Problem a: Equilibrium Points:

```
>> RRBOT
********** EQUILIBRIUM POINTS **********
values of theta1:
 0
pi
 0

values of theata2:
 0
 0
pi
```

Equilibrium points are –

[theta1, theta2, theta1_dot, theta2_dot] = [0, 0, 0, 0]

[theta1, theta2, theta1_dot, theta2_dot] = [pi, 0, 0, 0]

[theta1, theta2, theta1_dot, theta2_dot] = [0, pi, 0, 0]

- One solution that was not returned by MATLAB but is still an equilibrium point is – [theta1, theta2, theta1_dot, theta2_dot] = [pi, pi, 0, 0]

Problem b: Jacobian Linearization

```
********** LINEARIZATION AROUND EQUILIBRIUM POINTS **********
----Equilibrium Point (0,0,0,0):
State Matrix (A):
         0         0    1.0000         0
         0         0         0    1.0000
   12.5769  -11.9611         0         0
  -16.9227   46.1565         0         0

Input Matrix (B):
         0         0
         0         0
    1.7250   -4.4345
   -4.4345   14.8902

----Equilibrium Point (pi,0,0,0):
State Matrix (A):
         0         0    1.0000         0
         0         0         0    1.0000
  -12.5769   11.9611         0         0
   16.9227  -46.1565         0         0

Input Matrix (B):
         0         0
         0         0
    1.7250   -4.4345
   -4.4345   14.8902
```

```
----Equilibrium Point (0,pi,0,0):
State Matrix (A):
         0         0    1.0000         0
         0         0         0    1.0000
   12.5769  -11.9611         0         0
   -8.2310  -22.2343         0         0

Input Matrix (B):
         0         0
         0         0
    1.7250    0.9845
    0.9845    4.0522
```

```
----Equilibrium Point (pi,pi,0,0) [EXTRA SOLUTION - NOT RETURNED BY MATLAB]:
State Matrix (A):
         0         0    1.0000         0
         0         0         0    1.0000
  -12.5769   11.9611         0         0
    8.2310   22.2343         0         0

Input Matrix (B):
         0         0
         0         0
    1.7250    0.9845
    0.9845    4.0522
```

Problem c: Investigating Stability

```
********** CHECKING STABILITY **********
-- Eigen values of (A) around equilibrium point (0,0,0,0):
    7.1676
    2.7129
   -7.1676
   -2.7129
```

The system is UNSTABLE near the equilibrium point (0, 0, 0, 0)

```
-- Eigen values of (A) around equilibrium point (pi,0,0,0):
  -0.0000 + 7.1676i
  -0.0000 - 7.1676i
   0.0000 + 2.7129i
   0.0000 - 2.7129i
```

The stability of the system near the equilibrium point (pi, 0, 0, 0) is INCONCLUSIVE.

```
-- Eigen values of (A) around equilibrium point (0,pi,0,0):
  -3.8995 + 0.0000i
   3.8995 + 0.0000i
   0.0000 + 4.9864i
   0.0000 - 4.9864i
```

The system is UNSTABLE near the equilibrium point (0, pi, 0, 0)

```
-- Eigen values of (A) around equilibrium point (pi,pi,0,0) -- EXTRA SOLUTION:
   4.9864 + 0.0000i
  -0.0000 + 3.8995i
  -0.0000 - 3.8995i
  -4.9864 + 0.0000i
```

The system is UNSTABLE near the equilibrium point (pi, pi, 0, 0)

Problem d: Controllability around upward equilibrium (0,0,0,0)

```
********** INVESTIGATING CONTROLLABILITY AT UPWARD CONFIGURATION **********
rank of controllability matrix:      4
```

The controllability matrix is a full-rank matrix hence the system is CONTROLLABLE near the upward equilibrium.

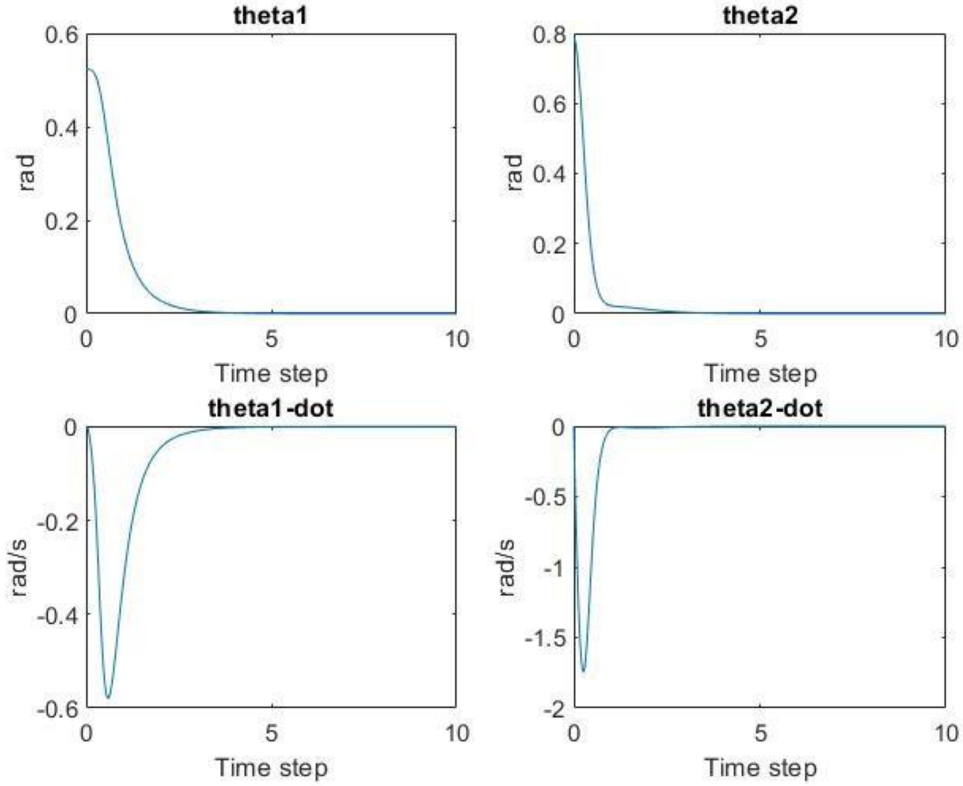
Problem e: Selected Eigen Values and resultant K-gains

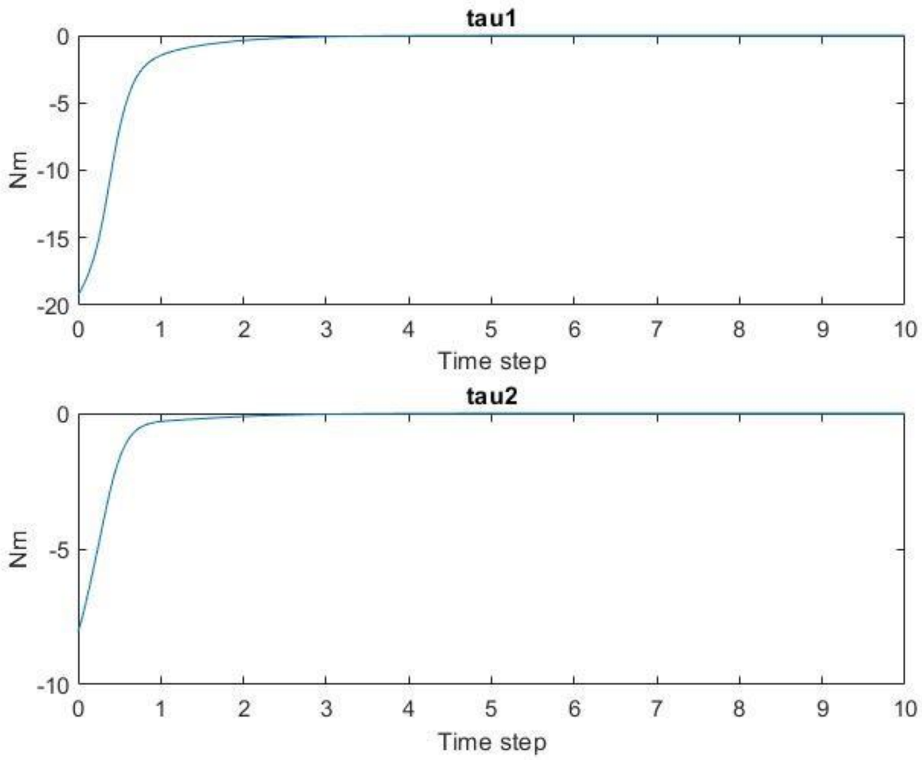
```
lambda = [-1.4,-2.6,-3.6,-6.7];
λ1 = -1.4 λ1 = -2.6 λ1 = -3.6
λ1 = -6.7
```

```
********** K Gains **********
K =

   37.2190   -0.3267   14.4845    0.7429
    9.5519    3.8883    4.1765    0.7474
```

<u>Problem f:</u> Simulation in MATLAB





Range of Maximum torc values (at the beginning) -

Editor - main.m

U

2x173 double

	1	2	3	4
1	-19.2312	-19.2312	-19.2312	-19.2311
2	-8.0552	-8.0552	-8.0552	-8.0551
3				
4				

Initial State Variables [theta1, theta2, theta1_dot, theta2_dot] in column wise order

y

173x4 double

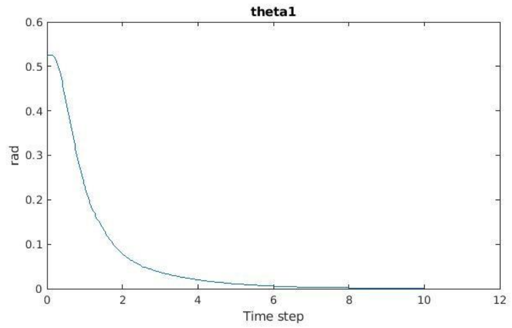
	1	2	3	4
1	0.5236	0.7854	0	0
2	0.5236	0.7854	5.8273e-07	-5.0237e-05
3	0.5236	0.7854	1.1653e-06	-1.0047e-04
4	0.5236	0.7854	1.7476e-06	-1.5071e-04
5	0.5236	0.7854	2.3298e-06	-2.0094e-04

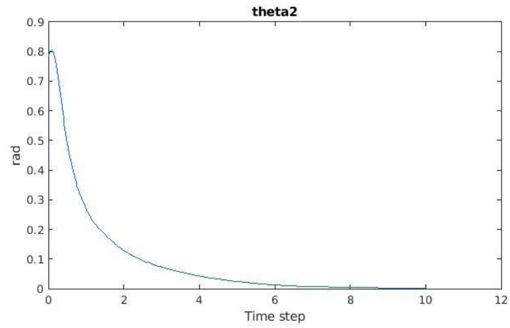
Final State Variables [theta1, theta2, theta1_dot, theta2_dot] in column wise order

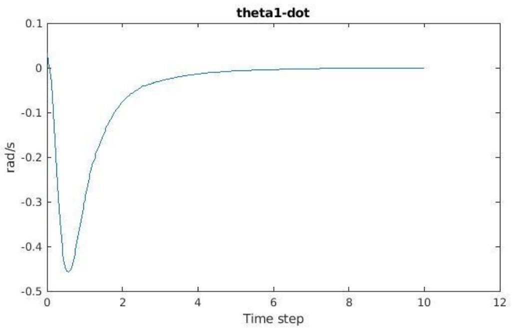
173x4 double

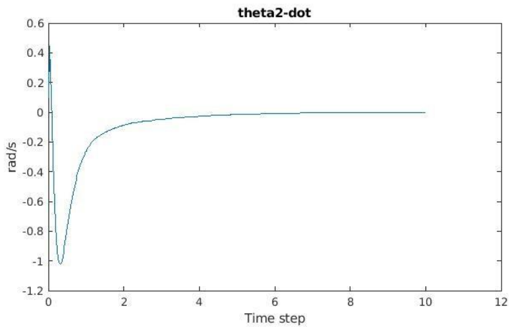
	1	2	3	4
172	4.0533e-07	2.9585e-07	-5.6355e-07	-4.1954e-07
173	3.1714e-07	2.3484e-07	-4.3251e-07	-3.4471e-07
174				
175				

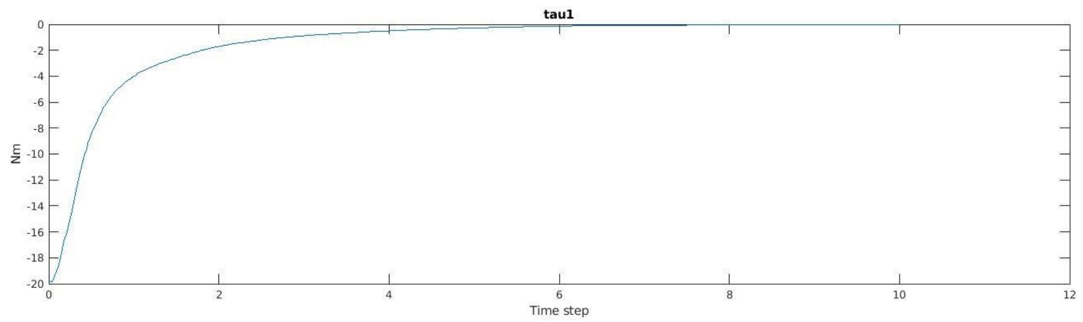
Problem f: Simulation in GAZEBO

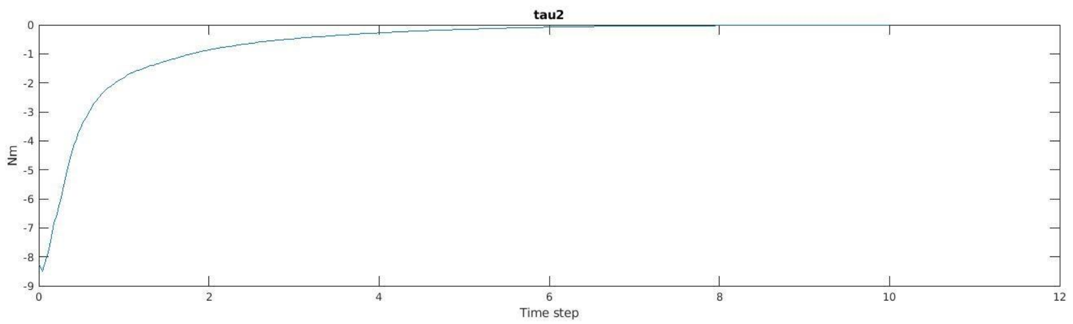












Range of Maximum torc values (at the beginning) -

X1 | X2 | X3 | X4 | U | TAU1

1x382 double

	1	2	3	4	5	6
1	-19.7848	-19.8323	-18.9955	-18.5288	-17.9670	-16.6
2						

Editor - rrbot_control.m

X1 | X2 | X3 | X4 | U | TAU1 | TAU2

1x382 double

	1	2	3	4	5	6
1	-8.2685	-8.5129	-8.0234	-7.7491	-7.4603	-6.8474
2						
3						
4						
5						

Final State Variables [theta1, theta2, theta1_dot, theta2_dot] in column wise order

X1

1x382 double

	374	375	376	377	378	379	380	381	382
1	.6612e-...	6.5772e-...	6.4943e-...	6.4126e-...	6.3322e-...	6.2527e-...	6.1744e-04	6.0974e-04	6.0212e-04
2									

X1 | X2

1x382 double

	374	375	376	377	378	379	380	381	382
1	0.0015	0.0015	0.0015	0.0015	0.0014	0.0014	0.0014	0.0014	0.0014
2									

X1 | X2 | X3

1x382 double

	374	375	376	377	378	379	380	381	382
1	4.2269e...	-4.1767e...	-4.1156e...	-4.0522e...	-3.9976e...	-3.9445e...	-3.8810e...	-3.8247e-04	-3.7903e-04
2									

Editor - rrbot_control.m

Variables - X4

X1 | X2 | X3 | X4

1x382 double

	374	375	376	377	378	379	380	381	382
1	9.5236e...	-9.3768e...	-9.2783e...	-9.1879e...	-9.0880e...	-8.9844e...	-8.9125e...	-8.8210e-04	-8.6618e-04
2									

<u>Discussion on ROS results:</u>

The output graphs generated by simulating the RRBot in GAZEBO is different from that of the one found in MATLAB because of multiple reasons which are listed below.

- There is some sort of damping (Friction or Air Drag) that is present in the Gazebo environment which is evident by the fact that the robot comes to rest after oscillating for some time when no control input is applied. This is not considered in our dynamic modelling of the system and we get little higher torques in ROS-GAZEBO implementation as compared to MATLAB implementation.

- We can also observe that in our MATLAB implementation after applying the control input the states converge very close to absolute zero, i.e in the power of 10e-7. And we can also observe that in the GAZEBO implementation we see that the convergence is little far form zero. We see that it is in the order of 10e-4. This can be because of the presence of some damping factor in the GAZEBO environment.

- We can also observe that the initial velocity in the GAZEBO implementation is not Zero. This is because there is a difference between the start time of the GAZEBO simulation and the application of Control input which leads to some gain in velocity due to free-fall under gravity till the time the controller kicks-in. This velocity gain at the beginning can also be the reason for higher initial torques then those found in the MATLAB implementation.

- However, the GAZEBO implementation shows that our controller is applicable and works perfectly on a real-world system.