Programming Assignment 3

By Chinmayee Prabhakar

a)To generate the trajectory as a cubic polynomial, the initial and final conditions were defined. The trajectories were calculated for each of the joints separately by using the following equation;

$$A * a = B$$

Where A is a 4 x 4 matrix where the initial and final time elements are substituted in the trajectory, B is a 4 x 2 matrix consisting of the initial and final states.

‘a’ constitutes the co-efficients of the cubic polynomial as a vector.

We compute ‘a’ by taking inverse of A and multiplying it with B.

‘a’ for theta1 :

```
a_th1 =  3.1416
         0
         -0.0942
          0.0063
```

‘a’ for theta2:

```
a_th2 =  1.5708
         0
         -0.0471
         0.0031
```

The trajectories obtained are:

```
theta1 =(pi*t^3)/500 - (3*pi*t^2)/100 + pi


theta2 =(pi*t^3)/1000 - (3*pi*t^2)/200 + pi/2

The angular velocities:

theta1_dot =(3*pi*t^2)/500 - (3*pi*t)/50


theta2_dot =(3*pi*t^2)/1000 - (3*pi*t)/100

The angular accelerations:

theta1_ddot =(3*pi*t)/250 - (3*pi)/50


theta2_ddot =(3*pi*t)/500 - (3*pi)/100
```

b) The different terms of the manipulator equation form:

M = [(m1*r1^2 + m2*r2^2 + 2*m2*cos(theta2)*r2*l1 + m2*l1^2 + I1 + I2), (m2*r2^2 + l1*m2*cos(theta2)*r2 + I2); (m2*r2^2 + l1*m2*cos(theta2)*r2 + I2), (m2*r2^2 + I2)];

C = [-(2*r2*theta2_dot*l1*m2*sin(theta2)), -(r2*theta2_dot*l1*m2*sin(theta2)); (r2*l1*m2*sin(theta2)*theta1_dot), 0];

G = [(- sin(theta1)*(r1*g*m1 + g*l1*m2) - r2*g*m2*sin(theta1 + theta2)); (- r2*g*m2*sin(theta1 + theta2))];

We can compute the Manipulator equation form as:

Tau = M*ddq + C * dq + G

Where ddq = [theta1_ddot; theta2_ddot]; and dq = [theta1_dot; theta2_dot] and Tau = [Tau1; Tau2]

c) By taking the virtual control input 'v' instead of ddq in the Manipulator Equation form we can compute the feedback linearization control law as follows:

When canceling out the non-linear terms we get : v = ddq

Treating this as a linear system equation and computing the state-feedback control law for each of the joints:

We get A = [0 0 1 0;

0 0 0 1;

0 0 0 0;

0 0 0 0]

B = [0 0;

0 0;

1 0;

0 1]

Taking lambda = [-2, -1.9, -1-2j, -1+2j];

We get the gains,
K = `[ 5.0000         0    2.0000         0; 0    4.6200   0    4.3000];`

The virtual control input, v = -k * (states-des_states) + vd

Where we get the desired states from the trajectory computed earlier as:

```
des_state = [(pi*t^3)/500 - (3*pi*t^2)/100 + pi;
             (pi*t^3)/1000 - (3*pi*t^2)/200 + pi/2;
             (3*pi*t^2)/500 - (3*pi*t)/50;
             (3*pi*t^2)/1000 - (3*pi*t)/100];
```

vd is obtained from the acceleration trajectory calculated by double differentiating the position trajectory over time.

```
vd = [(3*pi*t)/250 - (3*pi)/50;
    (3*pi*t)/500 - (3*pi)/100];
```

The feedback linearization control law is obtained as :

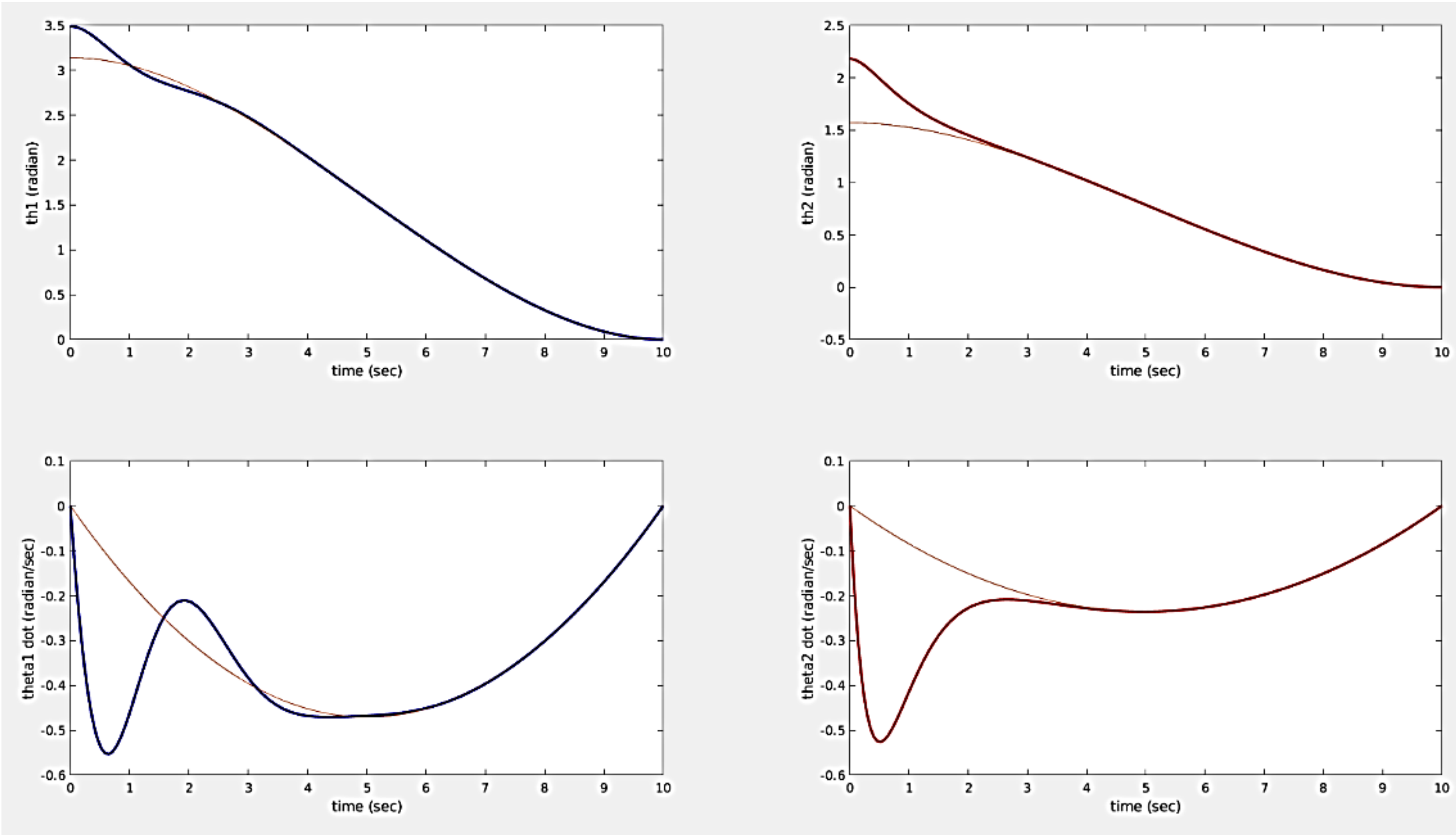
Tau = M*v + C*dq + G

d) We integrate the above control law : Tau = M*v + C*dq + G in the ode function.

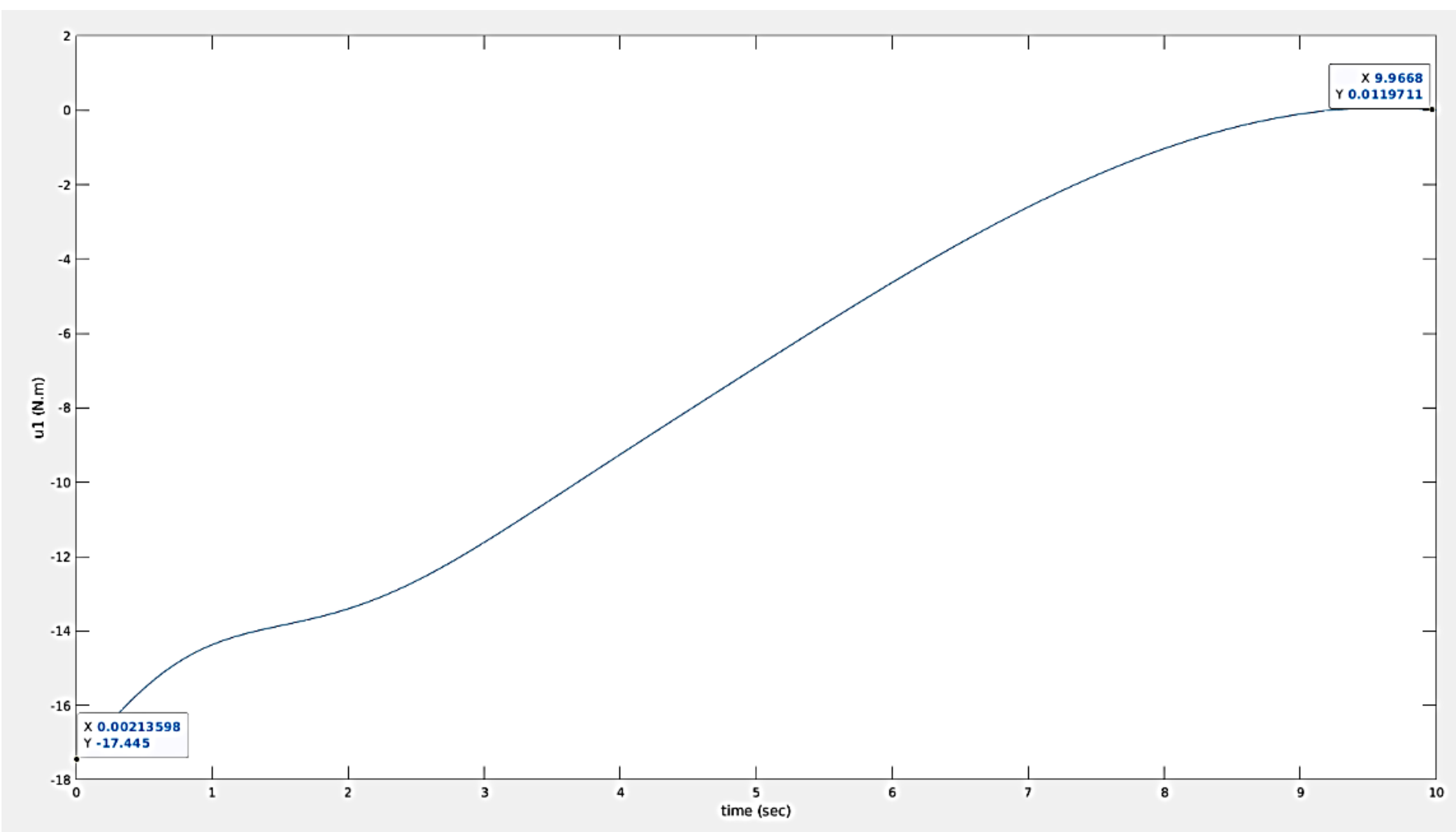
e) For a time duration of 10s and initial theta1 = 200°, theta2 = 125°, theta1_dot = theta2_dot = 0 at

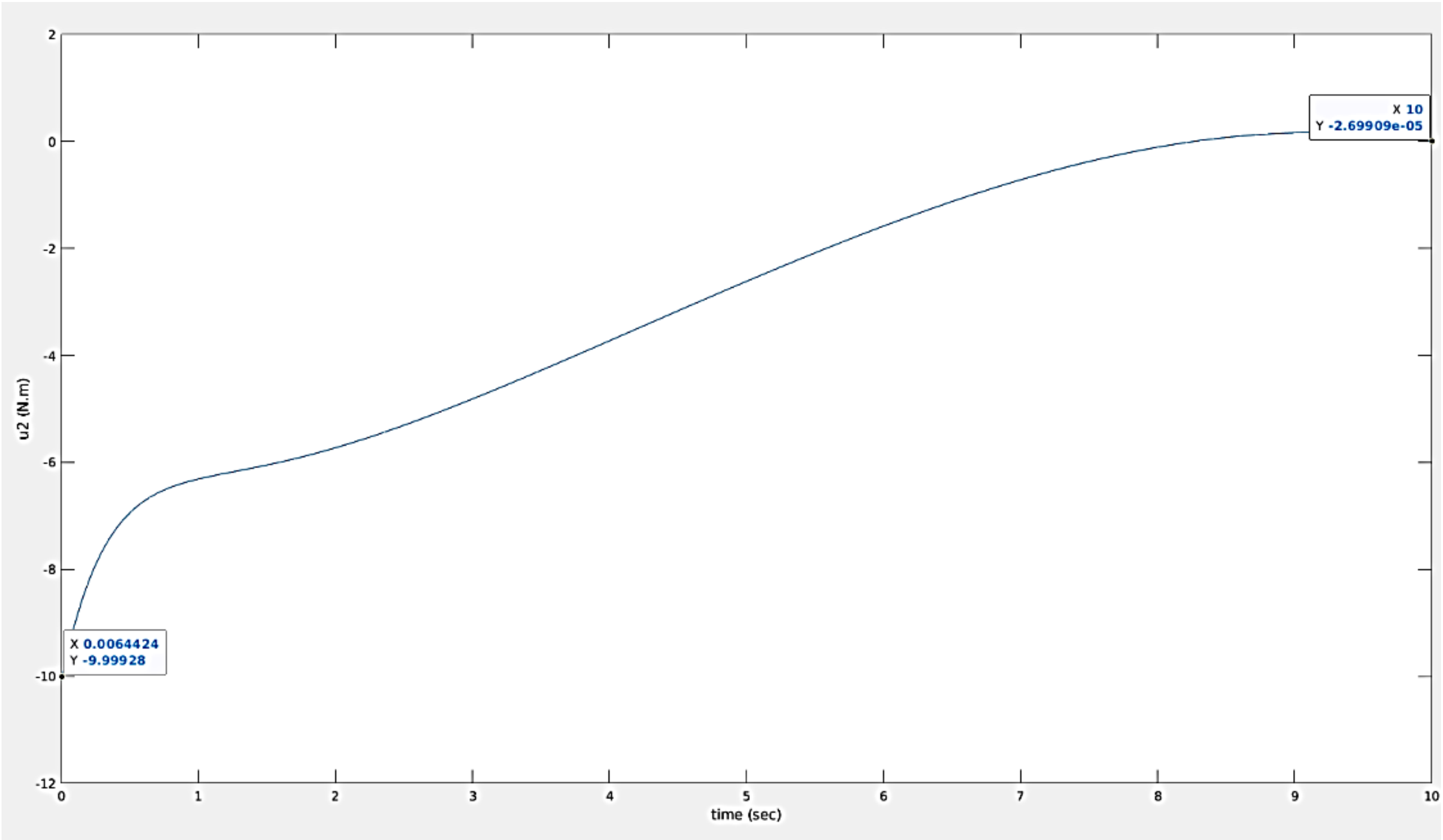
time = 0s

We obtain the following trajectory convergence of the position and velocity to the desired trajectories.

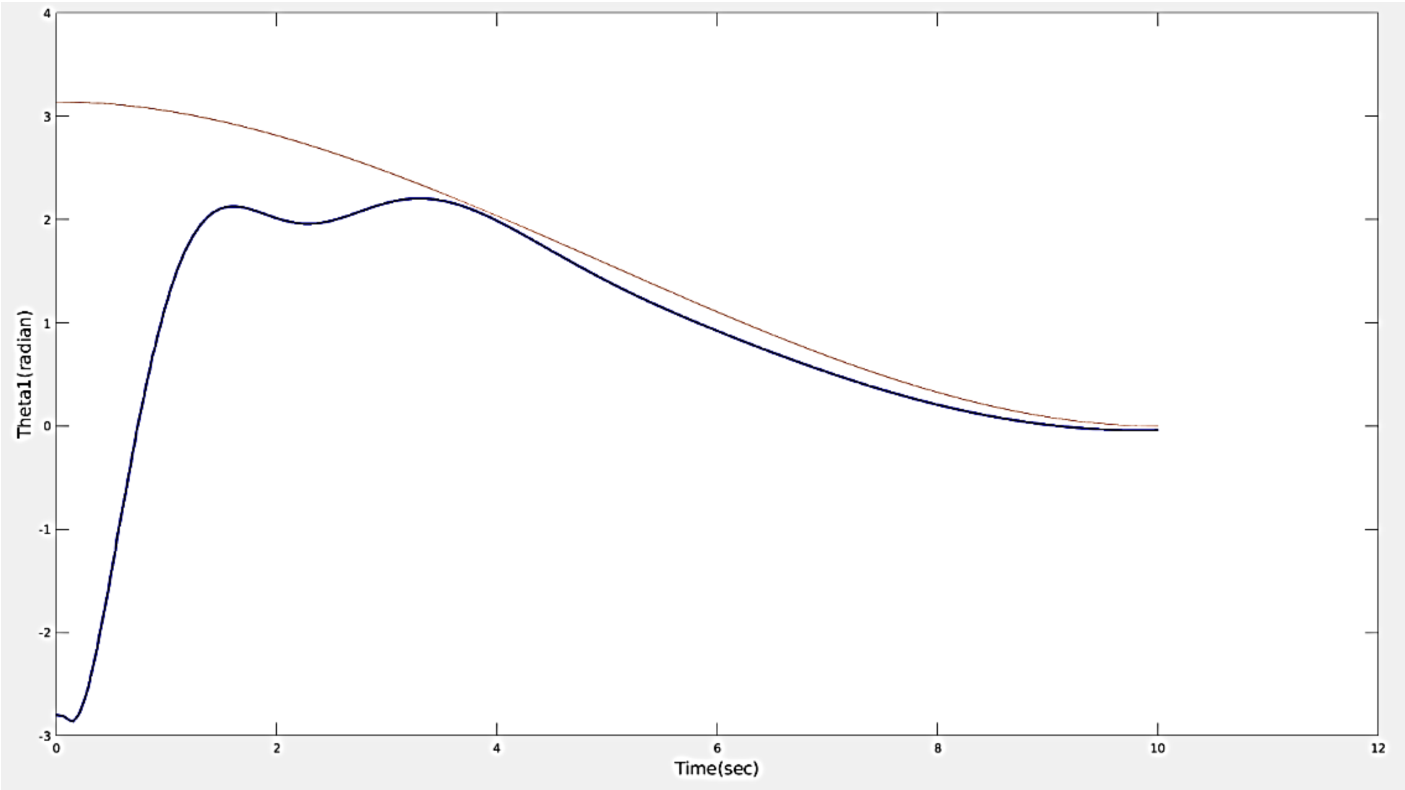


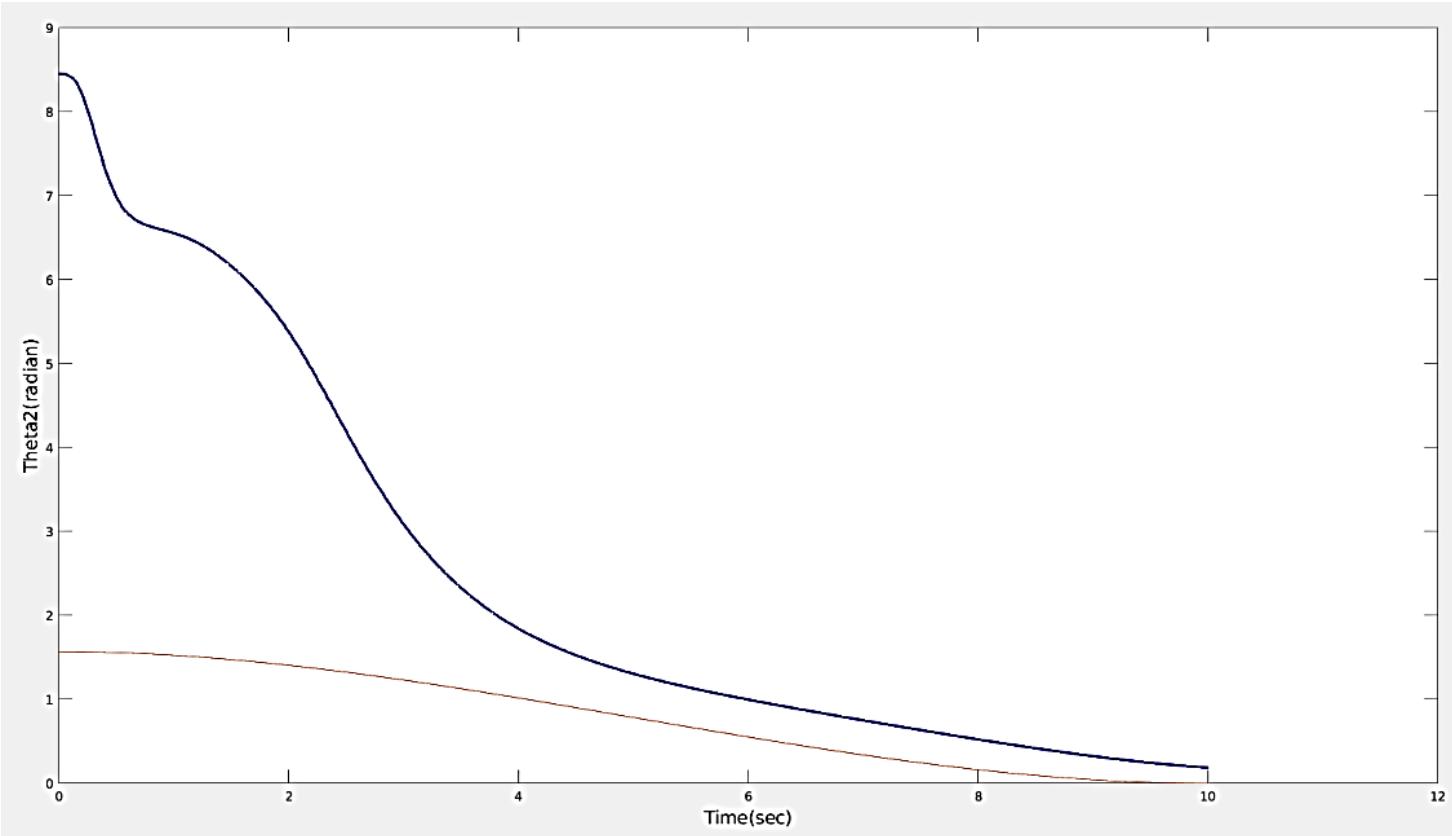
The light orange line shows the desired trajectory to which the output trajectory converges within the given time frame. The efforts are limited to as mentioned in the problem statement.



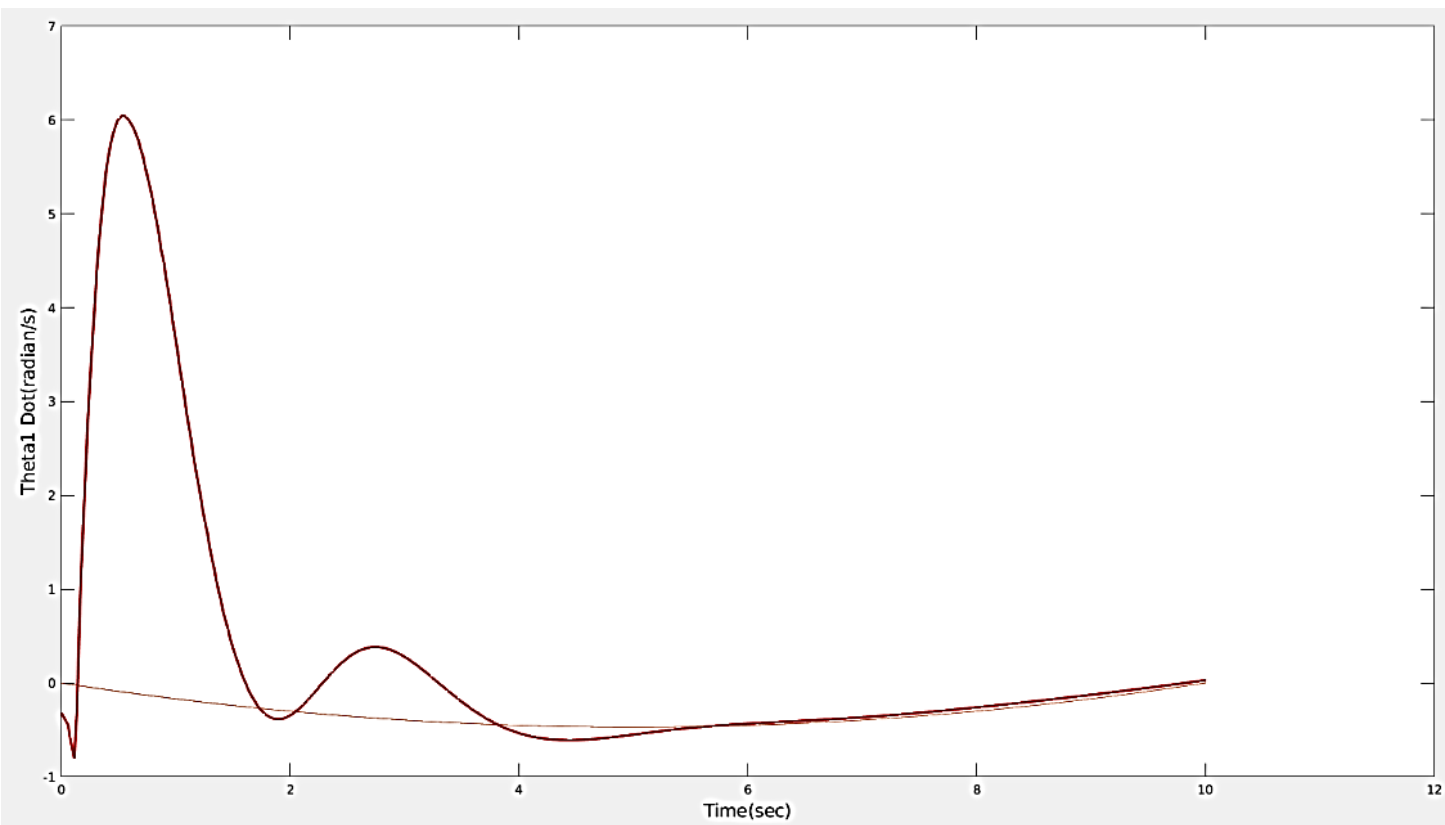


f) With the same conditions as above, the controller performance was tested on the RRBot in Gazebo and the following results were obtained:

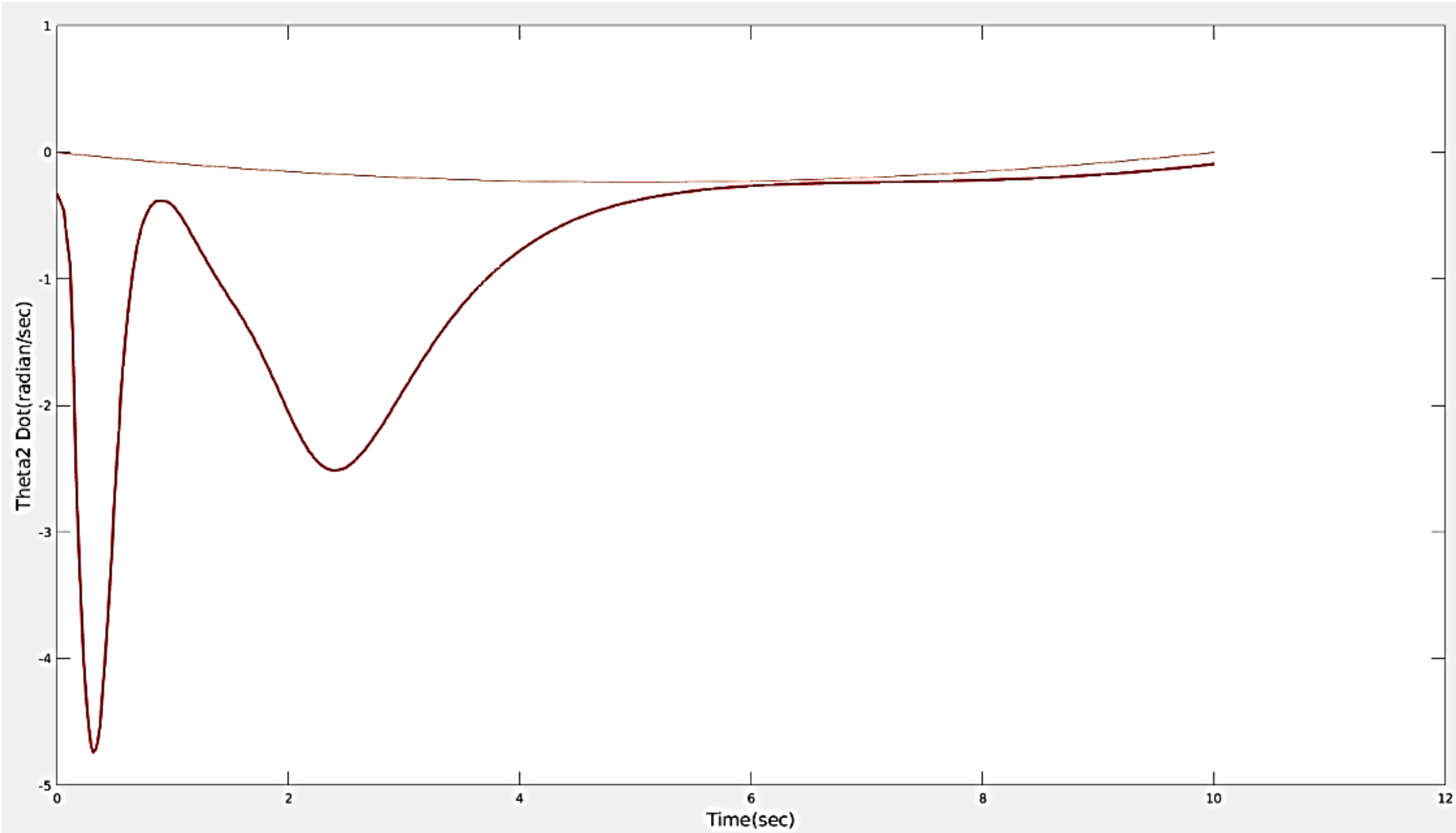


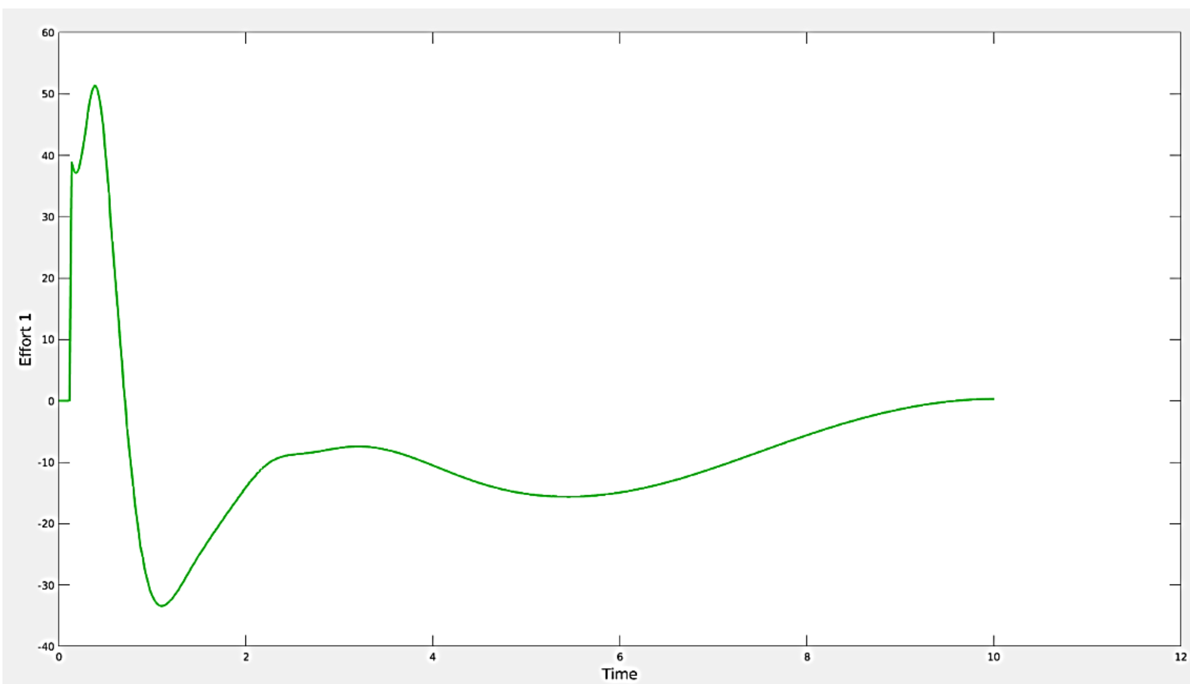


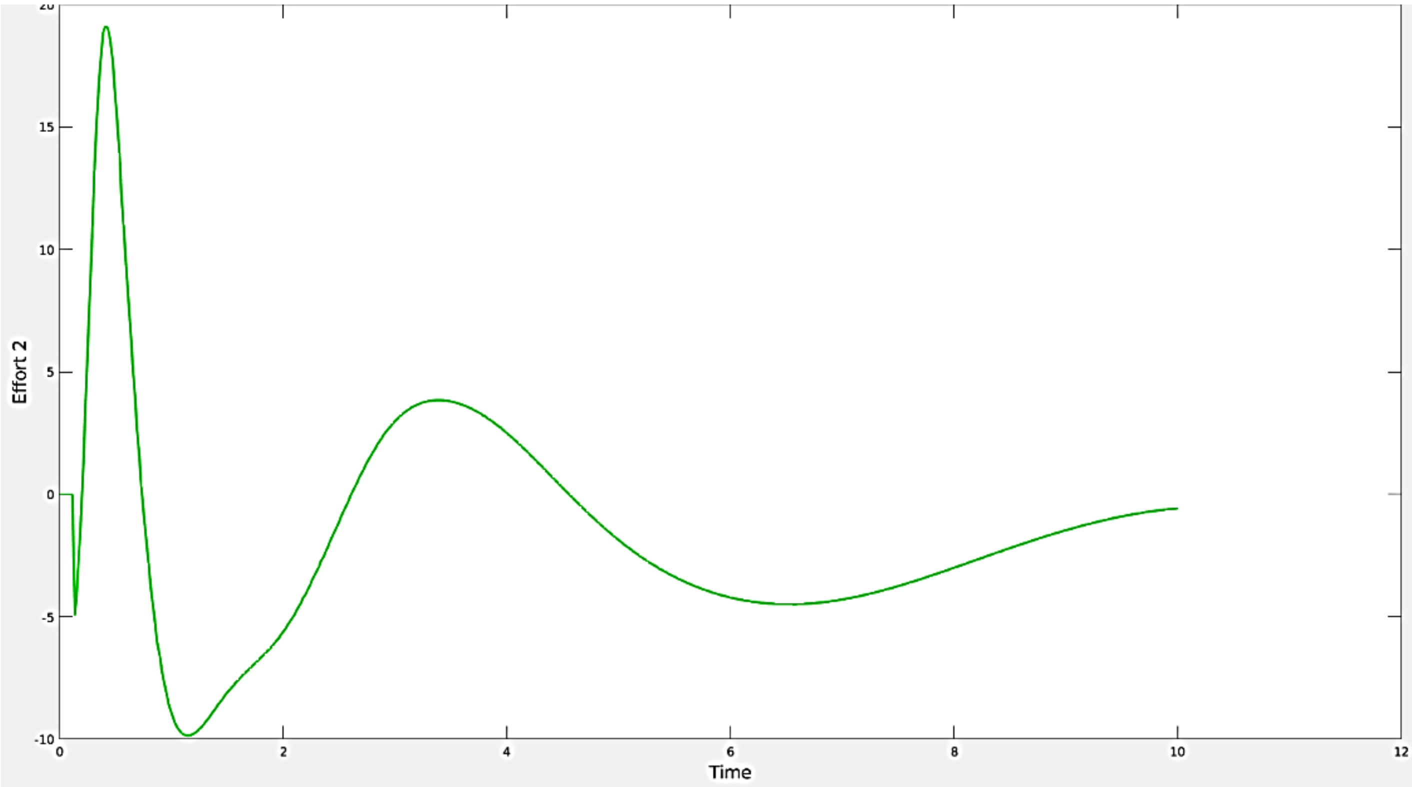
Theta2(radian)
Time(sec)
0
1
2
3
4
5
6
7
8
9
10
12



Theta1 Dot(radian/s)
Time(sec)
-1
0
1
2
3
4
5
6
7
8
10
12


Theta2 Dot(radian/sec)
Time(sec)
1
0
-1
-2
-3
-4
-5
0
2
4
6
8
10
12


Effort 1
Time
60
50
40
30
20
10
0
-10
-20
-30
-40
0
2
4
6
8
10
12



We observe that for the computed gains under our feedback linearization control law the output trajectories of the two joints in terms of their positions and angular velocities both converge to the desired trajectory as time tends to 10s.