Comparison of Controllers on a Planar RRR Manipulator for Different Trajectories

Evrim Selin Altınkaynak
Izmir Institute of Technology, Department of Electrical and Electronic Engineering, Izmir TURKEY

OBJECTIVE

The aim of this Project is to compare P, PI, PD, and PID controllers with lag, lead and lag-lead compensators on a planar RRR manipulator that is capable of following different trajectories of a line, a square, a triangle, and a circle. For each trajectory each controller is applied separately. The control applied is in joint space. The difference between the desired joint angle and the measured joint angle is the error and is expected to go to zero to control the robot in the desired way. The controller that gives the closest to zero is chosen for that specific trajectory.

Introduction

The project uses the dynamic model defined as;

$$\widehat{M}(q) * \ddot{q} + \overline{B}(q,\dot{q}) + \overline{G}(q) = \tau_a + \tau_{ext}$$

Where $\widehat{M}(q)$is the generalized inertia/mass matrix, $\bar{B}(q,\dot{q})$ is the vector that contains the centripedal and coriolis terms, $\bar{G}(q)$ is the vector that takes in account the gravitational effects on the robot, τ_{ext} is the external torques acting on it, and τ_a is the total torque vector fed back to the joint actuators.

The quasi-static force/torque analysis is performed as;

$$\begin{pmatrix} \delta Y1 \\ \delta Y2 \\ \delta Y3 \end{pmatrix} = \hat{J} * \begin{pmatrix} \delta q1 \\ \delta q2 \\ \delta q3 \end{pmatrix}$$

$$\hat{J}^t * \overline{F} = \overline{\tau}$$

Where

$$\overline{F} = \begin{pmatrix} -m1 * g \\ -m2 * g \\ -m3 * g \end{pmatrix}$$

The dynamic analyis is done by using Lagrange's Method .

$$L = K - U$$

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

;k=1,2…n for n-dof manipulator.

Methods

Solidworks is used to model the RRR manipulator. Using SimMechanics the model is extracted to Matlab. Then the controllers are added to the joints which are controlled .

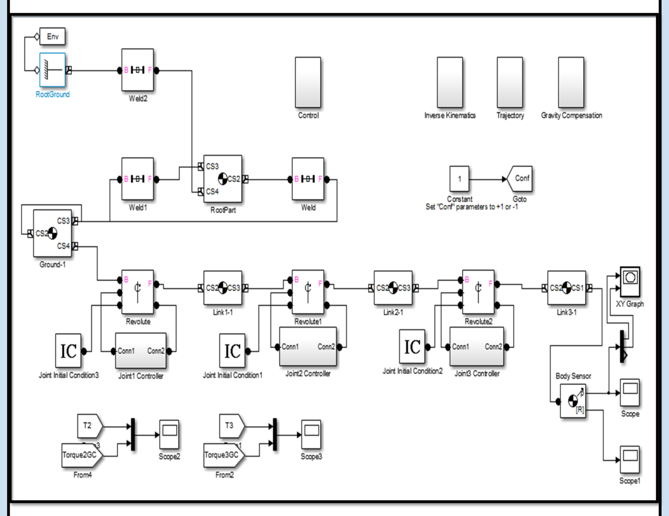

Figure 1: RRR manipulator in Simulink

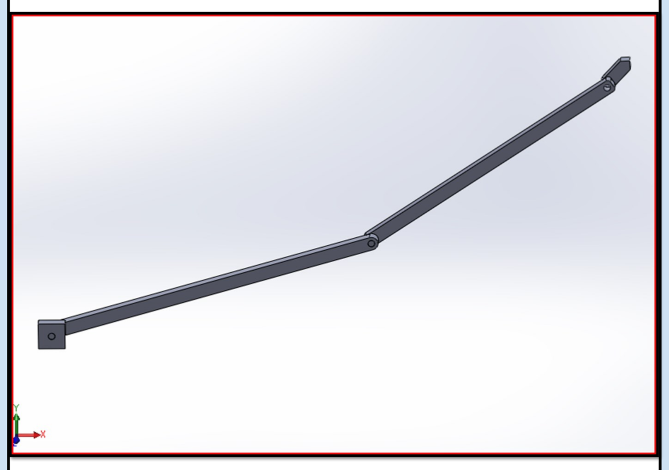

Figure 2: RRR model in Solidworks

Lag compensator is designed as;

$$D(s) = K * \frac{s+a}{s+b}$$

0<b<a<1.

- Acts like PD.
- Decreases the steady state error.

Lead compensator is designed as;

$$D(s) = K * \frac{s+a}{s+b}$$

1<a<b.

- Acts like PI controller.
- Decreases overshoot.

$$\overline{\tau} = -\begin{pmatrix} L1 * c1 & a1 * c1 + L2 * c12 & a1 * c1 + a2 * c12 + L3 * c123 \\ 0 & L2 * c12 & a2 * c12 + L3 * c123 \\ 0 & 0 & L3 * c123 \end{pmatrix} * \begin{pmatrix} -m1 * g \\ -m2 * g \\ -m3 * g \end{pmatrix}$$

Where c1=cos(q1), c12=cos(q1+ q2), and c123=cos(q1+ q2+ q3).

The torques obtained from the dynamic analysis are;

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

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

$$\tau_2 = \tau_{22} + \tau_{23}$$

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

Results

Lag, Lead and Lag-lead Compensators.

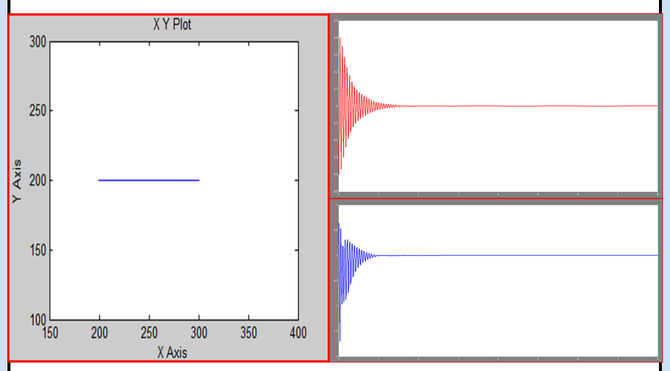

Figure 3: Lag Compensator with a= -0.01, b= -0.009, K=10000

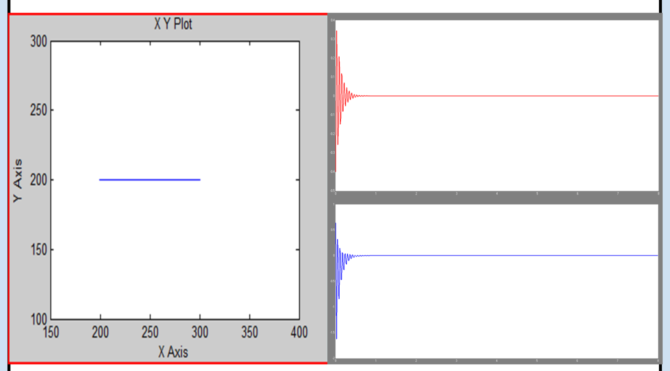

Figure 4: Lead Compensator with a= -10, b= -20, K= 10000

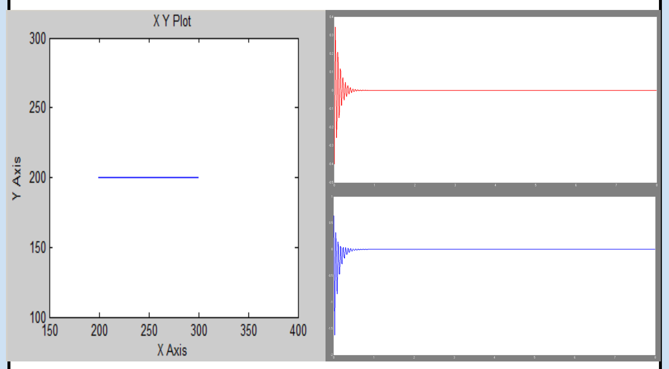

Figure 5: Lead-Lag compensator with lag parameters a= -0.01, b=-0.009, K=1 Lead parameters a= -10, b=-20, K=10000

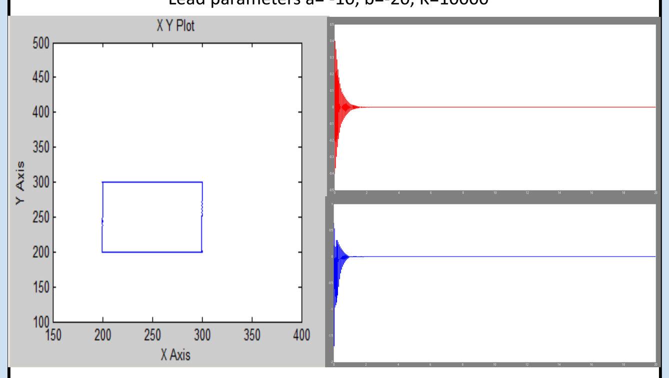

Figure 6: Lag Compensator with a= -0.01, b= -0.009, K=10000

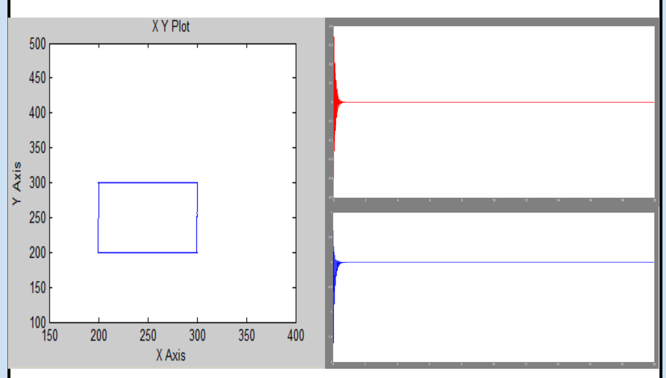

Figure 7: Lead Compensator with a= -10, b= -20, K= 10000

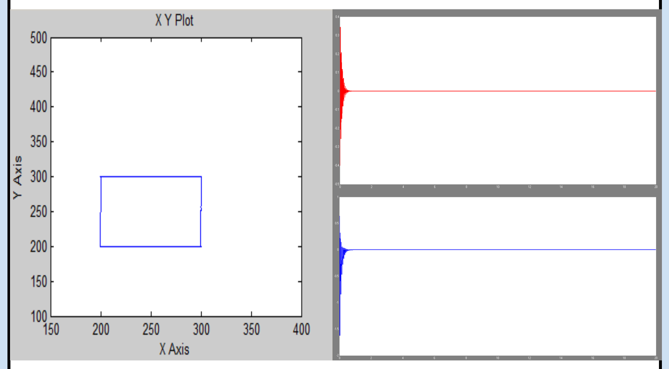

Figure 8: Lead-Lag compensator with lag parameters a= -0.01, b=-0.009, K=1 Lead parameters a= -10, b=-20, K=10000

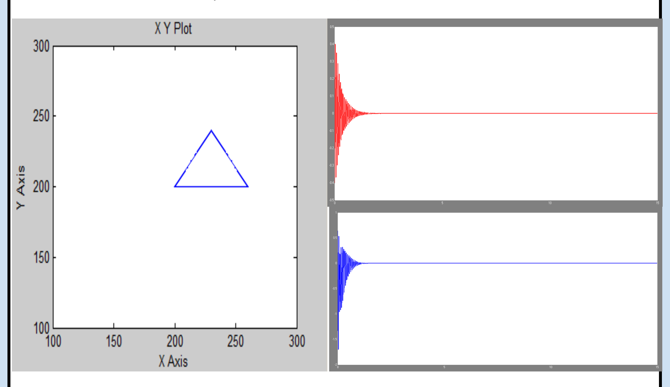

Figure 9: Lag Compensator with a= -0.01, b= -0.009, K=10000

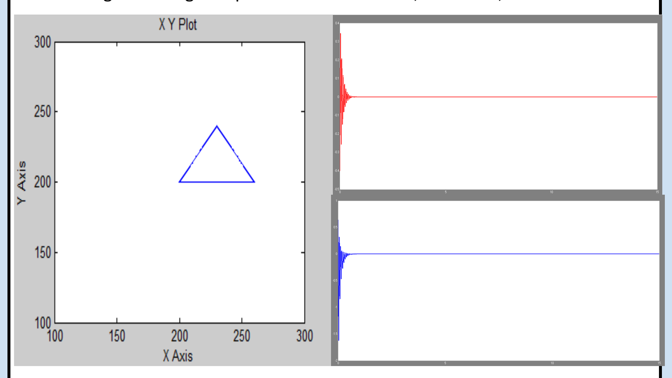

Figure 10: Lead Compensator with a= -10, b= -20, K= 10000

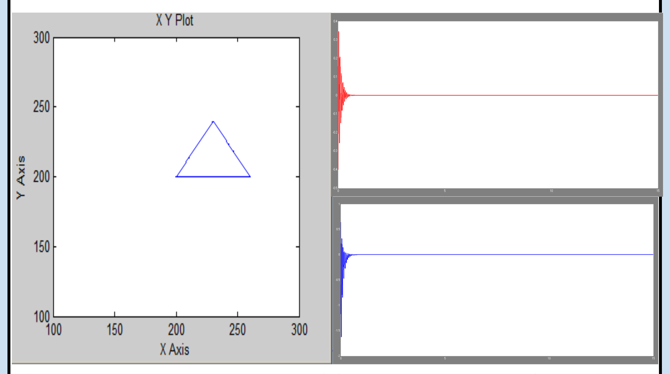

Figure 11: Lead-Lag compensator with lag parameters a= -0.01, b=-0.009, K=1 Lead parameters a= -10, b=-20, K=10000

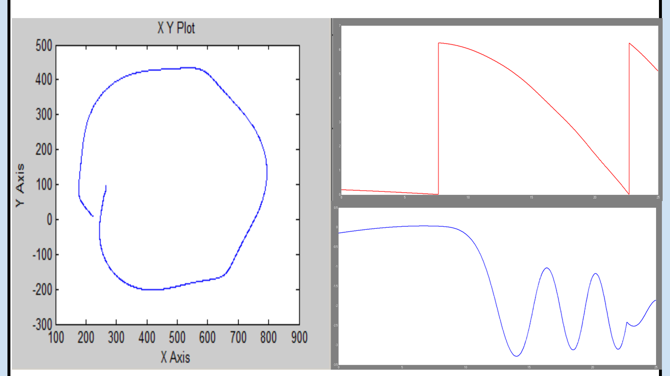

Figure 12: Lag Compensator with a= -0.01, b= -0.009, K=0.01

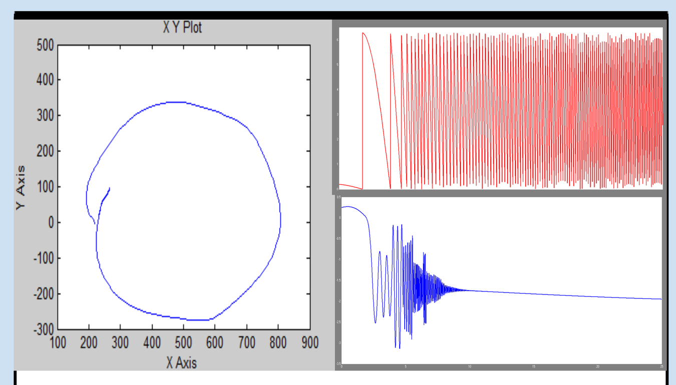

Figure 13: Lead Compensator with a= -10, b= -20, K= 1

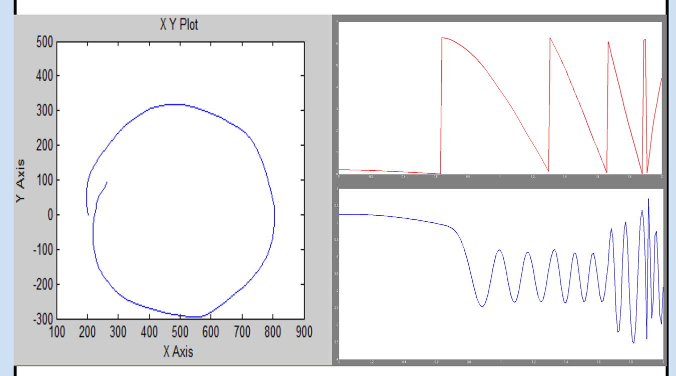

Figure 14: Lead-Lag compensator with lag parameters a= -0.01, b=-0.009, K=0.001 Lead parameters a= -10, b=-20, K=10000

P, PD, PI, and PID Controller

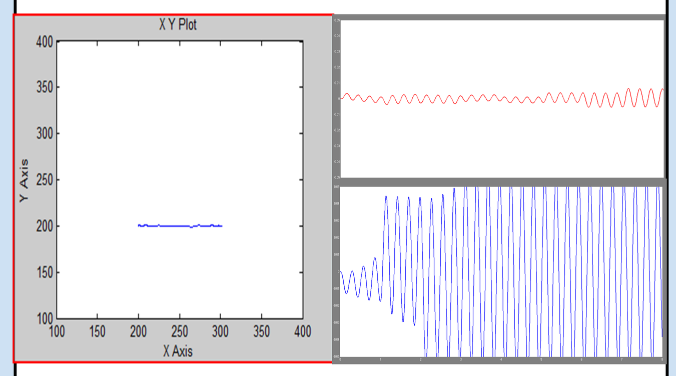

Figure 15: P control with kp1=kp2=kp3=500

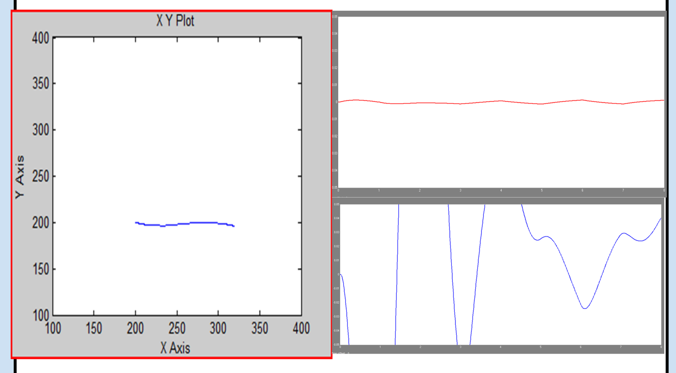

Figure 16: PD control with kp1=kp2=150, kp3=5, kd1=kd2=kd3=0.99

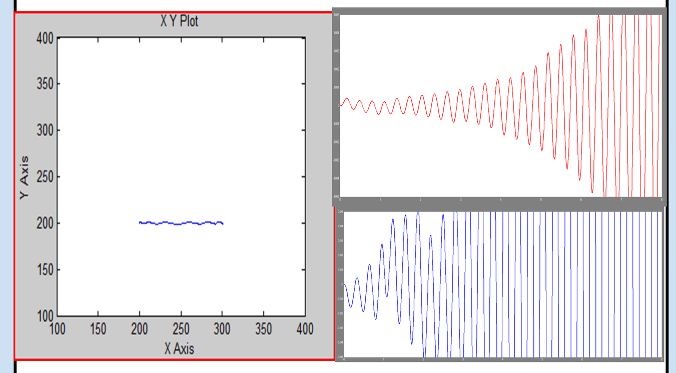

Figure 17: PI control with kp1=kp2=kp3=400,ki1=ki2=ki3=0.99

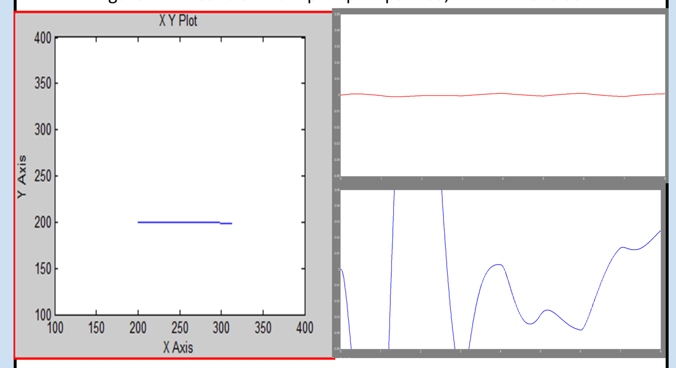

Figure 18: PID control with kp1=kp2=200, kp3=10, kd1=kd2=kd3=ki1=ki2=ki3=0.99

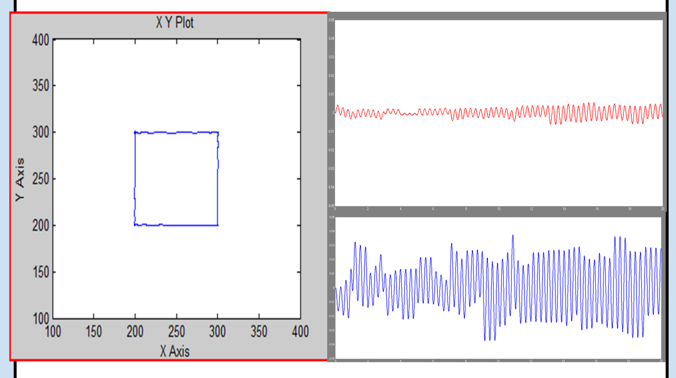

Figure 19: P control with kp1=kp2=kp3=400

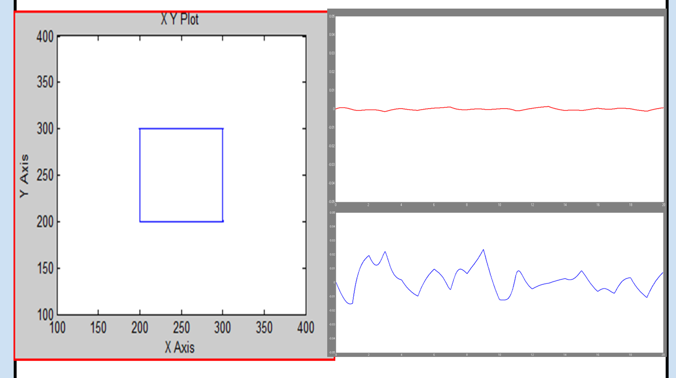

Figure 20: PD control with kp1=kp2=200, kp3=100, kd1=kd2=kd3=0.99

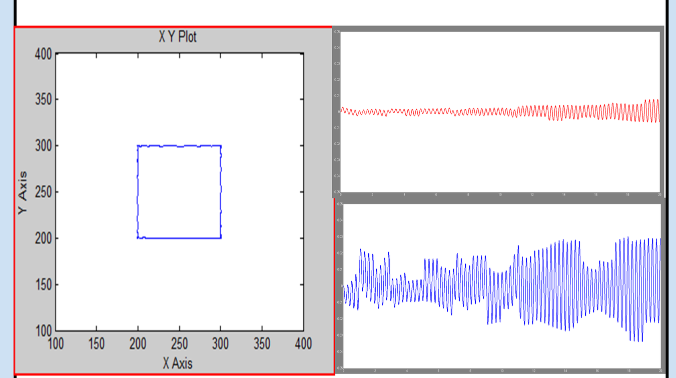

Figure 21: PI control with kp1=kp2=kp3=600, ki1=ki2=ki3=0.2

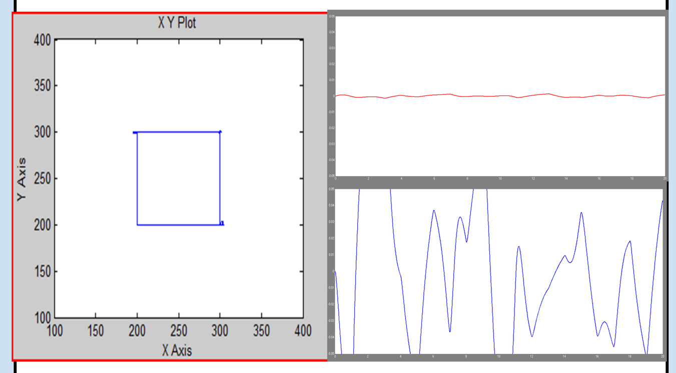

Figure 22: PID control with kp1=kp2=200, kp3=20,kd1=kd2=kd3=0.99, ki1=ki2=ki3=0.2

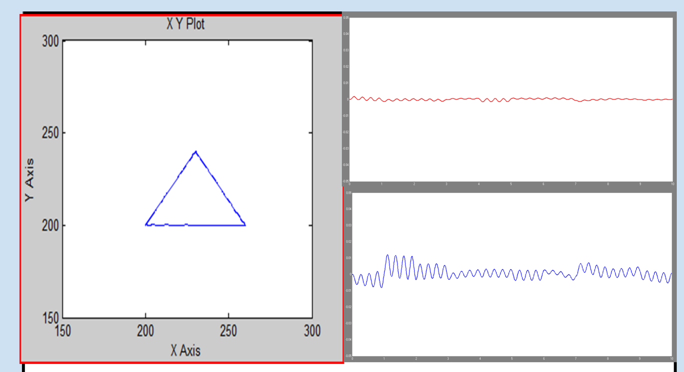

Figure 23: P control with kp1=kp2=kp3=600

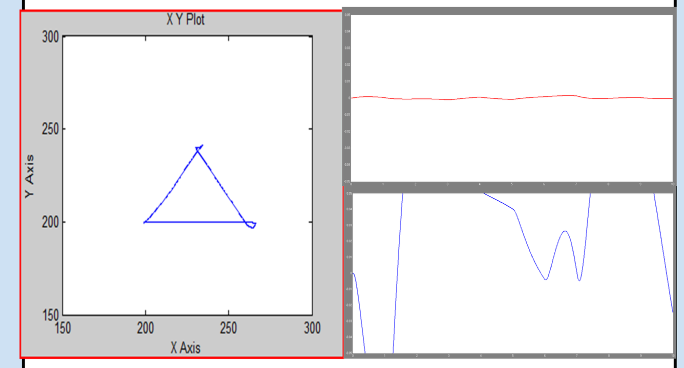

Figure 24: PD control with kp1=kp2=150,kp3=10, kd1=kd2=kd3=0.99

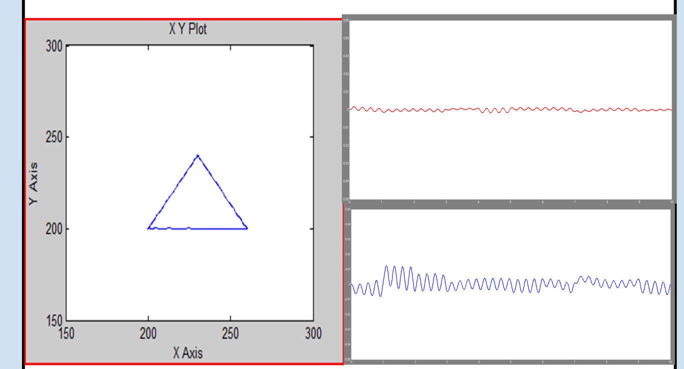

Figure 25: PI control with kp1=kp2=kp3=600,ki1=ki2=ki3=0.2

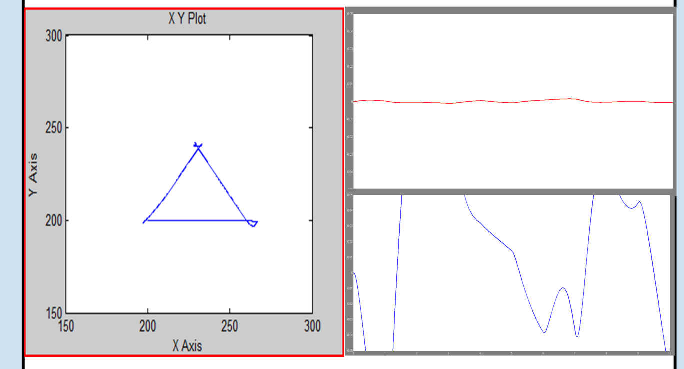

Figure 26: PID control kp1=kp2=150, kp3=10, kd1=kd2=kd3=0.99, ki1=ki2=ki3=0.2

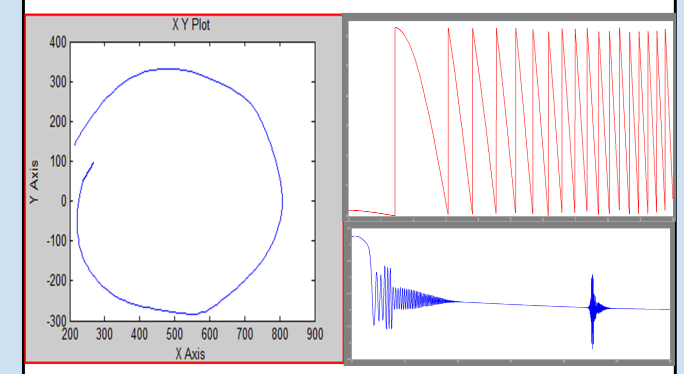

Figure 27: P control with kp1=kp2=kp3=0.9

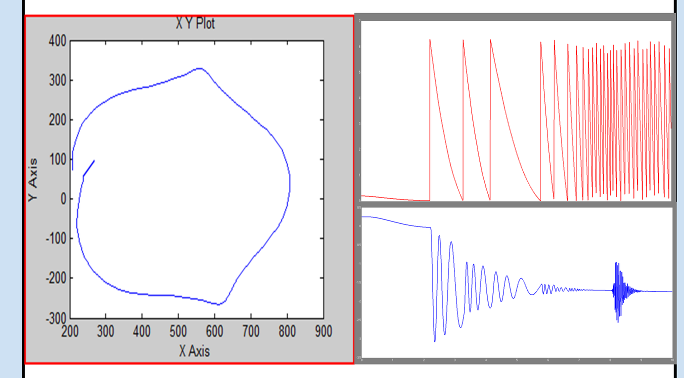

Figure 28: PD control with kp1=kp2=kp3=2,kd1=kd2=kd3=0.99

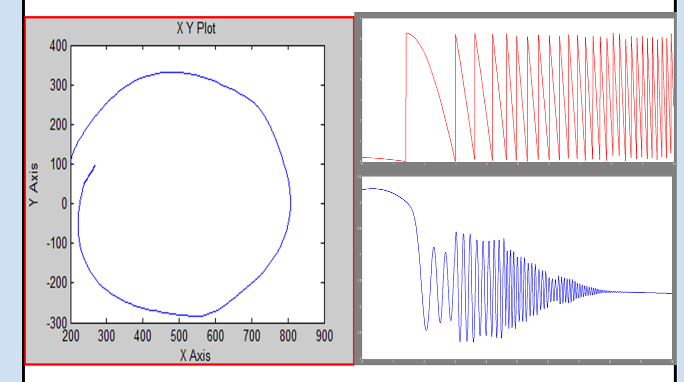

Figure 29: PI control with kp1=kp2=kp3=0.85, ki1=ki2=ki3=0.2

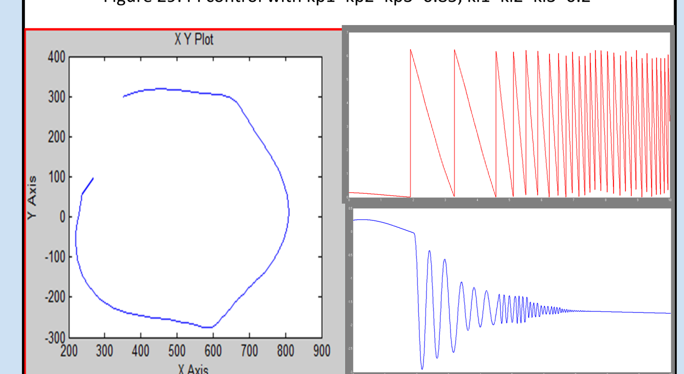

Figure 19: PID control with kp1=kp2=kp3=0.95, kd1=kd2=kd3=0.99, ki1=ki2=ki3=0.2

The plots represent the trajectory of robot (left), the error at joint 2 angle (top right), and error at joint 3 angle (bottom right).

Conclusions

In the end of this project, it is concluded that for every trajectory the preferred controller is different. Every controller follows the trajectory given. But not every controller converges the error to zero. Every controller has their own properties that make them better than the others. The lag compensator is known as decreasing the steady state error. The lead compensator decreases the amount of overshoot. The lead-lag compensator can do both. The P controller amplifies the error while increasing the output. But by adding the D to the controller we can decrease this error. The PD controller not only amplifies the error, but also amplifies the derivative of the error. For a constant error this term deletes that part from the error. The PI controller is known to summing the errors because of the integral part. In general PI controllers have greater errors. The PID controller amplifies the error, its derivative and its integral. Lag, lead, and lag-lead compensators give better results than the P controllers. But to use them, the transfer function of the system is needed. But most of the times this is not in our hand. The poles, zeros, and gains are found experimentally. The joint angle error for joint 1 wasn't put, because motion is given to its actuator.

For the line trajectory, the lead, lag and lag-lead compensator give a better result than the others. But because the error drops faster in lead-lag compensator, it is the best. For the square trajectory, again the lag-lead compensator is the best. For the triangle trajectory, again the lag-lead compensator is the best. The errors at the P controllers are close to zero. But we can't see the drop to exactly zero. This makes this controller to seem like a bad controller. But the parameters were not properly selected. For example, the cumulation of errors in PI is seen quite easily, which means that the Ki value is too high. With proper tuning the results will improve for them. But the effects they have at least can be seen in this way. The circle trajectory needs more tuning for each controller because of its complexity while changing angles.