Robot Control Programming Assignment 3

Submitted by: Purna Patel

WPI ID: 150062312

Part a:

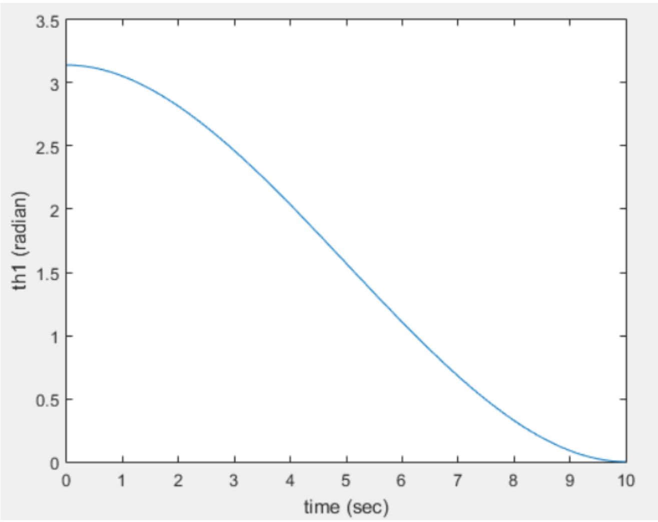
- In order to generate cubic polynomial trajectory for the two joints of the robot, initially the initial and final conditions of the trajectory were defined as mentioned in the assignment.
- A symbolic variable t was defined.
- Then using the general form of the cubic polynomial trajectory and its derivative, matrices A, a and B were defined such that A*a = B.
- A is a 4x4 matrix containing elements of trajectory substituted with t_0 and t_f.
- a is a 4x2 matrix consisting all the coefficients of both the trajectories (column 1 consists coefficients for trajectory for joint 1 and column 2 for joint 2 respectively) which are to be found.
- B is a 4x2 matrix consisting initial and final parameters of both trajectories.
- Now by calculating a = (A^{-1}) B, value of all the coefficients are calculated.
- Following are the final equations of the trajectories obtained in the matlab.

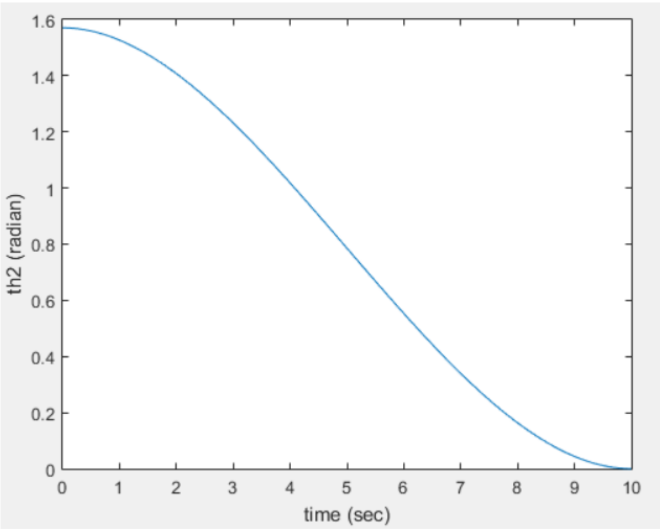
```
th1 = (pi*t^3)/500 - (3*pi*t^2)/100 + pi

th2 = (pi*t^3)/1000 - (3*pi*t^2)/200 + pi/2
```

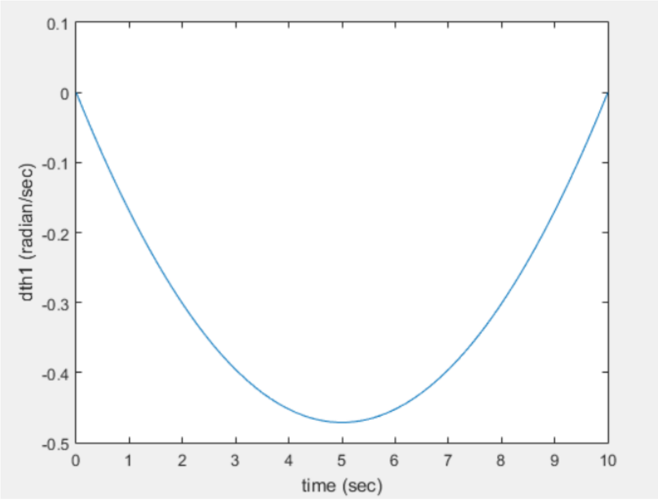
- On differentiating these equations with respect to time, equations for angular velocities and angular accelerations can be found as follows.

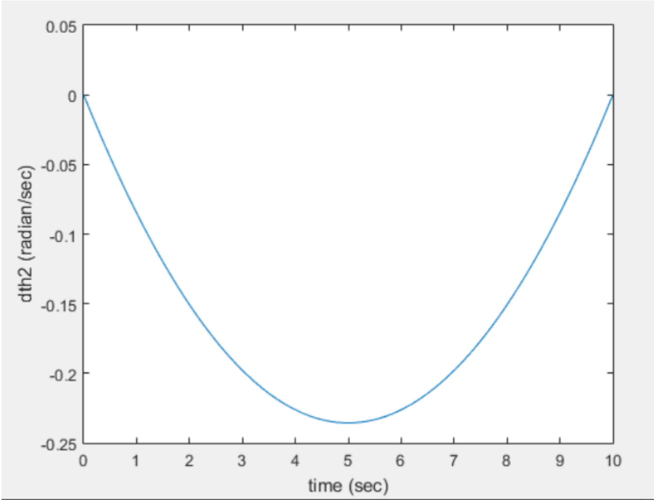
```
dth1 = (3*pi*t^2)/500 - (3*pi*t)/50

dth2 = (3*pi*t^2)/1000 - (3*pi*t)/100

ddth1 = (3*pi*t)/250 - (3*pi)/50

ddth2 = (3*pi*t)/500 - (3*pi)/100
```

- Following are the plots of the obtained trajectory.









Part b:

➢ Using the dynamic equations derived in previous assignments, matrices M(q), C(q, q·) and g(q) are obtained as follows.

```
M = [(m1*d1^2 + m2*d2^2 + 2*m2*cos(th2)*d2*l1 + m2*l1^2 + I1 + I2),
(m2*d2^2 + l1*m2*cos(th2)*d2 + I2);
    (m2*d2^2 + l1*m2*cos(th2)*d2 + I2), (m2*d2^2 + I2)];

C = [-(2*d2*dth2*l1*m2*sin(th2)), -(d2*dth2*l1*m2*sin(th2));
    (d2*l1*m2*sin(th2)*dth1), 0];

G = [(- sin(th1)*(d1*g*m1 + g*l1*m2) - d2*g*m2*sin(th1 + th2));
    (- d2*g*m2*sin(th1 + th2))];


u = M*ddq+C*dq+G;
```

Where u = [u1; u2], dq = [dth1; dth2], ddq = [ddth1, ddth2]

Part c:

➢ In order to design a stated feedback linearized controller, initially the feedback control input tau was considered as `tau = M*v+C*dq+G`
➢ Thus taking v = ddq
➢ Converting this equation into state space, we obtain A and B as follow.

```
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];
```

➢ Now lamda were assigned and using place function, values of elements of matrix k were found.

- Now using defined symbolic representations of states (i.e. matrix of th1, th2, dth1 and dth2), desired states (i.e. matrix of des_th1, des_th2, des_dth1 and des_dth2) and desired control input vd, v can be calculated using following command.

```
v = -k*(states-des_states)+vd;
```

- The desired states and vd are obtained from the trajectory.
- The final control law obtained is

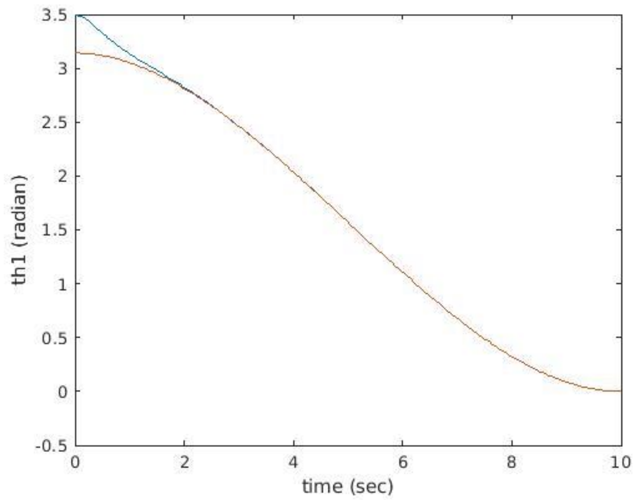
```
tau = M*v+C*dq+G.
```

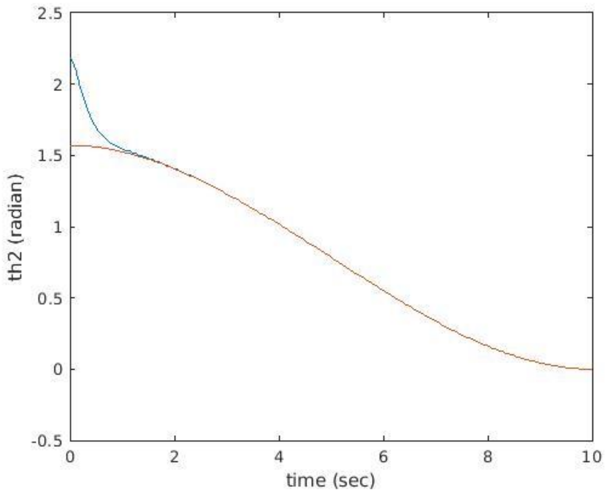
Part d:

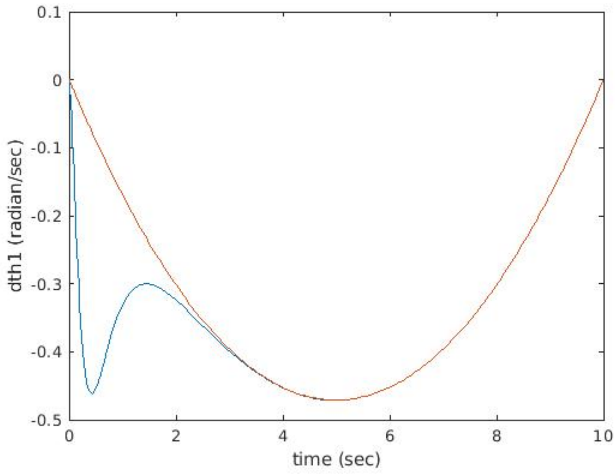
- Now in order to make the system follow a trajectory, the equations of the trajectory are added in the ode function from which decided states are obtained.
- The control law designed in part c is substituted in the ode function and the final control input is updated accordingly.

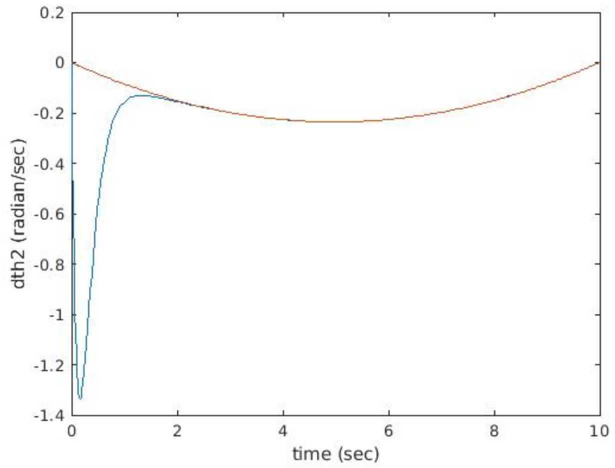
Part e:

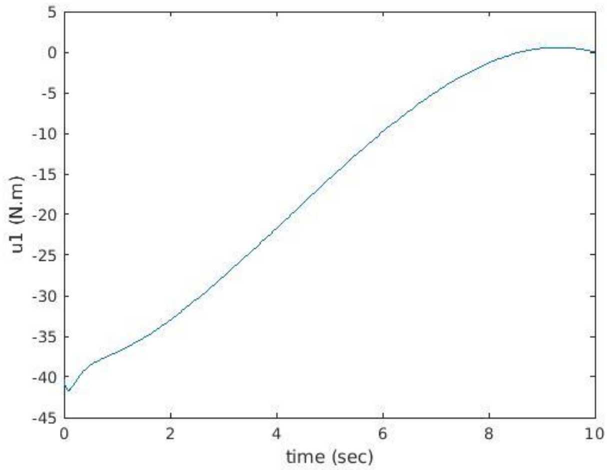
- Now using the ode function, simulation is run by providing the initial conditions th1 = 200^o, th2 = 125^o, dth1 = 0 and dth2 = 0
- Following plots were obtained from the simulation.

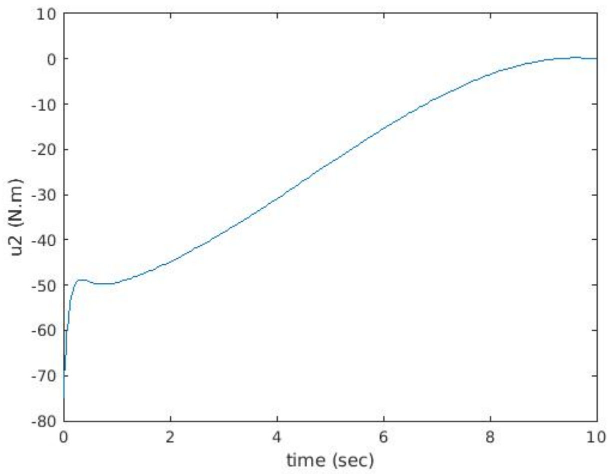








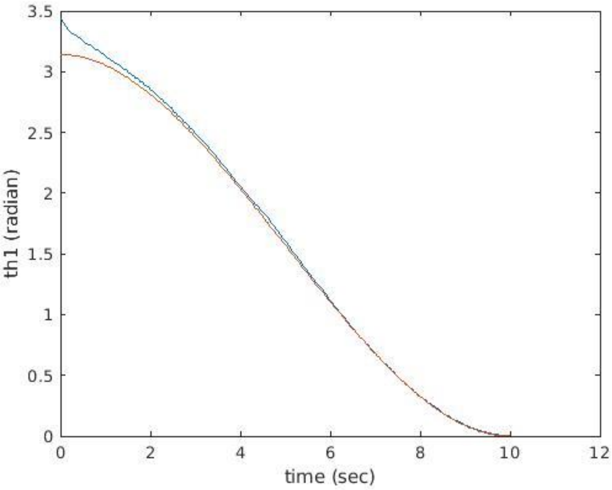


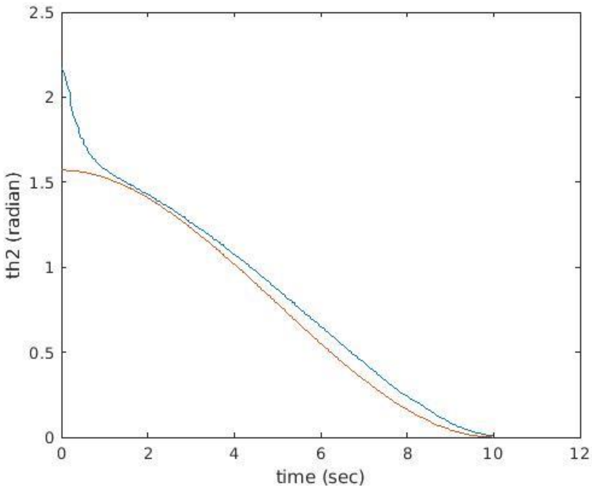


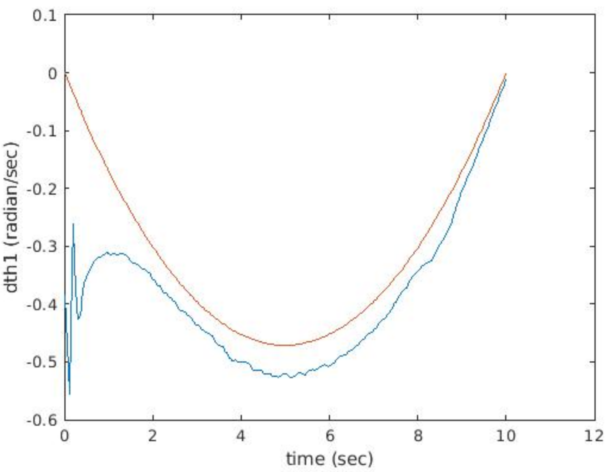
- Here the orange line represents the actual trajectory and the blue represents the actual output.
- It can be seen that initially the starting points of the system and the trajectory are different but as time proceeds, the system merges to the trajectory.

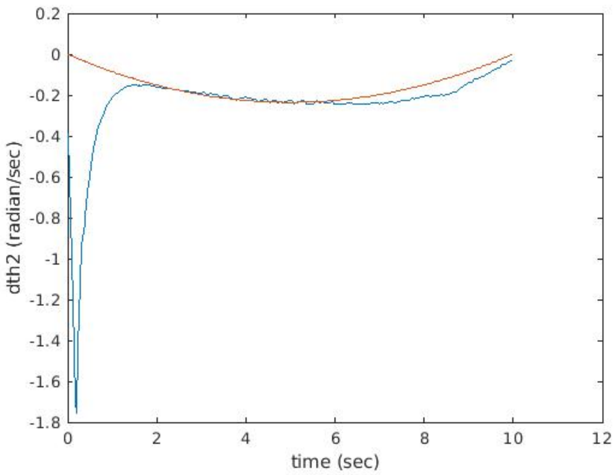
Part f:

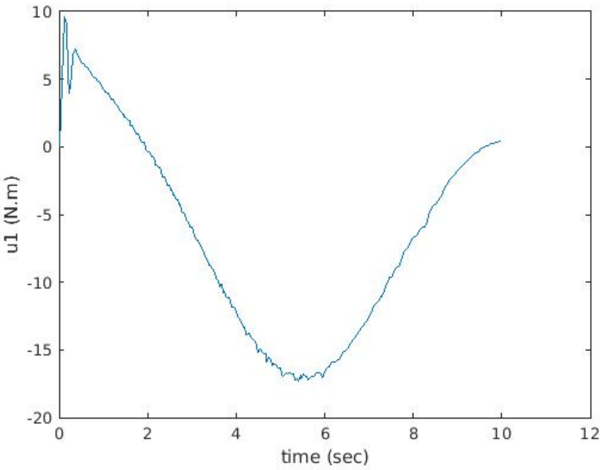
- New rrbot_traj_control file was created in order to test the controller and trajectory in gazebo simulation.
- The equations of the trajectory were added in order to obtain desired states and the control law was implemented inside the while loop.
- The initial starting conditions were provided as th1 = 200^o, th2 = 125^o, dth1 = 0 and dth2 = 0.
- The result obtained were stored and plotted in the end.
- The controller was successfully able to control the system through the trajectory.
- Following are the plots obtained.

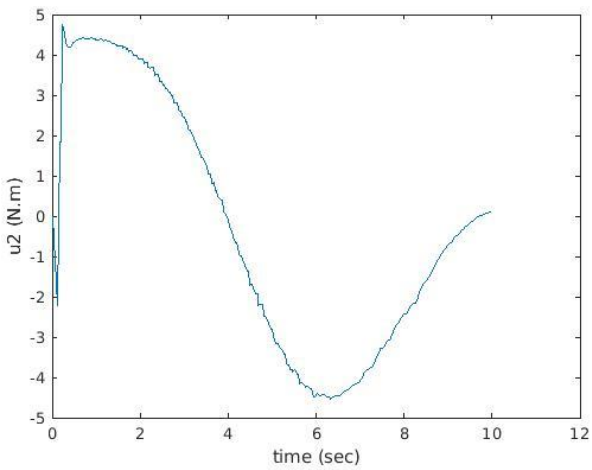












- The orange curve represents the trajectory and blue curve represents the actual output of the system.
- It can be seen from the plots that the starting points of the trajectory and the system were different but as the time proceeds, the system converges to the trajectory.