Project Report of Robotics 1

(Close loop feedback control of a two-link Manipulator)

By

Muhammad Arsalan

Submitted to:

Professor Dr. Riaz Akbar

DEPARTMENT OF MECHATRONICS ENGINEERING

UNIVERSITY OF ENGINEERING & TECHNOLOGY

PESHAWAR

INTRODUCTION:-

We have given the task to design the close loop feedback control of a two-link manipulator. We have given the various parameters of the two link manipulator, i.e

C m1 = frictional constant for motor1 =0.1

C m2 = frictional constant for motor2=0.1

J m1 = inertia constant for motor1=0.01

J m2 = inertia constant for motor2=0.01

M1 = mass of the link1=1 kg

M2 = mass of the link2= 0.5 kg

g = gravitational acceleration=9.8 m/s 2

K t motor constant (size and weight of motor) = 10

L1 =length of link1= 2 m

L2 =length of link2 = 1 m

K P =proportional gain= 80

K d =derivative gain= 1.0

The software used was SIMULINK MATLAB.

The flow chart of the close loop controller is below

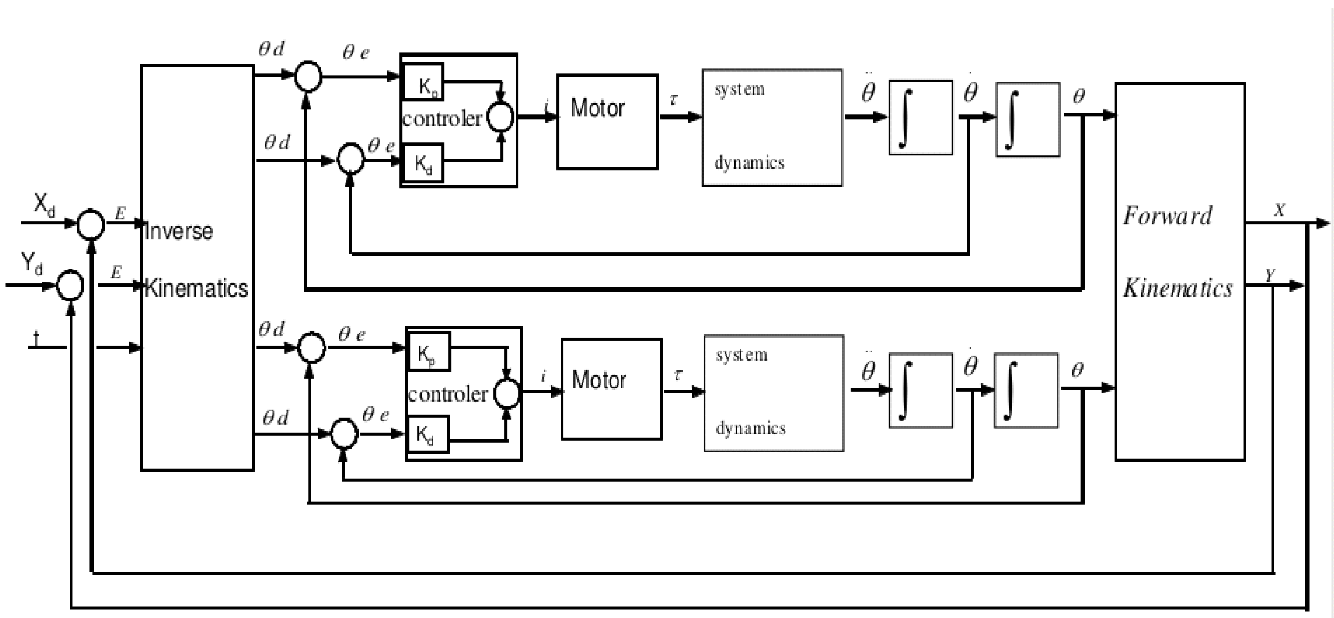


Figure 1: Flow Chart of the close loop feedback controller

PROCEDURE OF SOLVING THE PROBLEM:

The problem is that our robot should trace the path of circle having radius of 0.5 m and then we have to check the results whether the controller is doing its work fine or not.

First of all we have the circular path/ points on the circle.

INVERSE KINEMATICS:

After passing the circle points (x,y) we get the different number of joints angles for every point on the circle (ϴ1, ϴ2) for joint 1 and joint 2 respectively.

These are our desired joint positions.

PID CONTROLLER:

Then we have controller which finds the error in the system by comparing the desired values to the actual values.

The actual values are obtained after the passing values through system dynamics.

MOTOR:

After finding the error and the setting values in the controller i.e. kp and kd values, we get the current value which is the provided to the motor to rotate, the motor then gives us the torque value which motor generates.

SYSTEM DYNAMICS:

The torque obtained from motor and then pass through the system dynamics, which accounts all the masses and the lengths of the robot links, which then gives us the actual joint acceleration of the robot,

After integrating we get the actual joint velocities and again by integrating we get the joint positions, which is the actual joint position.

FORWARD KINEMATICS:

After obtaining the actual joint positions we pass these values through the forward kinematics to get the actual path which will follow the robot.

Error:

There are three types of error that we computed i.e,

1. **Error between Actual theta-1 and the desired theta-1**
2. **Error between Actual theta-2 and the desired theta-2**
3. **Error between the Actual-X-value and the Desired-X-value**
4. **Error between the Actual-Y-value and the Desired-Y-value**

SIMULINK MATLAB:

This time I did project of close loop feedback control of twolink manipulator in SIMULINK MATLAB.

Now some term to explain to understand the SIMULINK file I made.

x=desired x-values

y=desired y-values

Values after INVERSE KINEMATICS are the desired values

des_th1=Desired joint 1 position

des_th2=Desired joint 2 position

des_th1_d=Desired joint 1 velocity

des_th2_d=Desired joint 2 velocity

des_th1_dd=Desired joint 1 acceleration

des_th2_dd=Desired joint 2 acceleration

Values after SYSTEM DYNAMICS are the desired values

actual_th1= actual joint 1 position

actual_th2= actual joint 2 position

actual_th1_d= actual joint 1 velocity

actual_th2_d= actual joint 2 velocity

actual_th1_dd= actual joint 1 acceleration

actual_th2_dd= actual joint 2 acceleration

Tt1= motor 1 torque

Tt2=motor 2 torque

T1=total torque joint 1

T2=total torque joint 2

X=Actual X-values

Y=Actual Y-values

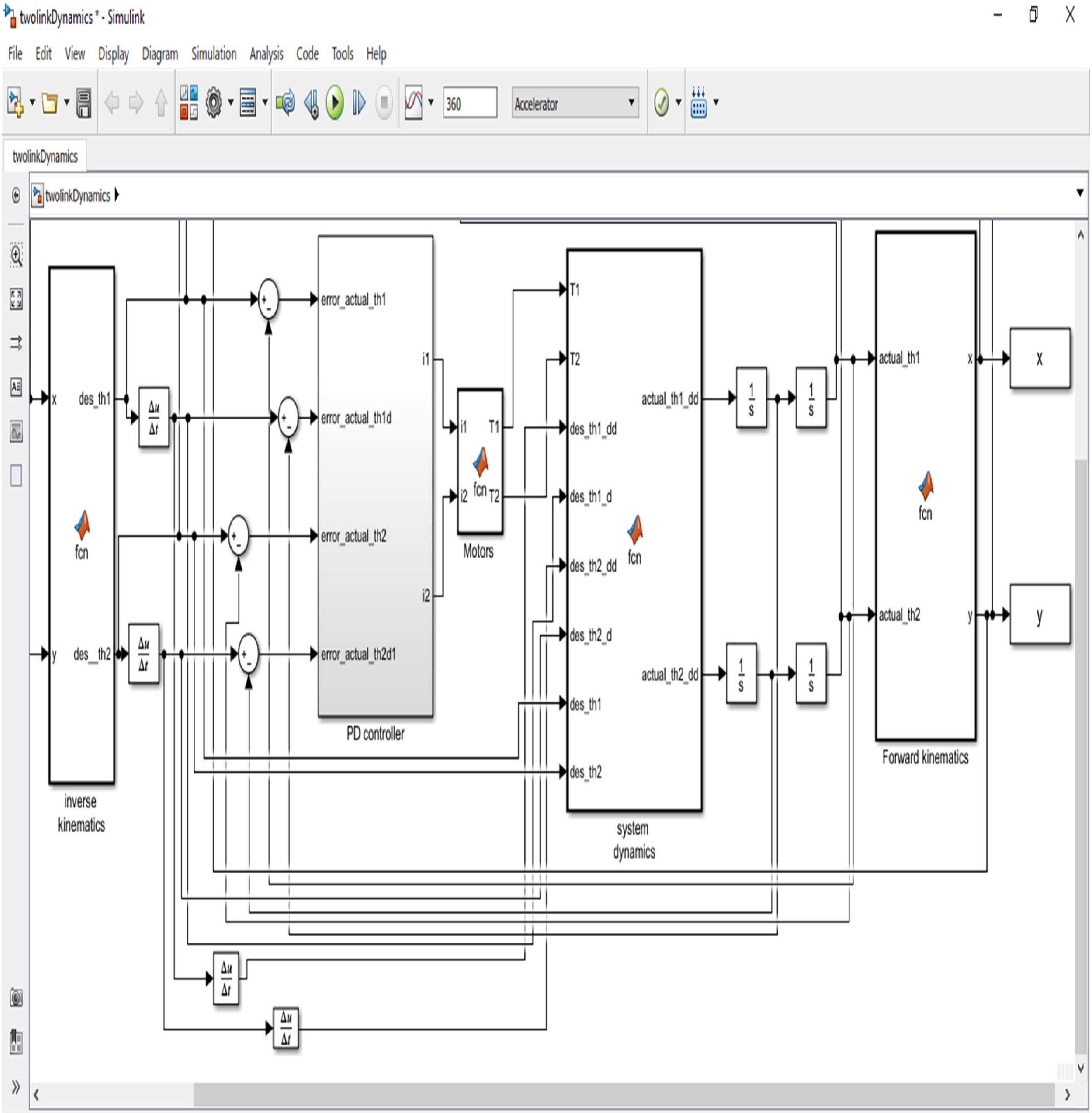


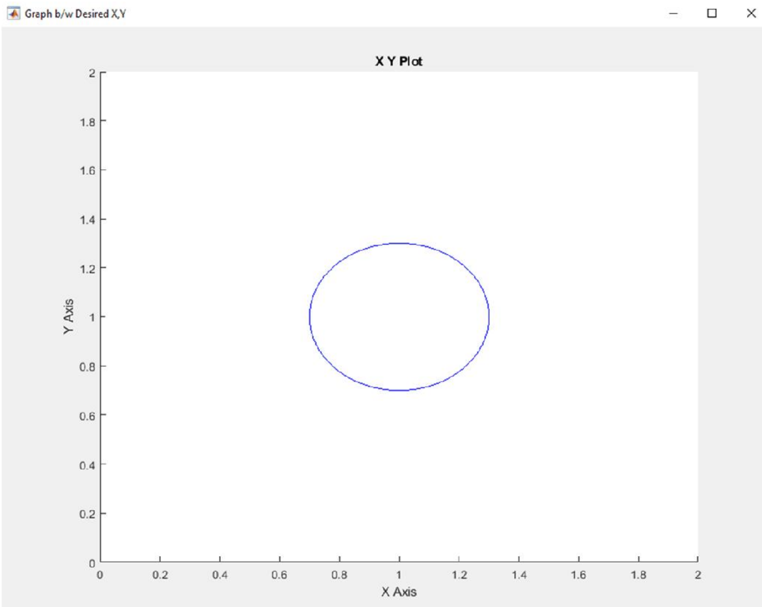
Figure 2: SIMULINK Model of the close loop feedback control of robot

This is the SIMULINK file which I created for the twolink manipulator project, after getting the circular trajectory coordinates we pass it through inverse kinematics, then through PD-controller, then through motor, then through system dynamics and after integrating two time we get the actual joint positions, and at last through forward kinematics to get the actual X and Y values.

RESULTS:

When PID controller (kp=100 & kd=0.5)

After adjusting the PID gains (kp,kd) we compare the graphs between the desired input circular path and the actual circle drawn by robot, below are the graphs



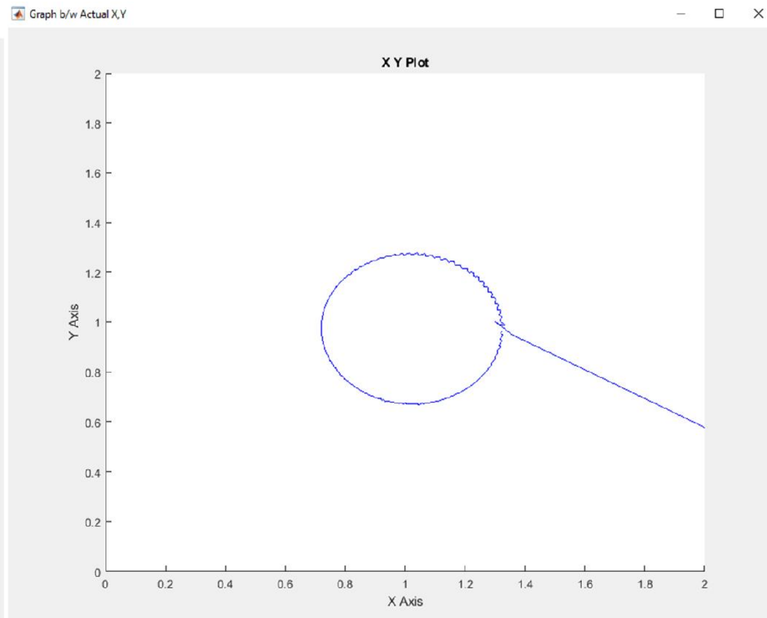
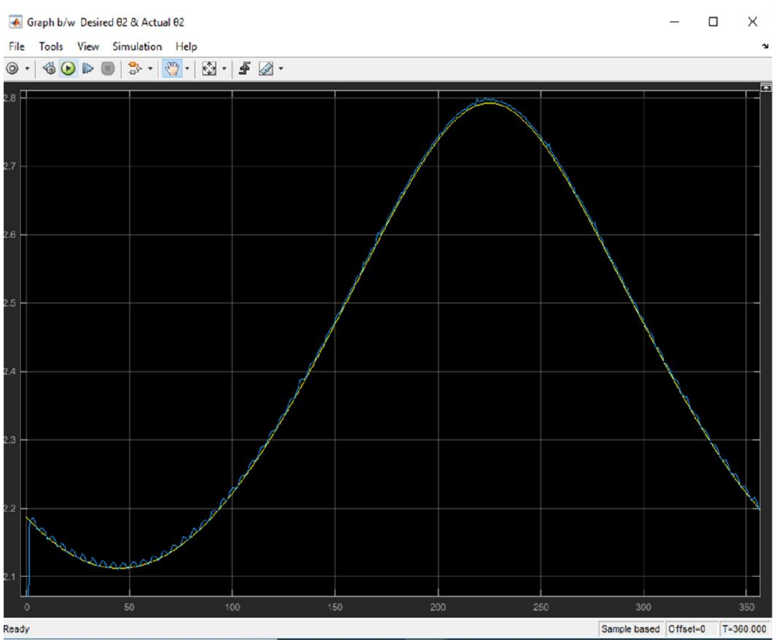


Figure 3: circular path input to the robot

Figure 4: circular path drawn by robot



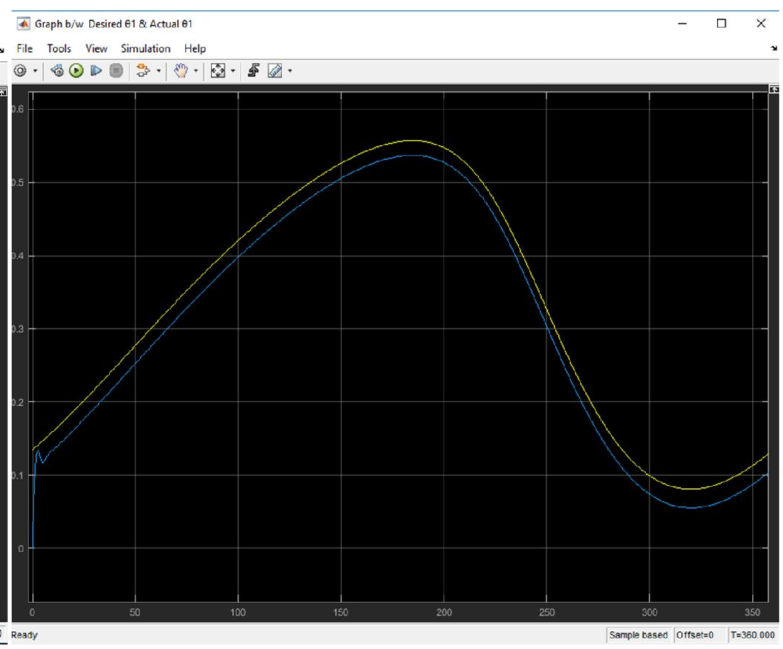
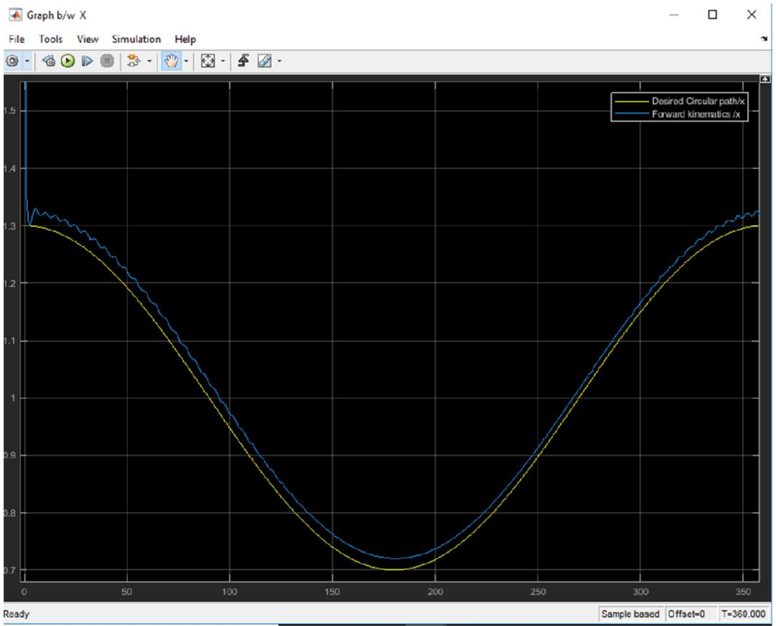


Figure 5: Graph b/w Desired and Actual theta2

Figure 6: Graph b/w Desired and Actual theta1



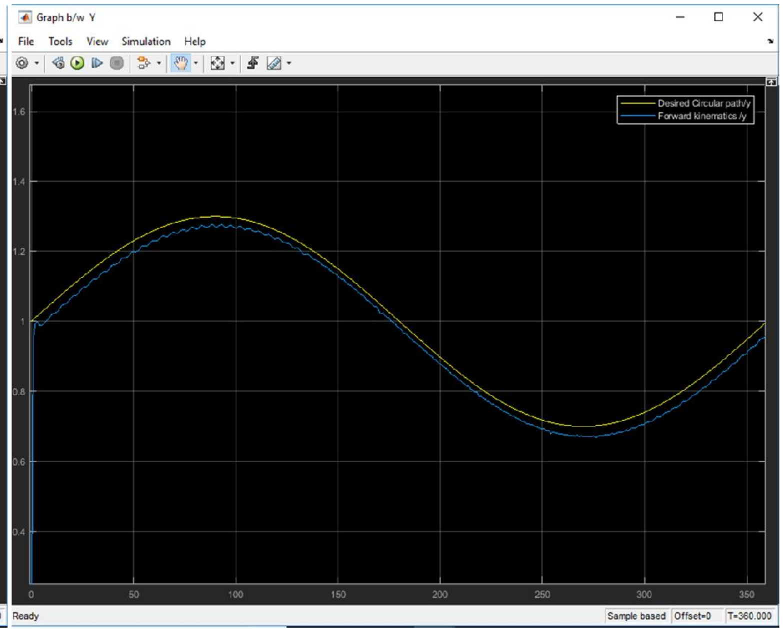
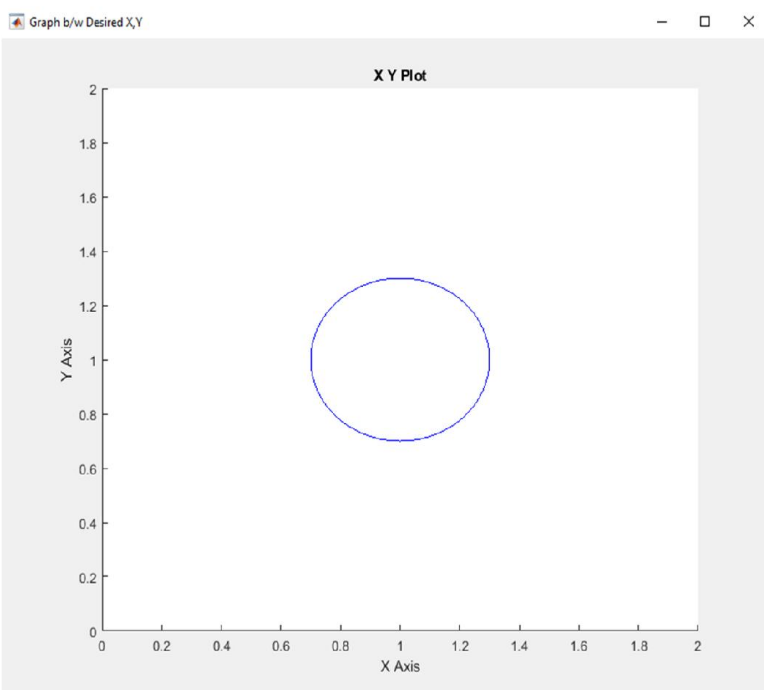


Figure 7: Graph b/w Desired and actual X-values

Figure 8: Graph b/w Desired and actual Y-values

When PID controller (kp=80 & kd=5.40)



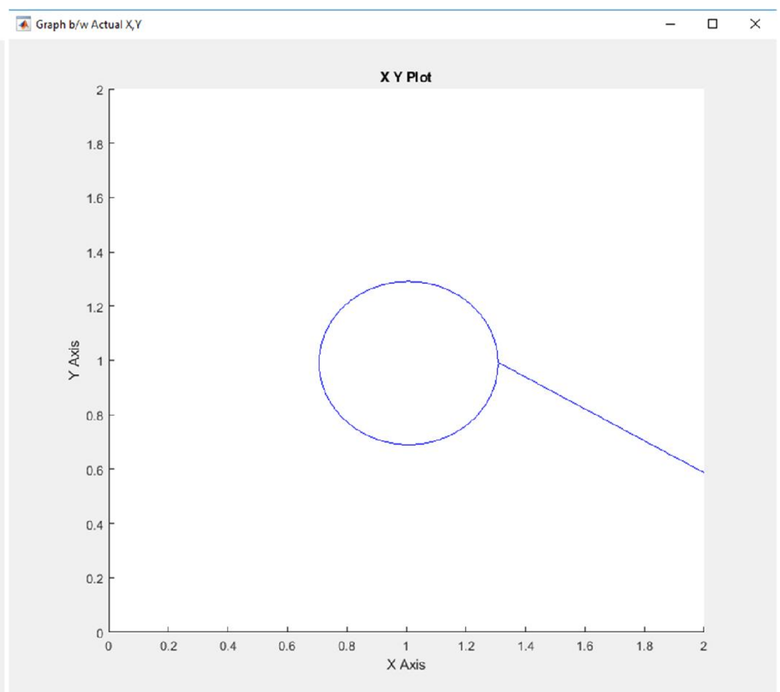


Figure 9 : Desired circle input to the robot

Figure 10: Actual circle drawn by robot

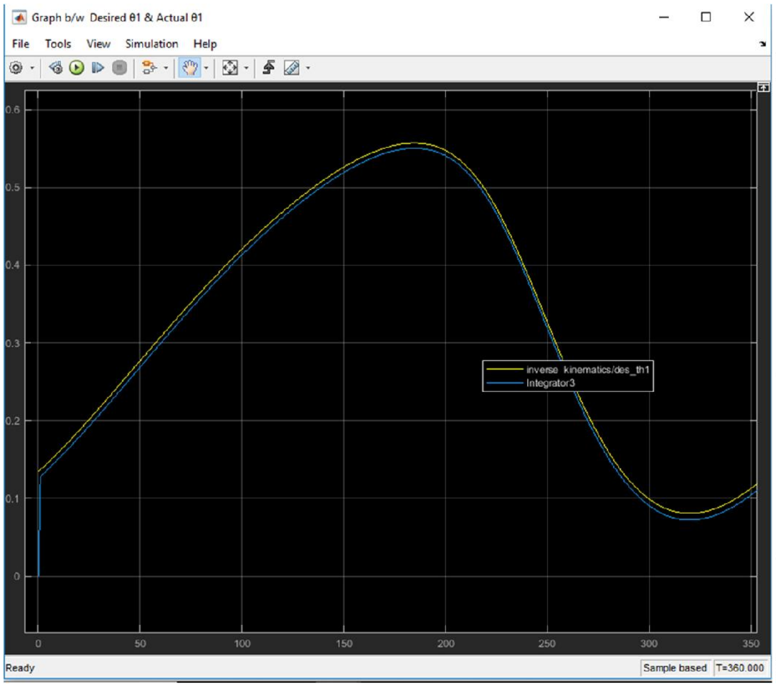


Figure 11: Graph b/w Desired and Actual theta_1

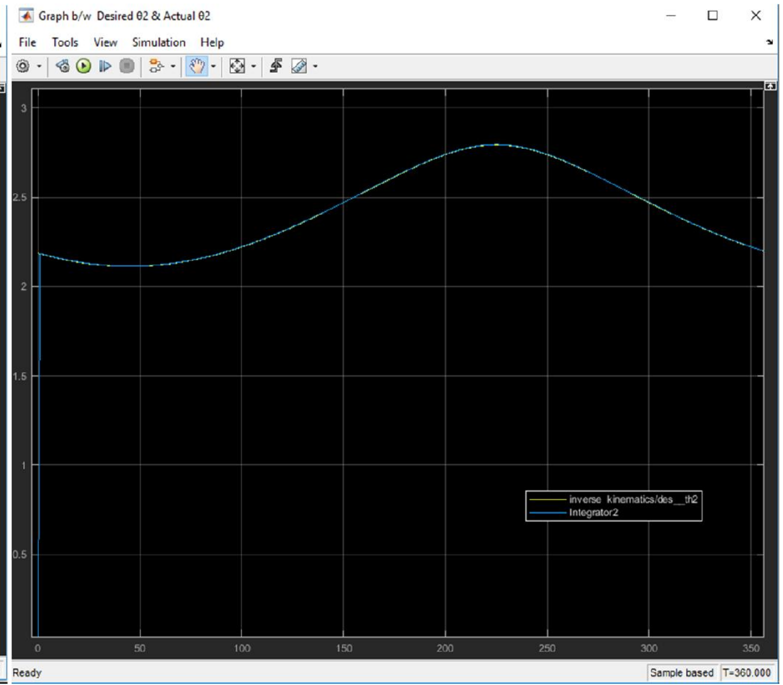


Figure 12: Graph b/w Desired and Actual theta_2

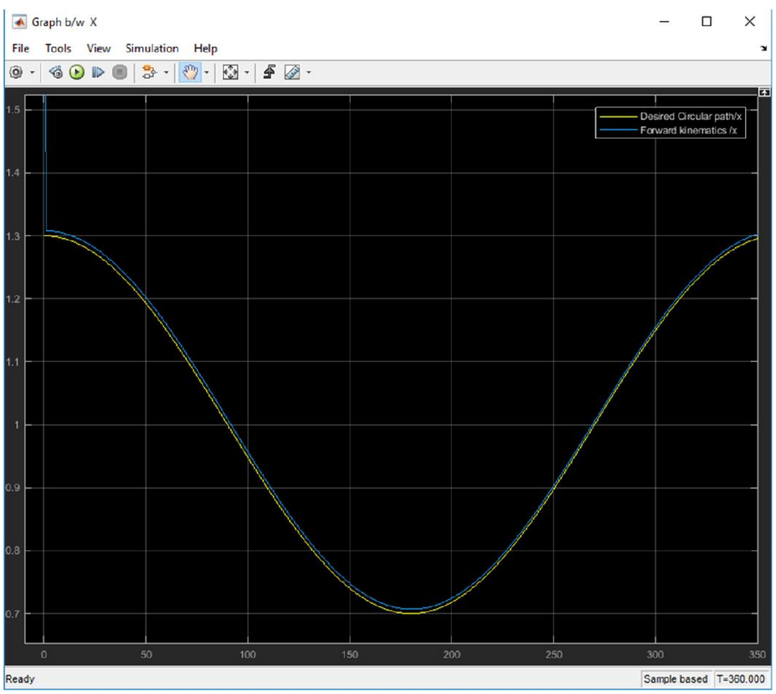


Figure 13: Graph b/w Desired and Actual X-values

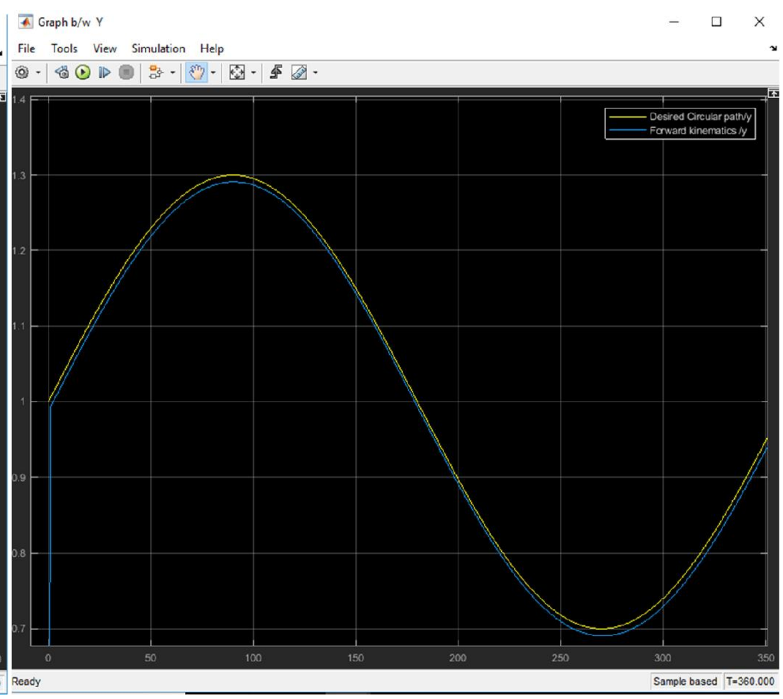


Figure 14: Graph b/w Desired and Actual Y-values

CONCLUSION:

As we can see from these two examples that in the first attempt there is overshoot and oscillation in the system, thus by adjusting the PD-controller gains then in 2nd attempt there is no overshoot and no oscillation in the system.

The system follows the desired trajectory accurately.