Robot Control Programming assignment 2

Submitted by: Purna Patel
WPI ID: 150062312

Part a:

- In order to find the equilibrium points, initially the values of theta_dot1, theta_dot2, T1 and T1 are substituted 0 using subs function.
- Taking the equations ==0, and solving for theta1 and theta2, we get value of thetas at equilibrium.
- But the solve function returns the first solution it gets as there are infinite possible of thetas. So in order to derive other values, we need to let the function search next equilibrium value neglecting the already achieved value.
- In order to achieve this, a nested for loop was implemented in the matlab code.
- The four equilibrium points obtained are (0, 0), (0, pi), (pi, 0) and (pi, pi).
- Please run the code in the newer version of matlab, because in the older versions, the solve function do not take 4 conditions which I used in my code in order to find the equilibrium.

Part b:

- Jacobian Linearization is done using jacobian function in matlab.
- Substituting the given values m1, m2, l1, l2, r1, r2 I1, I2 and g, and using jacobian function, Matrices A and B are obtained in symbolic structure.

```
A = jacobian (dX, x.');
B = jacobian (dX, u.');
```

Part c:

- In order to investigate stability at each equilibrium point, the value of equilibrium points are substituted instead of states and the eigen values are derived using eig function.
- Following are the obtained Eigen values:

eigenA1 = (0,0)

```
  7.1676
  2.7129
 -7.1676
 -2.7129
```

eigenA2 = (pi,0)

0.0000 + 7.1676i
0.0000 - 7.1676i
0.0000 + 2.7129i
0.0000 – 2.7129i

eigenA3 = (0,pi)

3.8995 + 0.0000i
-3.8995 + 0.0000i
-0.0000 + 4.9864i
-0.0000 – 4.9864i

eigenA4 = (pi,pi)

-4.9864 + 0.0000i
-0.0000 + 3.8995i
-0.0000 - 3.8995i
4.9864 + 0.0000i

- From the above mentioned eigen valued, we can conclude that all the states expect (pi,0) are unstable.
- But The state (pi,0) is marginally stable (All the real part of eigen values are less than or equal to zero).

Part d:

- Controllability matrix obtained around state (0,0) using ctrb command is as follows:

C =

0	0	1.7250	-4.4345	0	0	74.7378	-233.8759
0	0	-4.4345	14.8902	0	0	-233.8759	762.3251
1.7250	-4.4345	0	0	74.7378	-233.8759	0	0
-4.4345	14.8902	0	0	-233.8759	762.3251	0	0

- Rank 4 was obtained using Rank command, which is full rank so the system is controllable at state (0,0) which is the upward configuration.

Part e:

- For this assignment the values of lamda selected are [-1, -2, -2+i, -2-i]
- Substituting A and B matrices and lamda in the place function, the value of matrix k was obtained as follows:

k =

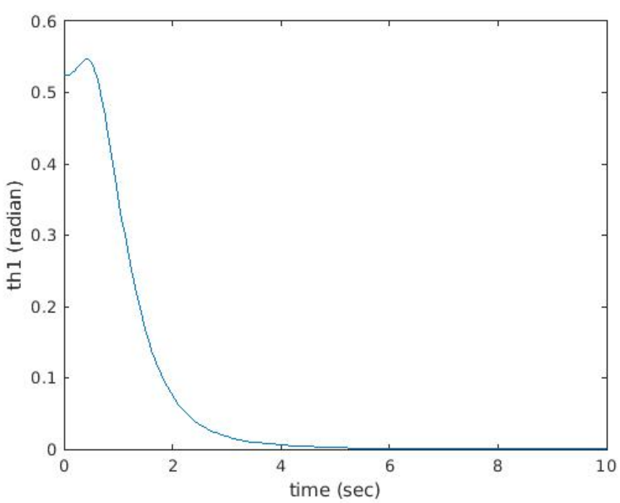
24.9780	5.0173	7.8833	1.9194
6.1250	4.8392	2.2887	0.8101

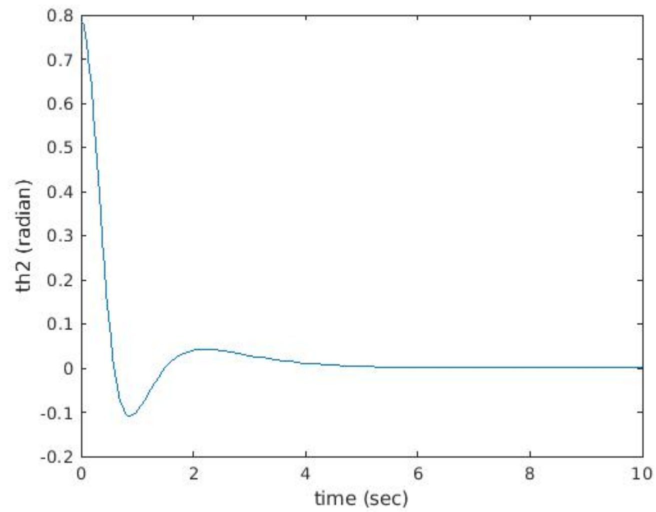
Part f:

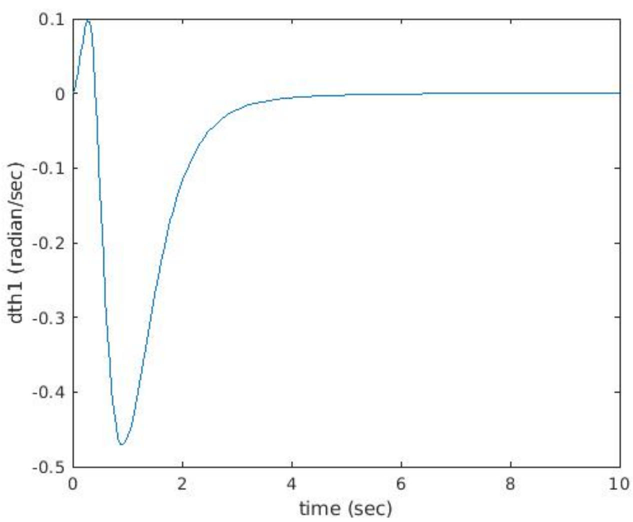
- This k was substituted in the ode function and the values of u1 and u2 were replaced with the state feedback controlled value.

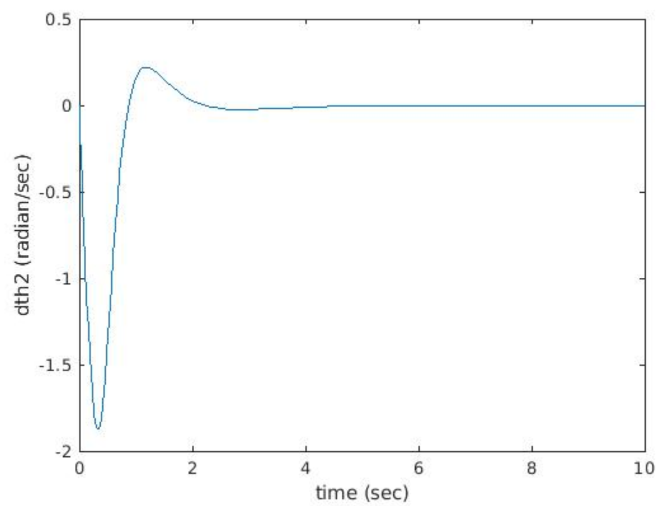
- u = -k*x

- The outputs were plotted for initial conditions (30, 45, 0, 0).



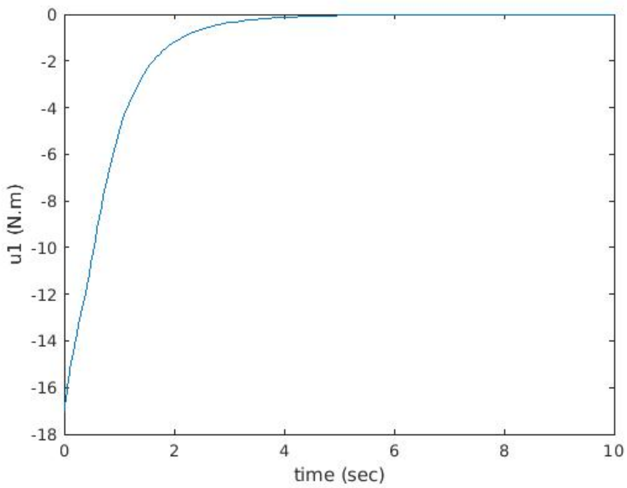


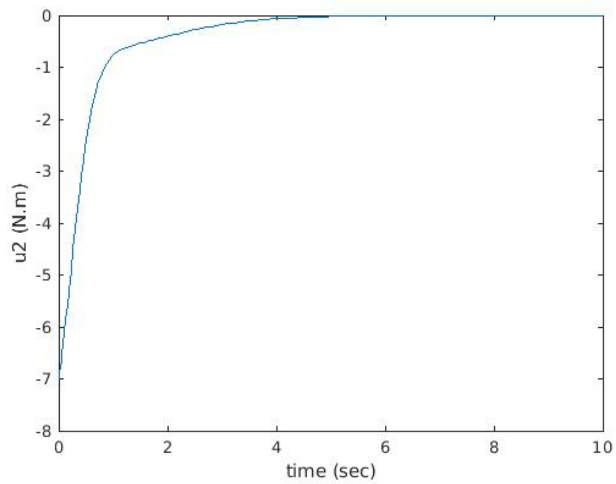




- All the states successfully converges to zero.

- To plot the efforts u1 and u2, there values are calculated and stored in a matrix using for loop and plotted. The plot obtained are as follows:



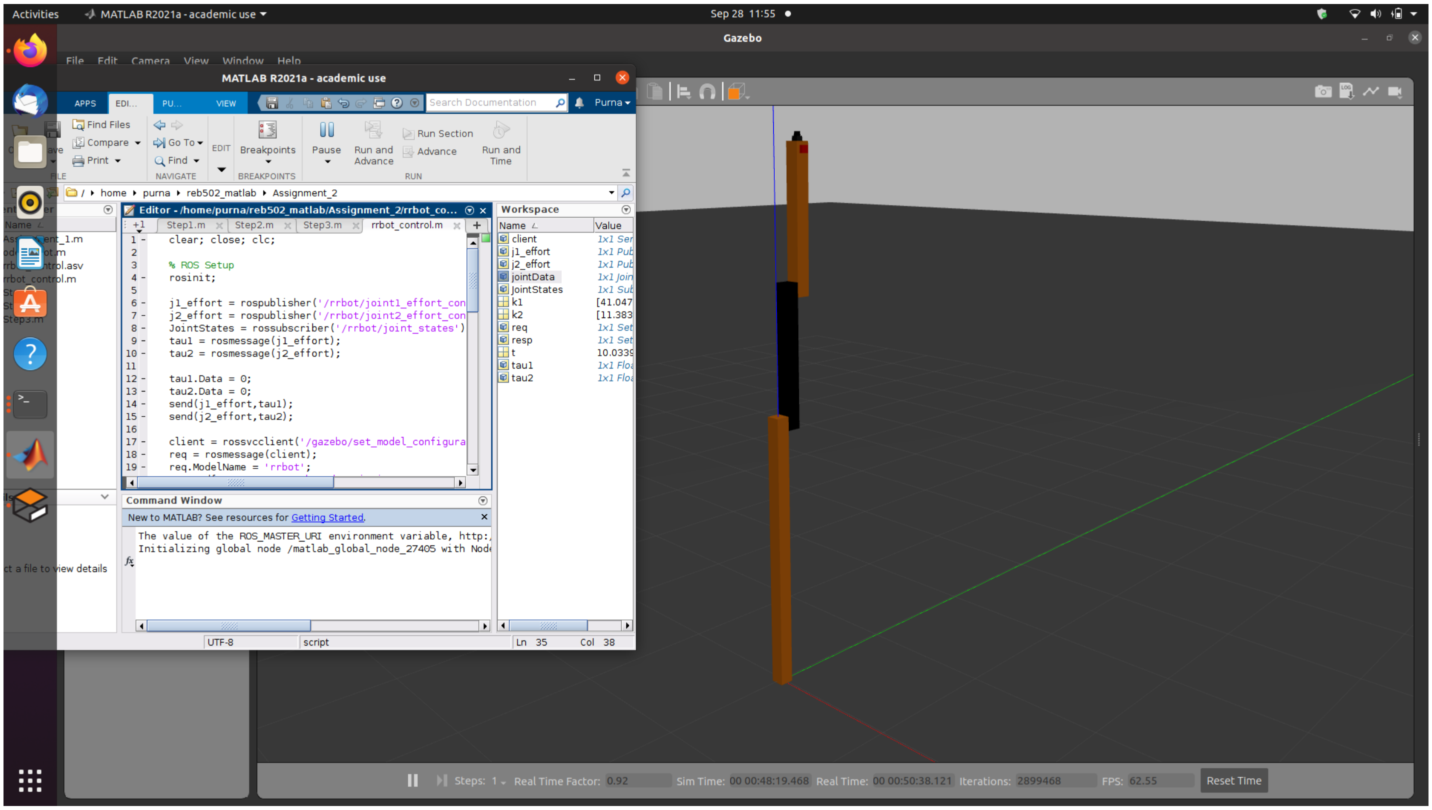


- u1 and u2 also successfully converges to zero.

Part g:

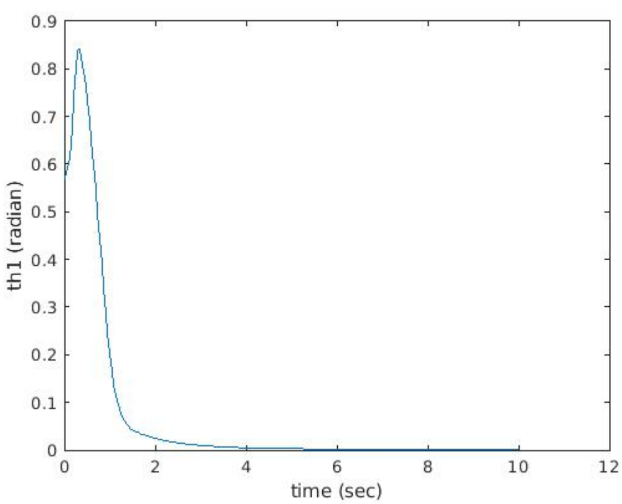
- In order to run the controller in the gazebo, the given code was copied in the matlab and the values of the state feedback controller were substituted.

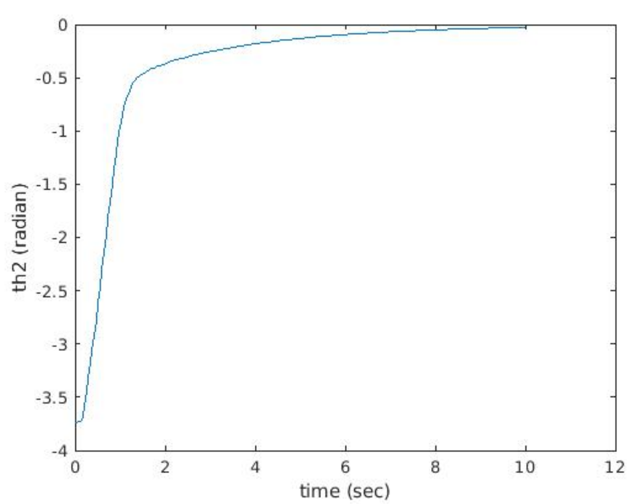
- The controller was successfully able to control the robot in the upward position.

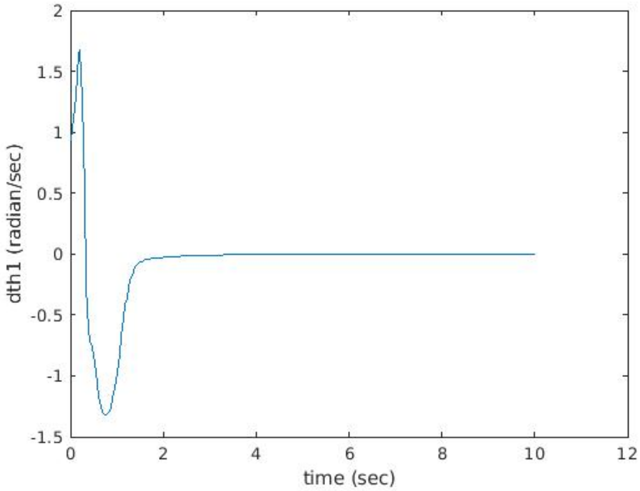


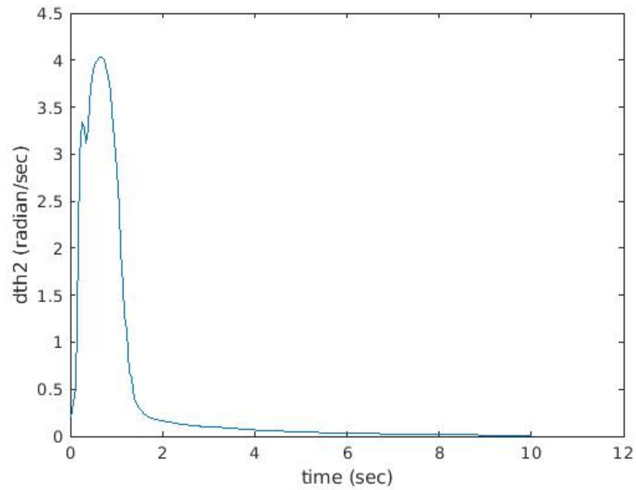
- The values of the states and the effort were stored in various matrices and were plotted.

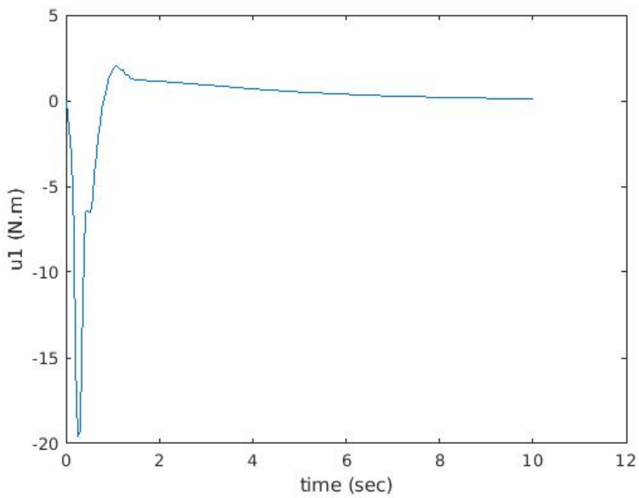
- The plots obtained were as follows.

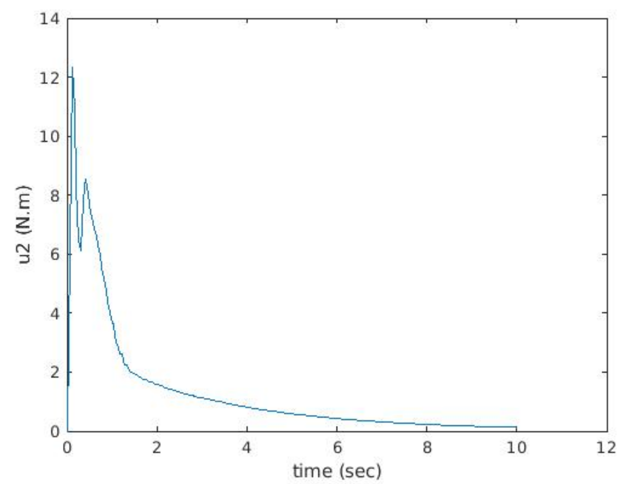












- As shown in the plots, the robot successfully stabilizes in the simulation proving the controller works.
- The curves obtained are not that smooth showing the variations and noise in the system.
- A weird problem was faced during the simulation was when the RRbot was initialized, the initial value of theta2 was automatically taken (-360 + 45), instead of 45.
- So the arm rotates in clockwise direction in order to compensate the negative value.
- This can be observed in the plots too, in the th2 vs time plot, the starting position of the arm is between -3.5 and -4 radian instead of the given position which is +0.785 (45 degrees).
- But still the controller is able to stabilize the robot at upward position.