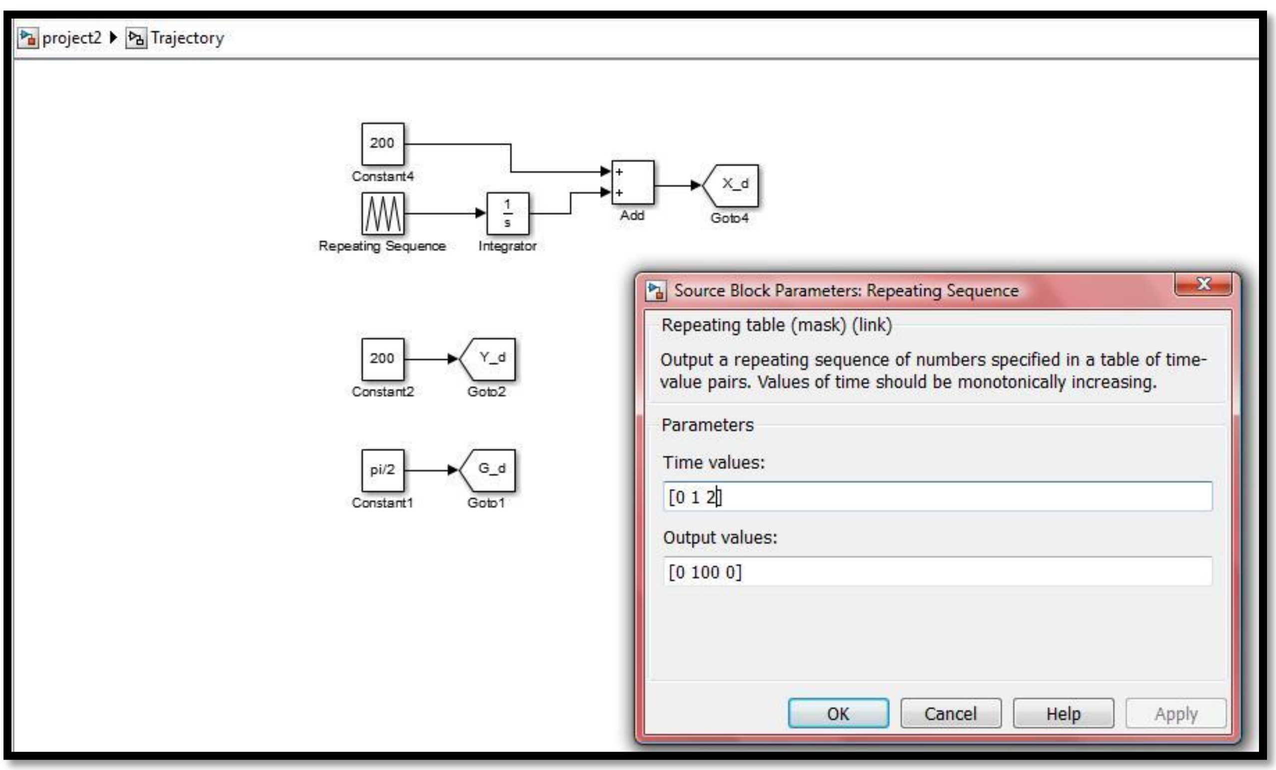

Figure 5: Trajectory of manipulator

The velocity of the manipulator is shown in Fig. 6.

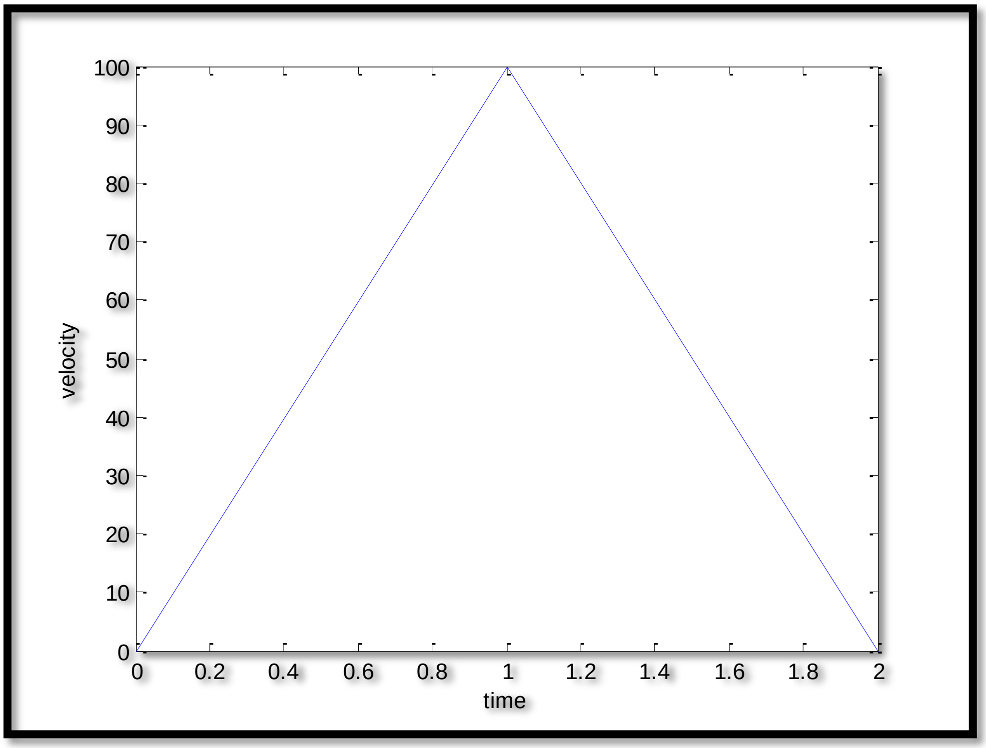

Figure 6: Velocity change in time for path following

The robot moved 10cm in the x-axis and kept the y-axis constant. This is seen in the xy plot below.

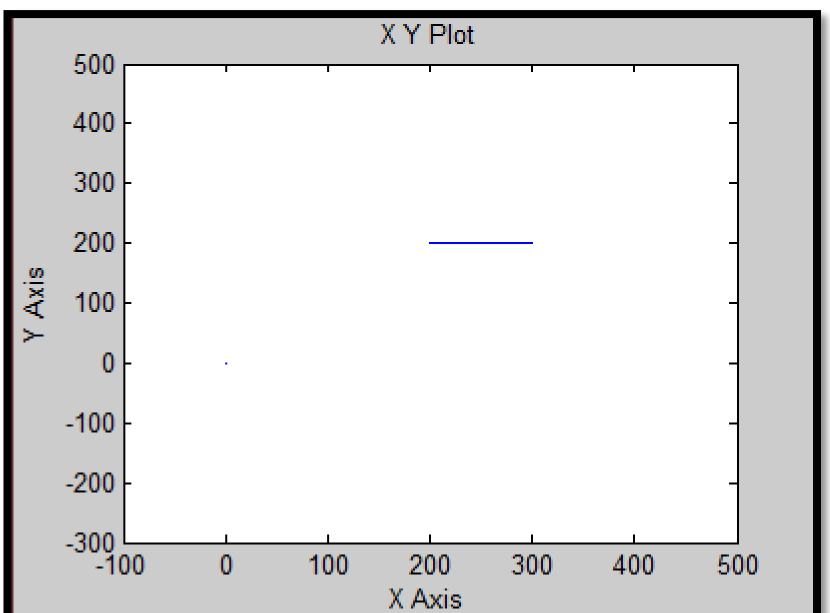

Figure 7: XY plot of robot

The inverse kinematic equations are the same as in the project 1.

$$Xe = a1 * \cos(q1) + a2 * \cos(q1 + q2) + a3 * \cos(q1 + q2 + q3) \quad (1)$$

$$Ye = a1 * \sin(q1) + a2 * \sin(q1 + q2) + a3 * \sin(q1 + q2 + q3) \quad (2)$$

Equations (1) and (2) are the end-effector positions. We first take the squares of both equation (1) and (2) separately and then sum them. The result obtained is shown below.

$$(Xe - a3 * \cos(q1 + q2 + q3))^2 + (Ye - a3 * \sin(q1 + q2 + q3))^2 = a1^2 + a2^2 + 2 * a1 * a2 * \cos(q2) = \beta \quad (3)$$

$$\sin(q2) = \pm\sqrt[2]{1 - \beta^2} \quad (4)$$

$$q2 = cos^{-1}(\frac{X^2+Y^2}{2*a1*a2}) \quad (5)$$

Equation (5) is found from equation (3) by first taking the first two terms to be X and Y and then leaving the second joint angle alone.

To find the first joint angle q1 we do the operations shown below.

$$X = a1 * \cos(q1) + a2 * \cos(q1) * \cos(q2) - a2 * \sin(q1) * \sin(q2) \quad (6)$$

$$Y = a1 * \sin(q1) + a2 * \sin(q1) * \cos(\mathrm{q2}) + \mathrm{a2} * \cos(\mathrm{q1}) * \sin(\mathrm{q2}) \quad (7)$$

We then write equations (6) and (7) in matrix form.

$$\begin{pmatrix} a1 + a2 * \cos(q2) & -a2 * \sin(q2) \\ a2 * \sin(q2) & a1 + a2 * \cos(q2) \end{pmatrix} * \begin{pmatrix} \cos(q1) \\ \sin(q1) \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \quad (8)$$

We then use Cramer's rule to obtain q1.

$$\cos(q1) = \frac{\begin{vmatrix} X & -a2*\sin(q2) \\ Y & a1+a2*\cos(q2) \end{vmatrix}}{\begin{vmatrix} a1+a2*\cos(q2) & -a2*\sin(q2) \\ a2*\sin(q2) & a1+a2*\cos(q2) \end{vmatrix}} \quad (9)$$

$$\sin(q1) = \frac{\begin{vmatrix} a1+a2*\cos(q2) & X \\ a2*\sin(q2) & Y \end{vmatrix}}{\begin{vmatrix} a1+a2*\cos(q2) & -a2*\sin(q2) \\ a2*\sin(q2) & a1+a2*\cos(q2) \end{vmatrix}} \quad (10)$$

We knew q1+q2+q3 and now know q1 and q2, so now we can find q3.

Then, I took account the gravitational effects. The torques were calculated as in the following.

$$Y1 = L1 * \sin(q1) \ (11)$$

$$Y2 = a1 * \sin(q1) + L2 * \sin(q1 + q2) \ (12)$$

$$Y3 = a1 * \sin(q1) + a2 * \sin(q1 + q2) + L3 * \sin(q1 + q2 + q3) \ (13)$$

where q1, q2, and q3 are the joint variables of the manipulator.

$$\mathrm{\delta Y1 = L1 * \cos(q1) * \delta q1} \quad (14)$$

$$\mathrm{\delta Y2 = a1 * \cos(q1) * \delta q1 + L2 * \cos(q1 + q2) * (\delta q1 + \delta q2)} \ (15)$$

$$\mathrm{\delta Y3 = a1 * \cos(q1) * \delta q1 + a2 * \cos(q1 + q2) * (\delta q1 + \delta q2) + L3 * \cos(q1 + q2 + q3) * (\delta q1 + \delta q2 + \delta q3)} \ (16)$$

$$\begin{pmatrix} \delta Y1 \\ \delta Y2 \\ \delta Y3 \end{pmatrix} = \begin{pmatrix} L1 * \cos(q1) & 0 & 0 \\ a1 * \cos(q1) + L2 * \cos(q1 + q2) & L2 * \cos(q1 + q2) & 0 \\ a1 * \cos(q1) + a2 * \cos(q1 + q2) + L3 * \cos(q1 + q2 + q3) & a2 * \cos(q2) + L3 * \cos(q1 + q2 + q3) & a3 * \cos(q1 + q2 + q3) \end{pmatrix} * \begin{pmatrix} \delta q1 \\ \delta q2 \\ \delta q3 \end{pmatrix} \quad (17)$$

$$\delta \bar{Y} = \hat{J} * \delta \bar{q} \ (18)$$

$$\bar{F}^t * \hat{J} * \delta \bar{q} = \bar{\tau}^t * \delta \bar{q} \ (19)$$

$$\hat{J}^t * \bar{F} = \bar{\tau} \ (20)$$

The t’s above the force and torques represent the transpose.

$$\bar{\tau} = -\begin{pmatrix} L1 * \cos(q1) & a1 * \cos(q1) + L2 * \cos(q1 + q2) & a1 * \cos(q1) + a2 * \cos(q1 + q2) + L3 * \cos(q1 + q2 + q3) \\ 0 & L2 * \cos(q1 + q2) & a2 * \cos(q1 + q2) + L3 * \cos(q1 + q2 + q3) \\ 0 & 0 & L3 * \cos(q1 + q2 + q3) \end{pmatrix} * \begin{pmatrix} -m1 * g \\ -m2 * g \\ -m3 * g \end{pmatrix} \ (21)$$

The parameters in the equations are shown in Fig. 8. This result was obtained using the virtual work principle. This analysis is known as the quasi-static force/torque analysis.

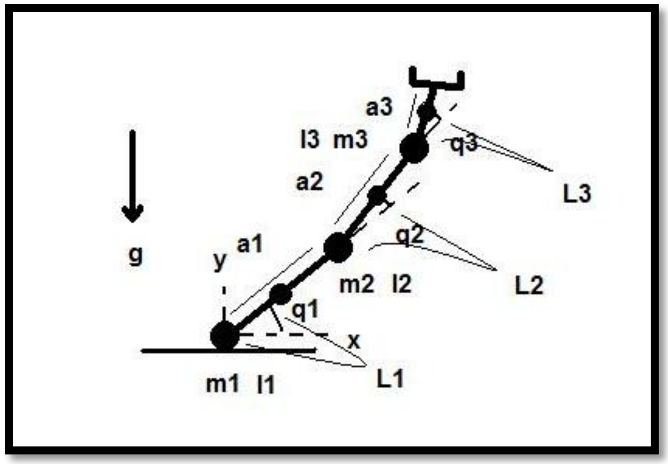

Figure 8: RRR manipulator

Next the dynamic analysis was done using the Lagrange's method. The Lagrangian term is defined as:

$$L = K - U \text{ (22)}$$

Where K is the kinetic energy and U is the potential energy. The Lagrangian equation is defined as below.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q_k}}\right) - \frac{\partial L}{\partial q_k} = Q_k \text{ (23)}$$

;k=1,2…n for n-dof manipulator.

The general equation of motion is as below.

$$\widehat{M}(q) * \dot{q} + \bar{B}(q, \dot{q}) + \bar{G}(q) = \tau_a + \tau_{ext} \text{ (24)}$$

Where M(q) is the generalized mass/inertia matrix. This matrix is 3x3 matrix for a 3-dof manipulator. B(q,$\dot{q}$) is matrix which contains the Centripetal and Coriolis terms. G(q) is the gravitational effect. τ_{ext} is the external torques.

In the computed torque method, the torque is computed and fed back to the joint actuators. The block diagram of this method can be shown as in below.

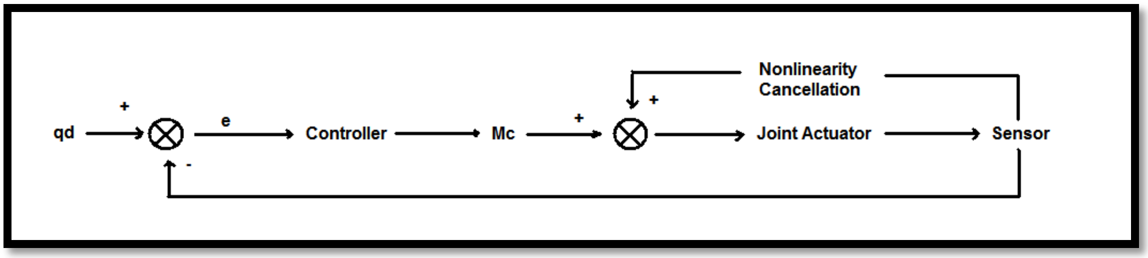

Figure 9: Computed Torque Method

The tuning of the controller is done by changing the gains of errors and its derivatives. This gains that were changed can be seen in Fig. 10.

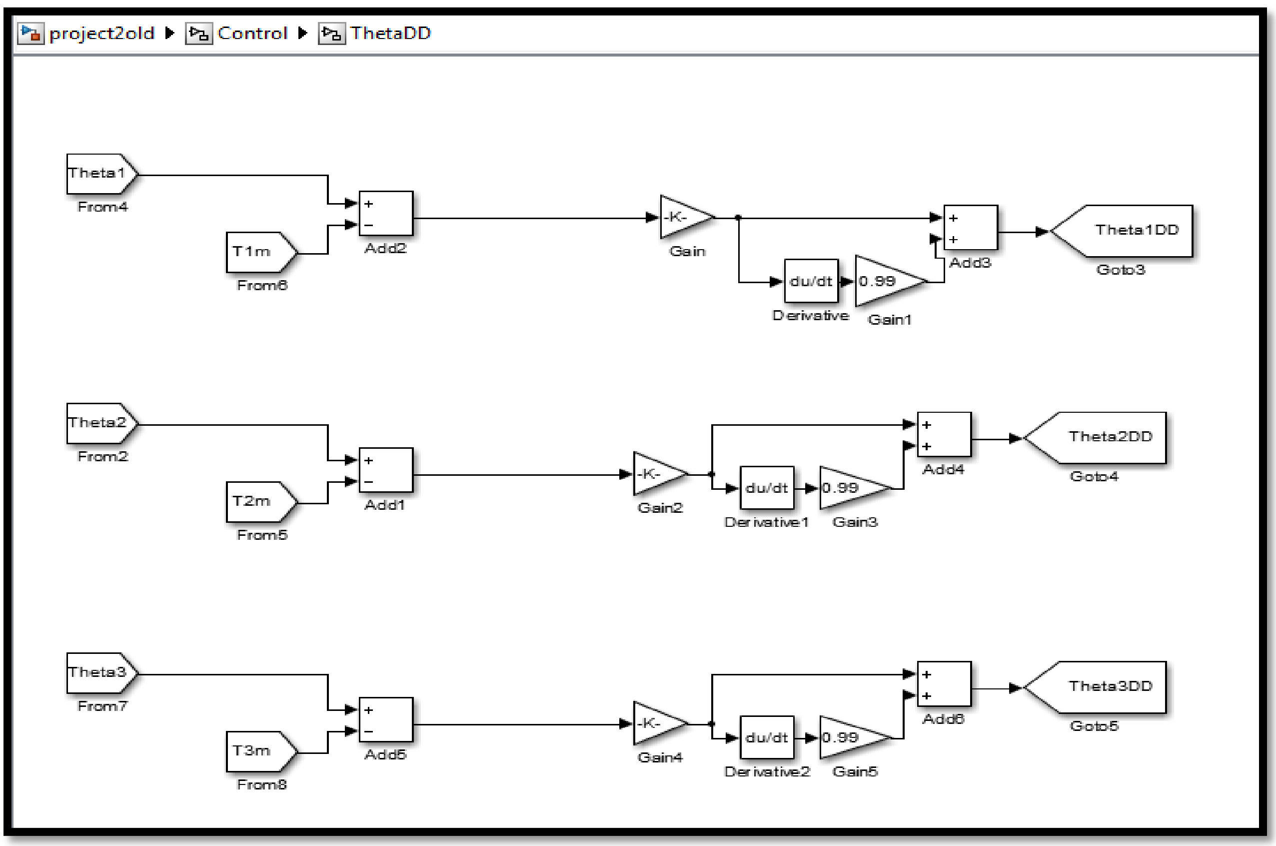

Figure 10: Calculation of the acceleration vector from the joint angles

The calculated torques were found with the equations below.

$\tau_{22} = \left(I3 + m3 * (a2^2 + L3^2 + a1 * a2 * \cos(q2) + a1 * L3 * \cos(q2 + q3) + 2 * a2 * L3 * \cos(q3))\right) * \ddot{q}1 + \left(I3 + m3 * (a2^2 + L3^2 + 2 * a2 * L3 * \cos(q3))\right) * \ddot{q}2 + \left(I3 + m3 * (L3^2 + a2 * L3 * \cos(q3))\right) * \ddot{q}3 - 2 * \dot{q}1 * \dot{q}3 * m3 * a2 * L3 * \sin(q3) - 2 * \dot{q}2 * \dot{q}3 * m3 * a2 * L3 * \sin(q3) - \dot{q}3^2 * m3 * a2 * L3 * \sin(q3) + \dot{q}1^2 * (m3 * a1 * a2 * \sin(q2) + m3 * a1 * L3 * \sin(q2 + q3))$ (25)

$\tau_{23} = \left(I1 + m2 * (L2^2 + a1 * L2 * \cos(q2))\right) * \ddot{q}1 + (I2 + m2 * L2^2) * \ddot{q}2 + \dot{q}1^2 * m2 * a1 * L2 * \sin(q2)$ (26)

$\tau_2 = \tau_{22} + \tau_{23}$ (27)

$\tau_3 = \tau_{33} = \left(I3 + m3 * (L3^2 + a1 * L3 * \cos(q2 + q3) + a2 * L3 * \cos(q3))\right) * \ddot{q}1 + \left(I3 + m3 * (L3^2 + a2 * L3 * \cos(q3))\right) * \ddot{q}2 + (I3 + m3 * L3^2) * \ddot{q}3$ (28)

I put some results to show on how the gain has an effect on the manipulator.

Results:

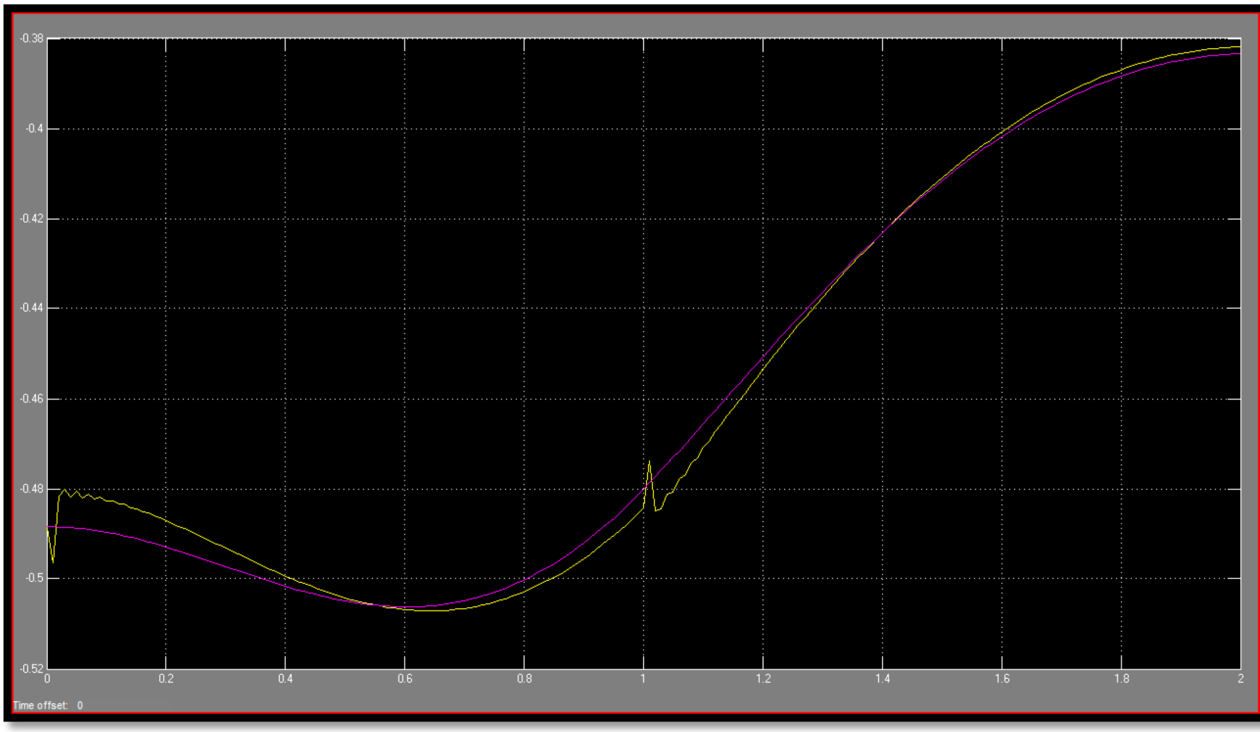

Figure 11: Torque2 for kp1=150 and kv1=0.99, kp2=200 and kv2=0.99, kp3=20 and kv3=0.99

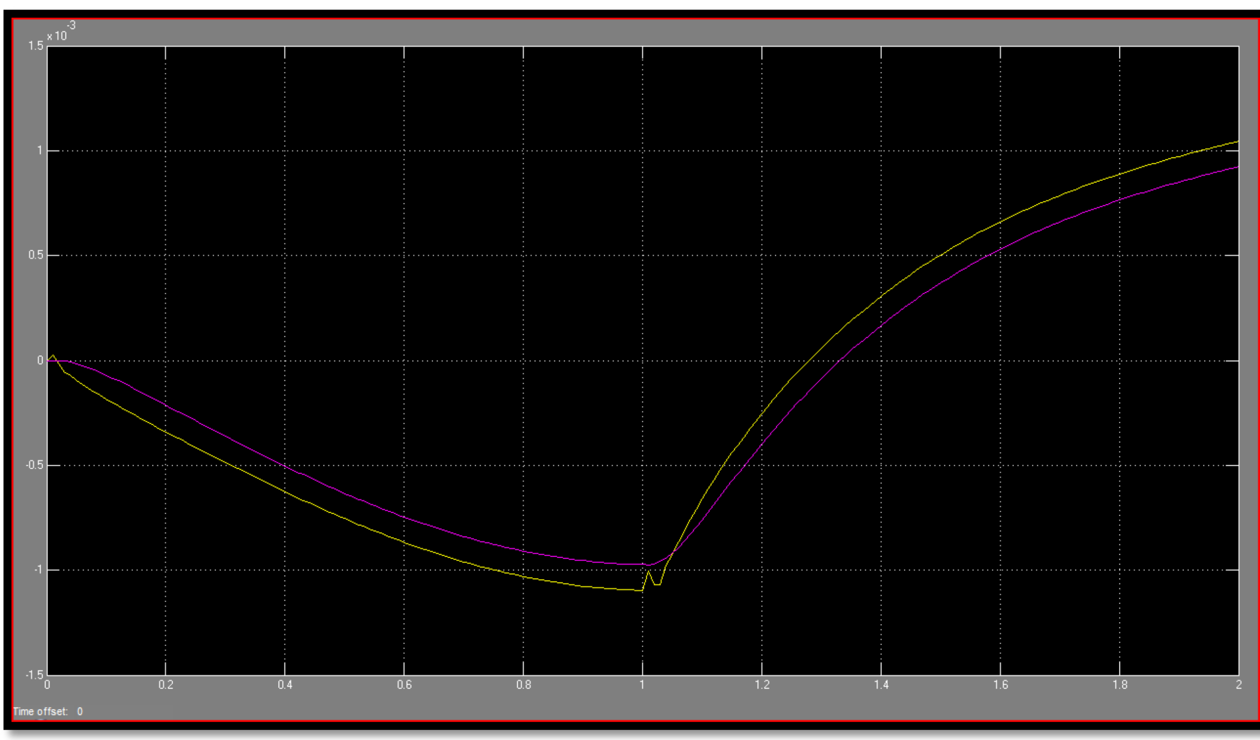

Figure 12: Torque3 for kp1=150 and kv1=0.99, kp2=200 and kv2=0.99, kp3=20 and kv3=0.99

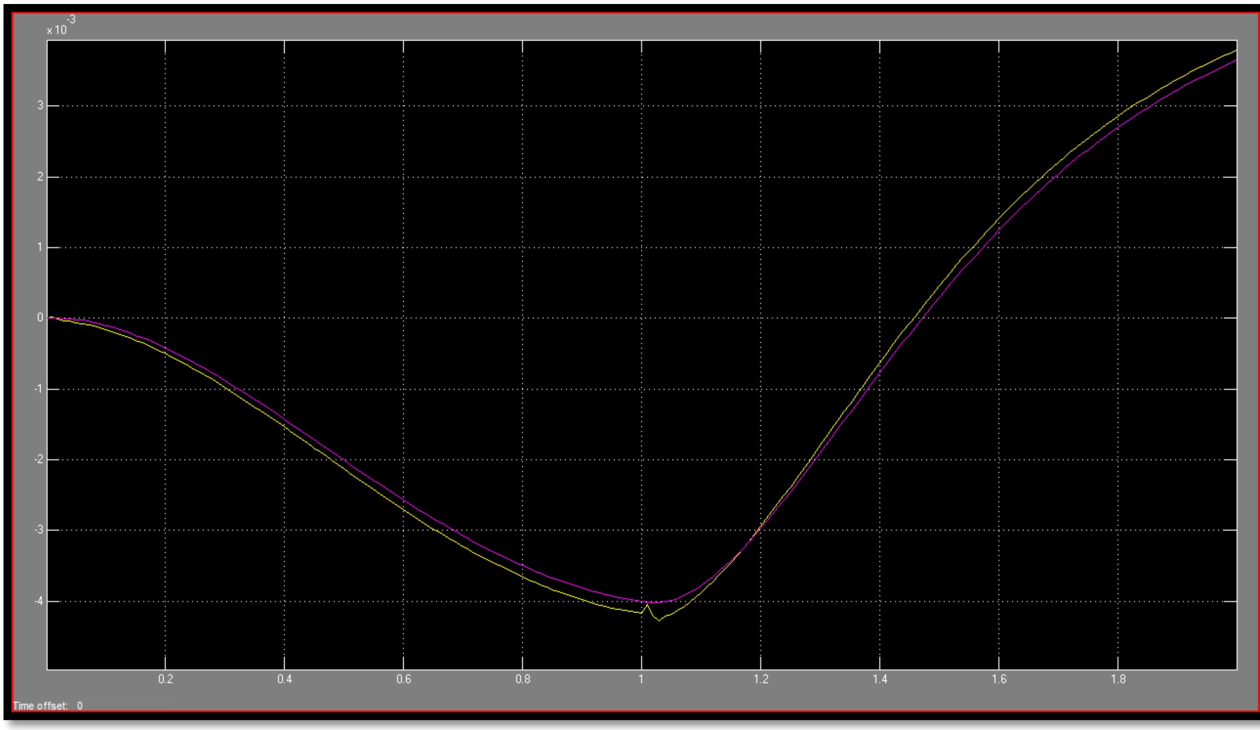

Figure 13: Torque2 for kp1=150 and kv1=0.99, kp2=150 and kv2=0.99, kp3=5 and kv3=0.99

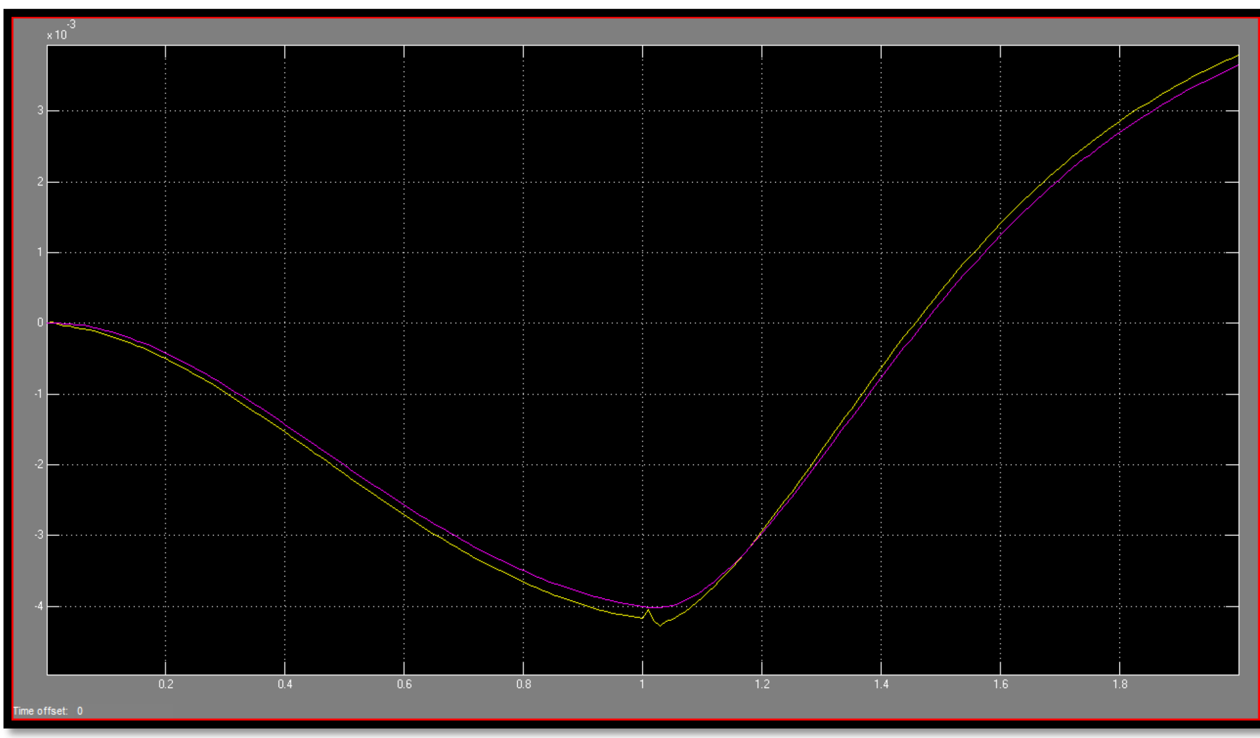

Figure 14: Torque3 for kp1=150 and kv1=0.99, kp2=150 and kv2=0.99, kp3=5 and kv3=0.99

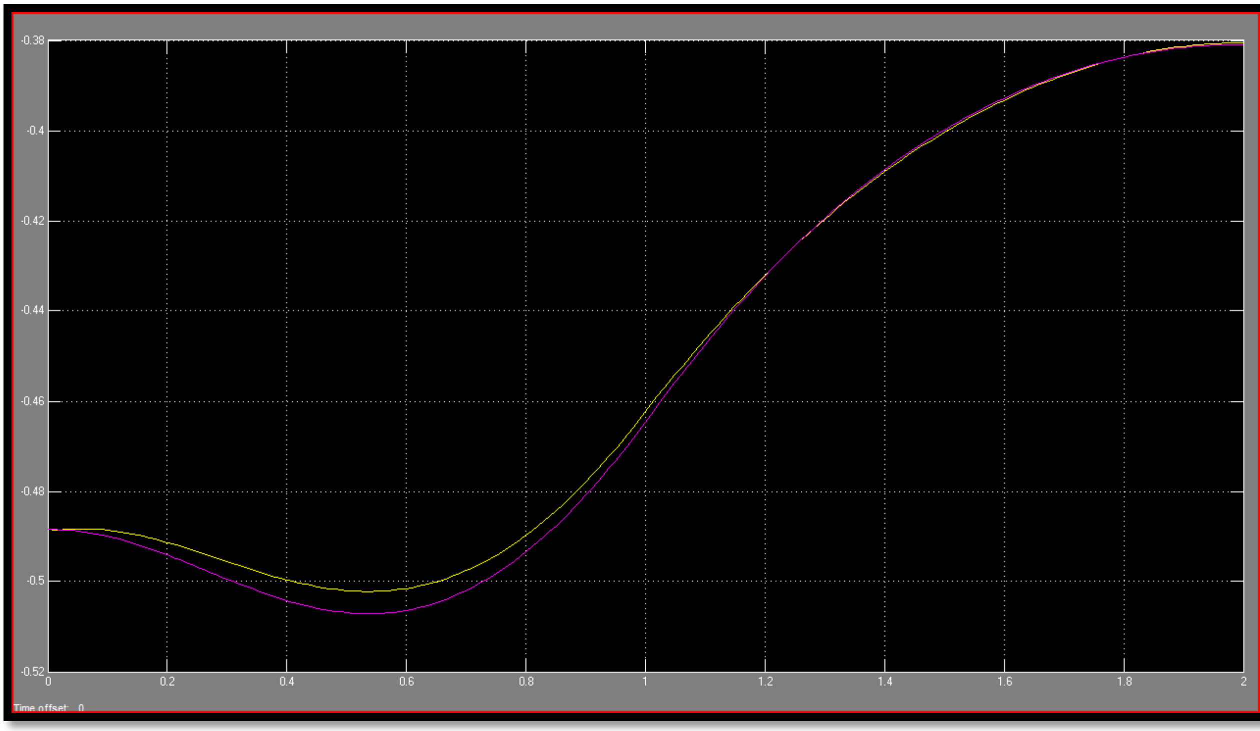

Figure 15: Torque2 for kp1=2 and kv1=0.99, kp2=2 and kv2=0.99, kp3=2 and kv3=0.99

Figure 16: Torque3 for kp1=2 and kv1=0.99, kp2=2 and kv2=0.99, kp3=2 and kv3=0.99

I observed that as I decreased the proportional gain, the total torque got closer to the gravitational torque. Also when the derivative gain is increased to be greater than one, the manipulator can't follow the desired trajectory. The maximum error at the task space was required to have an error less than 0.1mm. The gain values of Fig. 15 and Fig. 16 give the closest to that. These values were chosen as the controller parameters.

APPENDIX

```
%Link Parameter of RRR Planar Manipulator
a1=500; %mm
a2=260; %mm
a3=50; %mm
a1s=a1*0.001; %m
a2s=a2*0.001; %m
a3s=a3*0.001; %m
L1=0.246193; %in (m)Link 1 center of mass
m1=0.556965; %mass of Link 1 in kg
I1=0.0123571; %moment inertia of Link 1 about its center of mass
L2=0.626297-0.5; %in (m)Link 2 center of mass
m2=0.297765;  %mass of Link 2 in kg
I2=0.00189339; %moment inertia of Link 2 about its center of mass
L3=0.780098-0.76; %in (m)Link 3 center of mass
m3=0.0528412;  %mass of Link 3 in kg
I3=0.0000137671; %(kgm^2) moment inertia of Link 3 about its center of mass
g=9.81;
```

Figure 17: RRR Link Parameters