Final Project Report

ENPM808Q : Control of Robotic Systems
University of Maryland at College Park

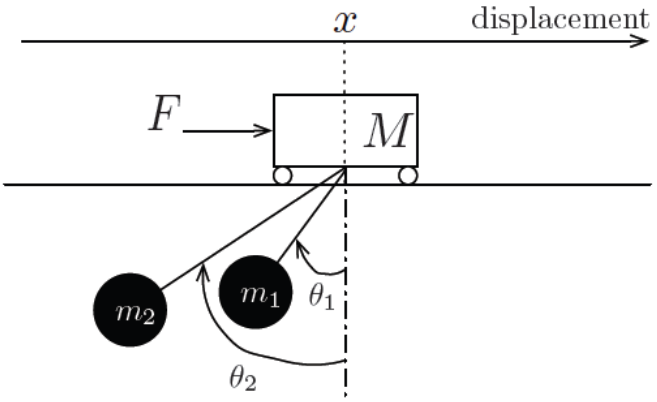
Ankur Jain (UID 114443489) Chirag Majithia (UID 113851896)

TABLE OF CONTENTS

Problem Overview

Given:

- A cart of Mass M on which we can apply a controlled force F(t)
- There are 2 masses, m1 and m2 suspended to the cart with mass-less string of length l1 and l2
- X is the displacement of the cart from a reference. Similarly θ_1 and θ_2 are the angles formed by masses m1 and m2 with the reference axis attached to the cart.



Things to Do:

- Check for observability of the system for given set of output vectors
- Create a Luenberger Observer for the system with the output vectors for which it was observable
- Design Output Feedback Controller for the System

Things we know and are superficially explained in the report

- Design of LQR and use of Q and R in getting controller gain matrix K
- Linearized State Space Model of the given system for equilibrium point θ_1 and $\theta_2 = 0$ and any point x
- The linearized state-space model of the given system was derived to be:

$$\dot{\overrightarrow{X(t)}} = A\overrightarrow{X(t)} + B\overrightarrow{U(t)}$$

$$\vec{Y}(t) = C\overrightarrow{X(t)} + D\overrightarrow{U(t)}$$

$$\dot{\overrightarrow{X(t)}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m1g/M & 0 & -m2g/M & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -(M+m1)g/Ml1 & 0 & -m2g/Ml1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -m1g/Ml2 & 0 & -(M+m2)g/Ml2 & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta_1}(t) \\ \theta_2(t) \\ \dot{\theta_2}(t) \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 1/Ml1 \\ 0 \\ 1/Ml2 \end{bmatrix} [F(t)]$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m1g/M & 0 & -m2g/M & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -(M+m1)g/Ml1 & 0 & -m2g/Ml1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -m1g/Ml2 & 0 & -(M+m2)g/Ml2 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 1/Ml1 \\ 0 \\ 1/Ml2 \end{bmatrix}$$

Matrices C and D are set according to required output

We have used M = 1000Kg, $m_1 = m_2 = 100$Kg, $l_1 = 20$m, $l_2 = 10$m g = 9.8 m/s^2

LQR Design Overview

For a given system $\dot{\overrightarrow{X(t)}} = A\overrightarrow{X(t)} + B\overrightarrow{U(t)}$, $\vec{Y}(t) = C\overrightarrow{X(t)} + D\overrightarrow{U(t)}$ we want to design a control such that U(t) = K x(t). The design of K is a trade-off between the desired response and the control effort. The cost function for the system is given by $J = \{ \int_0^\infty x^T(t)Qx(t) + u^T(t)Ru(t)\, dt\}$

Q and R are weighting matrices such that it allows us to put individual emphasis on the different components of the state- and controllable input.

$$Q = \begin{matrix} Q11 & & & & & \\ & Q22 & & & & \\ & & Q33 & & & \\ & & & Q44 & & \\ & & & & Q55 & \\ & & & & & Q66 \end{matrix} \quad \text{corresponding to states} \quad \begin{matrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \end{matrix}$$

Since we have only one input :

R = R_{11} corresponding to control input U(t) = F(t)

Selection of Q and R

If a reference is given to the system, the Q will effectively weight the error i.e. difference between the current state and the desired value. For our system, the stable equilibrium points of $\theta 1$ and θ_2 are found to be at 2*K*pi radians where K $\epsilon \{0,1,2,..\}$. Whereas it was found that the system can be stable at any given X. Thus, we weight Q_{33} ~ Q_{66} more to maintain the system within linearizable range. And we weight Q_{22} little more than Q_{11} since we need the system to smoothly come to 0 at the equilibrium point {(Q_{22}) weights velocity of the cart}

In Matlab:

[K, S, e] = lqr(A,B, Q, R) solves the Riccati equation : $A' P + P A - P BR^{-1}B' P = -Q$ and gives $K = R^{-1}B' P$

The return values: −K is the state feedback gain matrix.
S is the solution of the algebraic Riccati equation.
e are the resulting closed loop eigenvalues (i.e. the eigenvalues of A − BK).

One of the sample values of R and Q that is used rigorously in following simulations:

Using : R = 0.00001 ; and Q = [10 0 0 0 0 0;
0 100 0 0 0 0;
0 0 1000*l1 0 0 0;
0 0 0 100*l1^2 0 0;
0 0 0 0 1000*l2 0;
0 0 0 0 0 100*l2^2];

[K,S,e] = lqr(A,B,Q,R)

We get following K matrix and Eigen values :

K = 1.0e+04 * [0.1000 0.5535 4.8697 5.5201 4.0664 -1.5867]

e = -5.3080 + 0.0000i -0.3616 + 0.0000i -0.3619 + 0.4519i -0.3619 - 0.4519i
-0.1575 + 0.8494i -0.1575 - 0.8494i

Observability

Test for observability:

1. Output Vector = {x(t)}

   i.e.

$$Y(t) = [x(t)]$$

   From Eqn 2, we solve to get Matrix C

$$\vec{Y}(t) = [x(t)] = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta}_1(t) \\ \theta_2(t) \\ \dot{\theta}_2(t) \end{bmatrix} + [0][F(t)]$$

   Thus, $C = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

   We define Observability Matrix $O_m = [C \quad CA \quad CA^2 \quad CA^3 \quad CA^4 \quad CA^5]^T$

   Given system is observable for given A and C matrices of its equivalent state-space model, iff :

$$rank(O_m) = rank([C \quad CA \quad CA^2 \quad CA^3 \quad CA^4 \quad CA^5]^T) = n$$

   where n is the number of states.

   Using matlab, we perform the rank test :

```
%MATLAB SCRIPT
disp('Observability :: x(t)')
C_st = [1 0 0 0 0 0];
disp(['C = ' mat2str(C_st)]);
O_M = [C_st; C_st*A; C_st*A^2; C_st*A^3; C_st*A^4; C_st*A^5]
disp(['rank = ' num2str(rank(C_M))]);
```

```
%OUTPUT
```

```
Command Window

Observability :: x(t)
C = [1 0 0 0 0 0]

C_M =

    1.0000         0         0         0         0         0
         0    1.0000         0         0         0         0
         0         0   -0.9800         0   -0.9800         0
         0         0         0   -0.9800         0   -0.9800
         0         0    0.6243         0    1.1045         0
         0         0         0    0.6243         0    1.1045

rank = 6
```

2. Output Vector = $\{\theta_1(t), \theta_2(t)\}$
   i.e.

$$Y(t) = [\theta_1(t)\ \theta_2(t)]^T$$

From Eqn 2, we solve to get Matrix C

$$\vec{Y}(t) = \begin{bmatrix}\theta_1(t)\\ \theta_2(t)\end{bmatrix} = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x(t)\\ \dot{x}(t)\\ \theta_1(t)\\ \dot{\theta}_1(t)\\ \theta_2(t)\\ \dot{\theta}_2(t)\end{bmatrix} + [0][F(t)]$$

Thus, $C = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$

Using matlab, we perform the rank test:

```
%MATLAB SCRIPT
%% Output Vector :: Q1(t),Q2(t)
disp('Observability :: Q1(t) Q2(t)')
C_st = [0 0 1 0 0 0;
        0 0 0 0 1 0];
disp(['C = ' mat2str(C_st)])
O_M = [C_st; C_st*A; C_st*A^2; C_st*A^3; C_st*A^4; C_st*A^5]
disp(['rank = ' num2str(rank(C_M))])
pause;
```

```
%OUTPUT
```

```
Command Window
Observability :: Q1(t) Q2(t)
C = [0 0 1 0 0 0;0 0 0 0 1 0]

C_M =

         0         0    1.0000         0         0         0
         0         0         0         0    1.0000         0
         0         0         0    1.0000         0         0
         0         0         0         0         0    1.0000
         0         0   -0.5390         0   -0.0490         0
         0         0   -0.0980         0   -1.0780         0
         0         0         0   -0.5390         0   -0.0490
         0         0         0   -0.0980         0   -1.0780
         0         0    0.2953         0    0.0792         0
         0         0    0.1585         0    1.1669         0
         0         0         0    0.2953         0    0.0792
         0         0         0    0.1585         0    1.1669

rank = 4
fx
```

3. Output Vector = {x(t) $\theta_2(t)$}
   i.e.

$$Y(t) = [x(t)\ \theta_2(t)]^T$$

From Eqn 2, we solve to get Matrix C

$$\vec{Y}(t) = \begin{bmatrix} x(t) \\ \theta_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta}_1(t) \\ \theta_2(t) \\ \dot{\theta}_2(t) \end{bmatrix} + [0][F(t)]$$

Thus, $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Using matlab, we perform the rank test:

```
%MATLAB SCRIPT
disp('Observability :: x(t) Q2(t)')
C_st = [1 0 0 0 0 0;
        0 0 0 0 1 0];
disp(['C = ' mat2str(C_st)])
O_M = [C_st; C_st*A; C_st*A^2; C_st*A^3; C_st*A^4; C_st*A^5]
disp(['rank = ' num2str(rank(C_M))])
pause;
```

```
%OUTPUT
```

```
Command Window
Observability :: x(t) Q2(t)
C = [1 0 0 0 0 0;0 0 0 0 1 0]

C_M =

    1.0000         0         0         0         0         0
         0         0         0         0    1.0000         0
         0    1.0000         0         0         0         0
         0         0         0         0         0    1.0000
         0         0   -0.9800         0   -0.9800         0
         0         0   -0.0980         0   -1.0780         0
         0         0         0   -0.9800         0   -0.9800
         0         0         0   -0.0980         0   -1.0780
         0         0    0.6243         0    1.1045         0
         0         0    0.1585         0    1.1669         0
         0         0         0    0.6243         0    1.1045
         0         0         0    0.1585         0    1.1669

rank = 6
fx
```

4. Output Vector = {x(*t*), $\theta_1(t), \theta_2(t)$}

   i.e.

$$Y(t) = [x(t)\ \theta_1(t)\ \theta_2(t)]^T$$

From Eqn 2, we solve to get Matrix C

$$\vec{Y}(t) = \begin{bmatrix} x(t) \\ \theta_1(t) \\ \theta_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta}_1(t) \\ \theta_2(t) \\ \dot{\theta}_2(t) \end{bmatrix} + [0][F(t)]$$

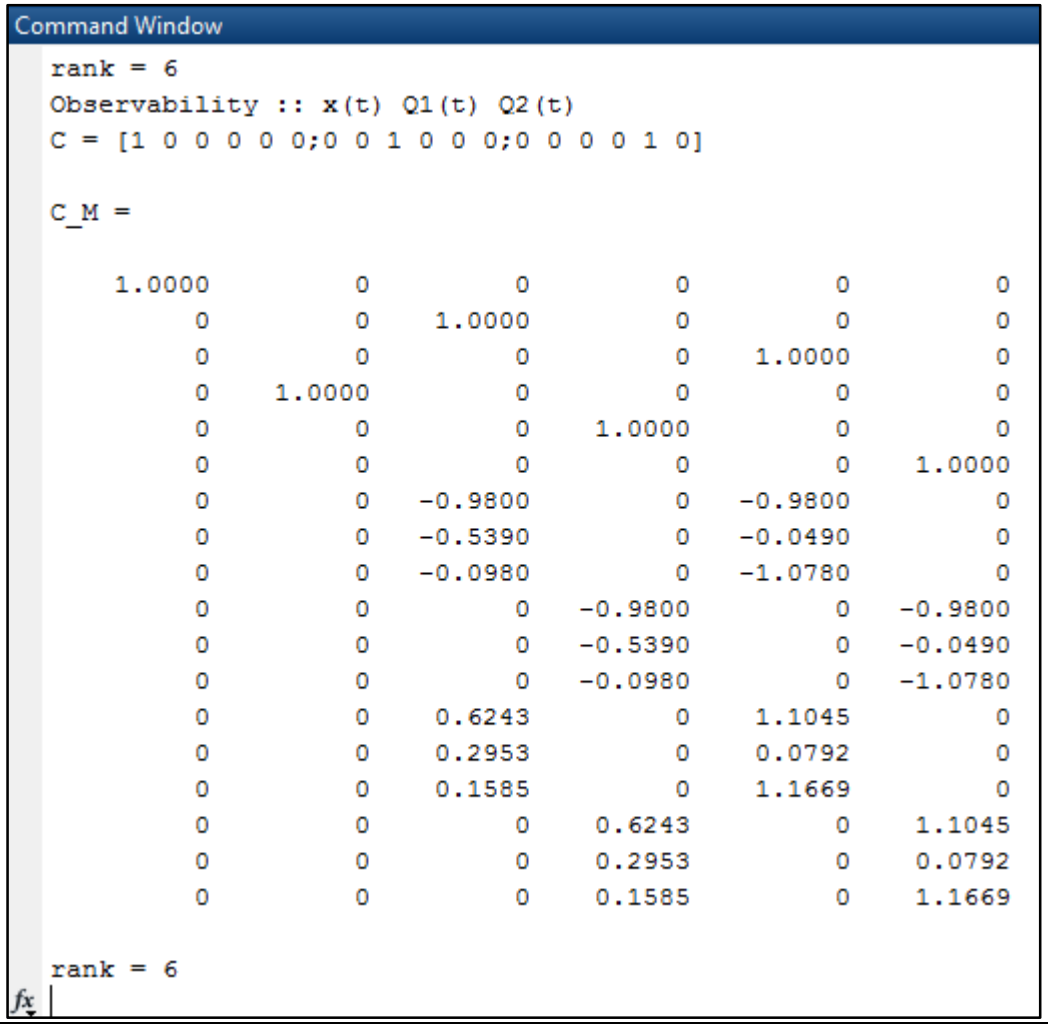
Thus, $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Using matlab, we perform the rank test:

```
%MATLAB SCRIPT
%% Output Vector :: x(t),Q1, Q2(t)
disp('Observability :: x(t) Q1(t) Q2(t)')
C_st = [1 0 0 0 0 0;
        0 0 1 0 0 0
        0 0 0 0 1 0];
disp(['C = ' mat2str(C_st)])
C_M = [C_st; C_st*A; C_st*A^2; C_st*A^3; C_st*A^4; C_st*A^5]
disp(['rank = ' num2str(rank(C_M))])
```

```
pause;
```

%OUTPUT



Command Window

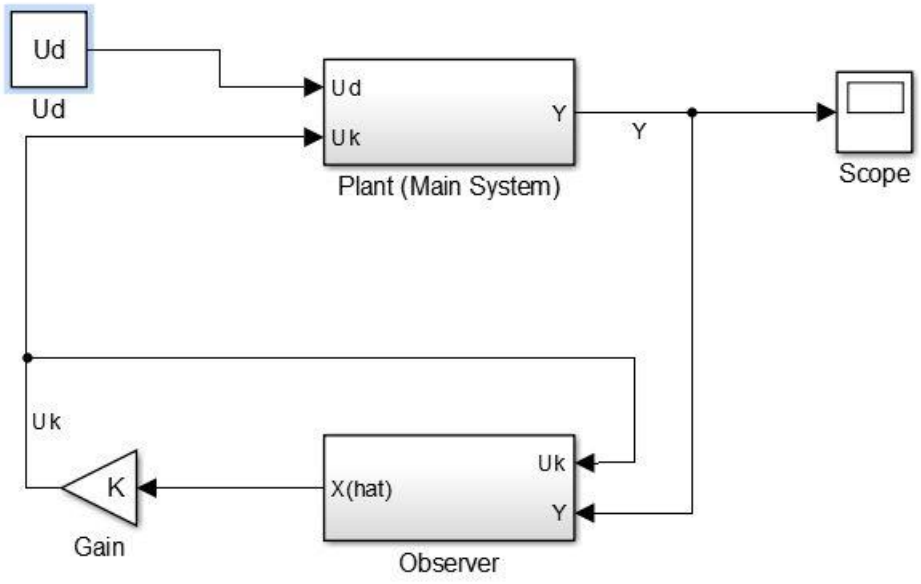
```
rank = 6
Observability :: x(t) Q1(t) Q2(t)
C = [1 0 0 0 0 0;0 0 1 0 0 0;0 0 0 0 1 0]

C_M =

    1.0000         0         0         0         0         0
         0         0    1.0000         0         0         0
         0         0         0         0    1.0000         0
         0    1.0000         0         0         0         0
         0         0         0    1.0000         0         0
         0         0         0         0         0    1.0000
         0         0   -0.9800         0   -0.9800         0
         0         0   -0.5390         0   -0.0490         0
         0         0   -0.0980         0   -1.0780         0
         0         0         0   -0.9800         0   -0.9800
         0         0         0   -0.5390         0   -0.0490
         0         0         0   -0.0980         0   -1.0780
         0         0    0.6243         0    1.1045         0
         0         0    0.2953         0    0.0792         0
         0         0    0.1585         0    1.1669         0
         0         0         0    0.6243         0    1.1045
         0         0         0    0.2953         0    0.0792
         0         0         0    0.1585         0    1.1669

rank = 6
```

Luenberger Observer

Design Process Overview:

The principle of separation of estimation and control:

- The problem of designing an optimal feedback controller can be solved by designing an optimal observer for the state of the system, which feeds into an optimal controller for the system
- Hence we use the Linear Quadratic Regulator Controller method to design optimal controller gain K and use pole placement method to determine L, the observer gain matrix for the error correction.



The estimated state of the system as obtained by the observer is given as: $\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - C\hat{x}\right)$

We chose the values of Q and R matrices similar to those obtained for the LQR controller designed in the mini-project 2.

The Eigen values of the new A matrix obtained for the closed loop system from above (given by (A-BK)) were then used as the reference poles to get the L matrix for the Observer.

As the pair (A, C) is observable for the output vectors: (x(t)); (x(t), θ2(t)) and (x(t), θ1(t), θ2(t)), we can place the Eigen values of the (A - LC) matrix as desired.

The reference poles for the L matrix should be between three to ten times that of the closed loop system to ensure a fast convergence of the Observer output with the System output. We tried a couple of values and settled on four times the reference poles as making the Control Gain of the Observer very high, while making it converge faster with the system output, will also make it very susceptible to noise.

Matlab Code for the Observer

The code used in Matlab to Initialise all the values used in the models for all the systems (linear and non-linear) of Observer:

Note:

- Q has been used to signify θ. Therefore, Q1 means $\theta 1$ and Q2 means $\theta 2$
- The Initial conditions for all the systems for the whole Question F are taken as x = 10m and all others as 0 units
- The unit step input is supplied at 5s for all
- All the graphs of the Observer outputs have been magnified to show only the starting points of the system to show how the outputs are being tracked and how much time is taken to do so.

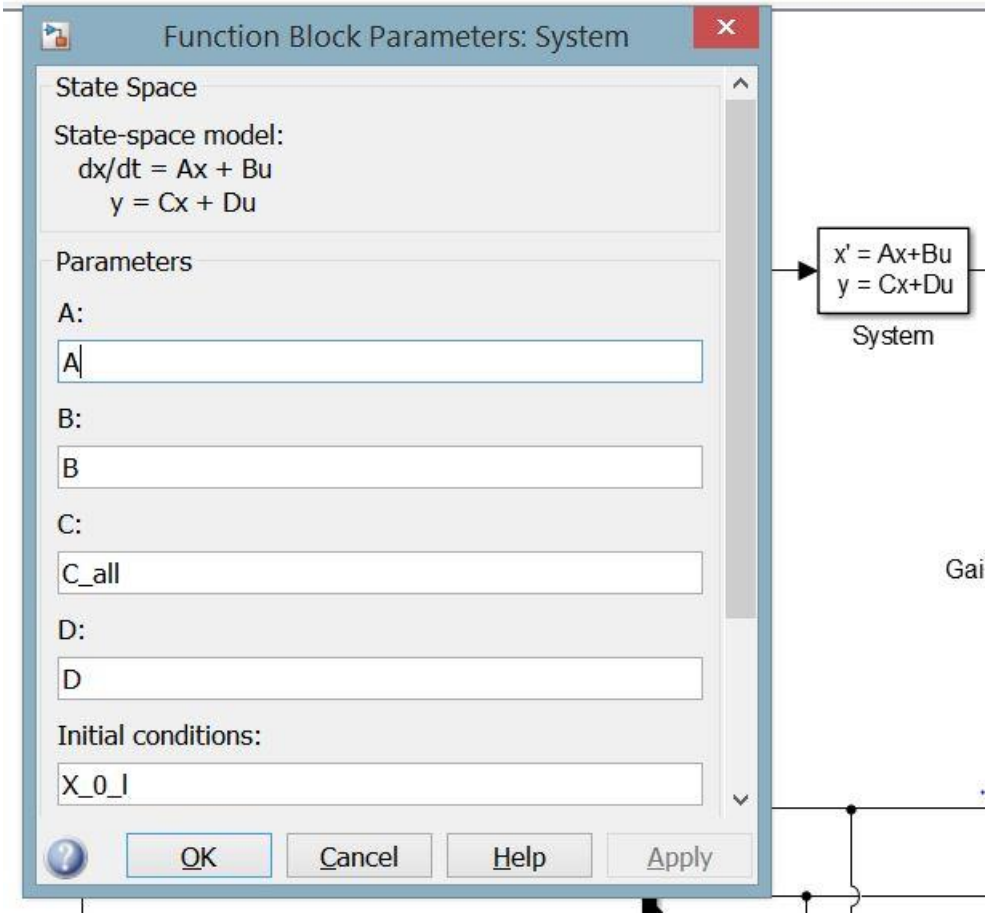
```
clear;
clc;

%% Initialize Parameters

M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.8;

%% Use A and B matrix from Q1(B)

A =[ 0, 1,                0,      0,                0, 0;
     0, 0,       -(m1*g)/M,      0,       -(m2*g)/M, 0;
     0, 0,                0,      1,                0, 0;
     0, 0,  -(M+m1)*g/M/l1,      0,    -(m2*g)/M/l1, 0;
     0, 0,                0,      0,                0, 1;
     0, 0,    -(m1*g)/M/l2,      0,  -(M+m2)*g/M/l2, 0];

B =[ 0;
     1/M;
     0;
     1/(M*l1);
     0;
     1/(M*l2)];

C_all = eye(6);

D = zeros(6,1);

%% Set Values of R and Q to get K matrix using LQR()

Q =[1 0 0 0 0 0;
    0 500 0 0 0 0;
    0 0 700 0 0 0;
```

```
    0 0 0 700 0 0;
    0 0 0 0 500 0;
    0 0 0 0 0 500];

R = 0.005;

[K,S,e] = lqr(A,B,Q,R)
PolesL = 4*e

%% Set Initial Condition X_0 in degrees --  X_0_1 is to convert X_0 in radians

X_0 = [10 0 0 0 0 0]';
X_0_1 = [X_0(1) X_0(2) X_0(3)*pi/180 X_0(4)*pi/180 X_0(5)*pi/180 X_0(6)*pi/180]';

%% Observer State Space for States (x(t))

Cx = [1 0 0 0 0 0];
Lx_t = place(A', Cx', PolesL)
Lx=Lx_t';
Ax_obs = [A-Lx*Cx];
Bx_obs = [B Lx];
Cx_obs = Cx;
Dx_obs = [0 0; 0 0; 0 0; 0 0; 0 0; 0 0];
X_0_x_obs = [0 0 0 0 0 0];

%% Observer State Space for States (x(t),Q2(t))

CxQ2 = [1 0 0 0 0 0;
        0 0 0 0 1 0];
LxQ2_t = place(A', CxQ2', PolesL)
LxQ2 = LxQ2_t';
AxQ2_obs = [A-LxQ2*CxQ2];
BxQ2_obs = [B LxQ2];
CxQ2_obs = CxQ2;
DxQ2_obs = [0 0 0; 0 0 0;0 0 0;0 0 0;0 0 0; 0 0 0];
X_0_xQ2_obs = [0 0 0 0 0 0];

%% Observer State Space for States (x(t),Q1(t),Q2(t))

CxQ1Q2 = [1 0 0 0 0 0;
          0 0 1 0 0 0;
          0 0 0 0 1 0];
LxQ1Q2_t = place(A', CxQ1Q2', PolesL)
LxQ1Q2 = LxQ1Q2_t';
AxQ1Q2_obs = [A-LxQ1Q2*CxQ1Q2];
BxQ1Q2_obs = [B LxQ1Q2];
CxQ1Q2_obs = CxQ1Q2;
DxQ1Q2_obs = [0 0 0 0; 0 0 0 0;0 0 0 0;0 0 0 0;0 0 0 0;0 0 0 0];
X_0_xQ1Q2_obs = [0 0 0 0 0 0];
```

Some of the Values received from the Matlab code:

```
K =

   14.1421  366.6576   85.4746  262.4992   86.6016  145.5497


e =

  -0.2637 + 0.0000i
  -0.0454 + 0.0000i
  -0.0274 + 1.0378i
  -0.0274 - 1.0378i
  -0.0152 + 0.7252i
  -0.0152 - 0.7252i


PolesL =

  -1.0549 + 0.0000i
  -0.1816 + 0.0000i
  -0.1097 + 4.1514i
  -0.1097 - 4.1514i
  -0.0607 + 2.9010i
  -0.0607 - 2.9010i

Lx =

    1.5773
   24.6882
 -339.6623
   88.5373
  305.7600
 -205.3749


LxQ2 =

    1.3727   -1.5040
   11.9822   -1.6094
  -15.1681    4.1299
    4.0429    2.9672
    1.5653    0.2046
    1.1456   10.0710


LxQ1Q2 =

    0.1420    0.1227   -0.0000
    9.3569    1.7249   -0.9800
   -0.0365    0.1989   -0.0000
    2.7132   15.7635   -0.0490
    0.0000    0.0000    1.2365
    0.0000   -0.0980   -0.8864
```

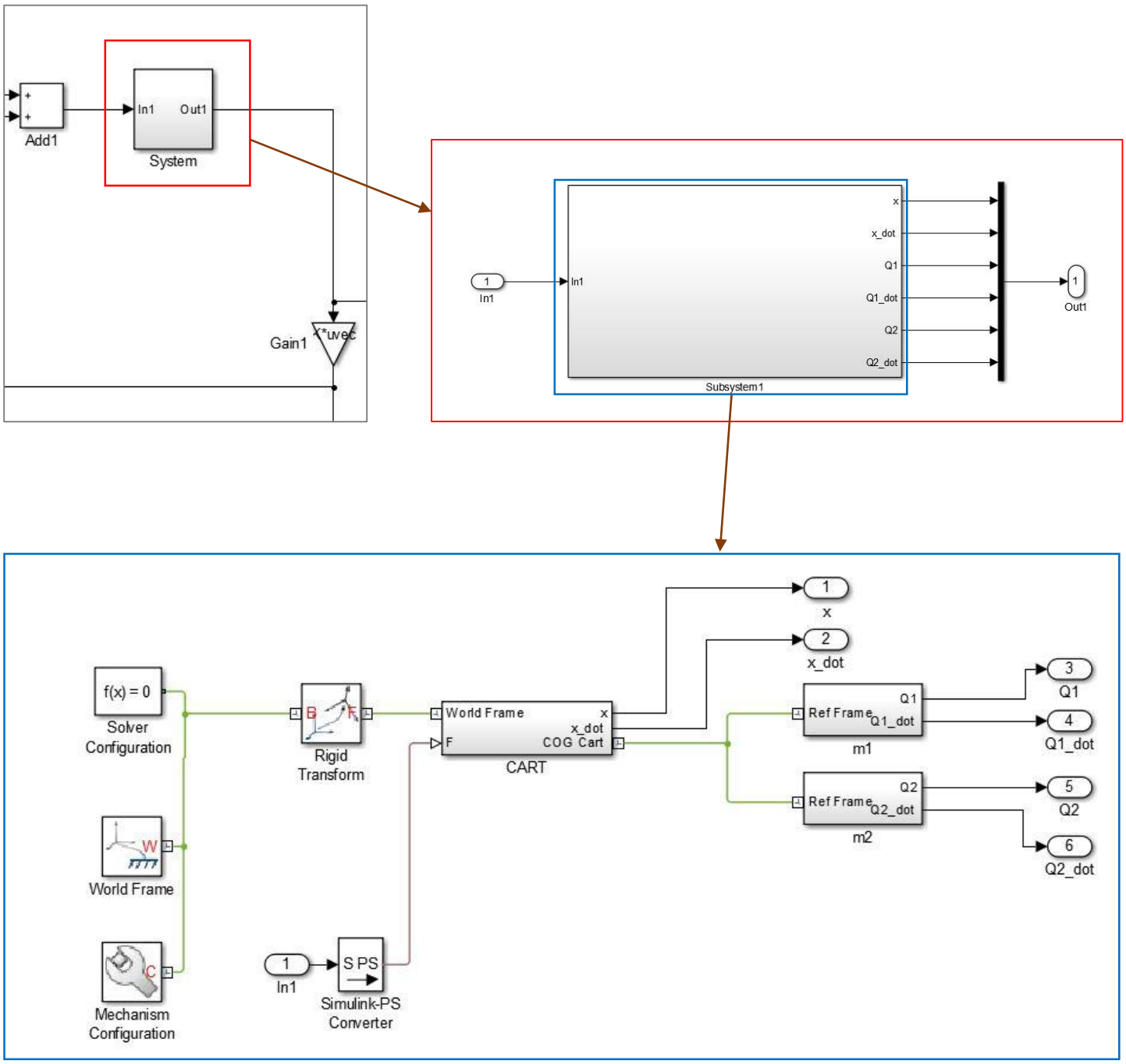
Linearized System Representation

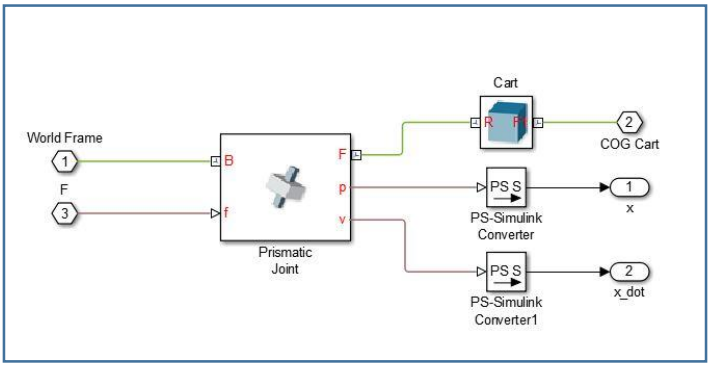


The Linearized system is represented by the State Space Block in Matlab and has the following values as shown in the figure. These variables have been defined in the Initializing code for all the systems.

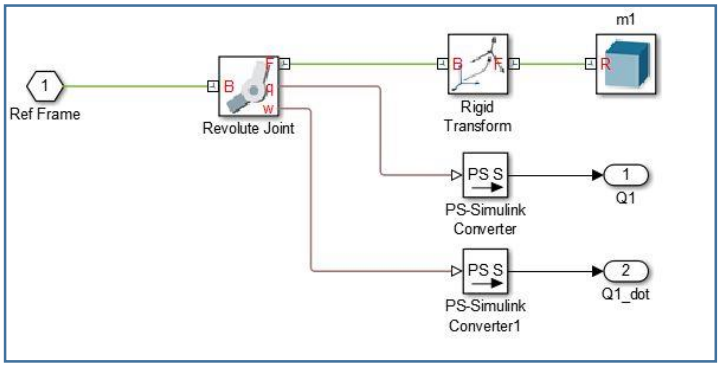
Non-Linearized System Representation

The Non-Linearized system is represented by the physical system made using SimMechanics simulation in Matlab and has the following blocks as shown in the figures. This System has been used throughout the project whenever the Non-Linearized system was required.

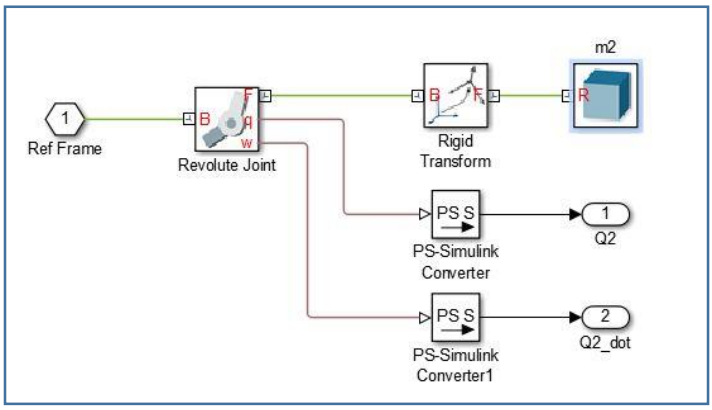




Cart (M)

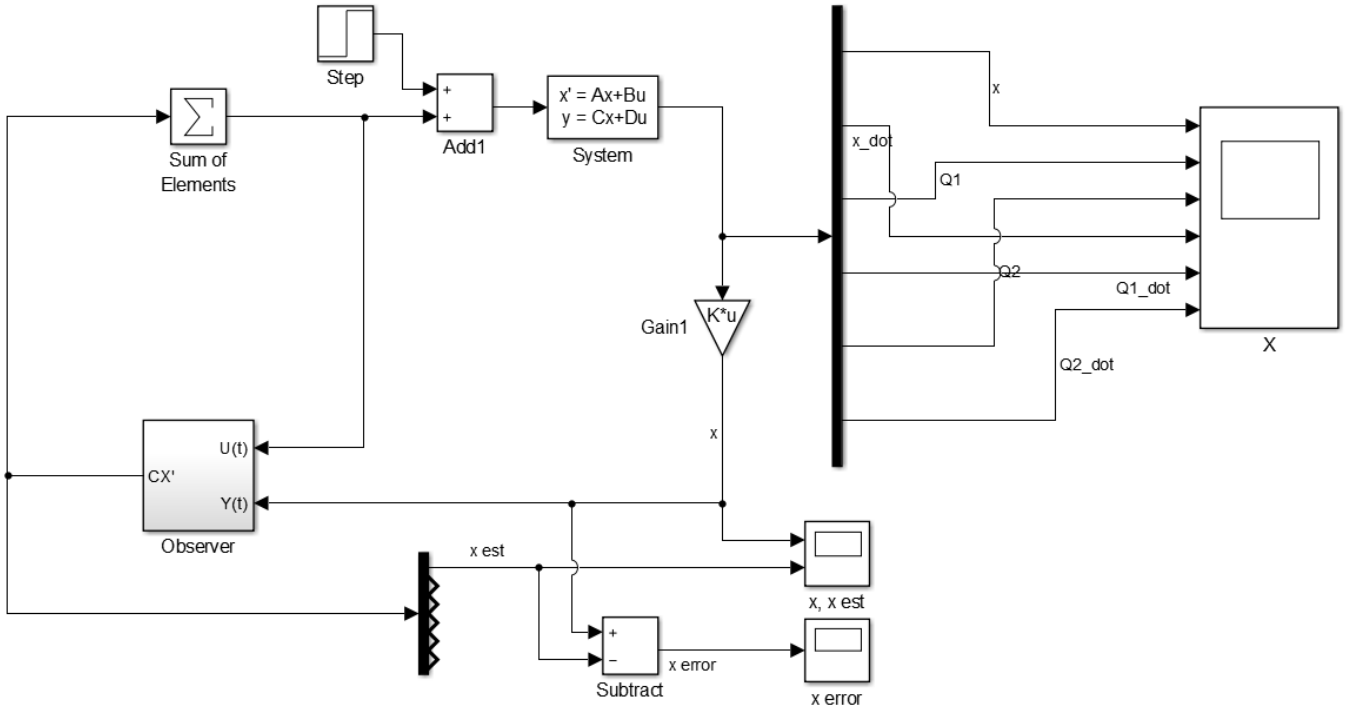


Mass 1 (m1)

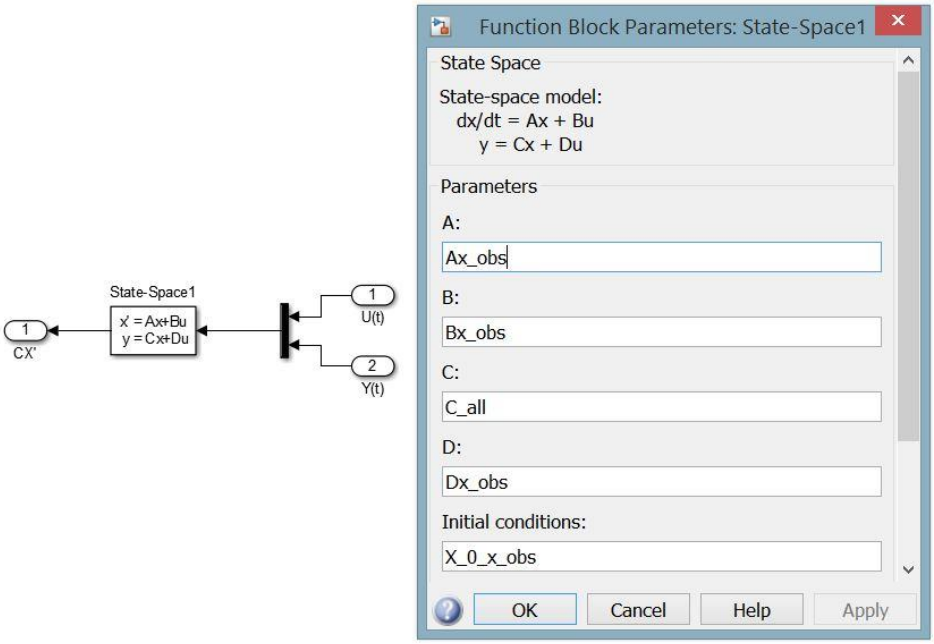


Mass 2 (m2)

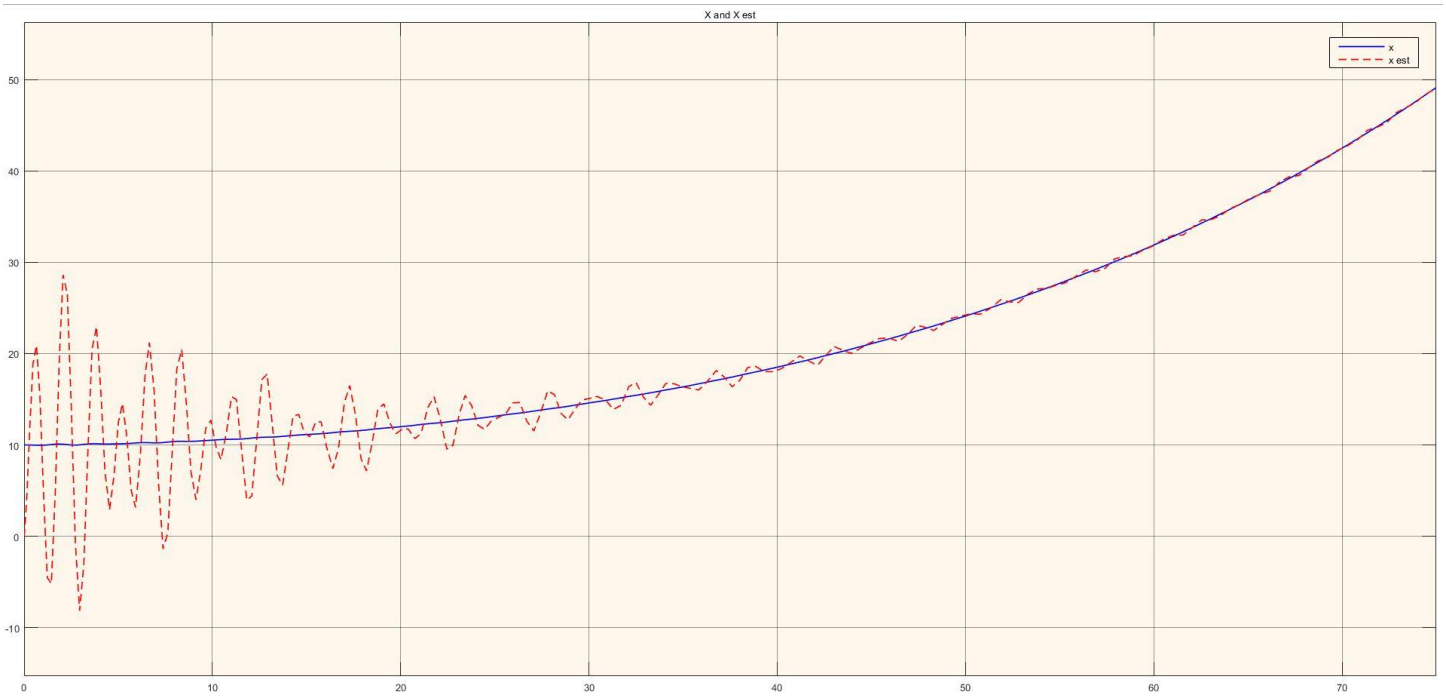
Linearized System with only Observer and no controller for Output Vector x(t)



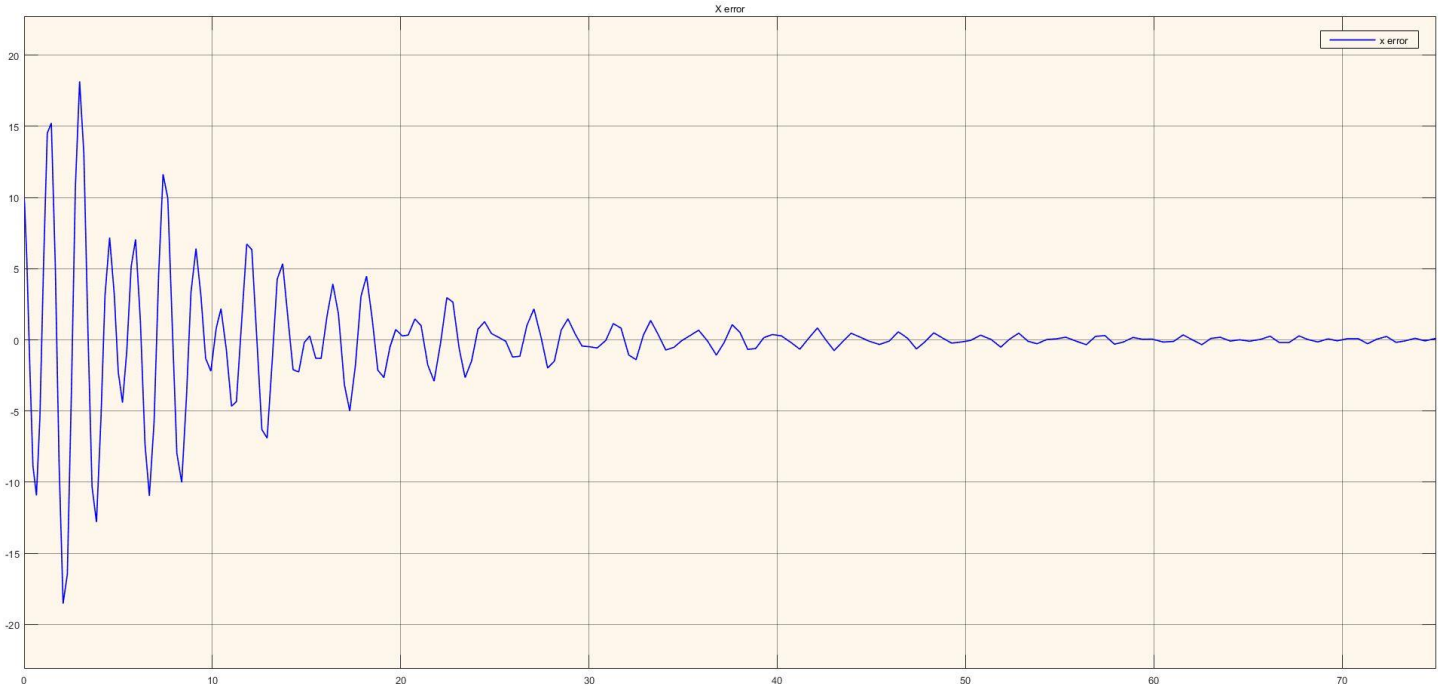
Linearized System with only Observer



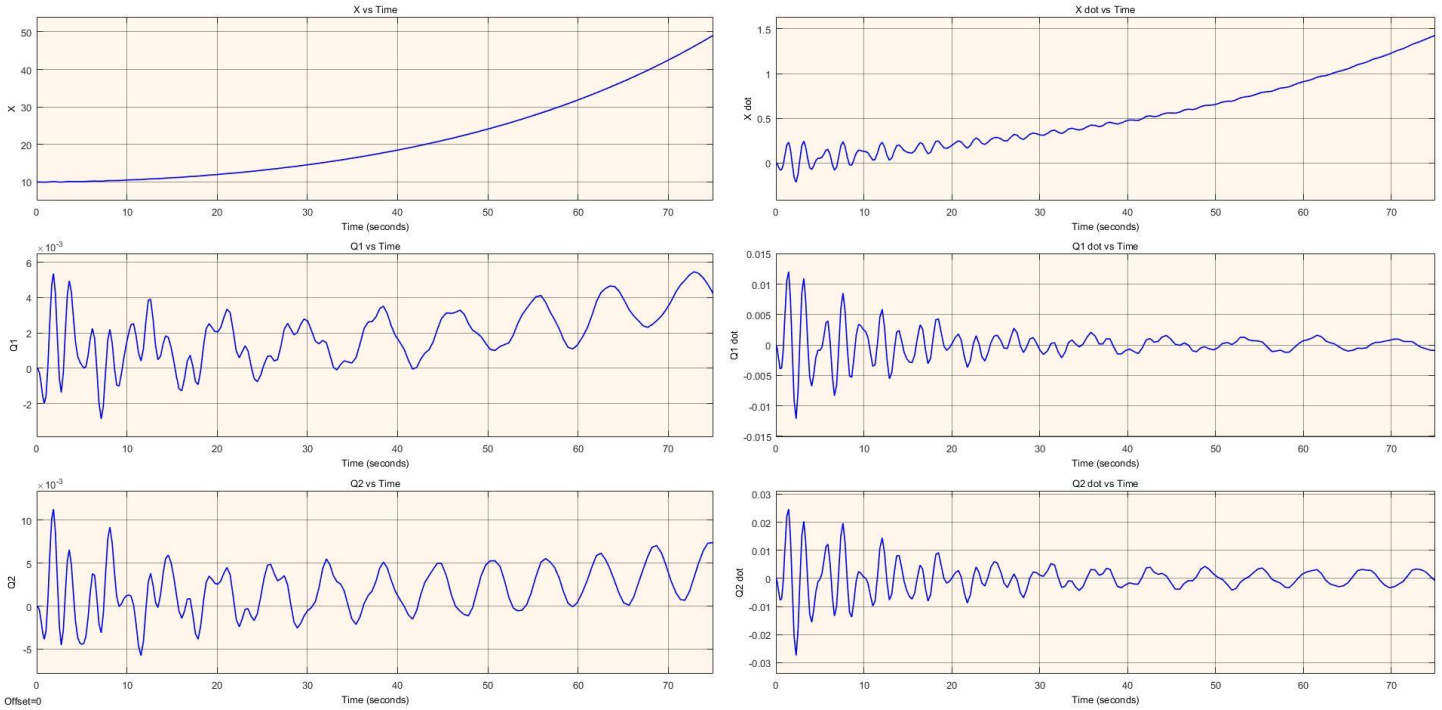
Observer's State Space Model



X from the system and X_{est} from the observer



X_{error} given be ($X - X_{est}$)

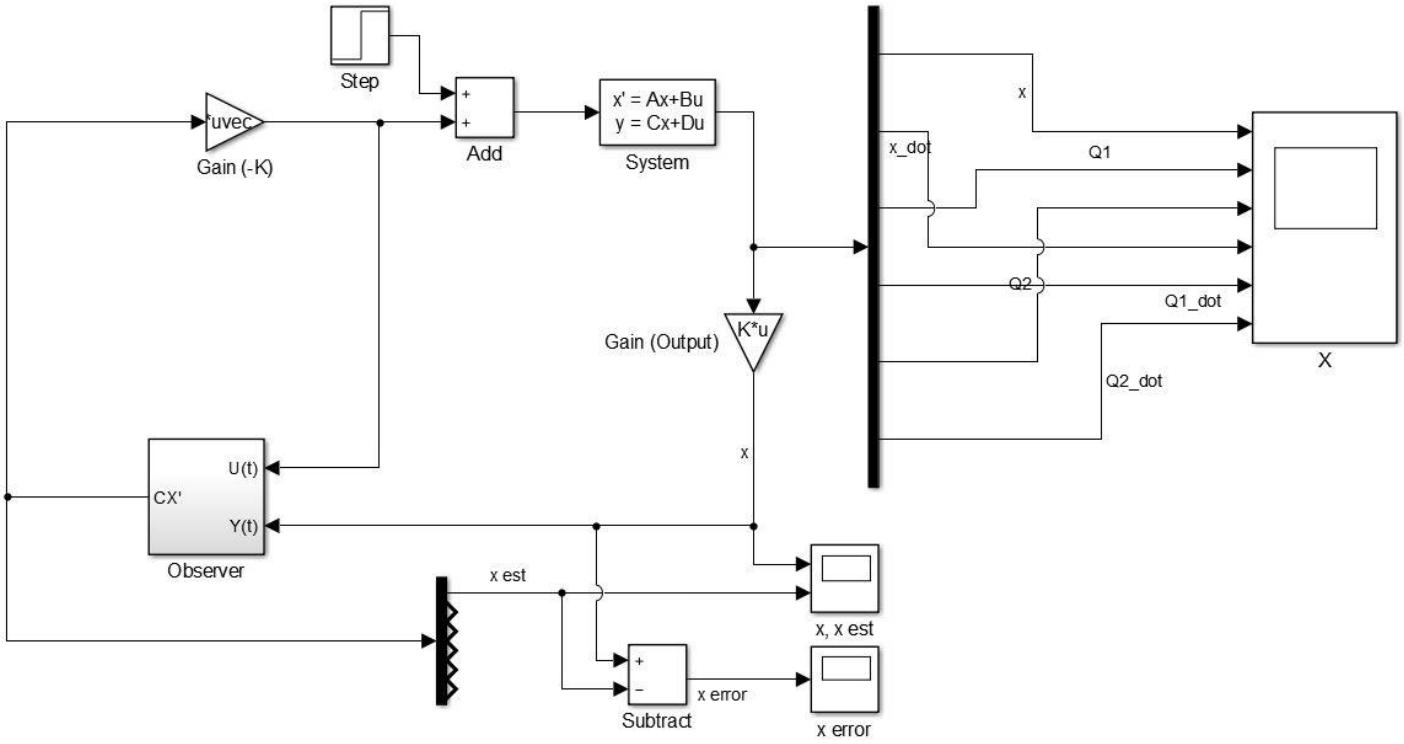


All the states of the system

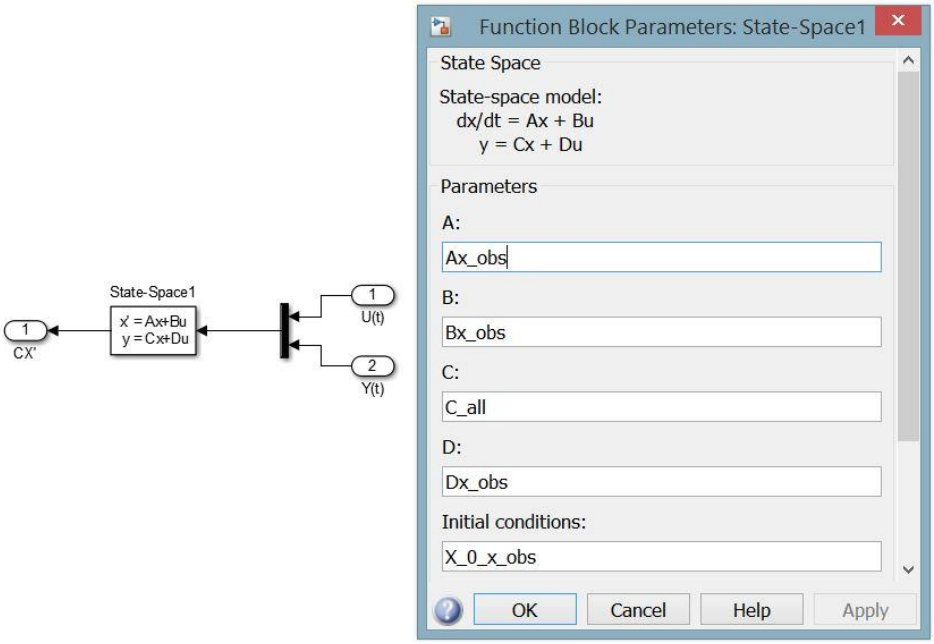
Observations:

As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 60s. Also as only an observer is being used without a controller and a constant input of 1N is being supplied, the x value goes on increasing exponentially.

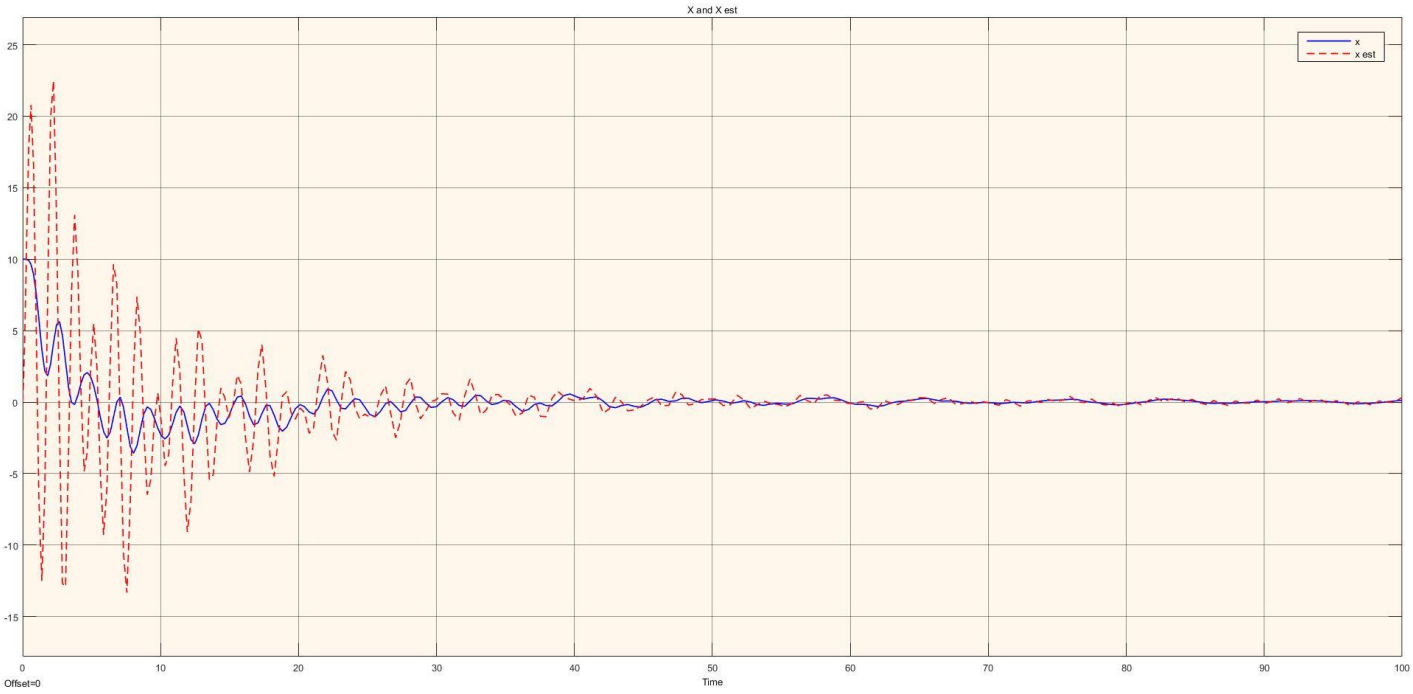
Linearized System with Observer and a controller for Output Vector x(t)



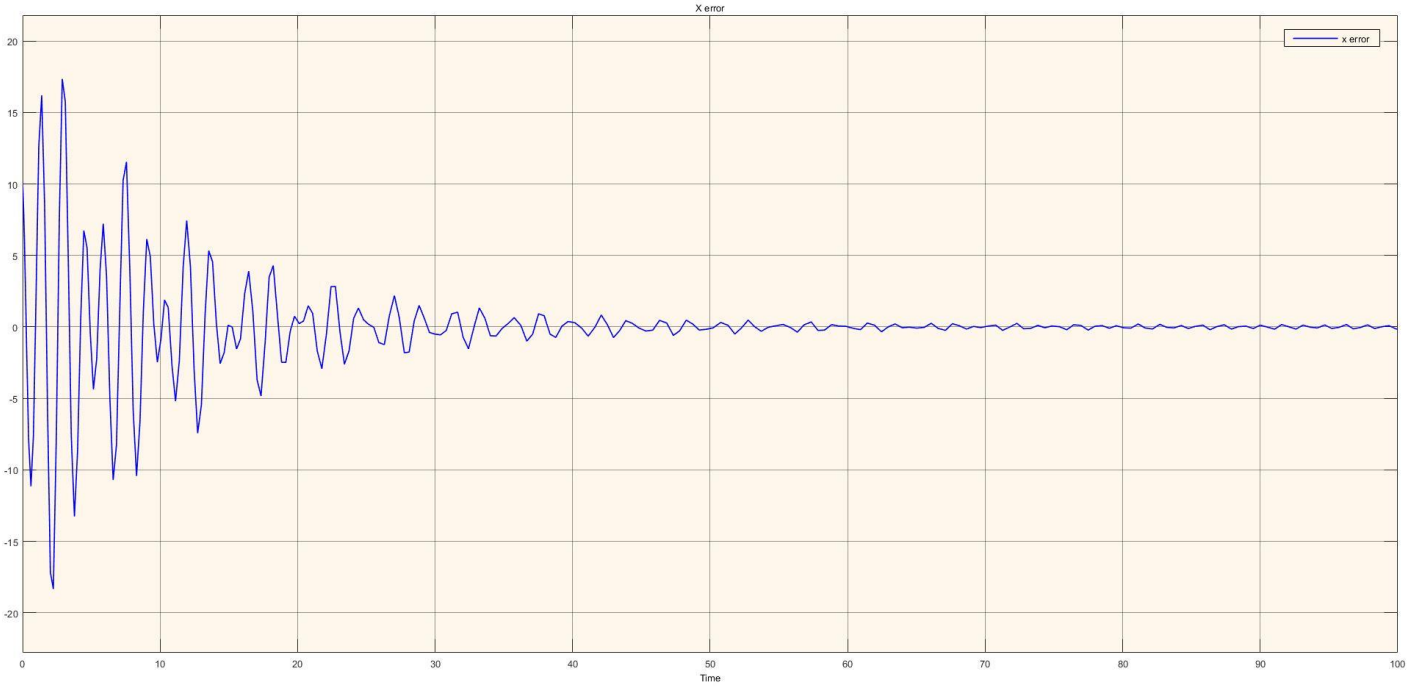
Linearized System with Observer and Controller



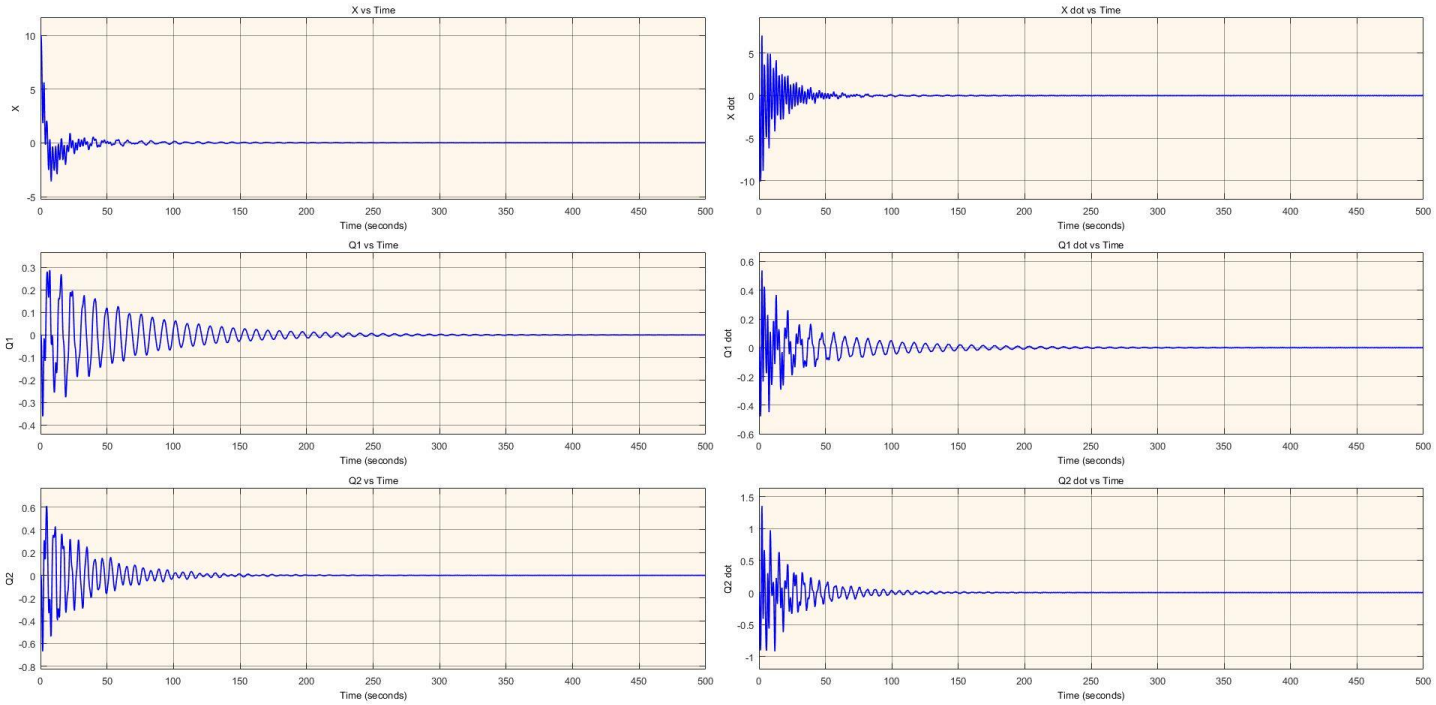
Observer's State Space Model



X from the system and X_{est} from the observer



X_{error} given be ($X - X_{est}$)



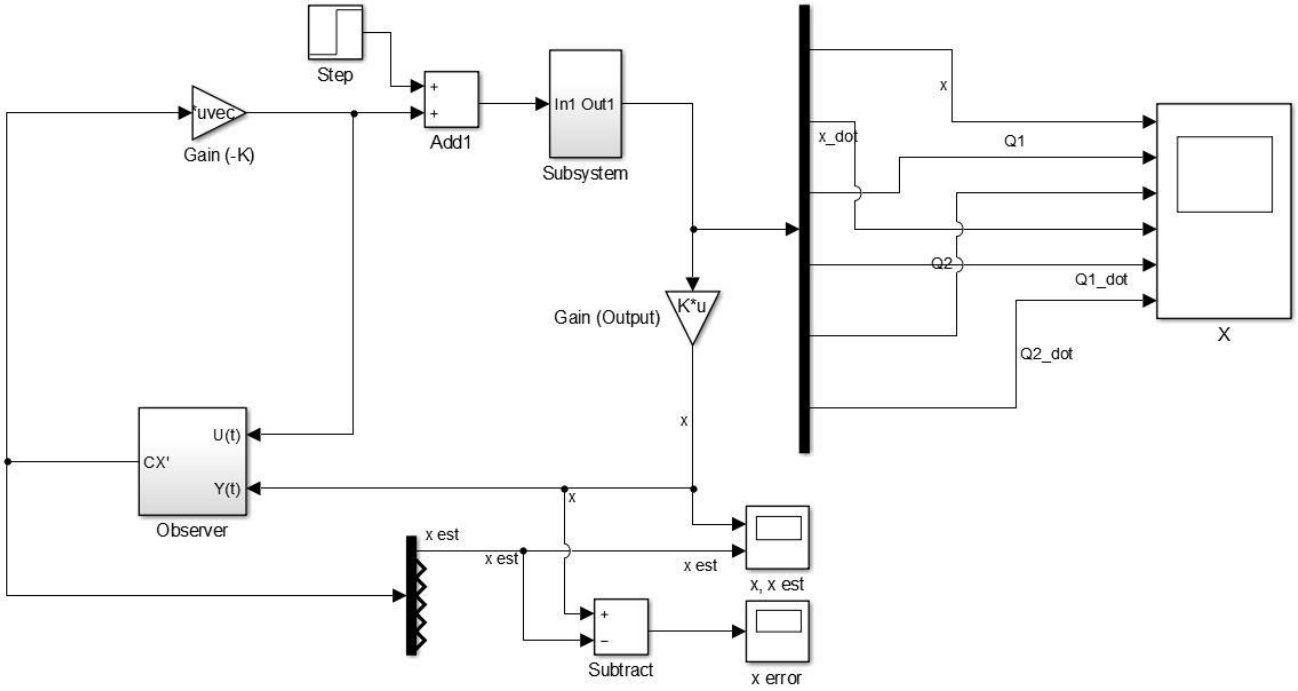
All the states of the system

Observations:

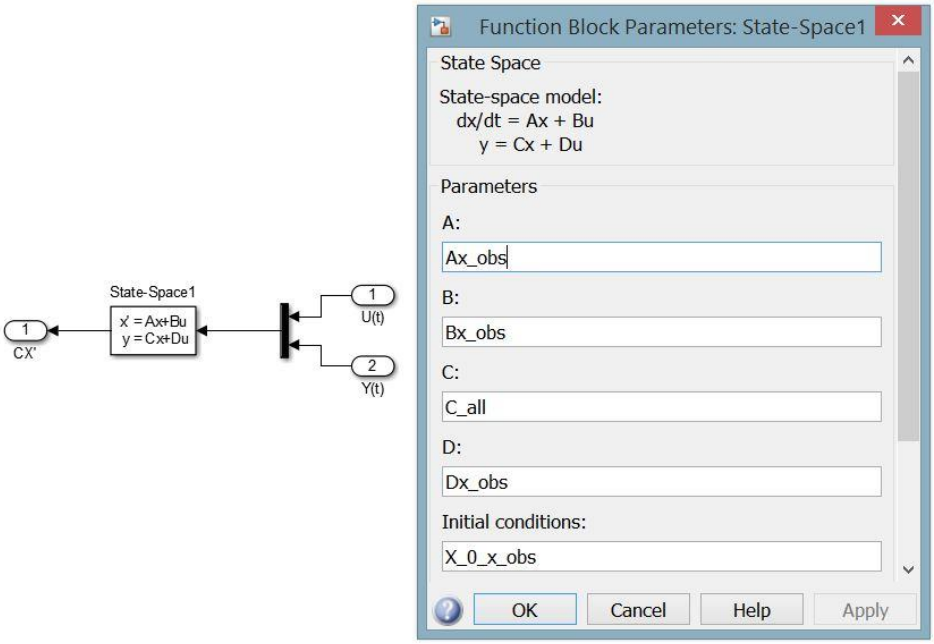
As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 80s.

Now after adding a controller to the observer's output, our output x stabilizes to 0.02 around 400s.

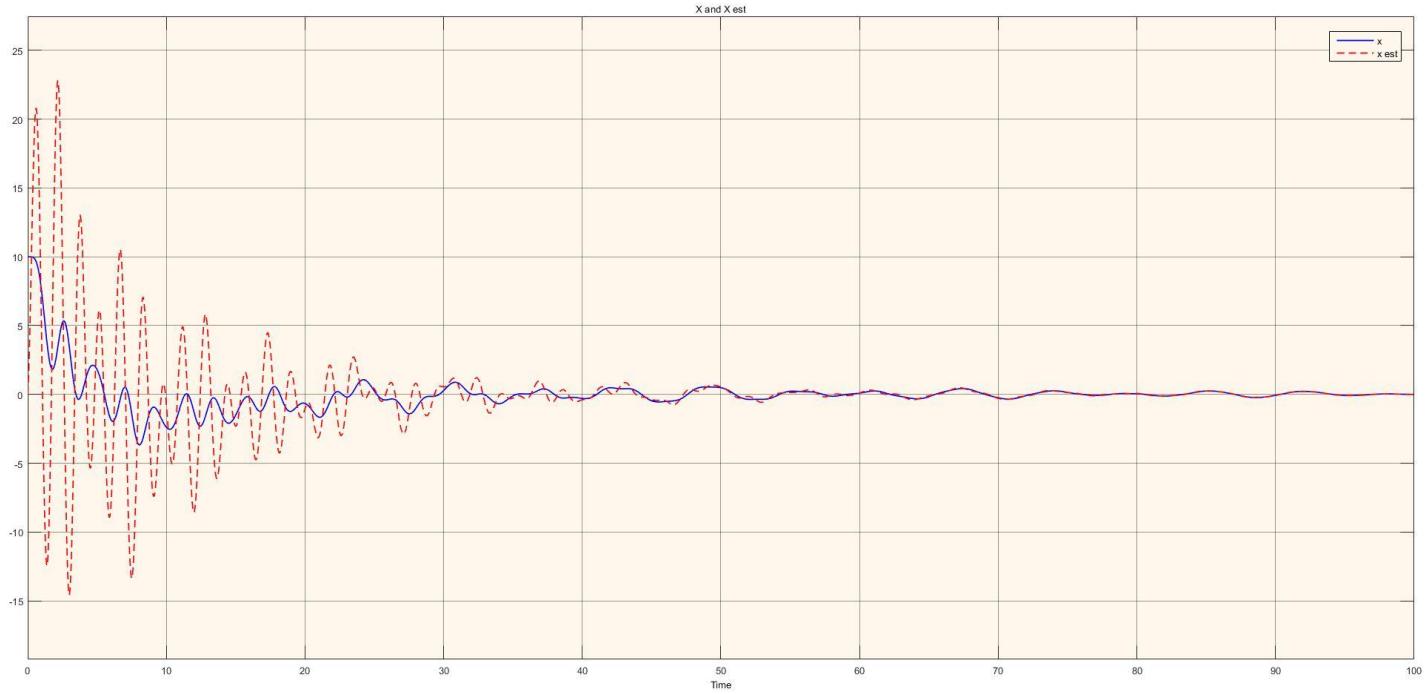
Non-Linearized System with Observer and a controller for Output Vector x(t)



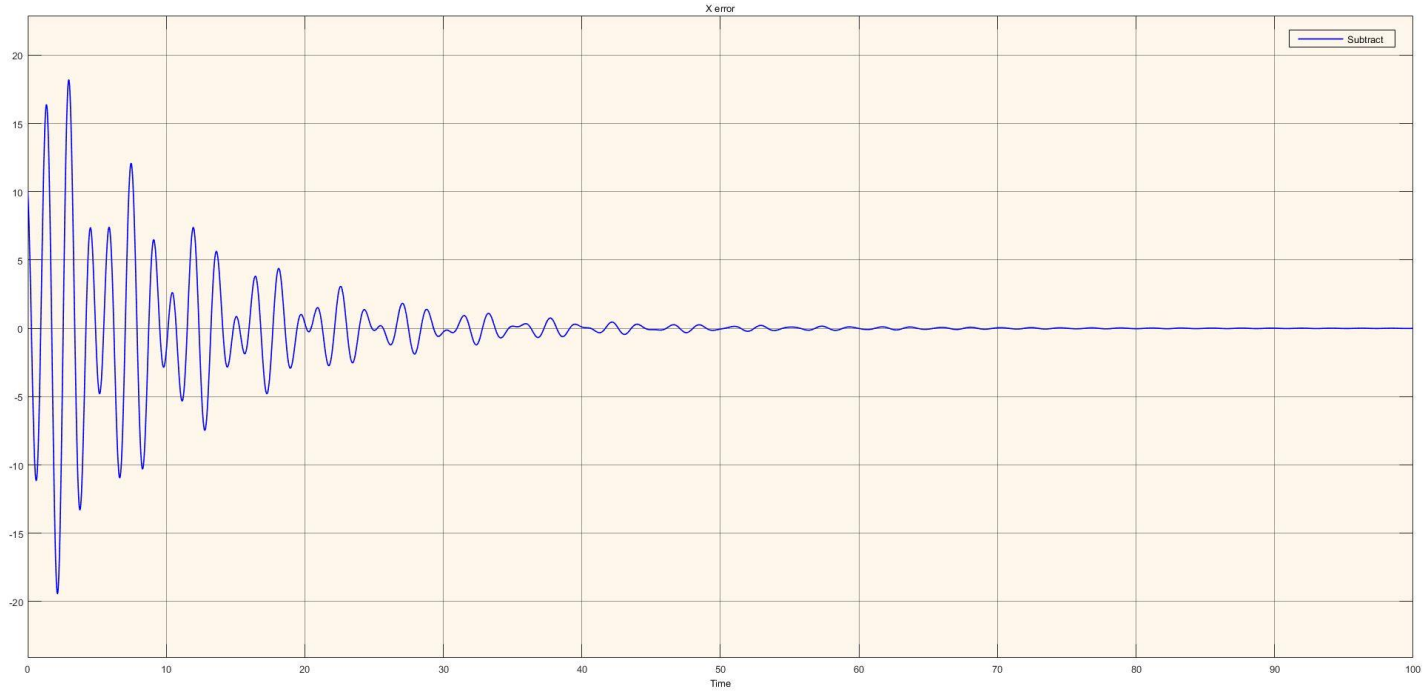
Linearized System with Observer and Controller



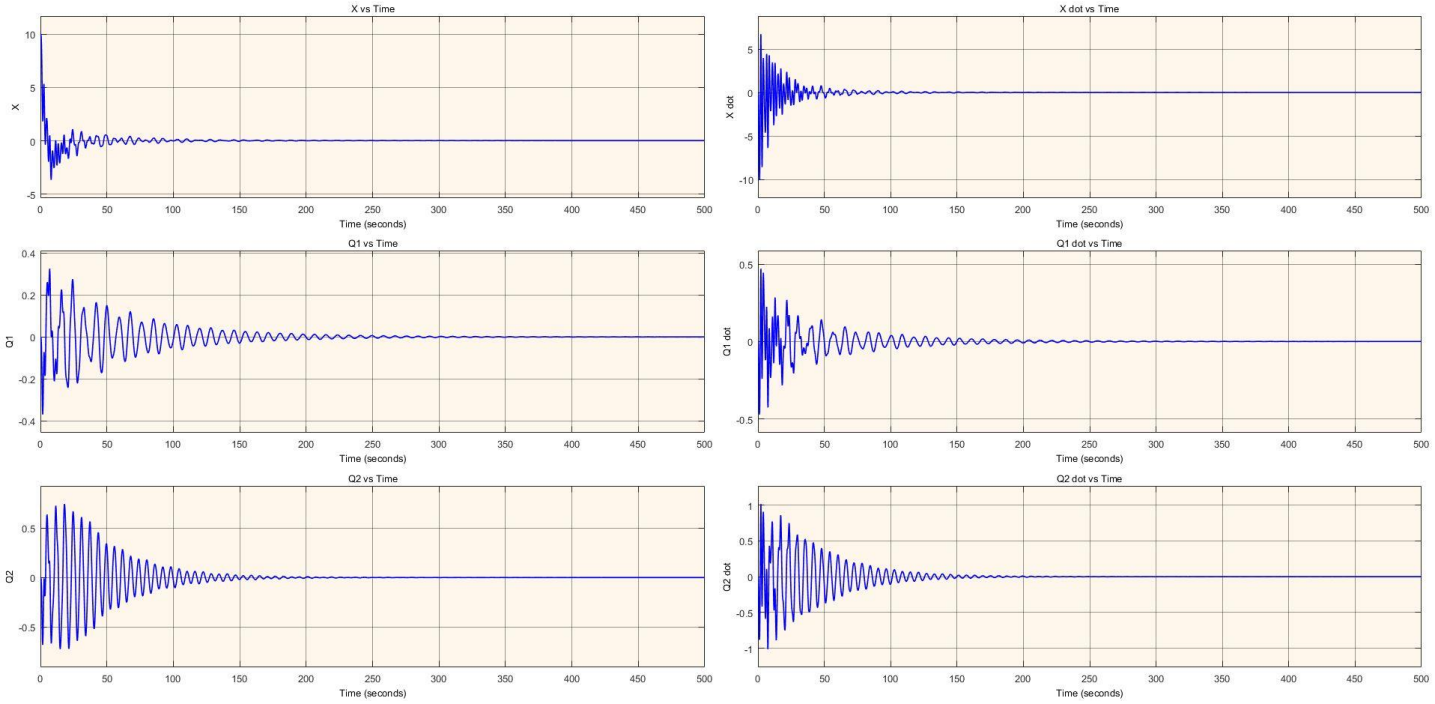
Observer's State Space Model



X from the system and X_{est} from the observer



X_{error} given be (X-X_{est})



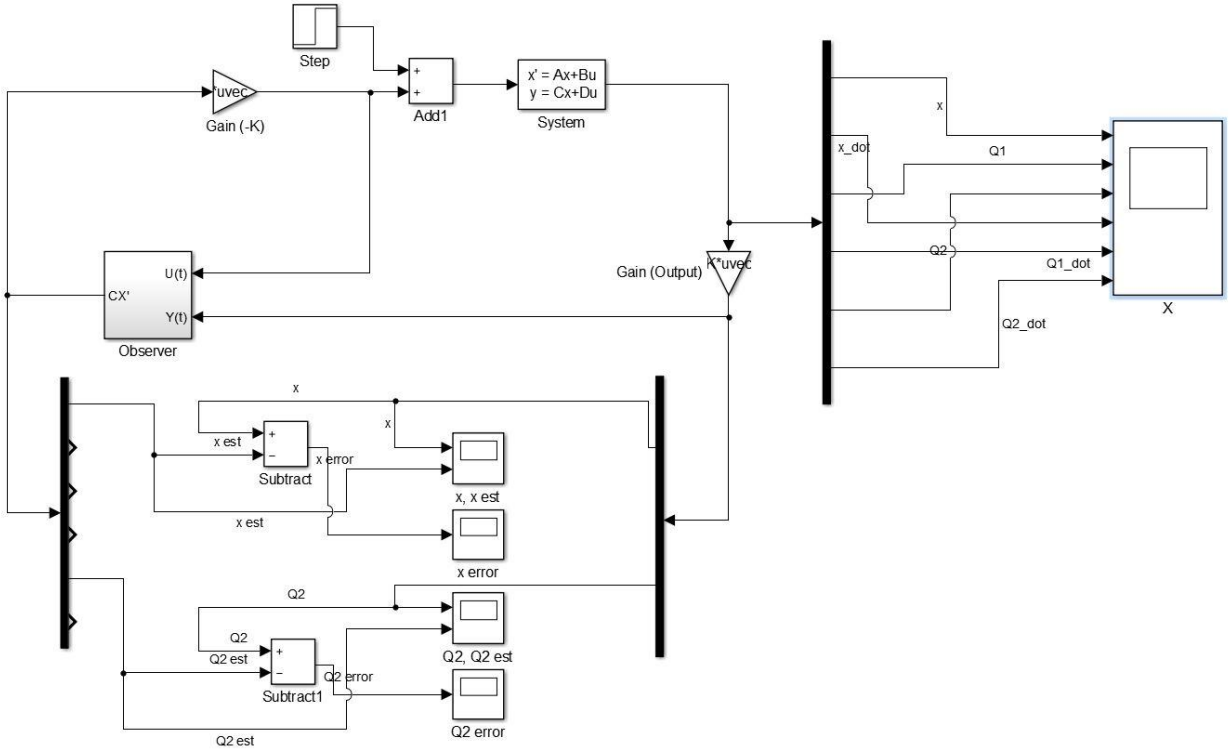
All the states of the system

Observations:

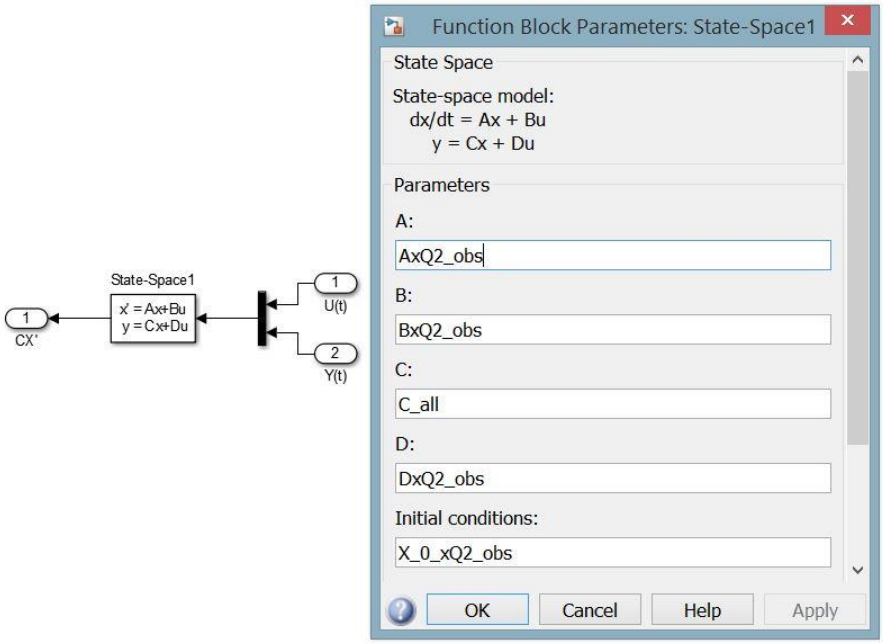
As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 55s.

Our output x stabilizes to 0.02 around 400s.

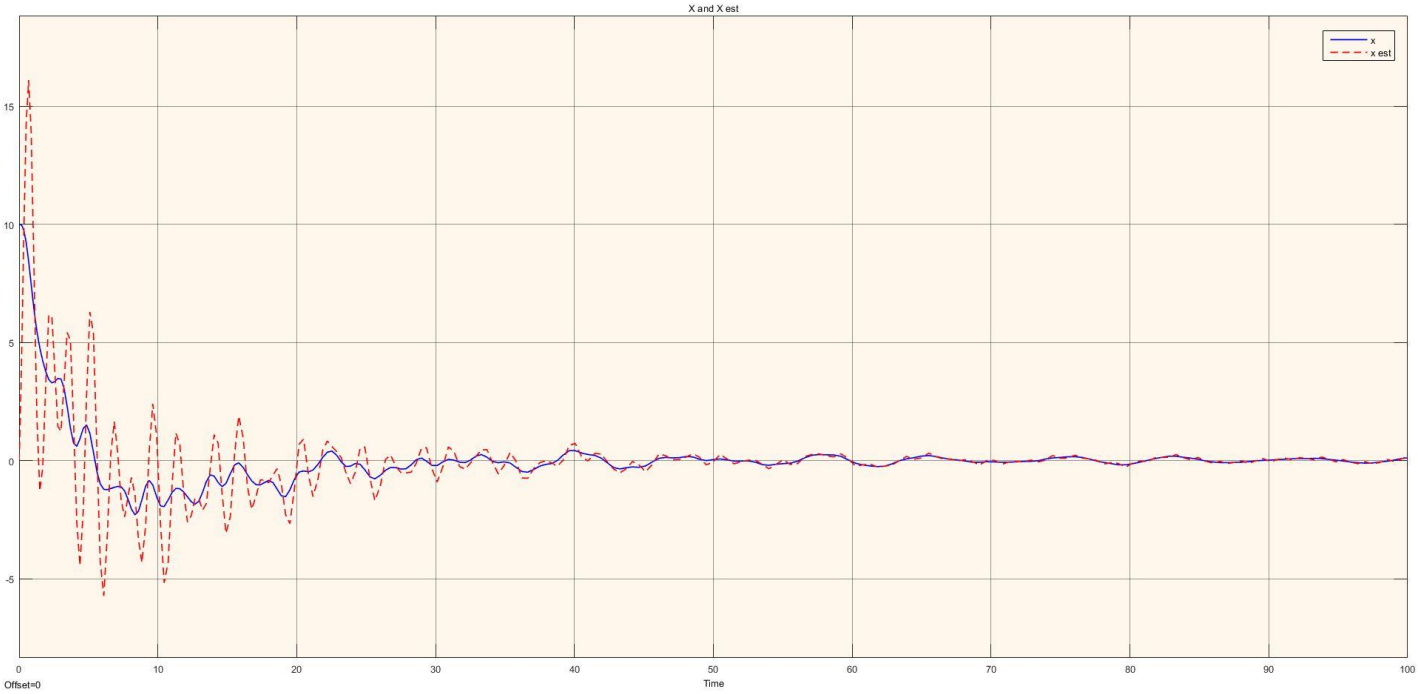
Linearized System with Observer and a controller for Output Vector (x(t), θ2(t))



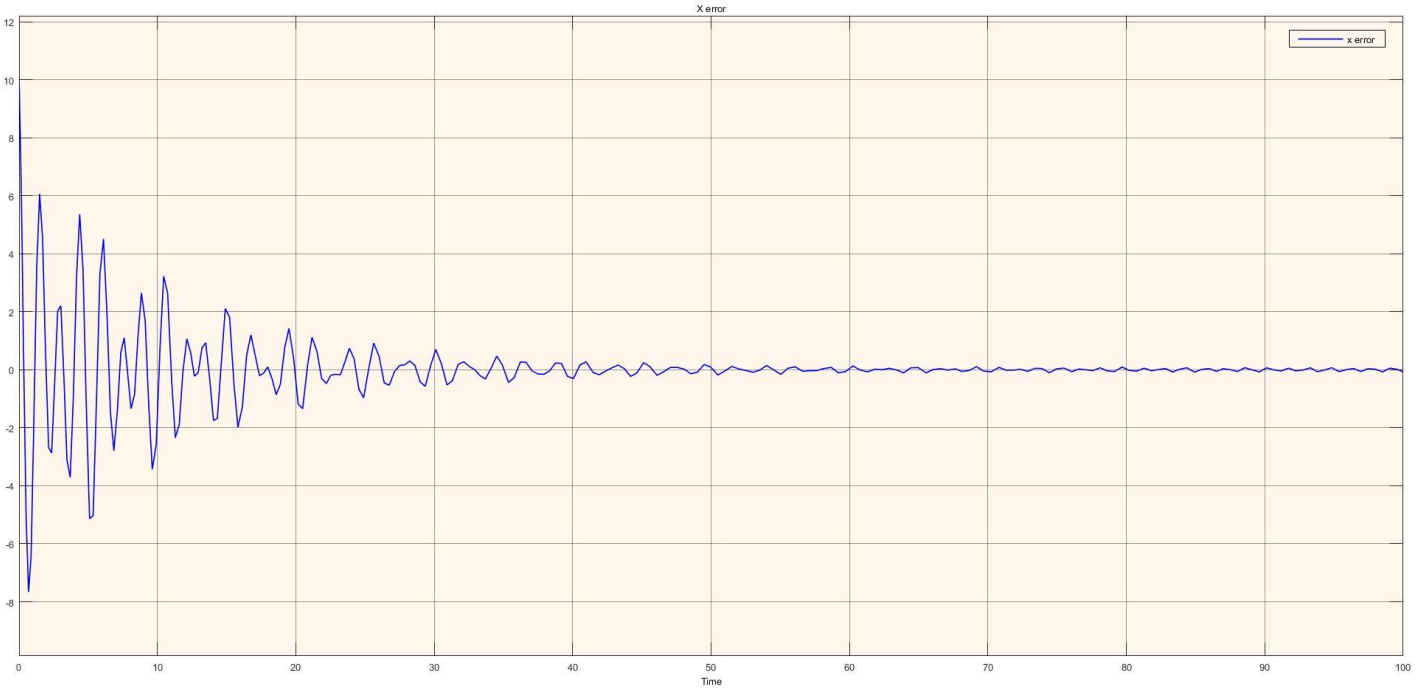
Linearized System with Observer and Controller



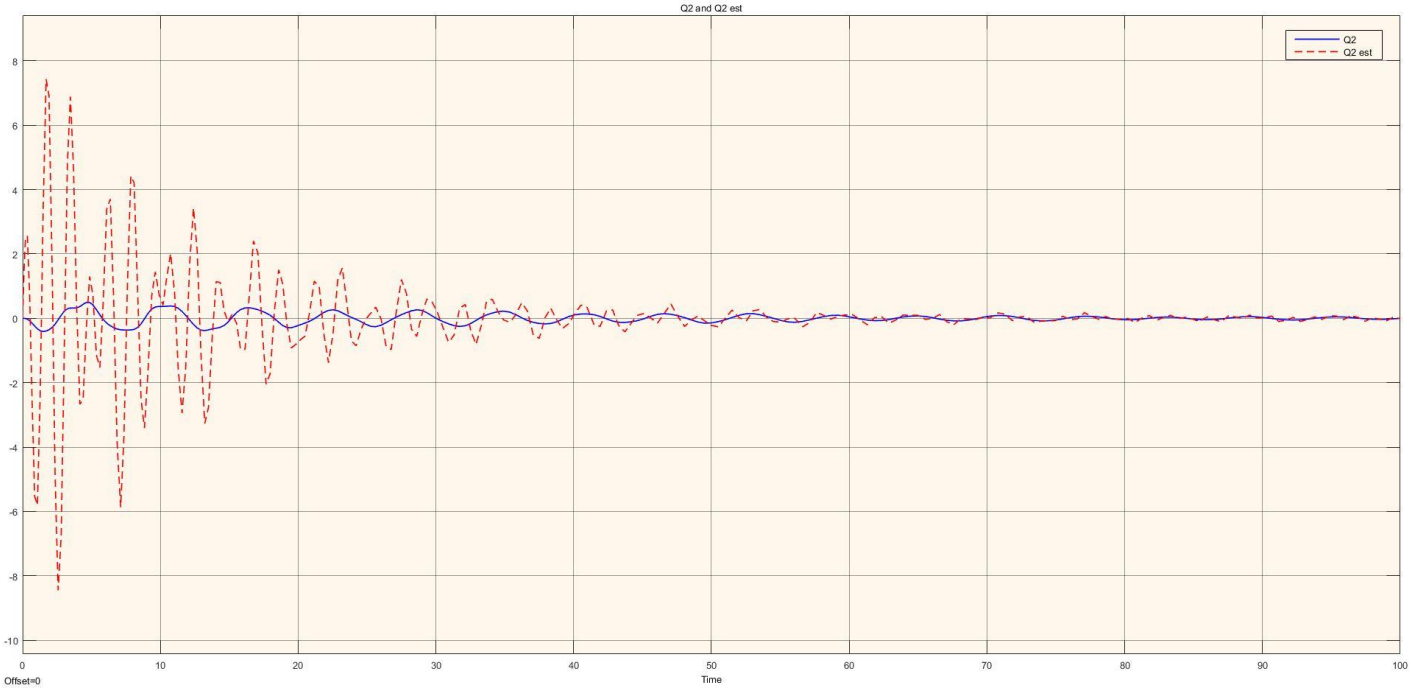
Observer's State Space Model



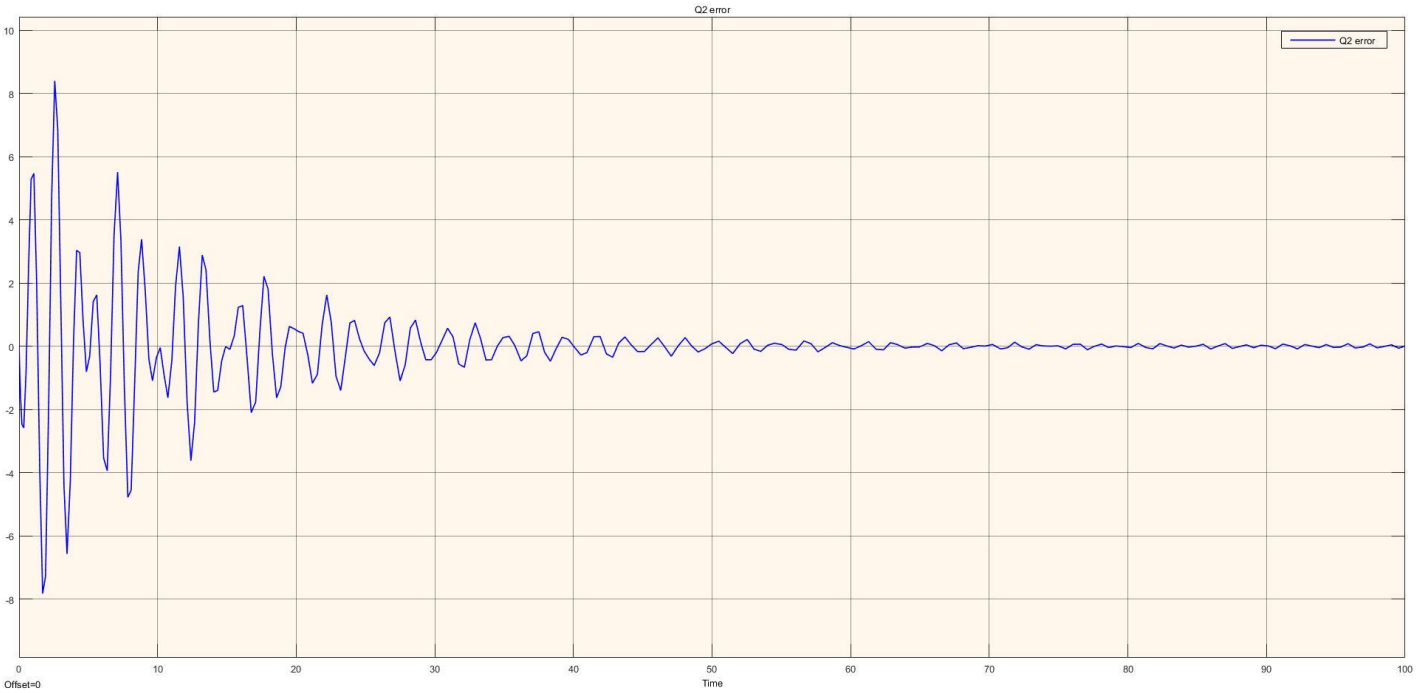
X from the system and X_{est} from the observer



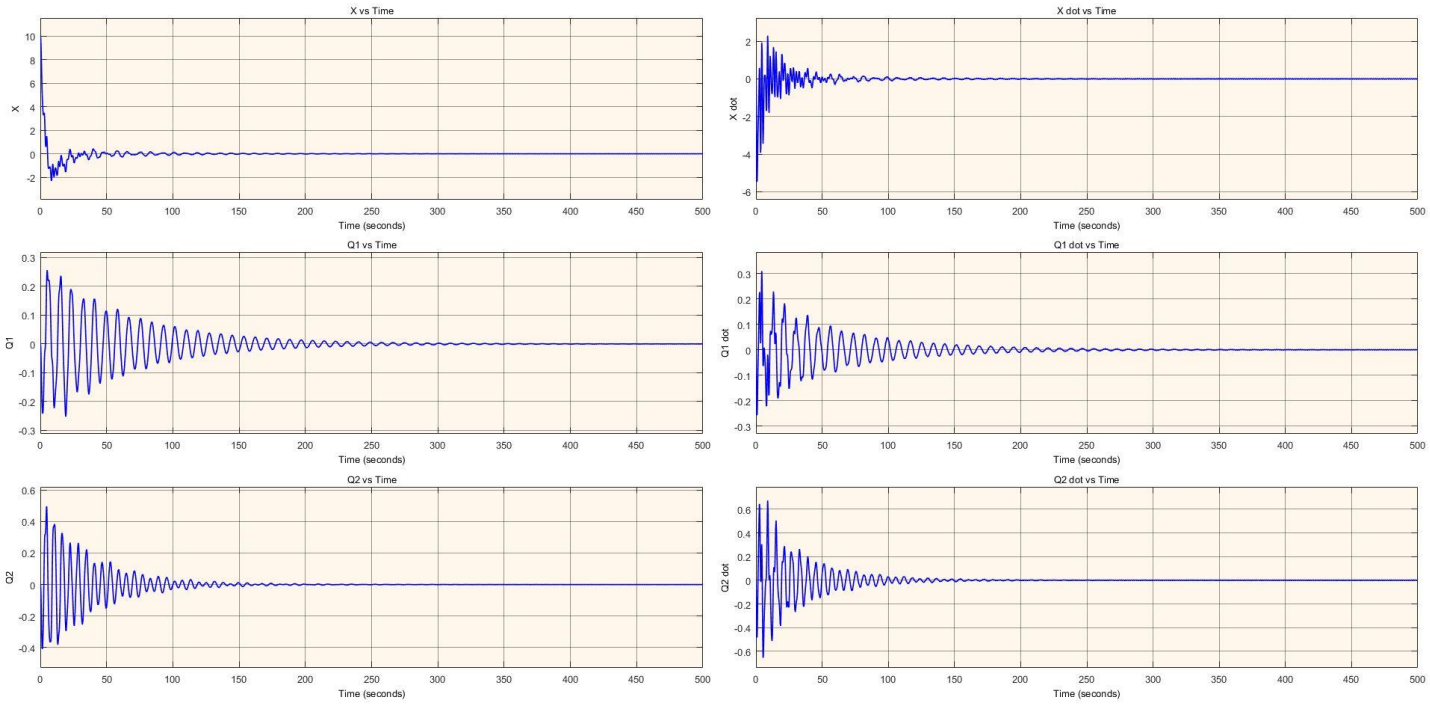
X_{error} given be (X-X_{est})



$\theta 2$ from the system and $\theta 2_{est}$ from the observer



$\theta 2_{error}$ given be ($\theta 2$- $\theta 2_{est}$)



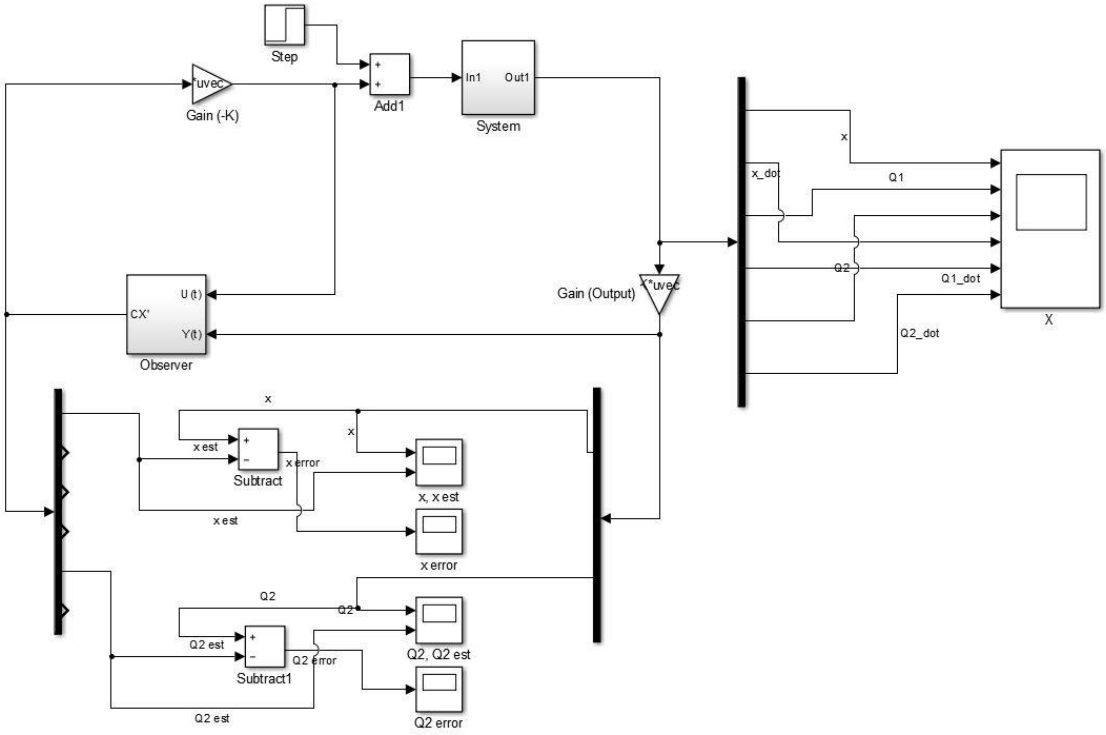
All the states of the system

Observations:

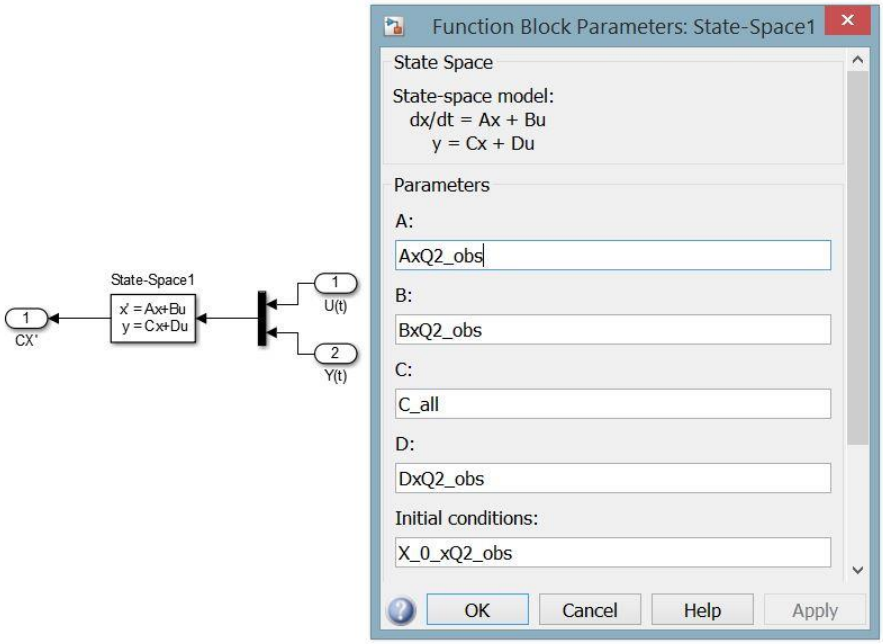
As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 55s and the observer's estimated state $\theta 2_{est}$ converges with that from the system's output θ2 around 65s.

Our output x stabilizes to 0.02 around 390s

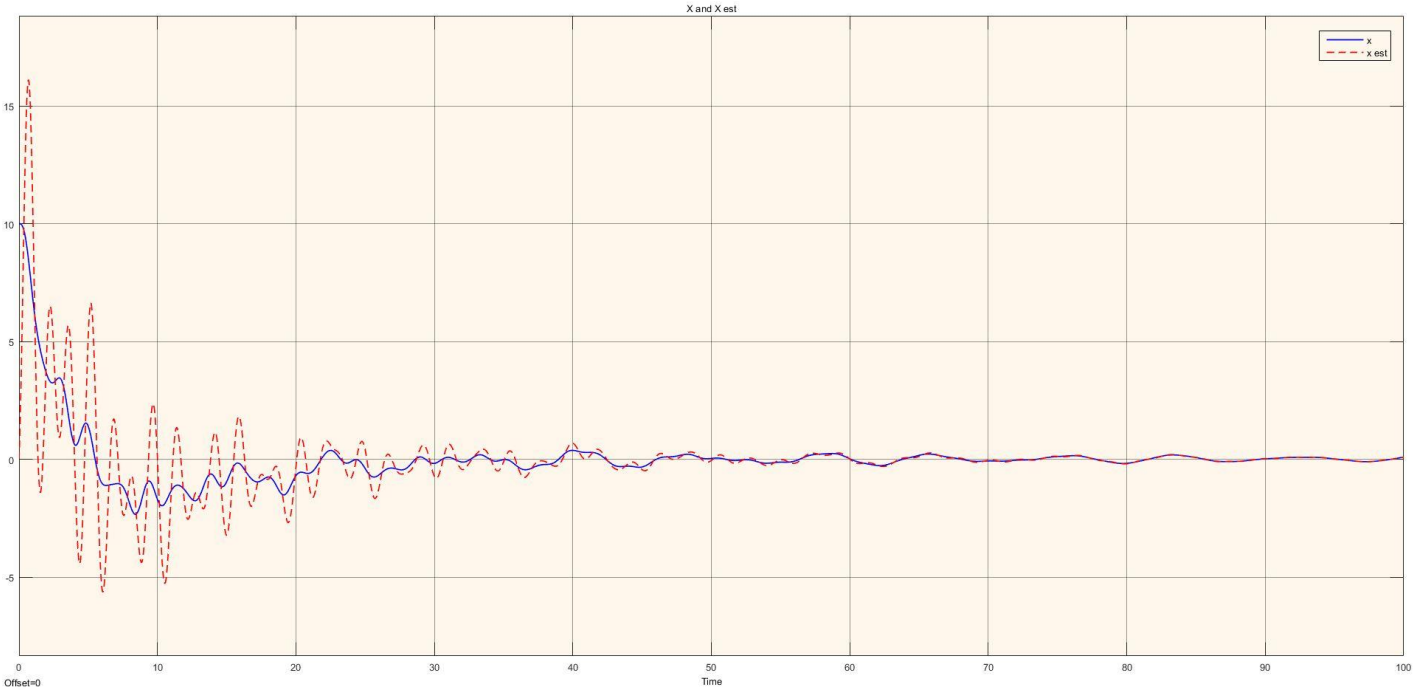
Non-Linearized System with Observer and a controller for Output Vector (x(t), θ2(t))



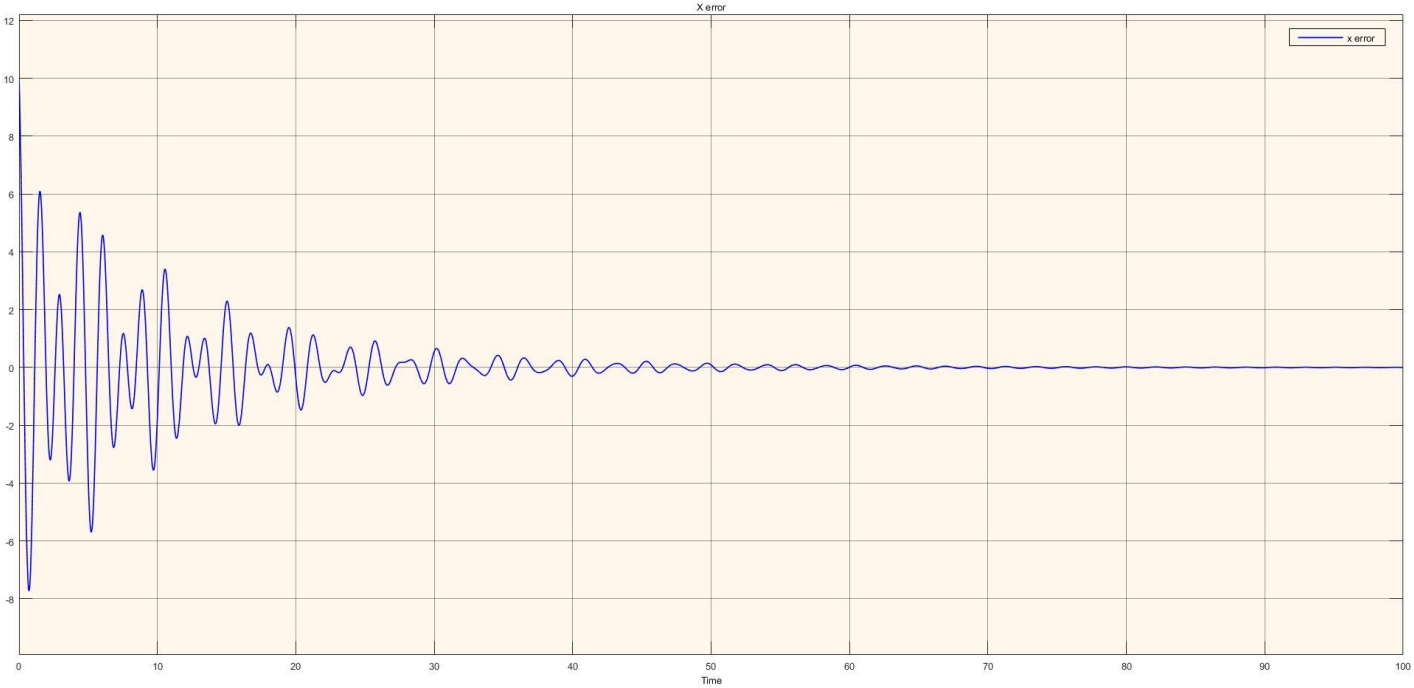
Non-Linearized System with Observer and Controller



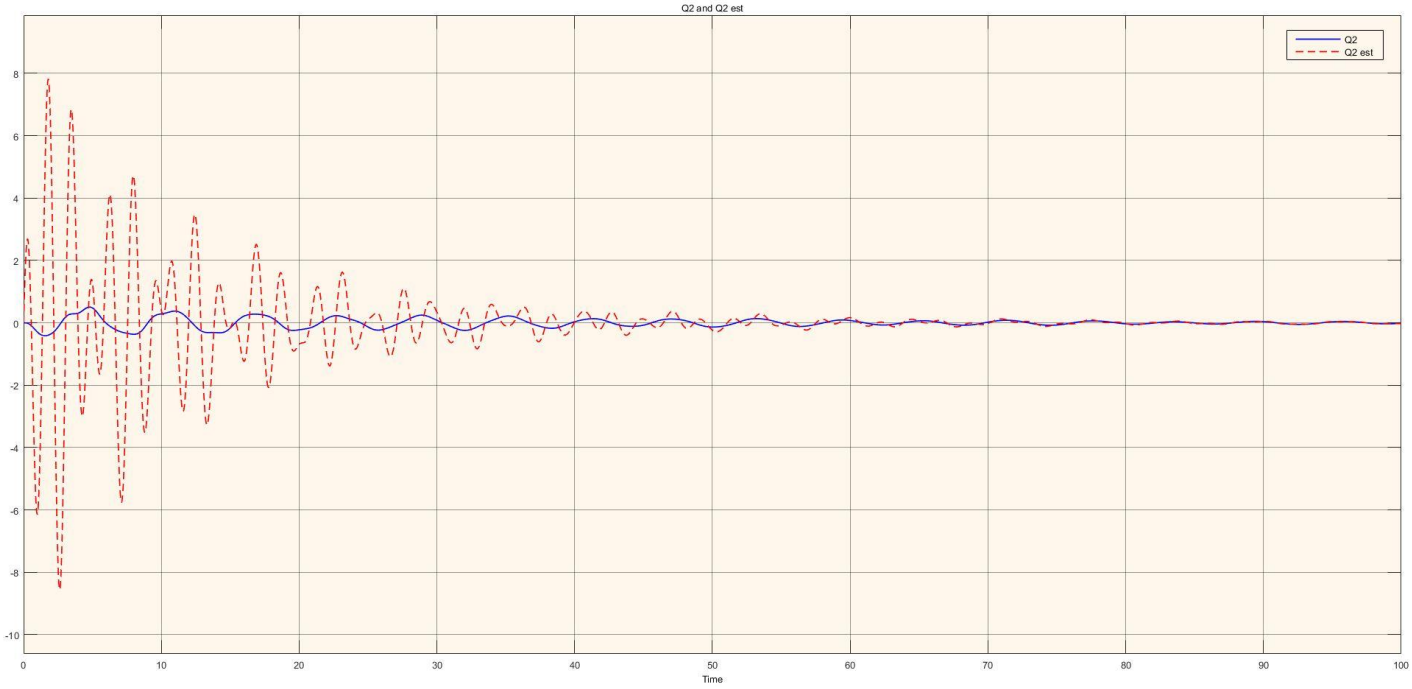
Observer's State Space Model



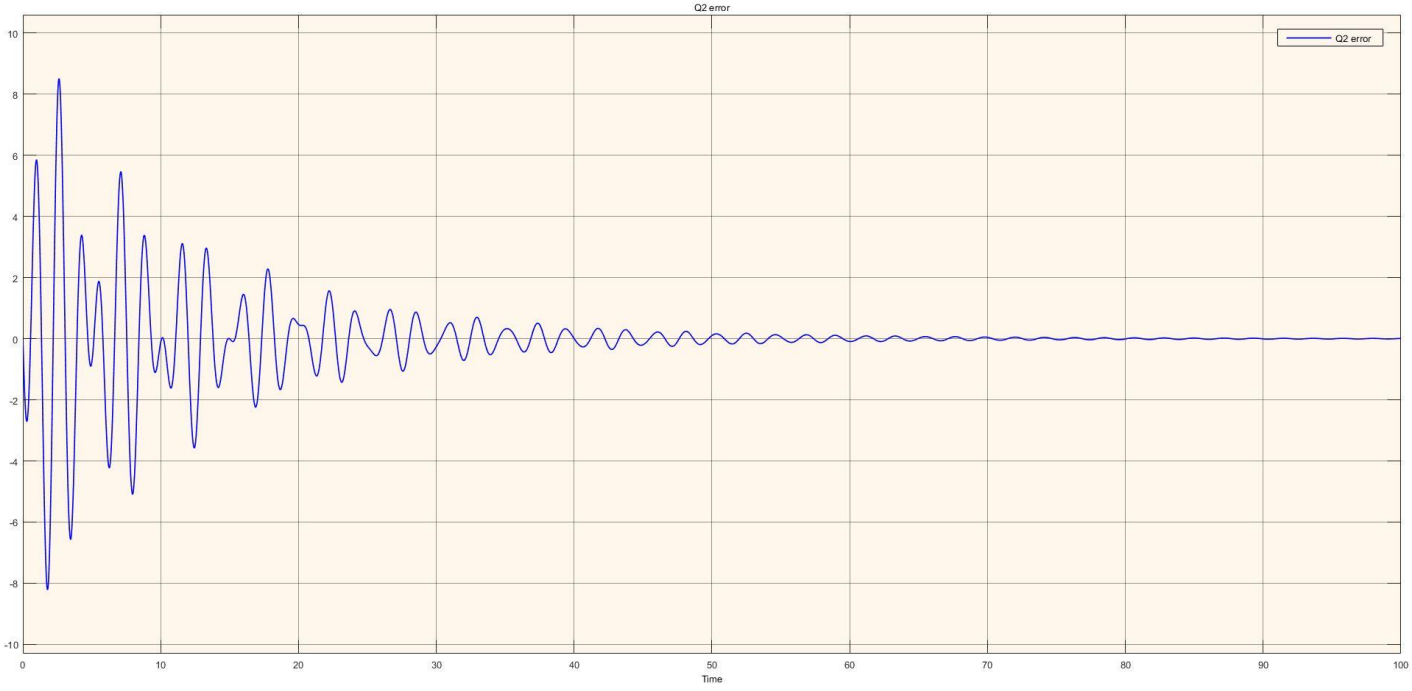
X from the system and X_{est} from the observer



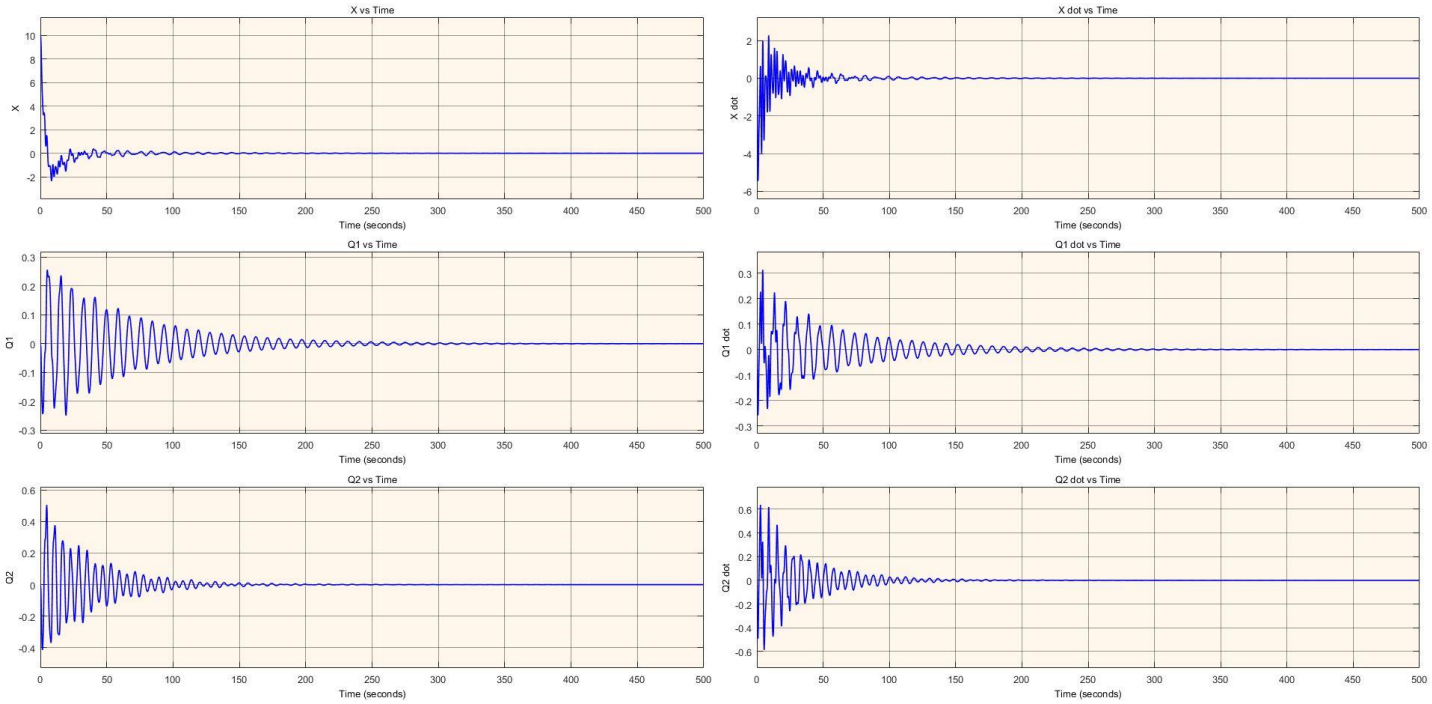
X_{error} given be (X-X_{est})



$\theta 2$ from the system and $\theta 2_{est}$ from the observer



$\theta 2_{error}$ given be ($\theta 2$- $\theta 2_{est}$)



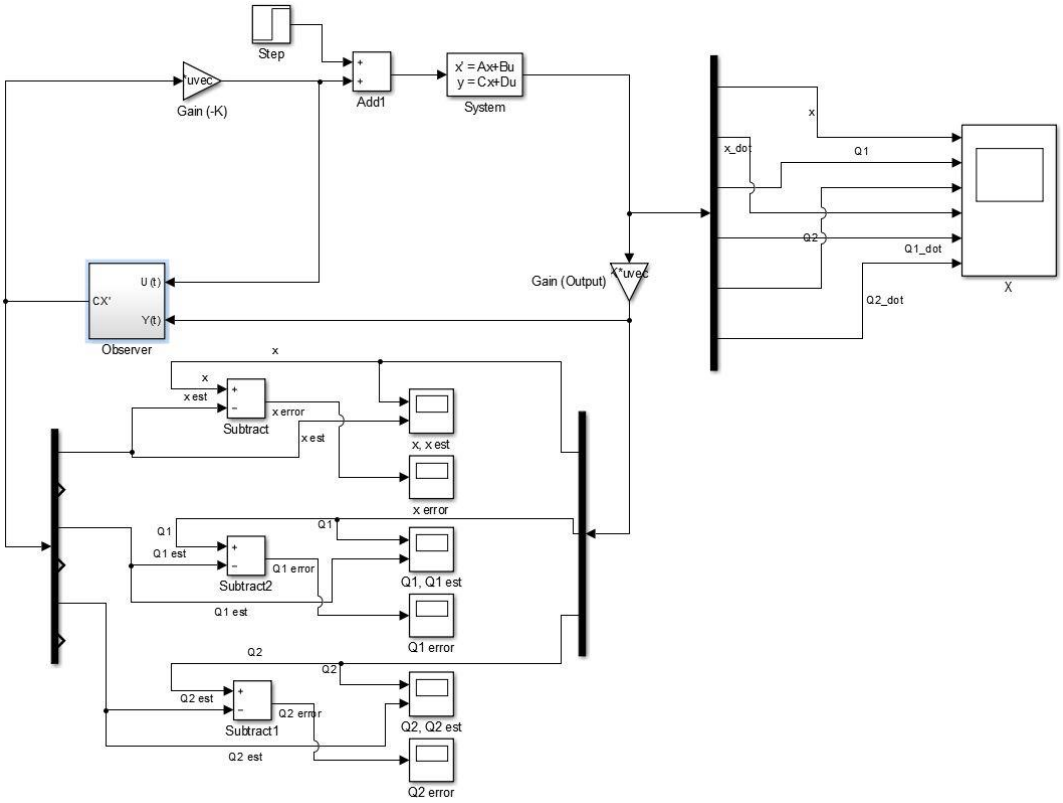
All the states of the system

Observations:

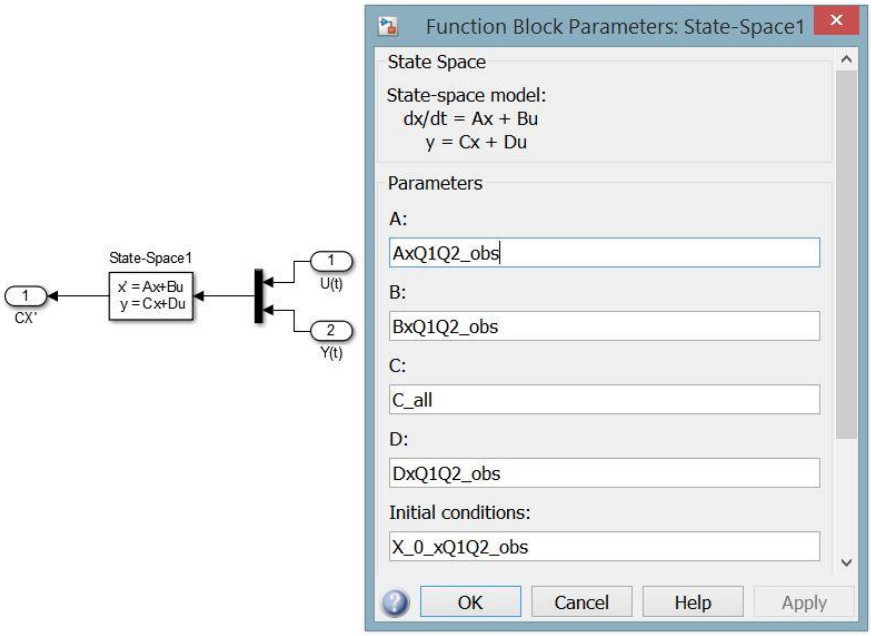
As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 55s and the observer's estimated state $\theta 2_{est}$ converges with that from the system's output θ2 around 65s.

Our output x stabilizes to 0.02 around 380s.

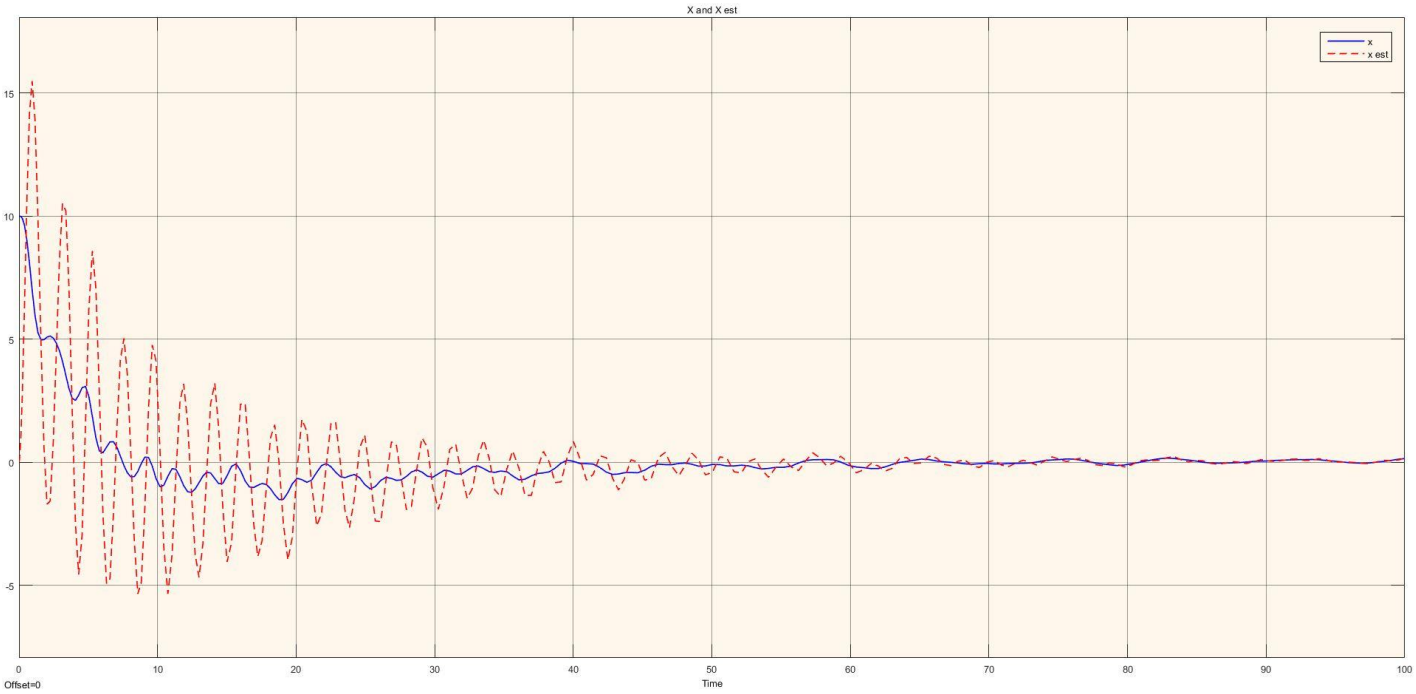
Linearized System with Observer and a controller for Output Vector (x(t), θ1(t), θ2(t))



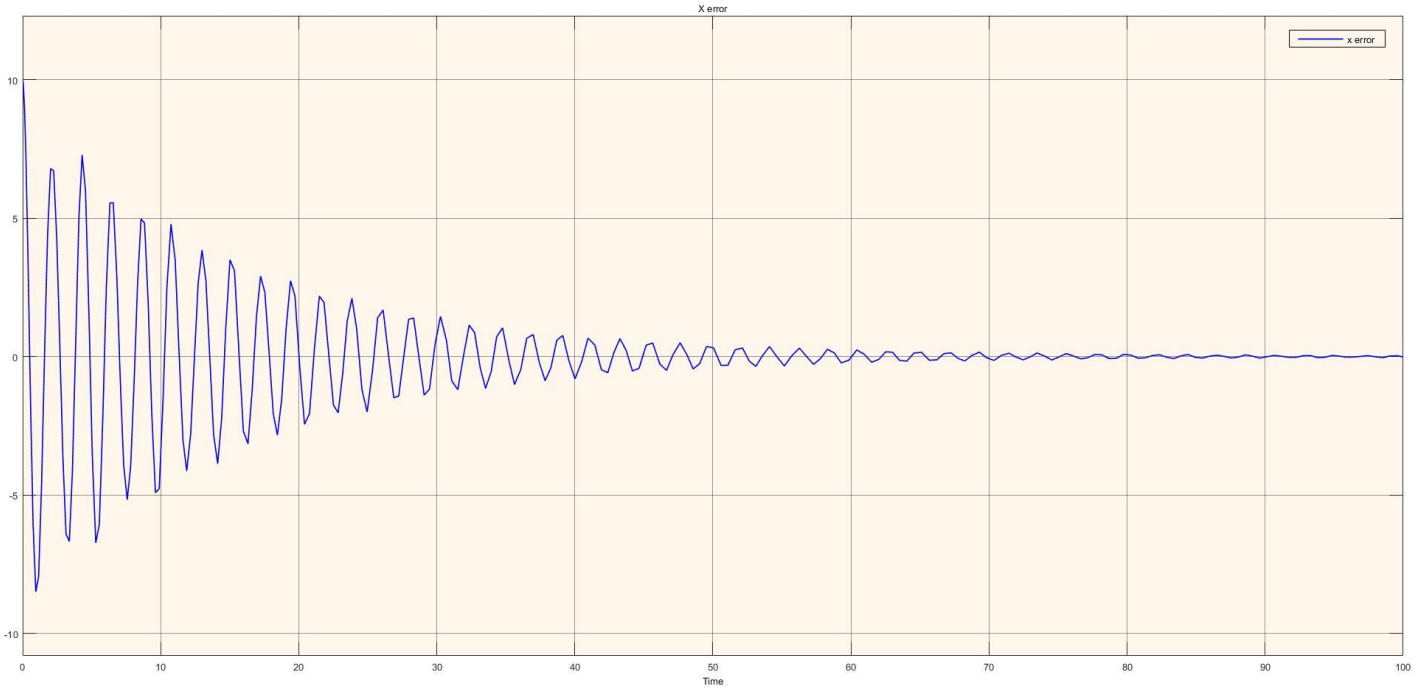
Linearized System with Observer and Controller



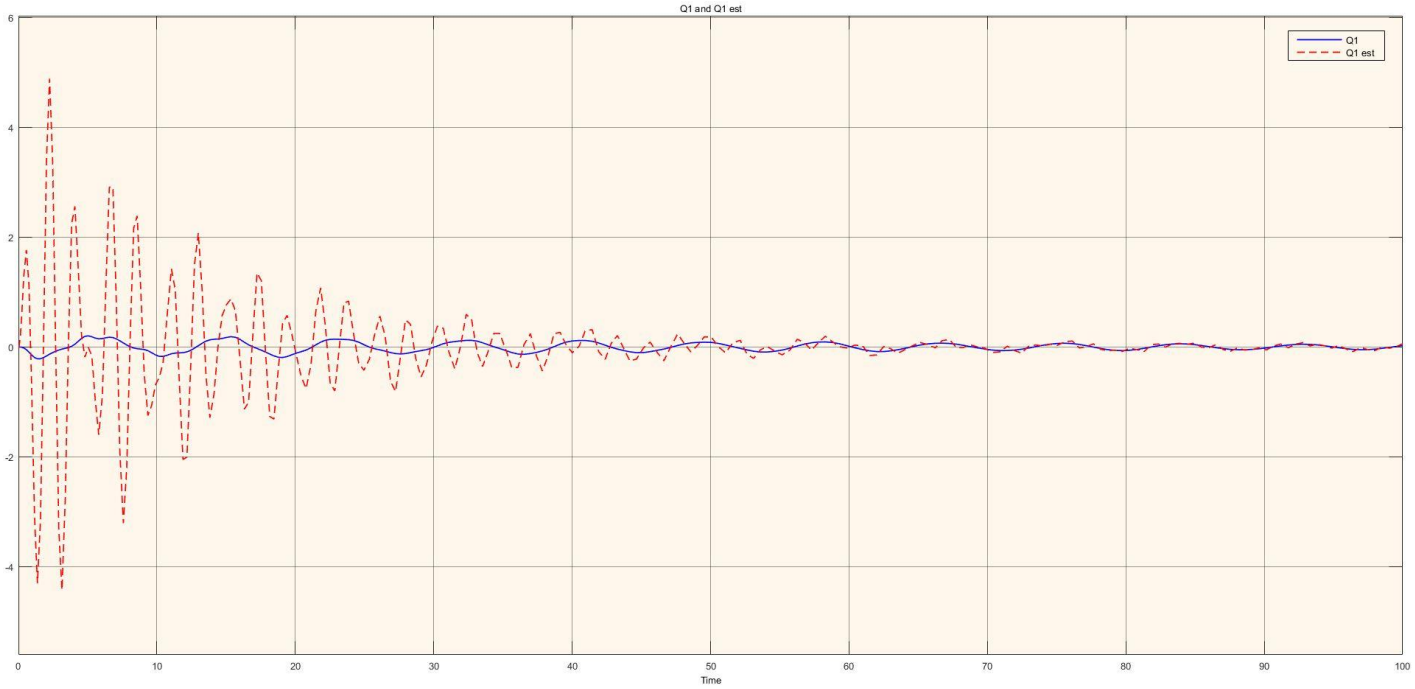
Observer's State Space Model



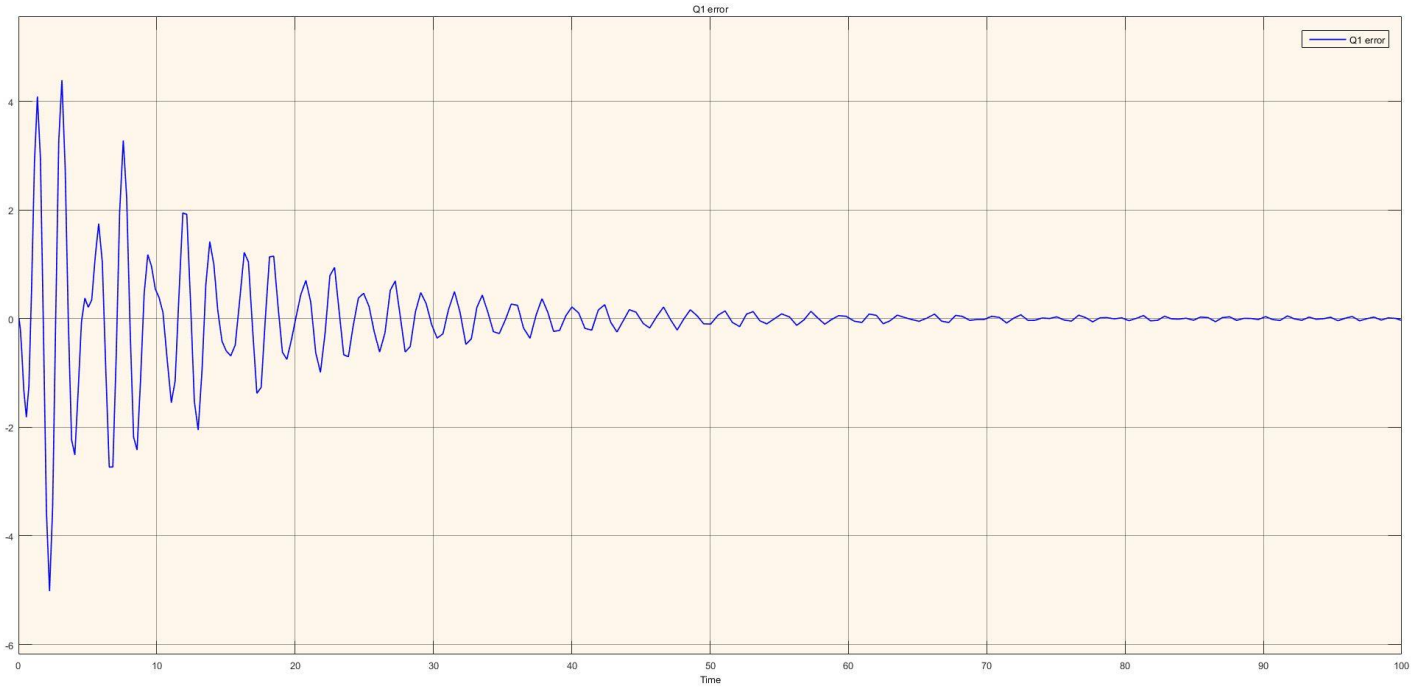
X from the system and X_{est} from the observer



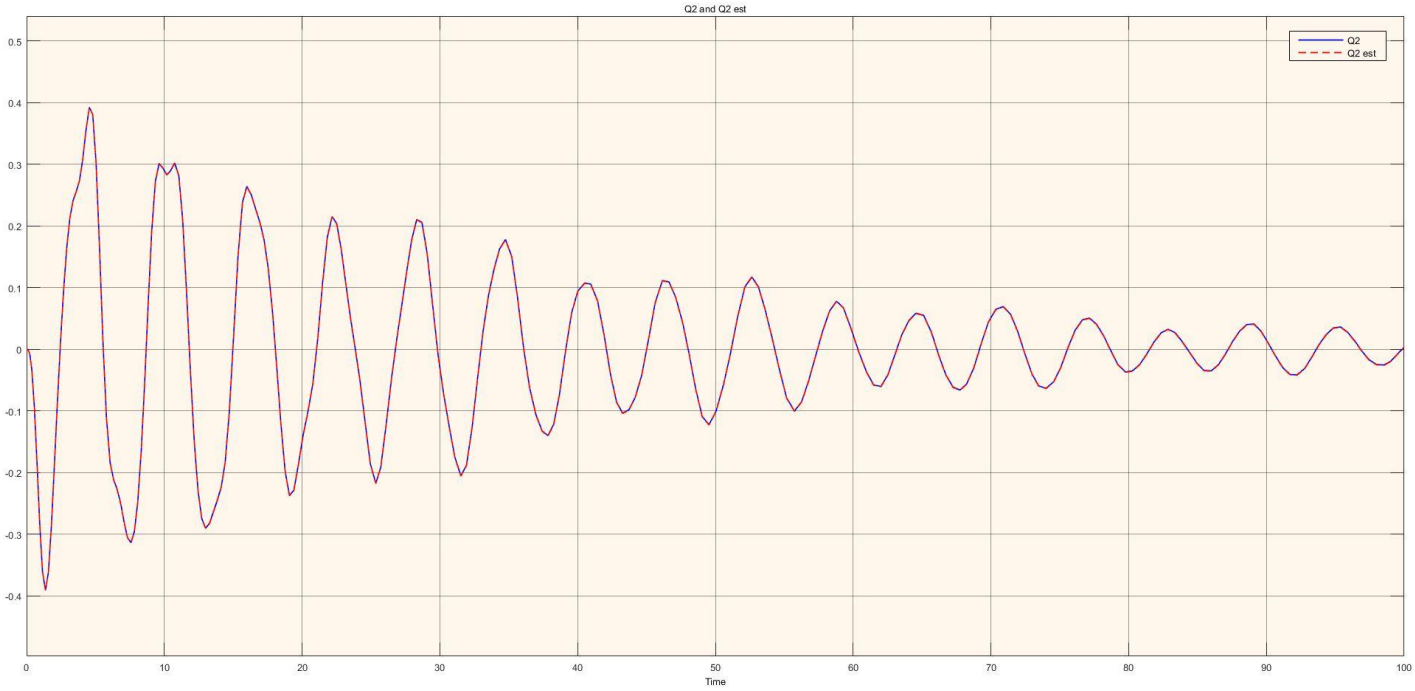
X_{error} given be ($X - X_{est}$)



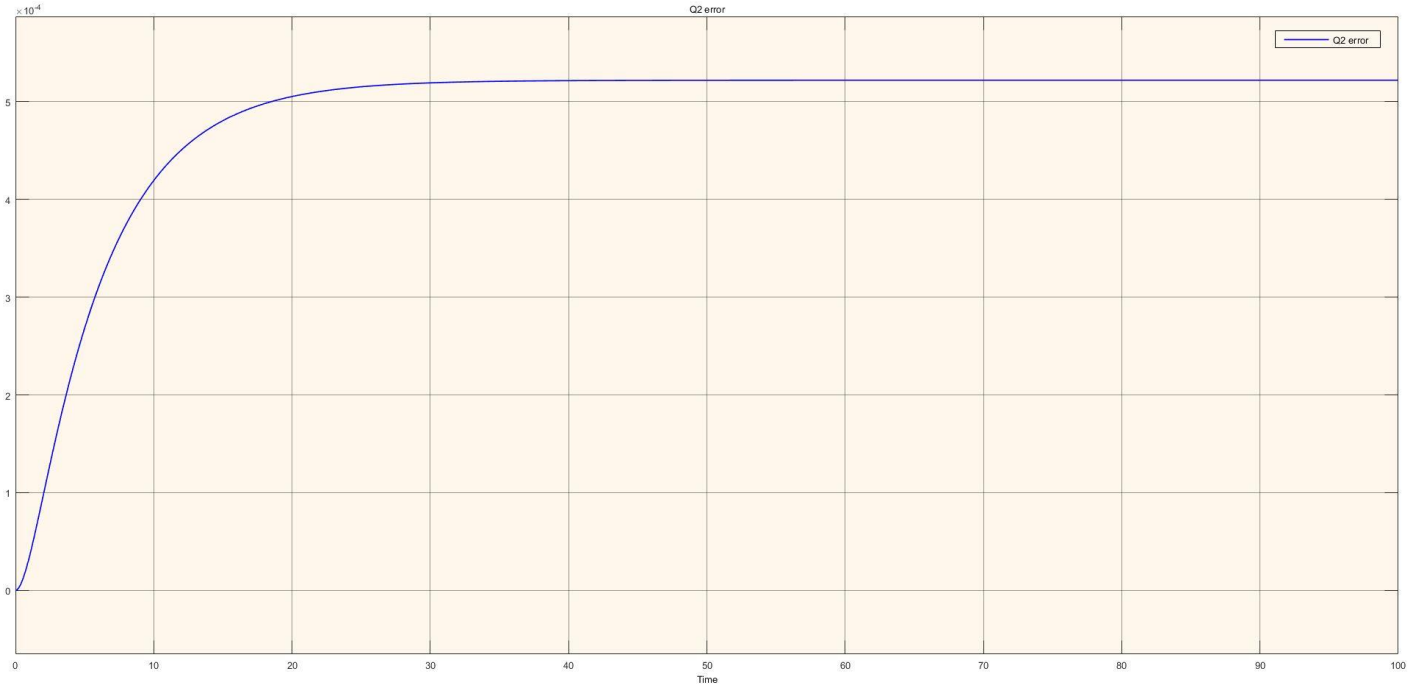
$\theta 1$ from the system and $\theta 1_{est}$ from the observer



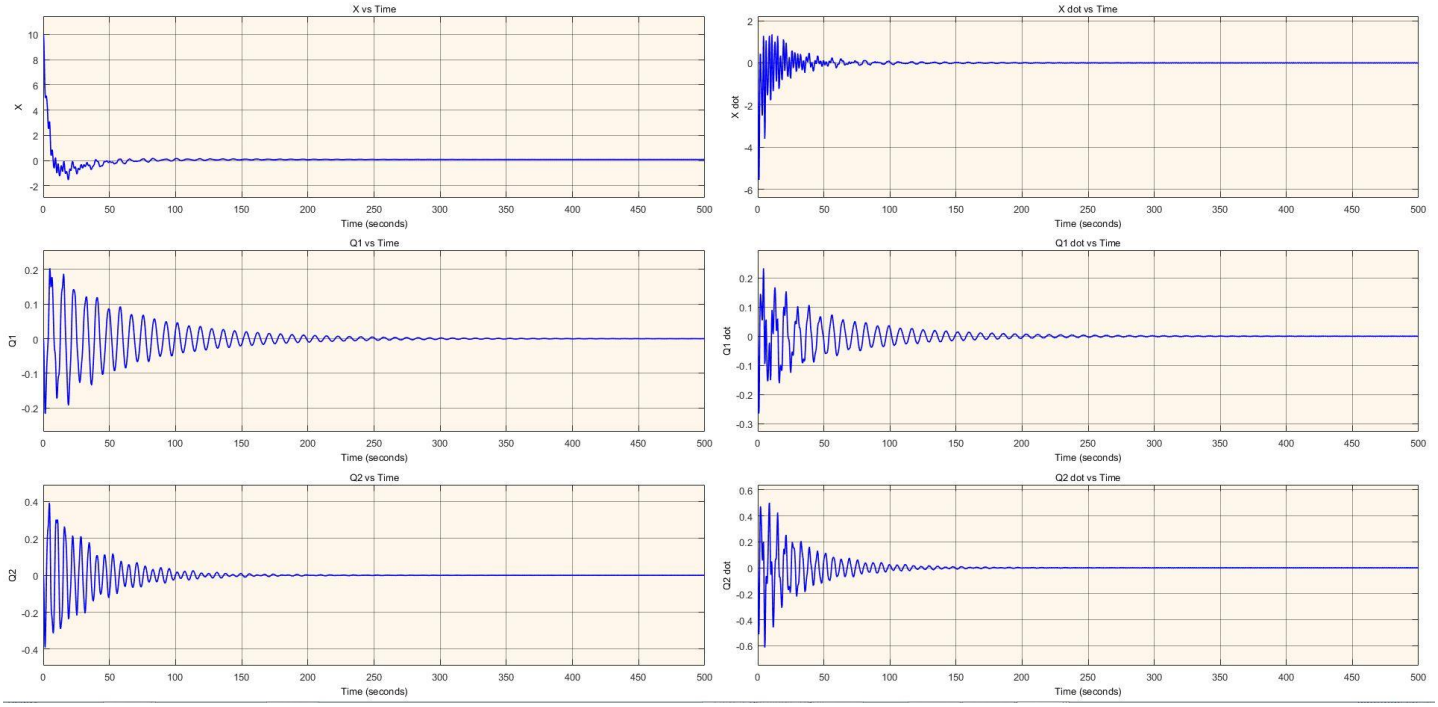
$\theta 1_{error}$ given be $(\theta 1 - \theta 1_{est})$



$\theta 2$ from the system and $\theta 2_{est}$ from the observer



$\theta 2_{error}$ given be ($\theta 2$- $\theta 2_{est}$)



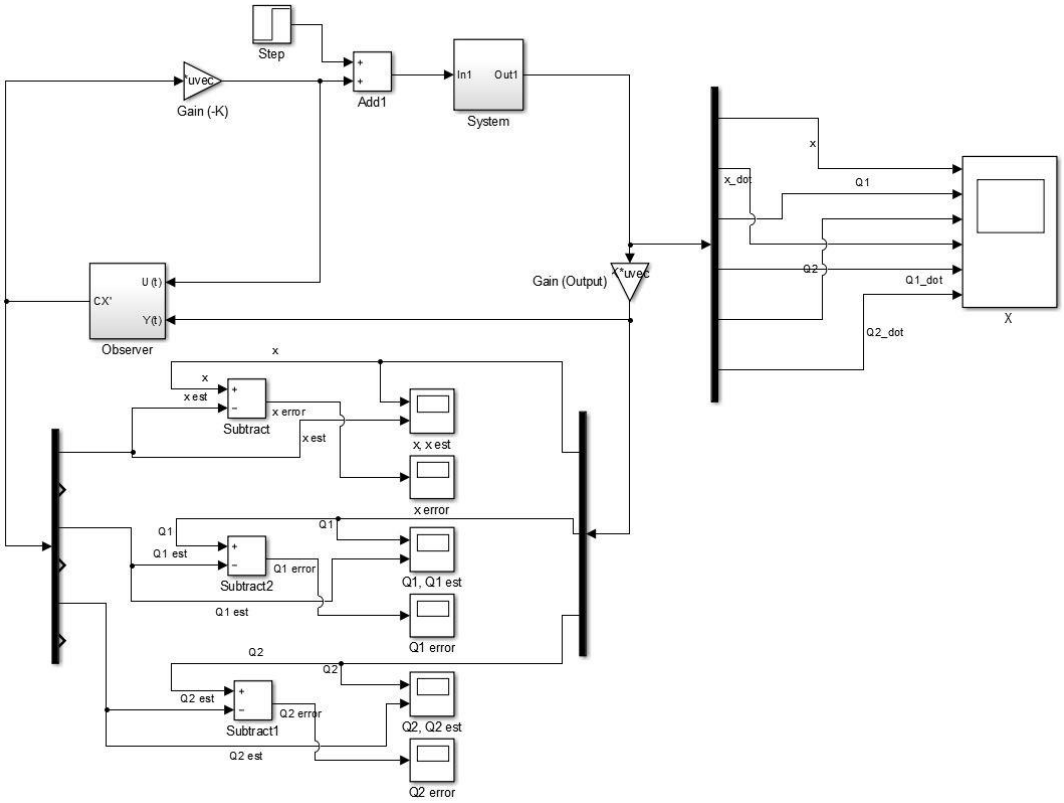
All the states of the system

Observations:

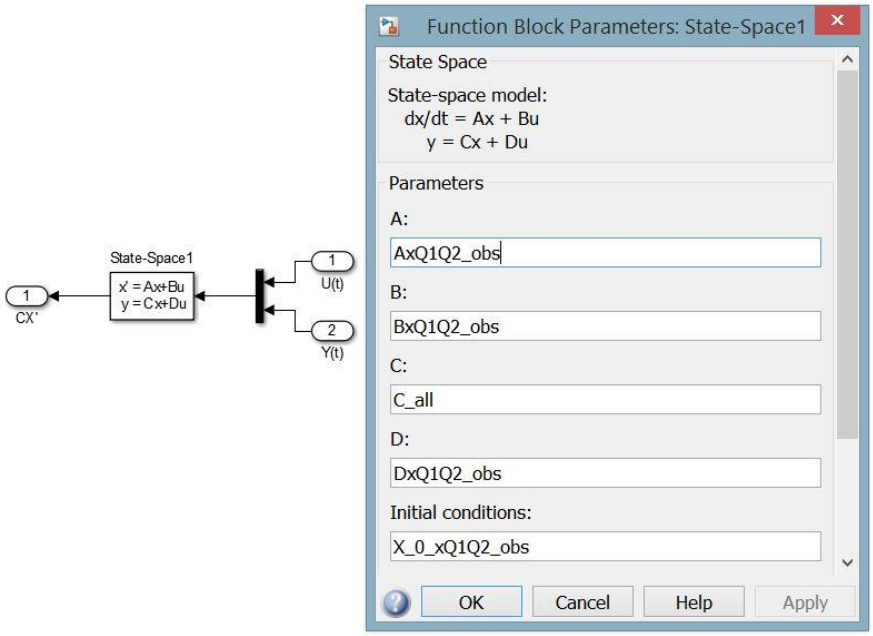
As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 75s, the observer's estimated state $\theta 1_{est}$ converges with that from the system's output $\theta 1$ around 75s and the observer's estimated state $\theta 2_{est}$ converges with that from the system's output $\theta 2$ around 0s (Right from the start!).

Our output x stabilizes to 0.08 around 380s.

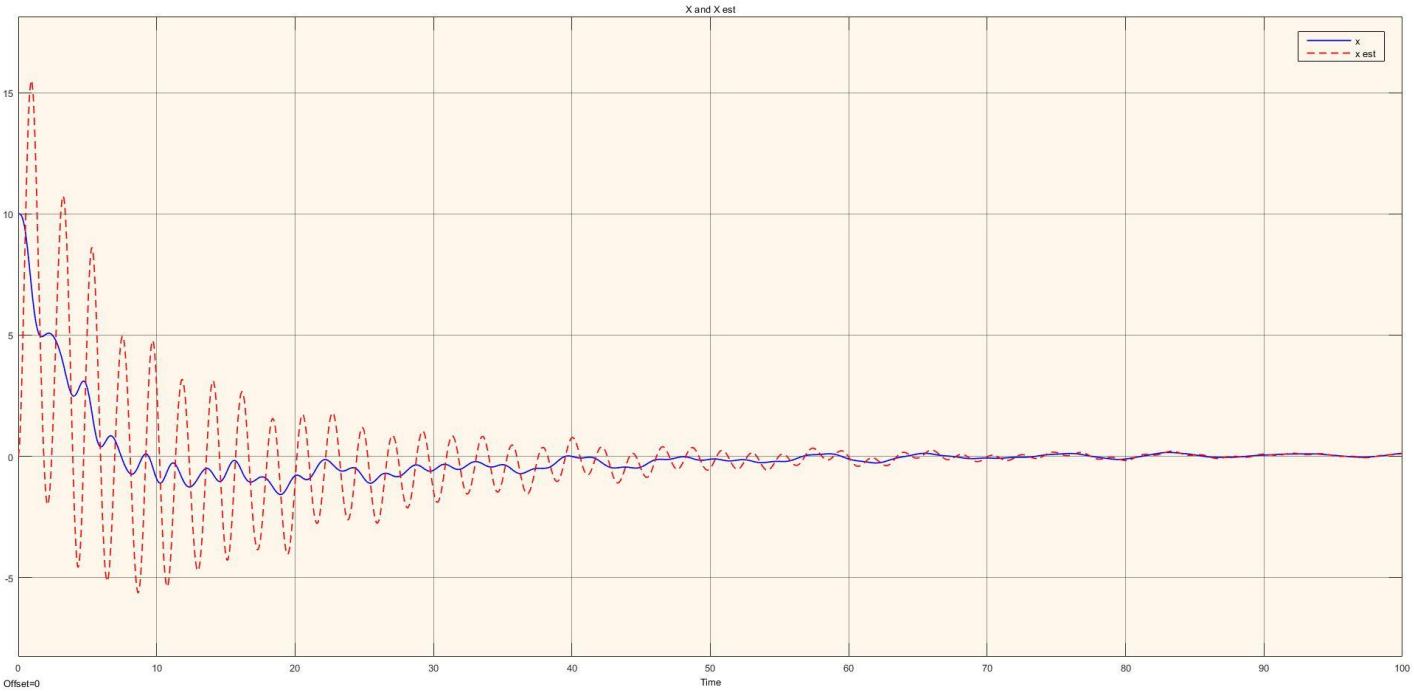
Non-Linearized System with Observer and a controller for Output Vector (x(t), θ1(t), θ2(t))



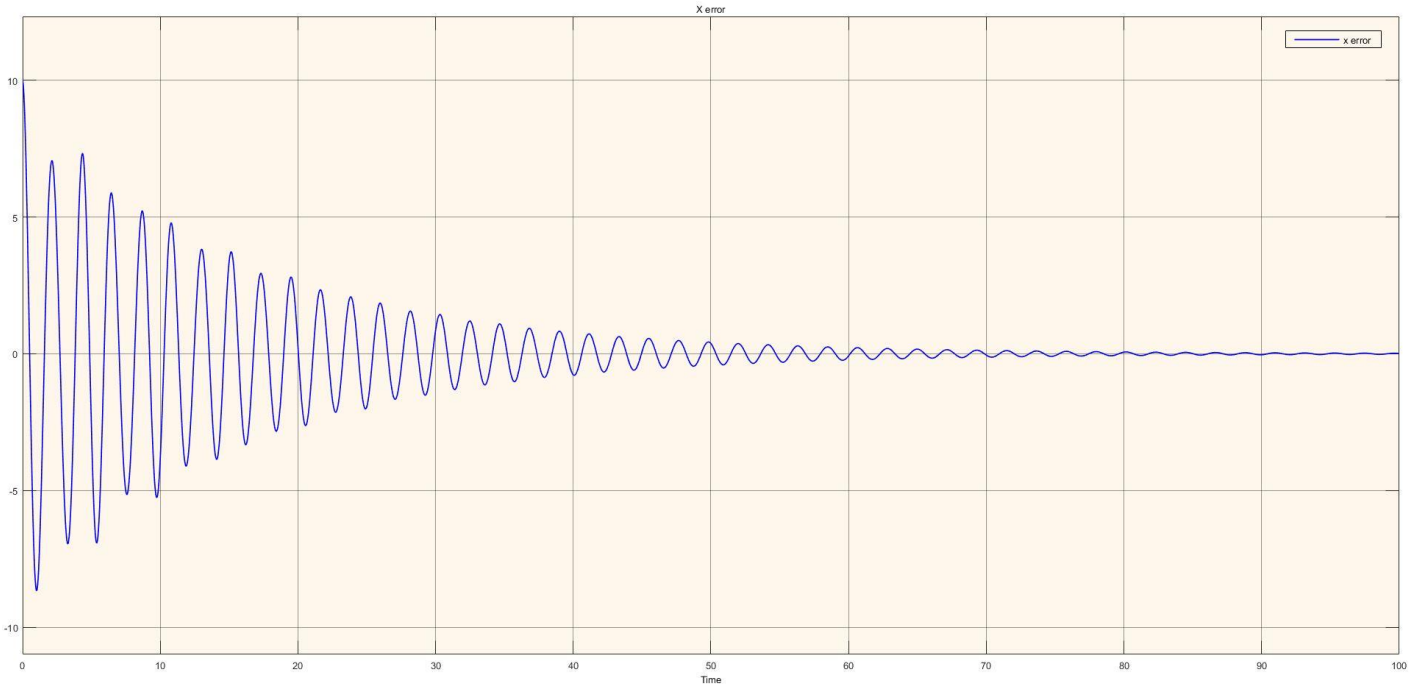
Non-Linearized System with Observer and Controller



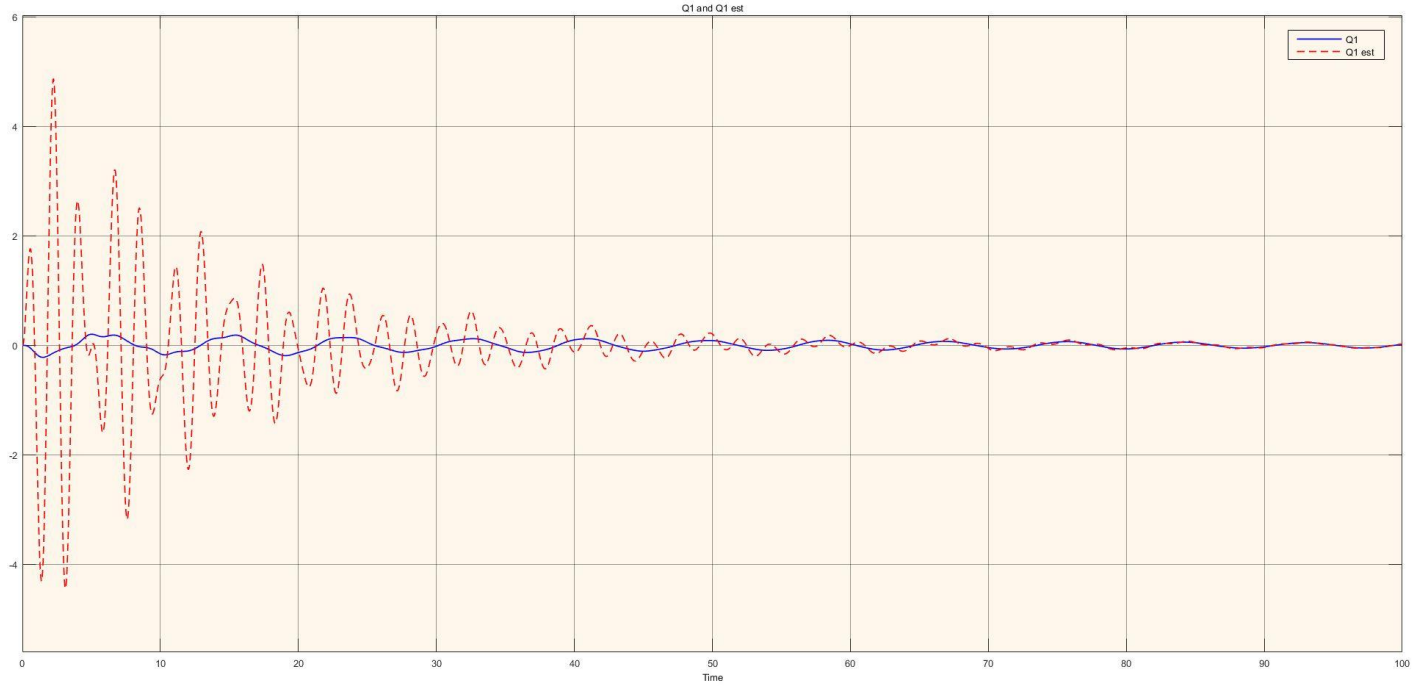
Observer's State Space Model



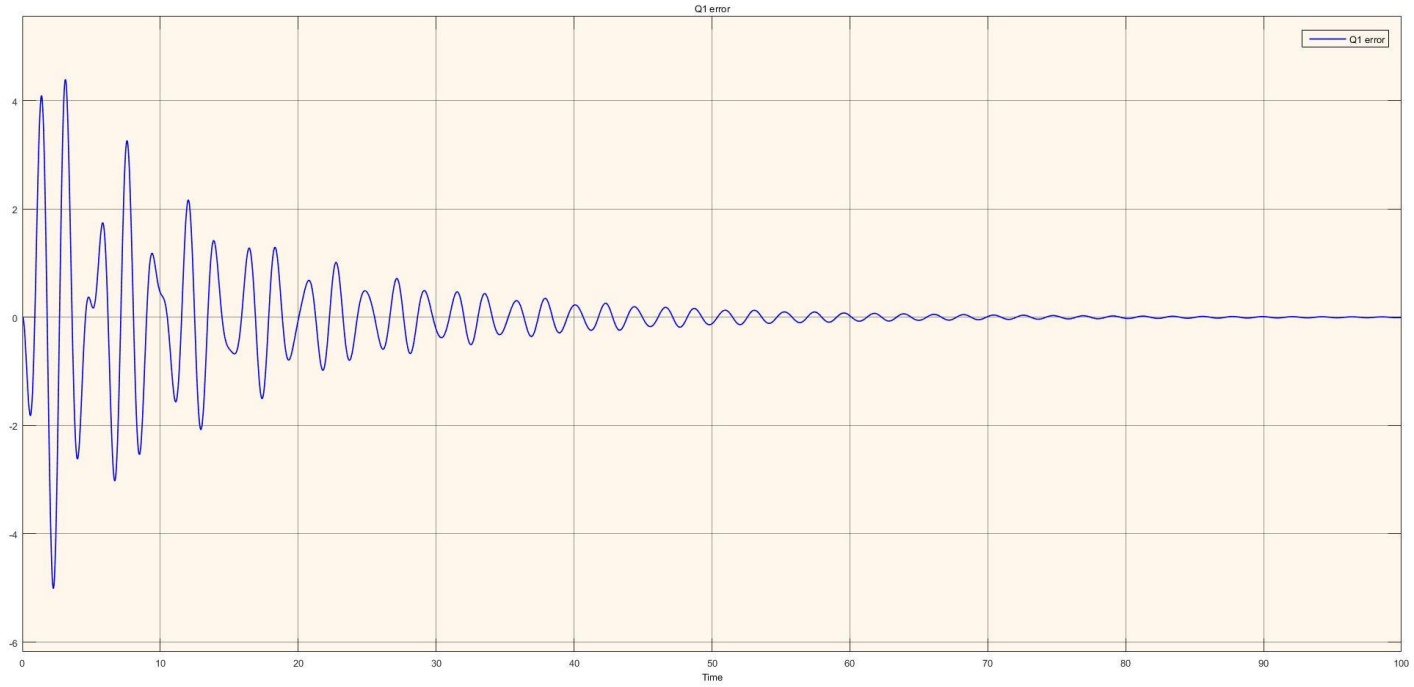
X from the system and X_{est} from the observer



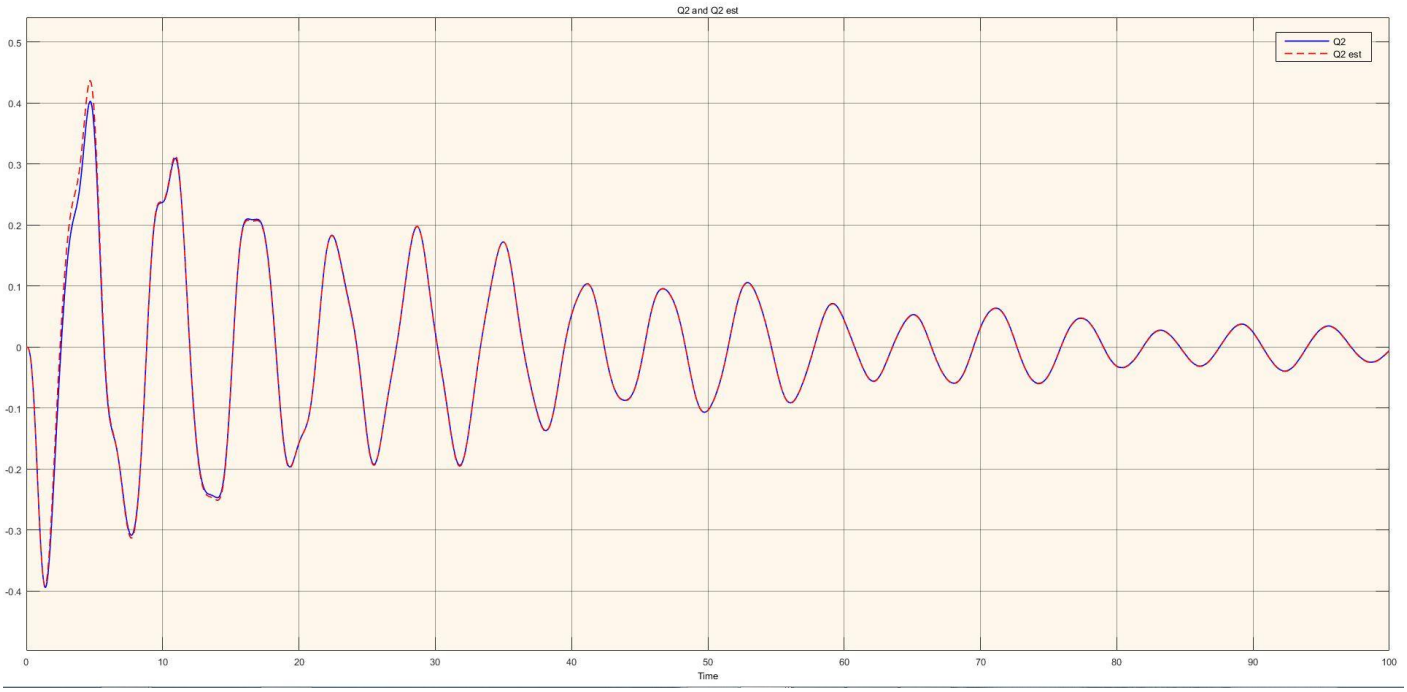
X_{error} given be ($X - X_{est}$)



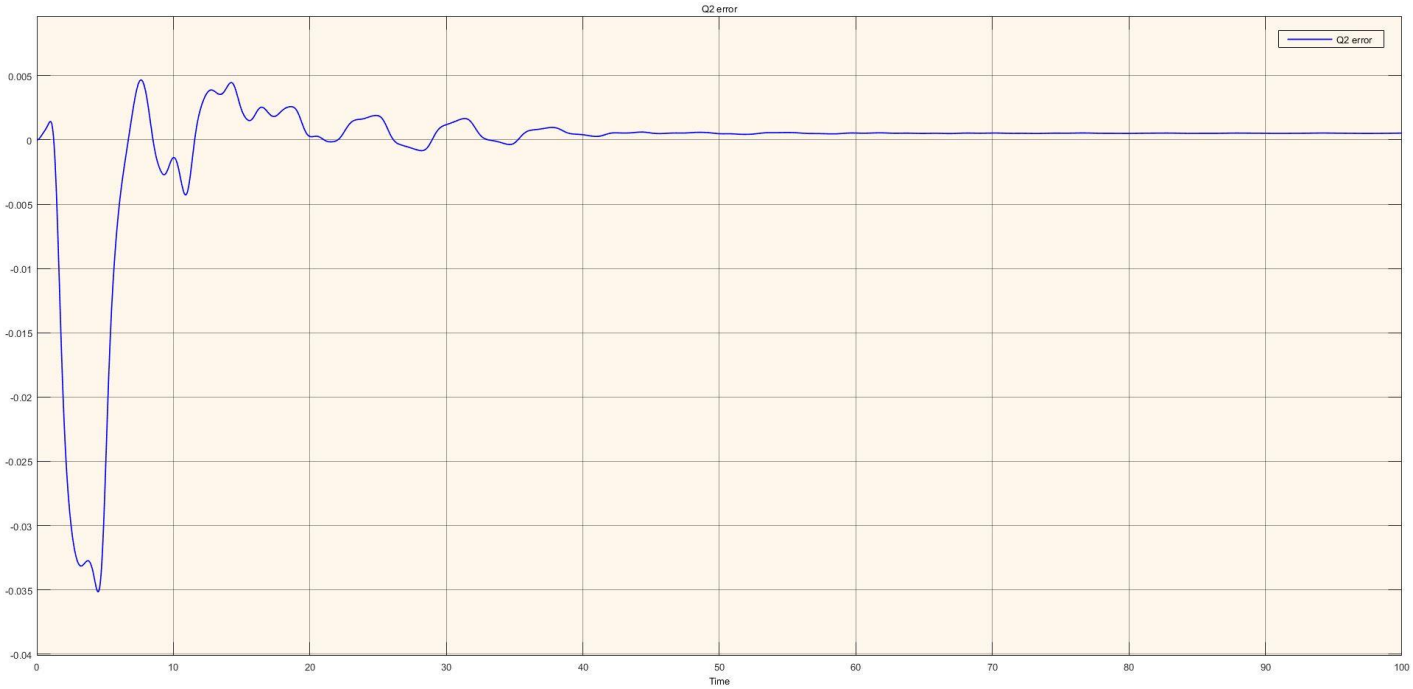
$\theta 1$ from the system and $\theta 1_{est}$ from the observer



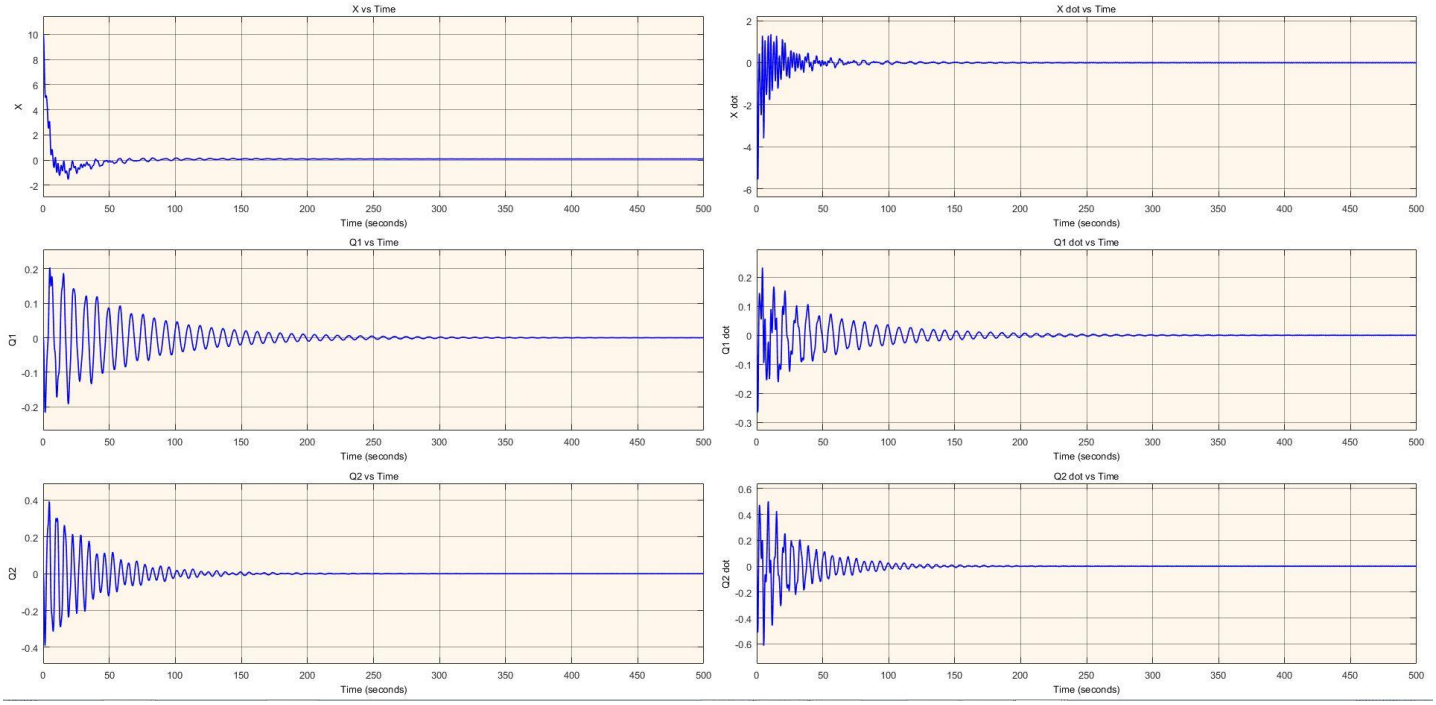
$\theta 1_{error}$ given be ($\theta 1$ - $\theta 1_{est}$)



$\theta 2$ from the system and $\theta 2_{est}$ from the observer



$\theta 2_{error}$ given be ($\theta 2$- $\theta 2_{est}$)



All the states of the system

Observations:

As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 75s, the observer's estimated state $\theta 1_{est}$ converges with that from the system's output θ1 around 75s and the observer's estimated state $\theta 2_{est}$ converges with that from the system's output θ2 around 0s (Right from the start!).

Our output x stabilizes to 0.08 around 360s.

Summary

Keeping the same values of Q and R for all the three output vectors showed that as the number of outputs in the output vector increased:

- Amplitude of the oscillations decreased
- The amount of time taken by the system to converge to zero was also reduced

The reason the x output of the system does not stabilize to 0 is because the controller is a PD controller and hence has an offset error. This can be fixed by adding an Integrator term to the controller and making it a PID controller which will drive the offset error to 0.

The outputs of the Non-Linearized systems when compared to their Linearized counterparts show a very slight decrease in the amplitude of the oscillations.

Further Trials

Now we decided to play around more with the values of Q and R and also with the multiplier for the Eigen values of L from K matrix and check how the system responded for different values. What we saw was interesting.

Each system (with different output vectors) had a better response for a different set of Q and R matrices which were unique for that system (Linear and Non-Linear) and made the other systems (with lesser number of outputs) unstable when used for them.

Also, the limit of the multiplier used to set the poles of Observer matrix L increased as the number of outputs in the output vector increased.

The maximum value of the multiplier that we were able to use (for the obtained Q and R values) was 3 for output vector (x(t)) and the multiplier 4 for output vector (x(t), θ2(t)) and for the output vector (x(t), θ1(t), θ2(t)), we could use any multiplier (even till 20) but decided to use 5 as going above that would have made our system more susceptible to noise.

As the outputs from the Linearized and the Non-Linearized systems are very similar, we have shown only Non-Linearized system's output for the above statements.

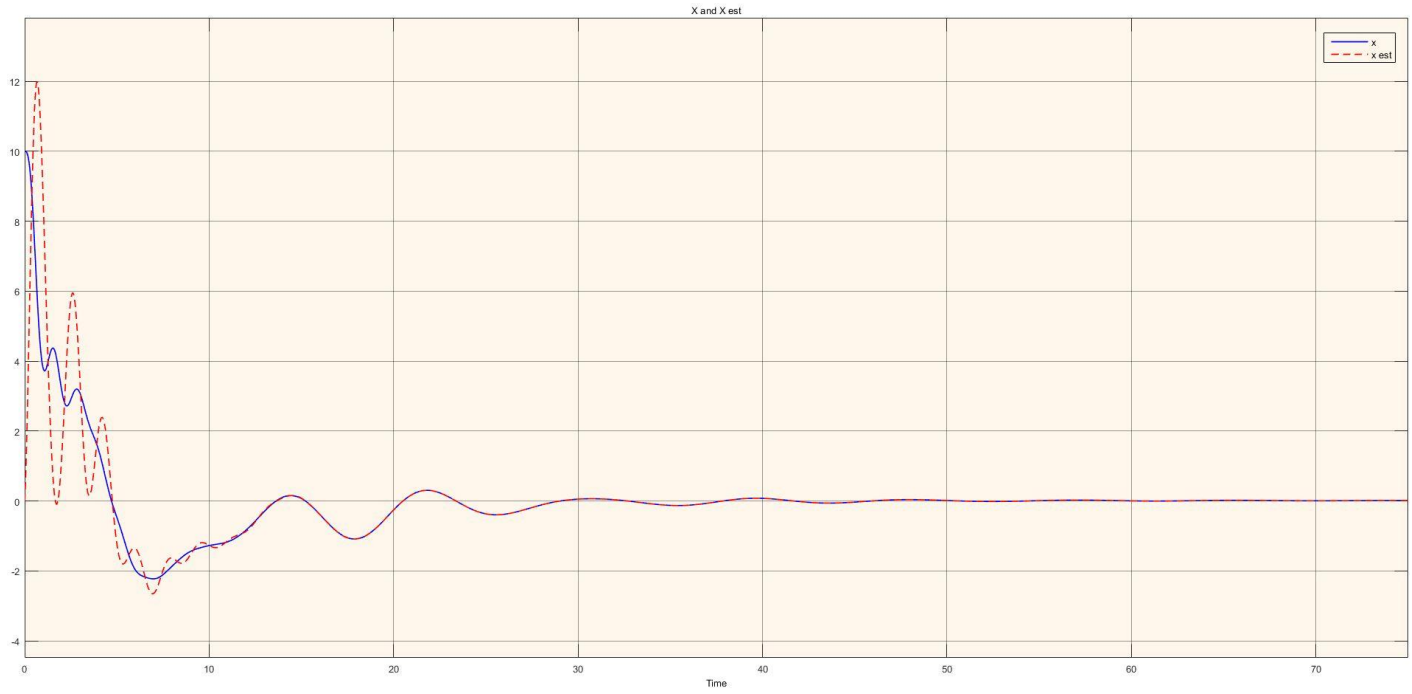
Non-Linearized System with Observer and a controller for Output Vector (x(t),θ1(t),θ2(t))

Matlab Code:

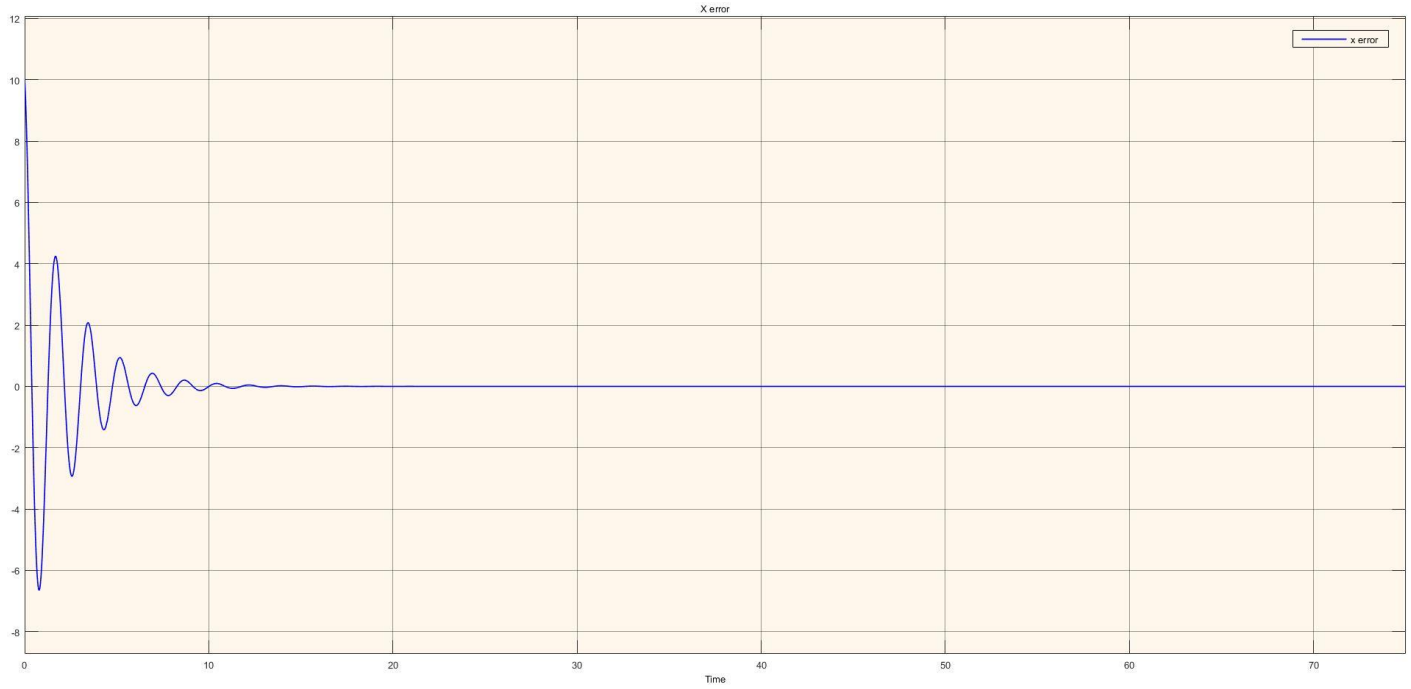
```
%% Set Values of R and Q to get K matrix using LQR()
Q= [1 0 0 0 0 0;
    0 60 0 0 0 0;
    0 0 1500 0 0 0;
    0 0 0 500 0 0;
    0 0 0 0 500 0;
    0 0 0 0 0 500];
R=  0.0001;

K,S,e] = lqr(A,B,Q,R)
PolesL= 5*e
```

The values obtained for the poles:

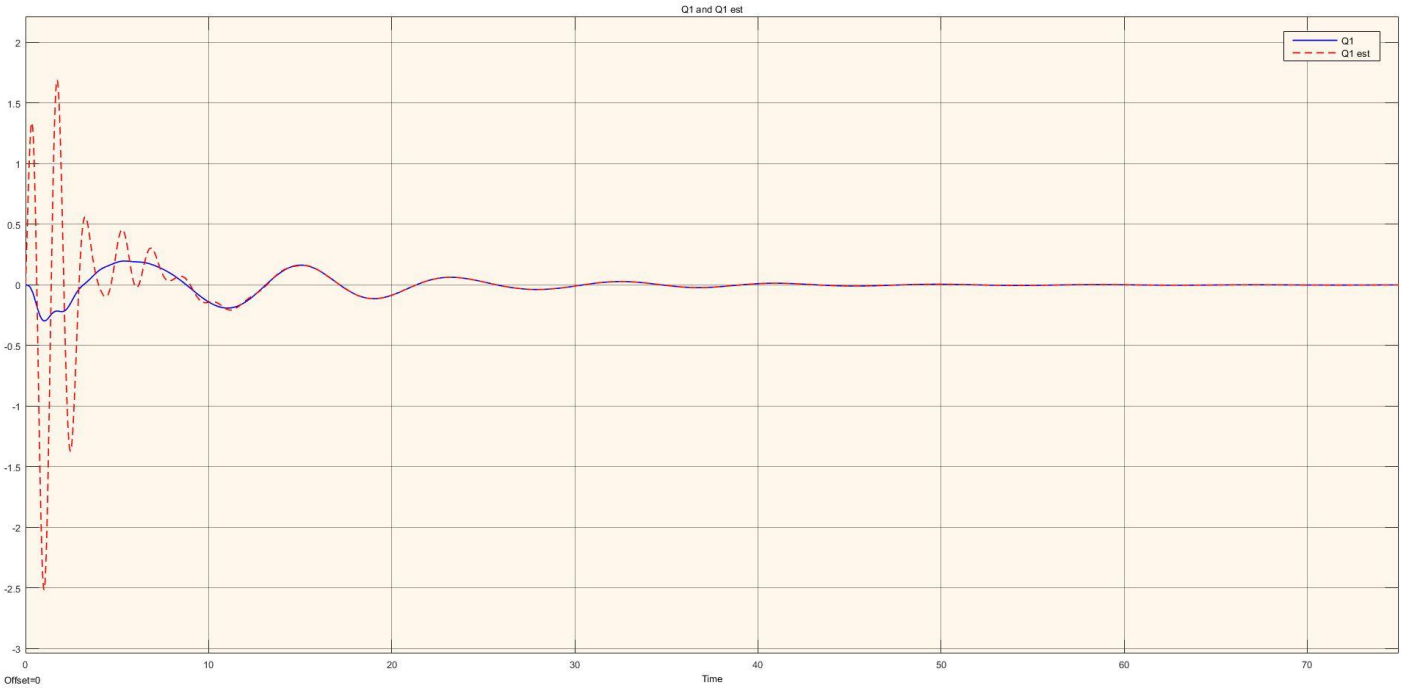
```
e =

  -0.6642 + 0.0000i
  -0.1323 + 0.0000i
  -0.1340 + 1.0120i
  -0.1340 - 1.0120i
  -0.0884 + 0.7188i
  -0.0884 - 0.7188i


PolesL =

  -3.3210 + 0.0000i
  -0.6613 + 0.0000i
  -0.6698 + 5.0599i
  -0.6698 - 5.0599i
  -0.4421 + 3.5942i
  -0.4421 - 3.5942i
```



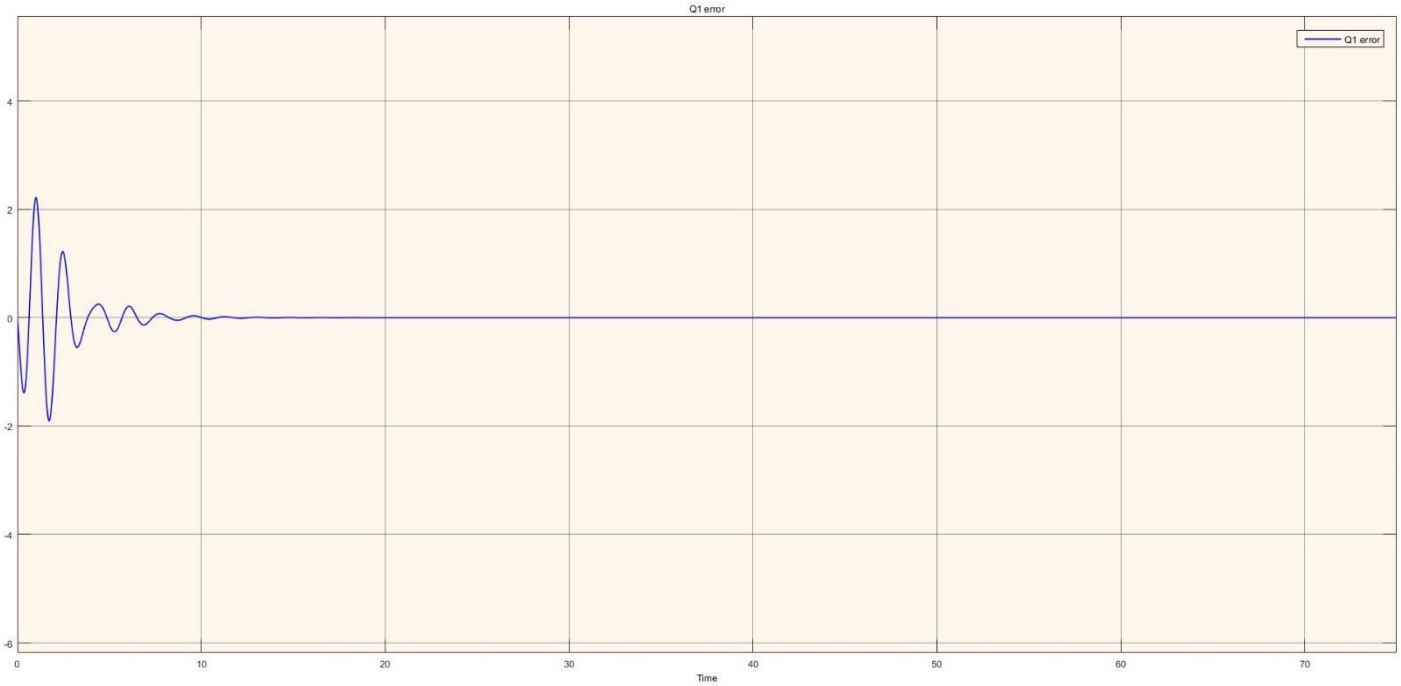
X from the system and X_{est} from the observer



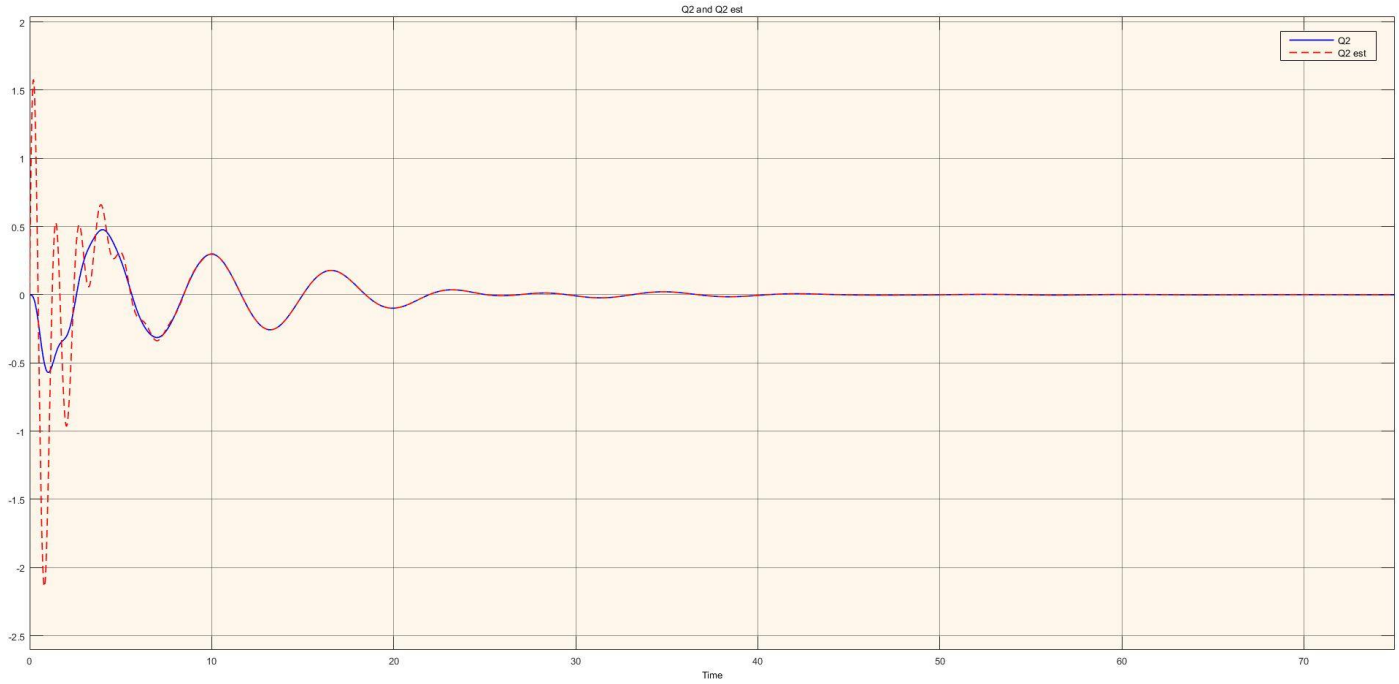
X_{error} given be (X-X_{est})



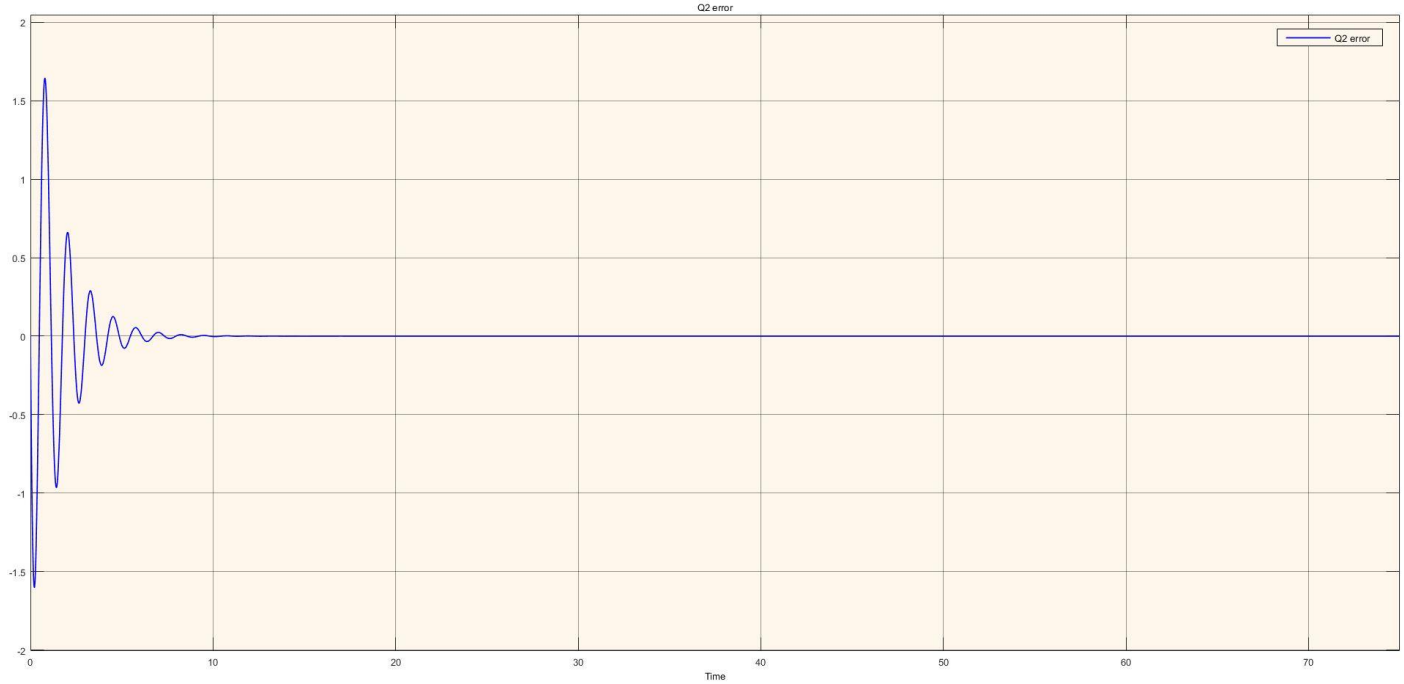
$\theta 1$ from the system and $\theta 1_{est}$ from the observer



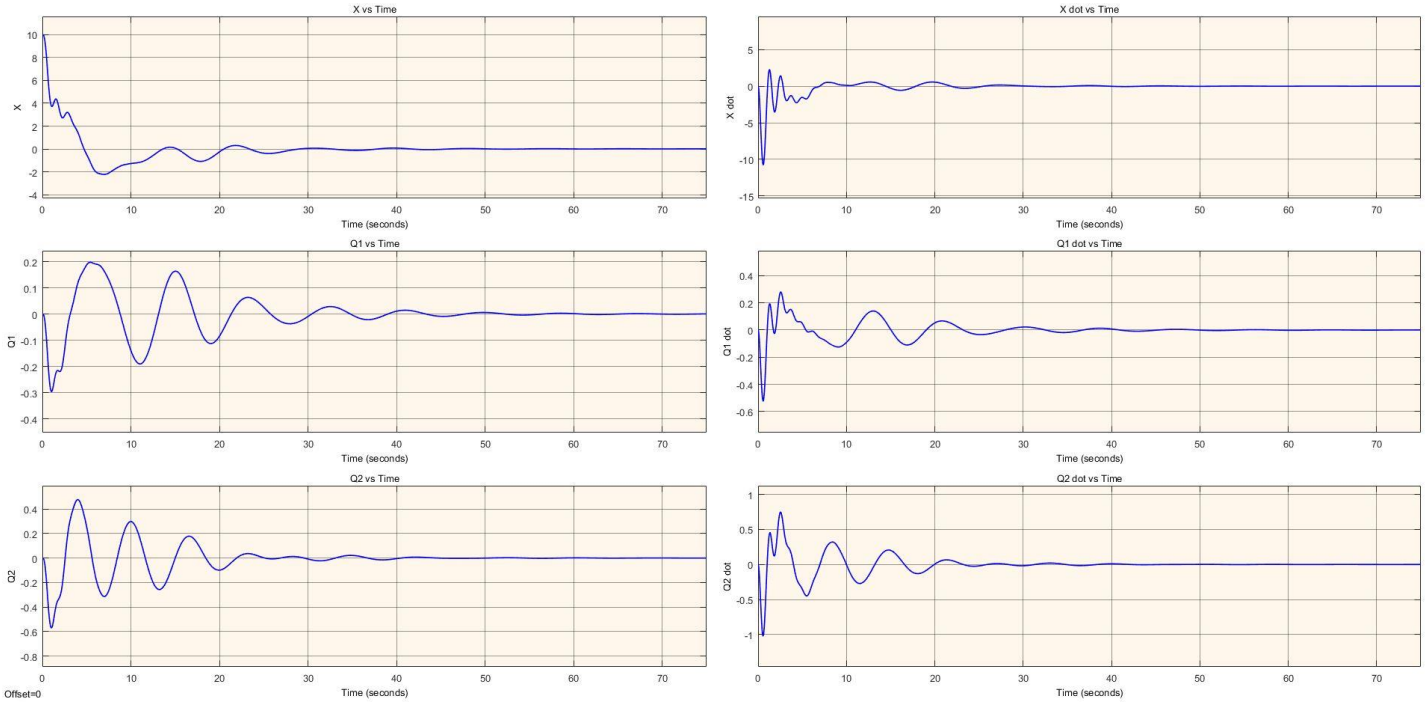
$\theta 1_{error}$ given be $(\theta 1 - \theta 1_{est})$



$\theta 2$ from the system and $\theta 2_{est}$ from the observer



$\theta 2_{error}$ given be ($\theta 2 - \theta 2_{est}$)



All the states of the system

As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 14s, the observer's estimated state $\theta 1_{est}$ converges with that from the system's output θ1 around 14s and the observer's estimated state $\theta 2_{est}$ converges with that from the system's output θ2 around 10s.

The whole system stabilizes around 60s with x at an offset of around 0.01

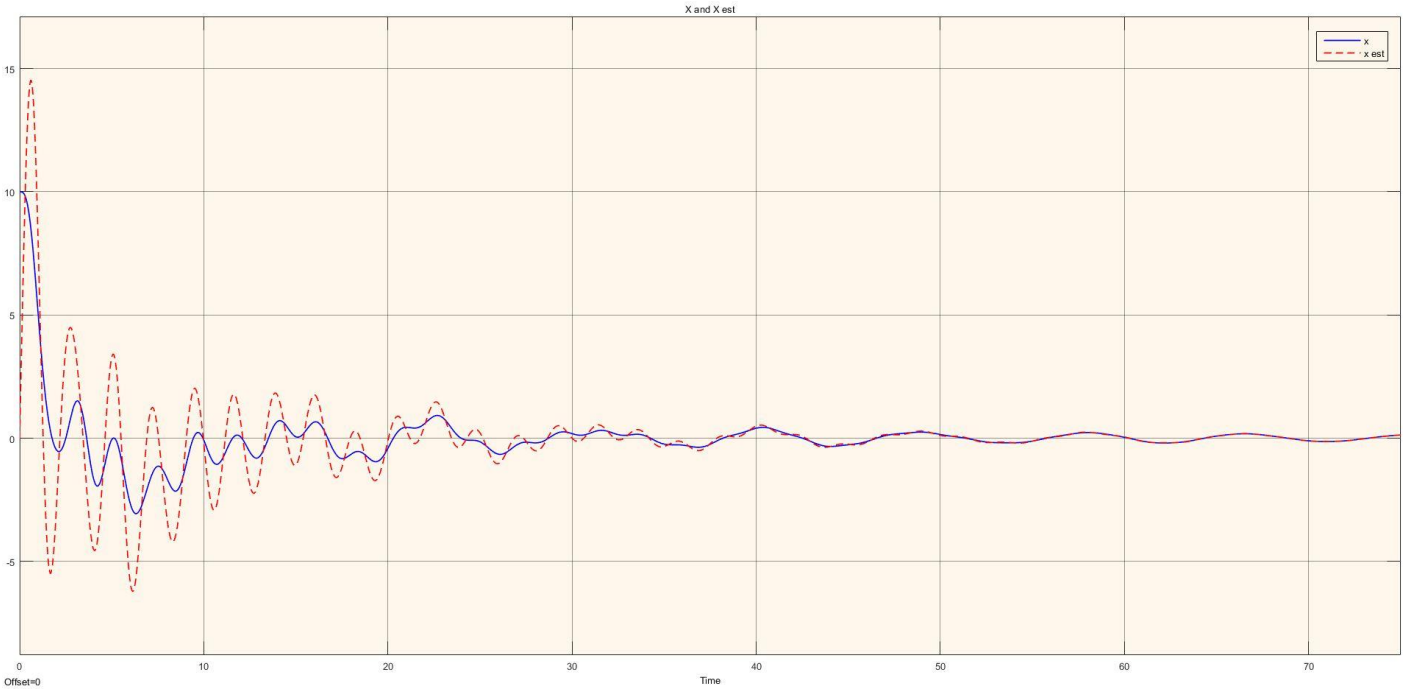
Non-Linearized System with Observer and a controller for Output Vector (x(t), θ2(t))

Matlab Code:

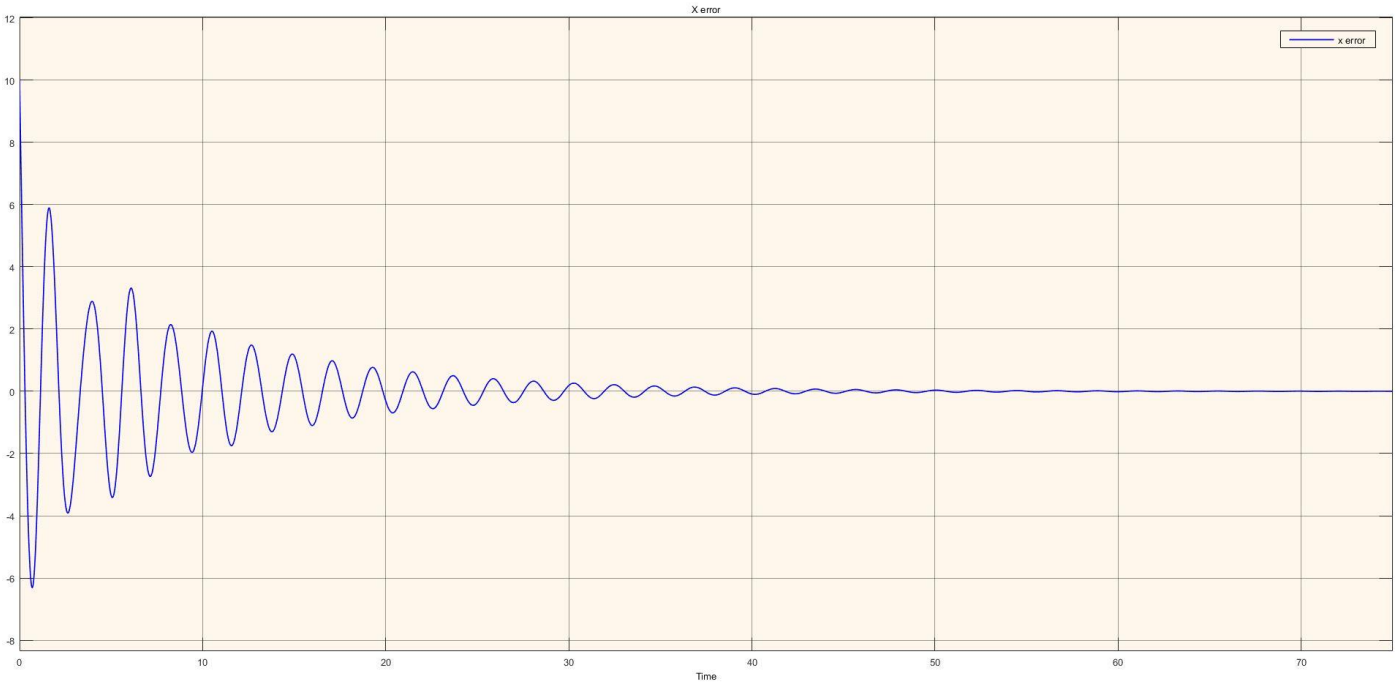
```
%% Set Values of R and Q to get K matrix using LQR()
Q=[ 5 0 0 0 0 0;
    0 60 0 0 0 0;
    0 0 50 0 0 0;
    0 0 0 70 0 0;
    0 0 0 0 150 0;
    0 0 0 0 0 100];
R=  0.0001;

[K,S,e] = lqr(A,B,Q,R)
PolesL= 4*e
```

The values obtained for the poles:

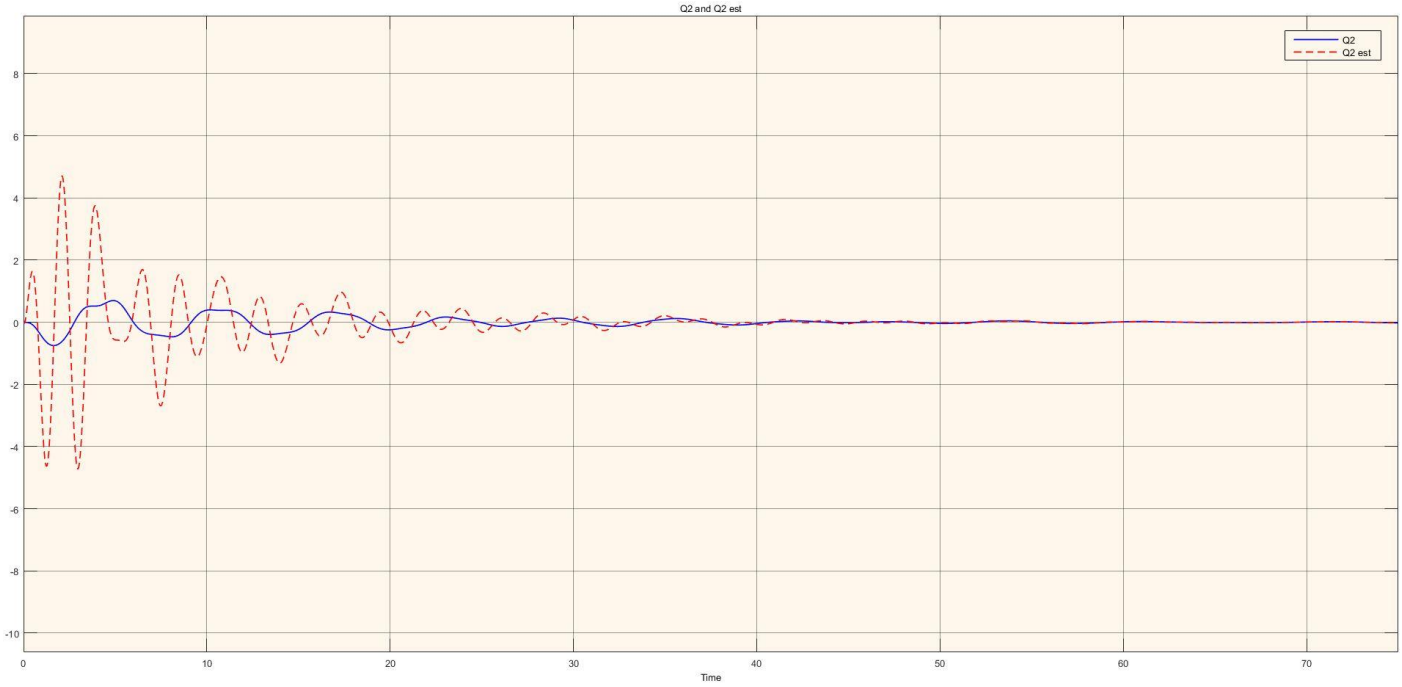
```
e =

  -0.5981 + 0.0000i
  -0.3356 + 0.0000i
  -0.0714 + 1.0205i
  -0.0714 - 1.0205i
  -0.0242 + 0.7145i
  -0.0242 - 0.7145i


PolesL =

  -2.3924 + 0.0000i
  -1.3424 + 0.0000i
  -0.2858 + 4.0820i
  -0.2858 - 4.0820i
  -0.0966 + 2.8582i
  -0.0966 - 2.8582i
```



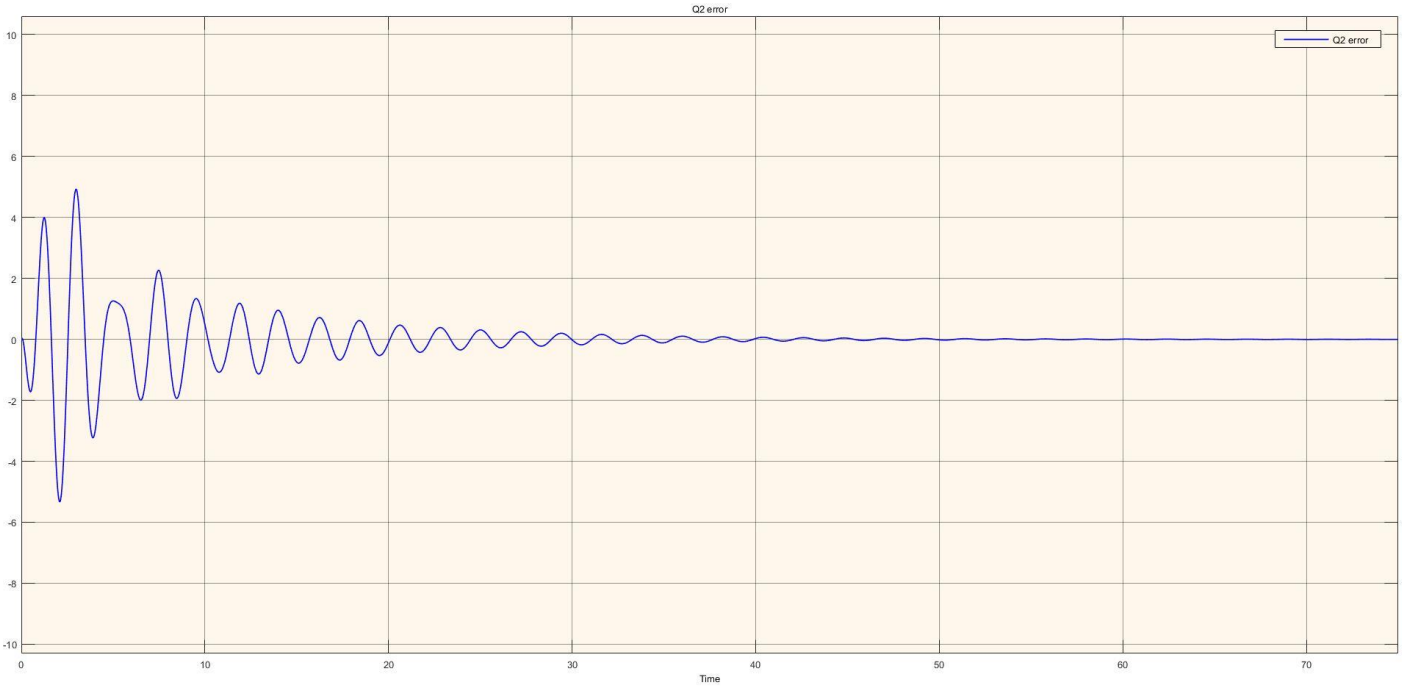
X from the system and X_{est} from the observer



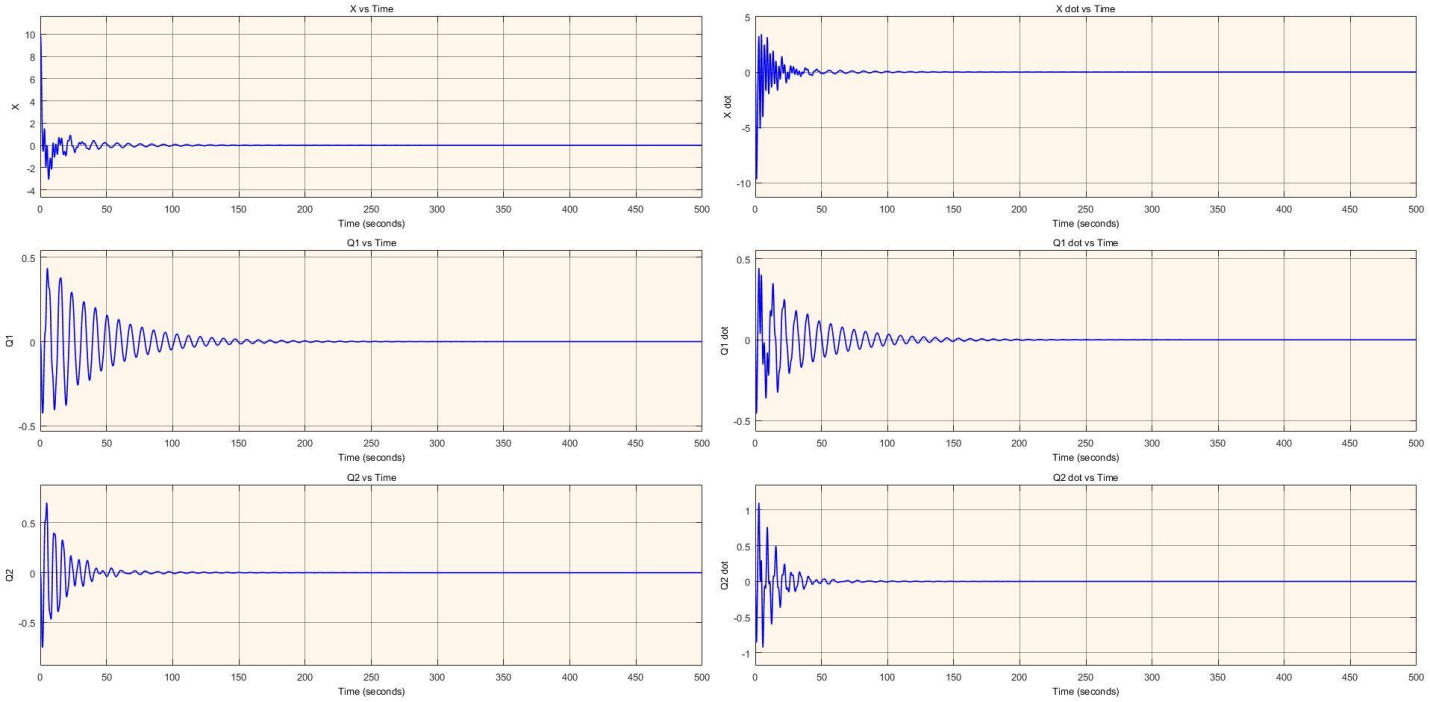
X_{error} given be (X-X_{est})



$\Theta 2$ from the system and $\theta 2_{est}$ from the observer



$\Theta 2_{error}$ given be $(\theta 2 - \theta 2_{est})$



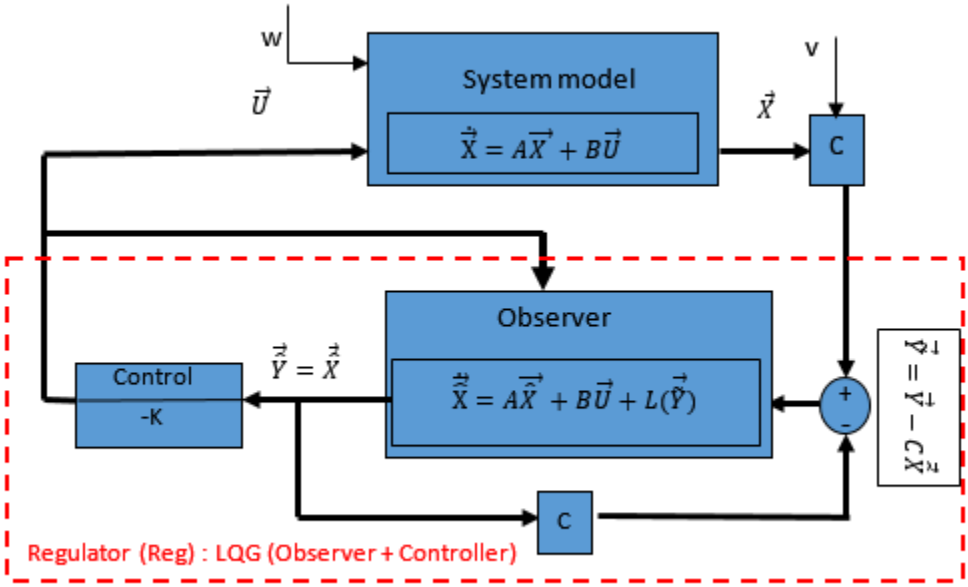
All the states of the system

As can be seen from the graphs above, the observer's estimated state x_{est} converges with that from the system's output x around 55s and the observer's estimated state $\theta 2_{est}$ converges with that from the system's output θ2 around 55s.

The whole system stabilizes around 250s with x at an offset of 0.02

Linear Quadratic Gaussian Controller

Design



The figure shows the block diagram of an output-feedback controller. The **LQG controller** is the combination of a linear-quadratic estimator (LQE) i.e. a Kalman estimator, with a linear-quadratic regulator (LQR).

Computing Matrix L

The System-State Space parameters are used to replicate the system behavior by the observer. The ***L*** parameter is used to compensate for the error between the actual system-output and the output generated by the observer (estimator). It is selected to compensate for the initial conditions (to which the observer is blind), system process noise and system measurement noise (which generates error in the states and measured output of the system).

The only difference between the Kalman estimator and pole-placement ("Robust Pole Assignment in Linear State Feedback" (1)) is in how the gain matrix (L) is derived. The driving force of any estimator is the location of the estimator poles. If the poles are well-known then Ackerman's formula should be employed to determine the gain matrix. However, in practice this is hardly ever the case. The Kalman estimator also is a "pole-placement" method, but the poles are selected through rigorous use of known statistical properties of the process and measurement noise. (2)

Computing Matrix K

Again, LQG is equivalent to combination of LQE and LQR, hence the computation of the matrix K is done according to method explained with State-Space model with full-state feedback i.e. with actual feedback of all states without estimated states from the output.

MATLAB command

We use MATLAB to get the complete state space representation of regulator with output as $\vec{U}$(t) i.e. the input to the main system.

Syntax :
reg = lqg(sys,QXU,QWV);

'reg' computes an optimal linear-quadratic-Gaussian (LQG) regulator (Reg) given a state-space System Model (sys) and weighting matrices Q_{XU} and Q_{WV}. The dynamic regulator 'reg' uses the measurements $\vec{Y}(t)$ to generate a control signal $\vec{U}$(t) that regulates $\vec{Y}(t)$ around the zero value. (3)

LQG minimizes the cost function:

$$J = E\left\{\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^{\tau} [x^T \quad u^T] * Q_{xu} * \begin{bmatrix} x \\ u \end{bmatrix} dt\right\}$$

subject to the plant equations

dx/dt = Ax + Bu + w

y = Cx + Du + v

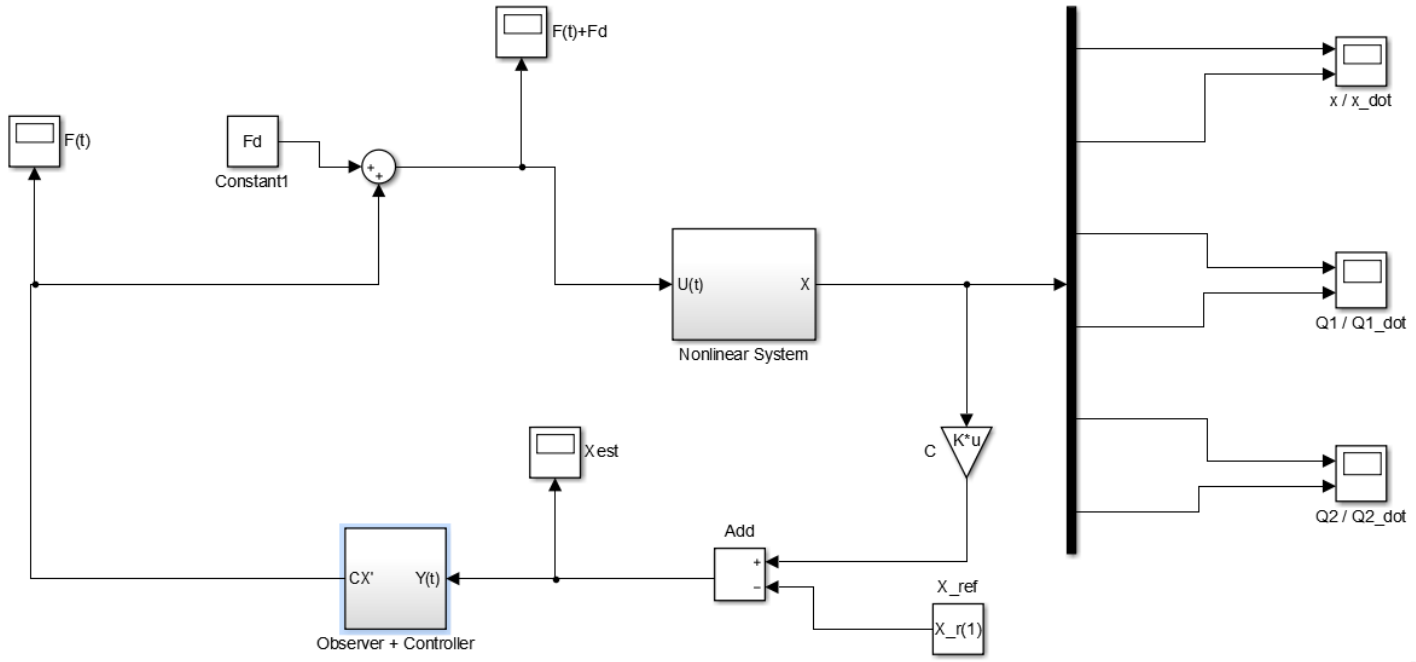
Thus comparing the above cost equation to the equation mentioned in LQR design,

$$\boldsymbol{Qxu} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix}$$

Thus $\boldsymbol{Q_{xu}}$ is the block diagonal matrix with Q and R are the weights set in LQR method to get the controller gain matrix K.

where the process noise w and measurement noise v are Gaussian white noises with covariance:
And $\boldsymbol{Q_{wv}}$ = E([w;v] * [w',v']);

Model

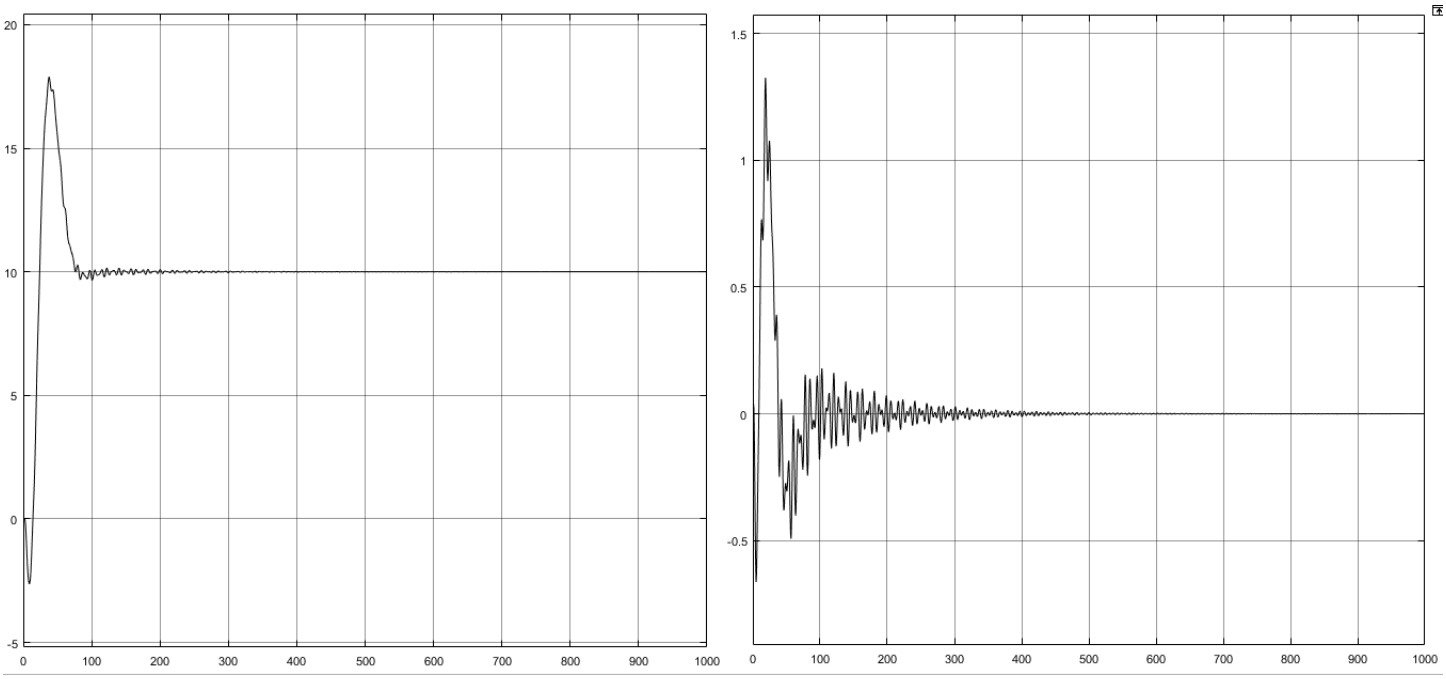


With Initial Conditions, Reference X_{ref} and No disturbance

- X(0) = [10,0,0,0,0,0] and F_d = 0

Since we know that our system is devoid of noise, we choose very small values of w and v.

```
w = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1]*0.01';
wwT = w * w';
v = 0.1;
vvT = v*v';
```



The system settles around 150 seconds at X_ref = 10 with 0 disturbance force.

- X(0) = [10,0,0,0,0,0], X_ref = 0 and F_d = 0

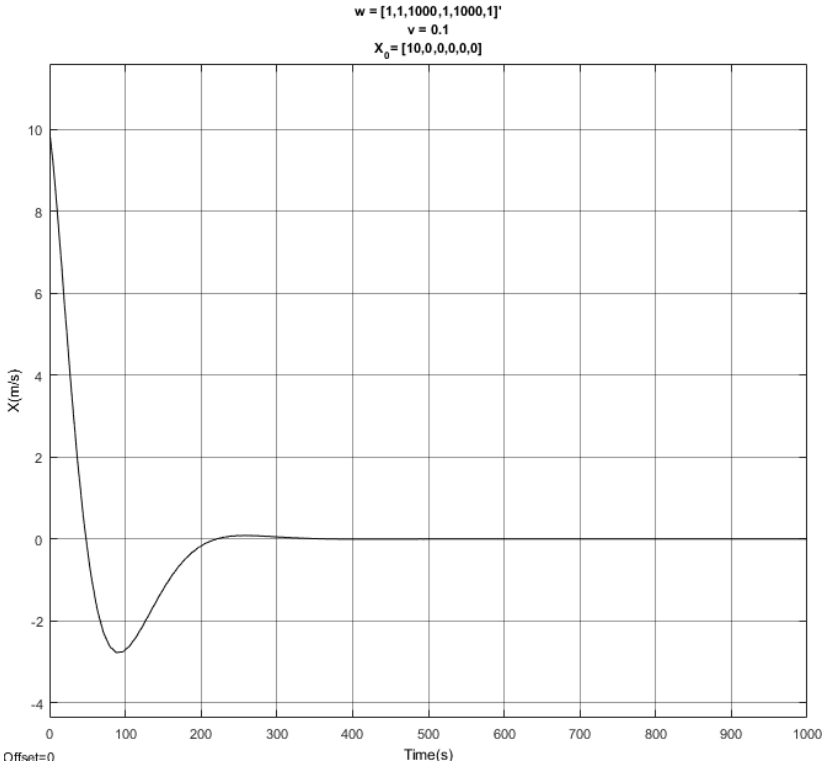
Since our system does not have any noise, we can use w and v to increase or decrease the convergence time between x_estimate and x_actual.

Thus we use,

```
w = [1, 1, 1000, 1, 1000, 1]';
wwT = diag(w'); %removing the cross-correlation terms
v = 0.1;
vvT = v;
```

With initial conditions X (0) = [10,0,0,0,0,0] i.e. x(0) = 10, $\dot{x}(0)$ =0, $\theta_1(0) = 0$, $\dot{\theta_1}(0) = 0$, $\theta_2(0) = 0$, $\dot{\theta_2}(0) = 0$
We get following results

.

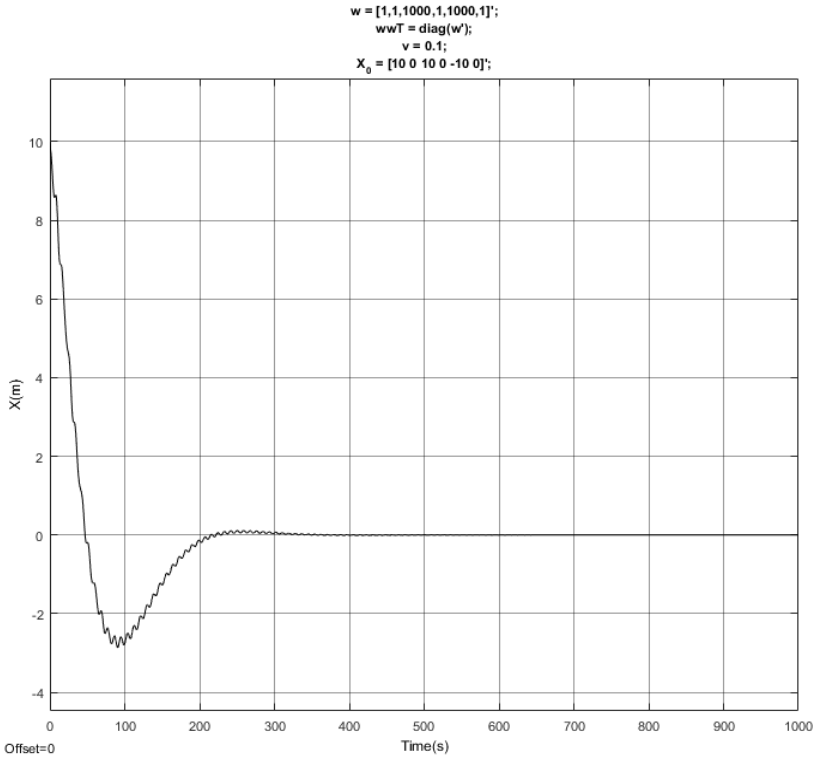


The systems appears to converge to the desired state at around 280 secs. Since there are no visible oscillations in x, all other states converge to zero with negligible oscillations, if any i.e. $\dot{x}(0)$ =0, $\theta_1(0) = 0, \dot{\theta_1}(0) = 0, \theta_2(0) = 0, \dot{\theta_2}(0) = 0$.

Now, we test the system with more rigorous initial conditions.

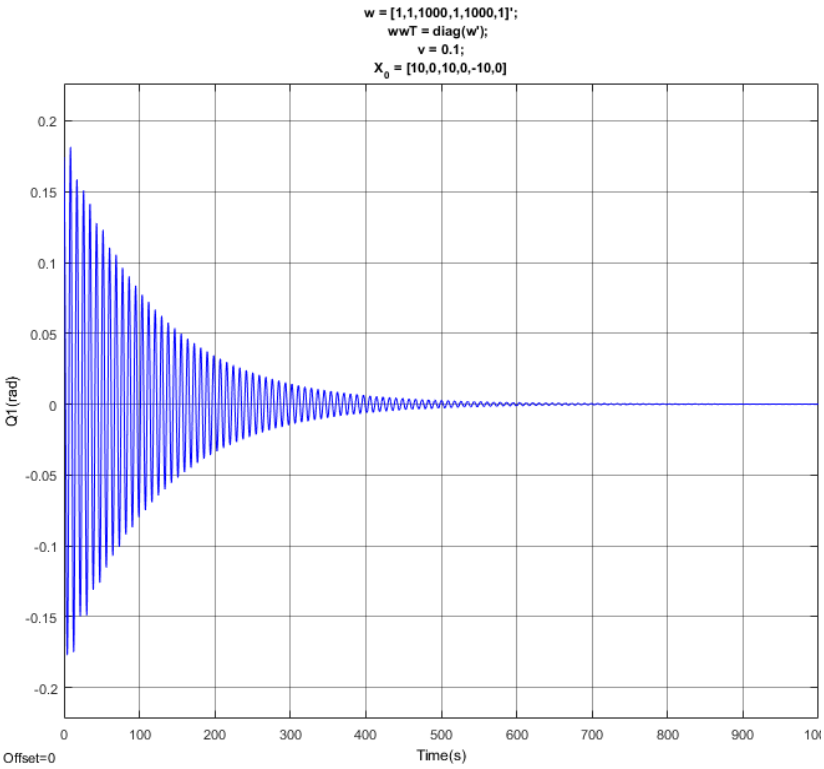
- X(0) = [10,0,10,0,-10,0] and X_{ref} = 0

The system shows more oscillations, compared to previous example, in x before reaching its reference state X (t) = 0. This is because the two masses start with some initial angles and the observer is blind to the initial conditions of the system. Hence the observer takes time to converge the estimated states to actual states and hence the control finds it difficult to give a quick response to damp out the oscillations.



.

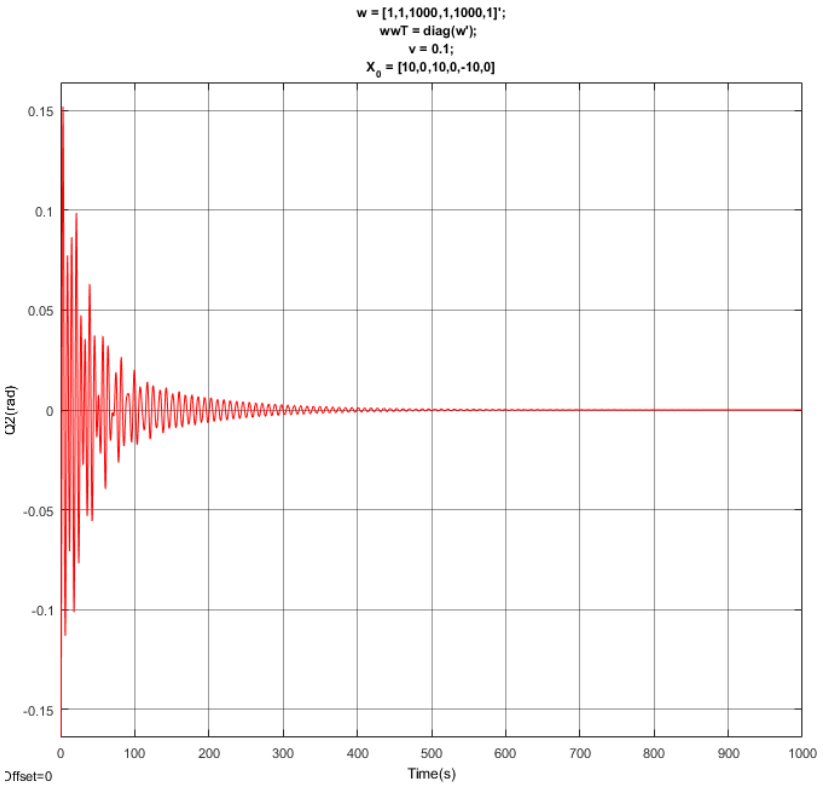
Here, x(t) starts from 10 and converges to the 0 with little more oscillations (due to oscillations in m1 and m2) at around 300s



.

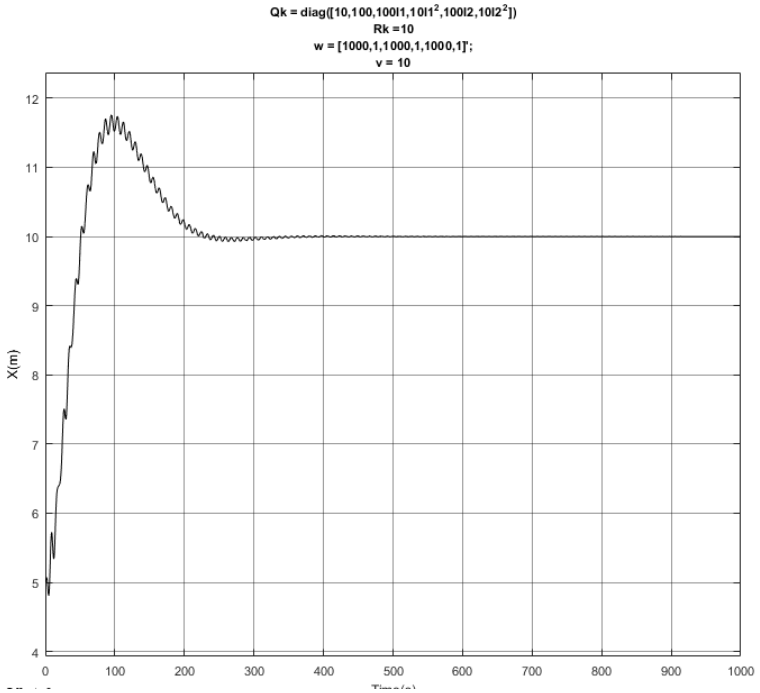
We see here, that the Q1 starts from 10 degrees = 0.174 rad and the system converges around 400s when the oscillations of order 10^{-3} are present, which later damp out to 0 at around 500s

We see here, that the Q2 starts from -10 degrees =- 0.174 rad and the system converges around 350 s when the oscillations of order 10^{-3} are present, which later damp out to 0 at around 450s



It is thus important to note that the system stabilizes with varied values of initial conditions and system reaches to 0 in desired time with completely damped out oscillations over a period of time. We now test the system to reach $X_{ref} > 0$ with initial conditions to which the observer is blind.

- X(0) = [5,0,10,0,-10,0] and X_{ref} = 10



The value of QR to compute the controller gain are :

```
Qk = [ 10    0    0    0    0    0;
            0    1000    0    0    0   0;
            0    0   100*l1   0    0    0;
            0    0    0   10*l1^2   0    0;
            0    0    0    0   100*l2   0;
              0    0    0    0    0   10*l2^2]';
Rk = 10;
```

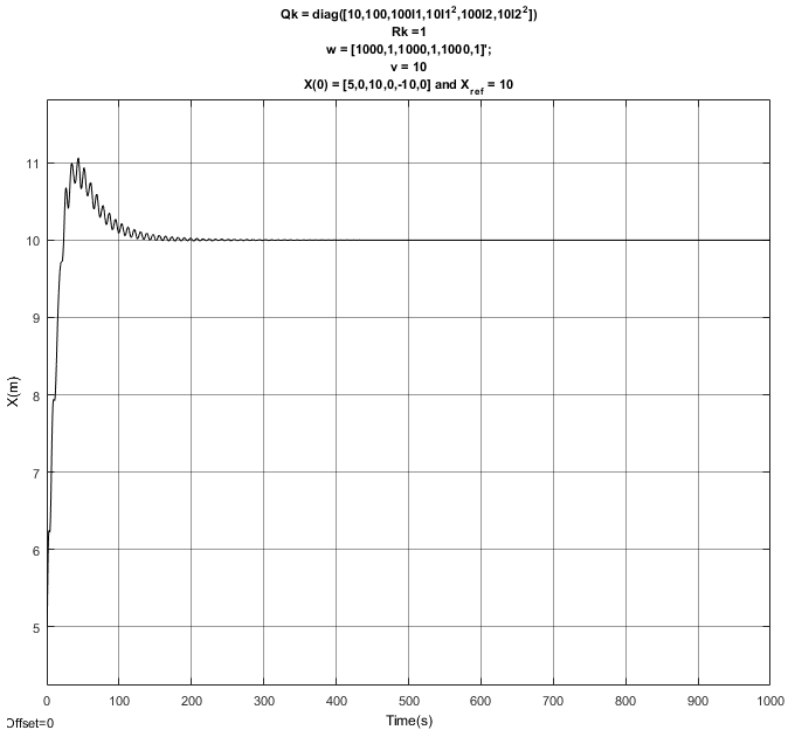
We find the response of the non-linear system as shown in figure on right side.

Thus we feel that there is need to improve the settling time of the system and hence we decrease Rk = 1;

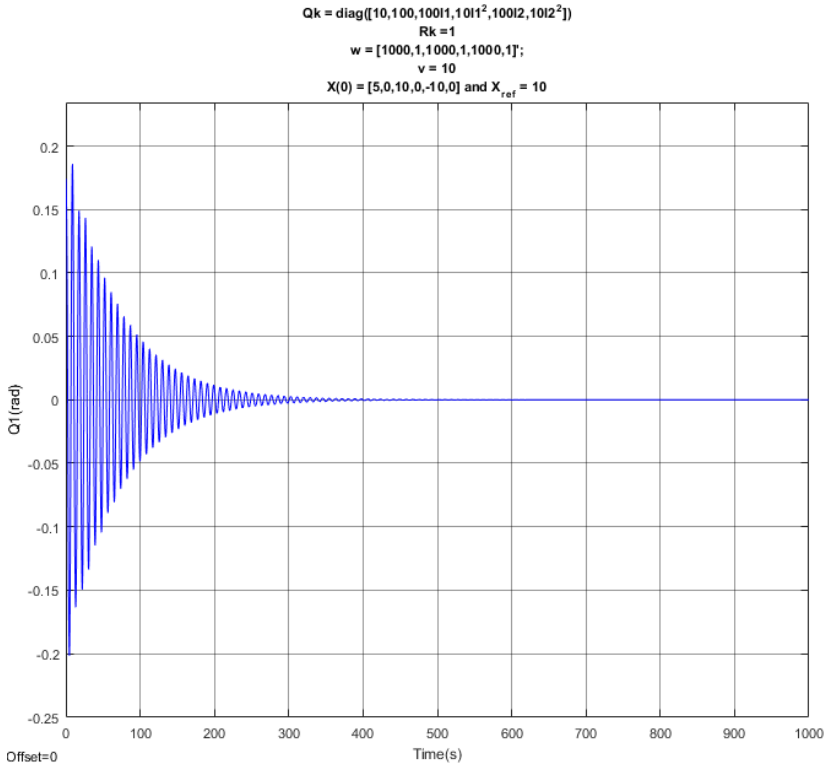
We should also compare the effect of values w and v with the first graph in this section. This response is much better since the graph traces the desired trajectory accurately. Hence we only change the value of Q and R and not of w and v.

The value of QR to compute the controller gain are :

```
Qk = [ 10    0    0    0    0    0;
            0    1000    0    0    0   0;
            0    0   100*l1   0    0    0;
            0    0    0   10*l1^2   0    0;
            0    0    0    0   100*l2   0;
              0    0    0    0    0   10*l2^2]';
Rk = 1;
```

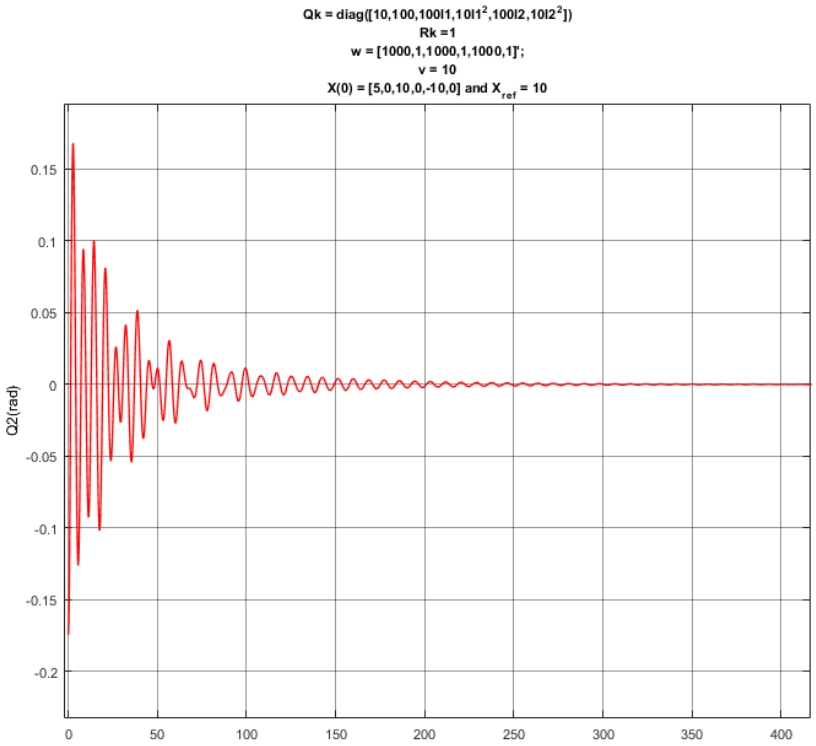


As seen from the figure on the right, the settling time is now much improved. We can also tweak QR to reduce the offset and w and v to improve estimation performance, but for now, we take this as our best result.



We see here, that the Q1 starts from 10 degrees = 0.174 rad and the system converges around 200s when the oscillations of order 10^{-3} are present, which later damp out to 0 at around 350s

We see here, that the Q1 starts from 10 degrees = 0.174 rad and the system converges around 100s when the oscillations of order 10^{-3} are present, which later damp out to 0 at around 250s
(Note* Time axis in above to graphs is changed, the second is zoomed till 400s to view it properly)



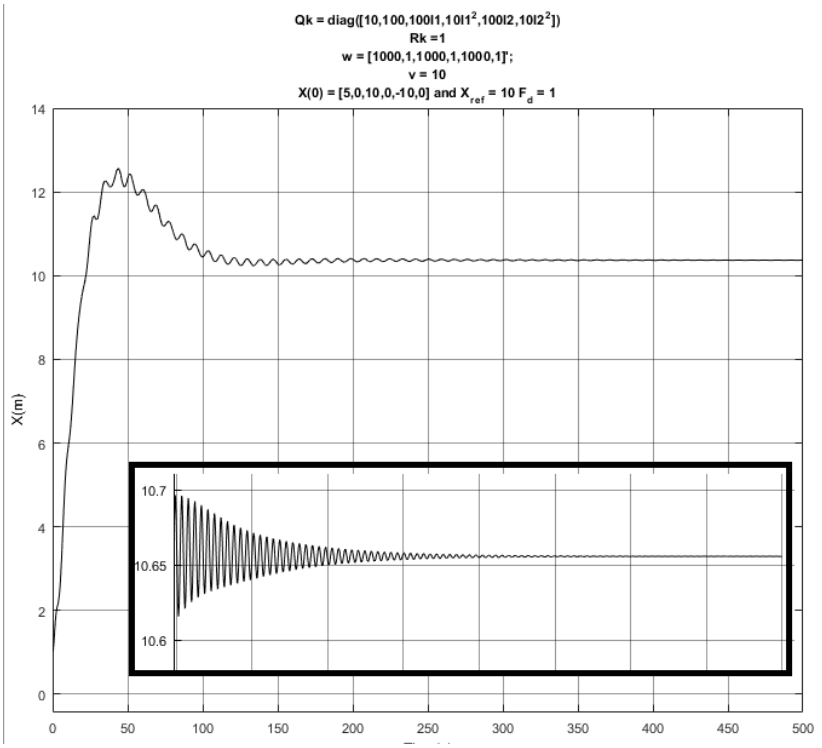
- With Reference x(*t*) and F_d disturbance

For the tuned value of LQG and LQR, we find that a disturbance of 1N produces and offset in the steady state value of the Cart. From the graph to the right, it is evident that the system settles at 10.65, after approximately 200s

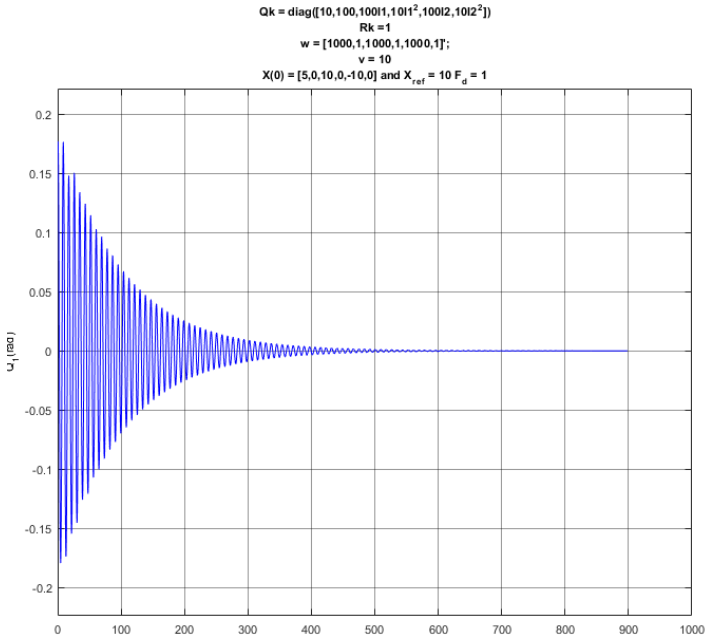
It is evident that as the x(t) settles, θ_1 and θ_2 also settles to the equilibrium point.

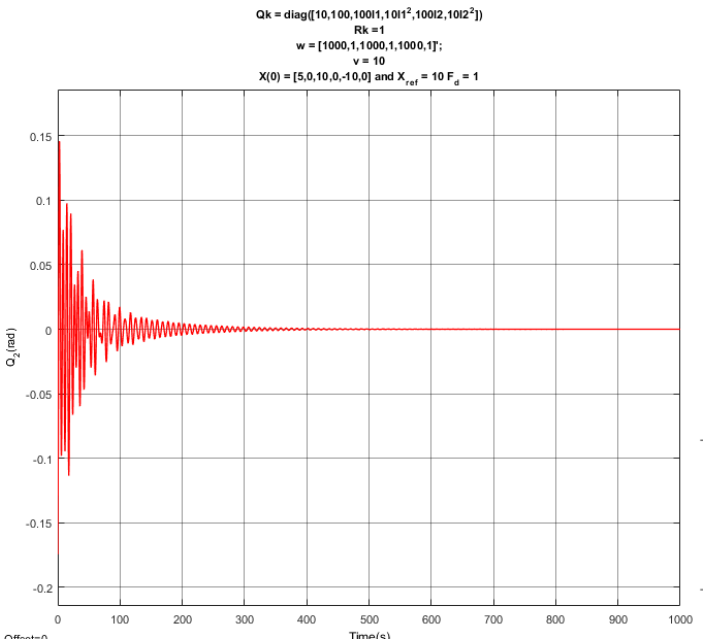
Thus, there is still scope of improvement in values of Q and R. The steady state error of a PD controller is inversely proportional to K_p.

Thus, if we increase the 1st element of controller gain matrix from LQR, we would achieve a better settling time and also reduce the steady state error. However, it might increase the swings of θ_1 and θ_2



For the given LQR, following response for θ_1 and θ_2 is found. The system settles at around 150 with oscillations in θ_1 and θ_2 are of the order 10^{-3}





We, than changed the LQR gain to get the desired response. We increased the weight corresponding to state x(t) and also reduced the R, so that we have large force to compensate for the overall error thereby, increasing K_{px} and over-all gain matrix K. As explained earlier, this may result in larger oscillations of θ_1 and θ_2, hence we increase the weight on $K_{p\theta 1}$ and $K_{p\theta 2}$

New Q,R

```
Qk = [ 1000     0    0         0       0      0;
          0    10    0         0       0      0;
          0     0  100*l1      0       0      0;
          0     0    0      10*l1^2    0      0;
          0     0    0         0     100*l2   0;
          0     0    0         0       0    10*l2^2])';
```

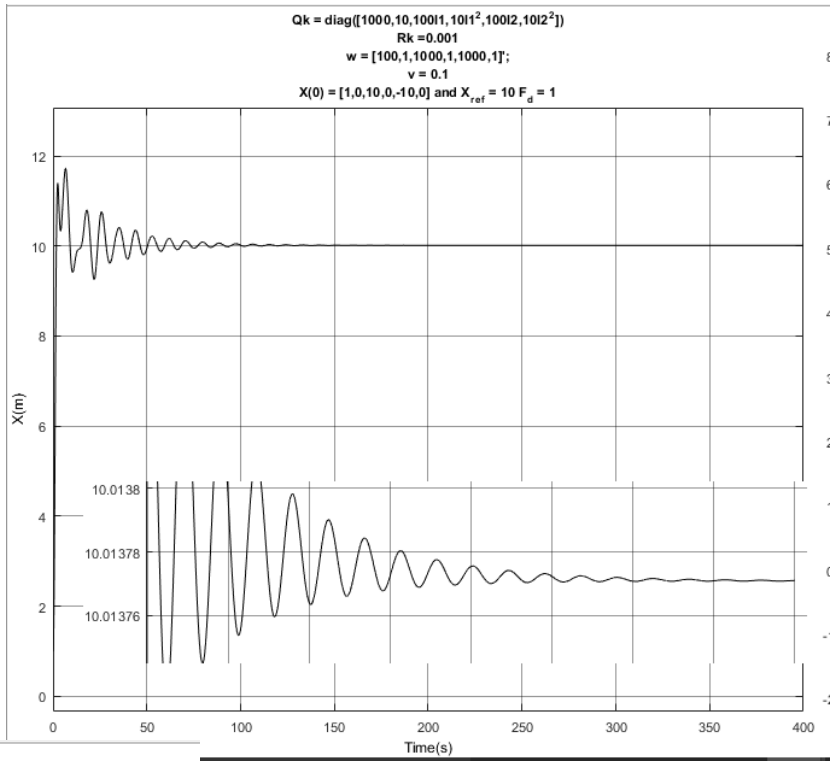
```
Rk = 0.001;
```

New w,v

```
w =  [100 1 1000 1 1000 1]';
```

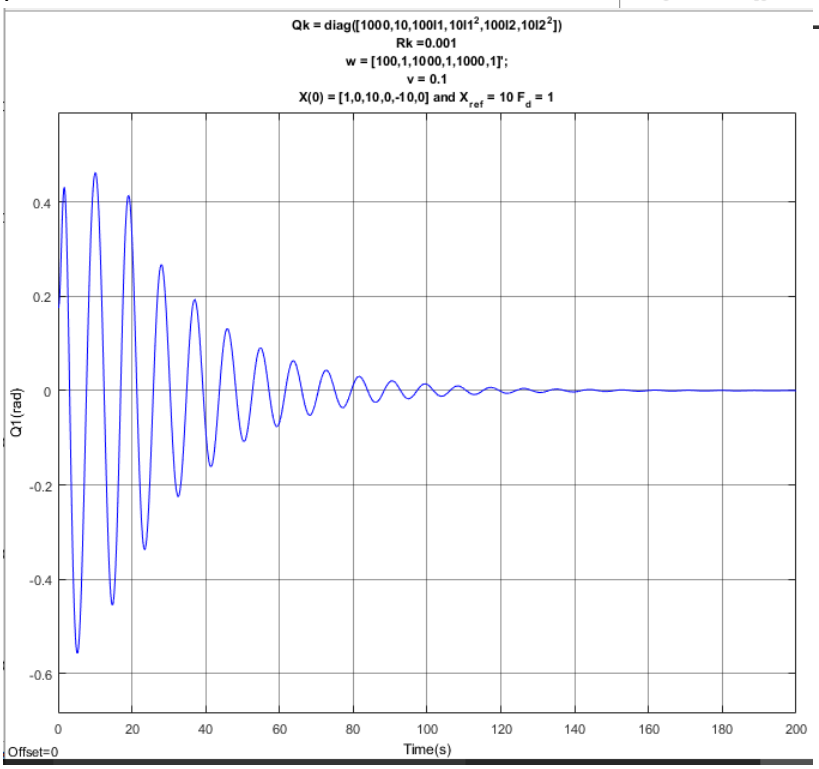
```
v = 0.1;
```

.



The figure on the right shows the output of the system with suggested values of QR and wv. The rise time is drastically reduced as compared to previous trial, overshoot has also come dome by some margin. The x(t) steady-state error for the system is around 10.01376

.

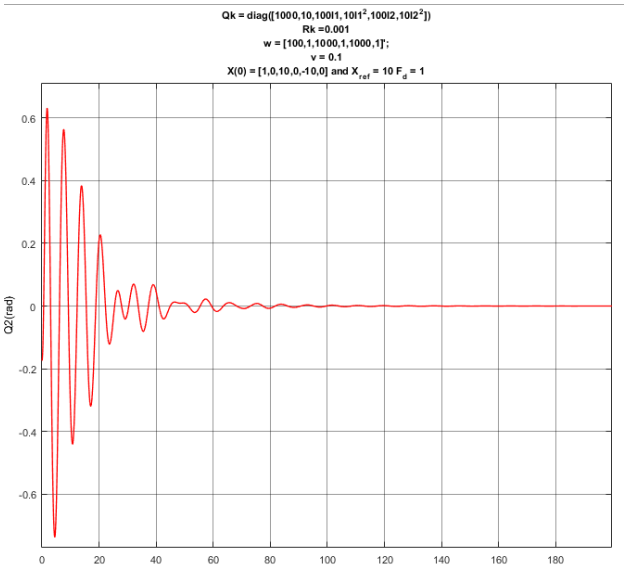


We observe that since the controller gain matrix has high values, the input makes the system shoot to reach X_{ref} as soon as possible. This leads to increase in θ_1 and θ_2 initially. This is the trade-off that we need to do to get quick rise and settling time and also to minimize steady state error.

Since the maximum value of θ_1 is little over 0.4 i.e. θ_1 is always less than 25 degrees.
This seems to be an acceptable slag for our system.

.

Again, the maximum value of θ_2 is little over 0.6 i.e. θ_2 is always less than 35 degrees.
This seems to be an acceptable slag for our system given the initial conditions of the system with unit step continuous force acting.



Since this system computes gain matrix G = [k_1, k_2, k_3, k_4, k_5, k_6] such that the input to the system $\vec{U}(t)$ is given by following:

$$\vec{U}(t) = k_1(x(t) - x_{ref}) + k_2(\dot{x}(t) - \dot{x}_{ref}) + k_3(\theta_1(t) - \theta_{1ref}) + k_4(\dot{\theta}_1(t) - \dot{\theta}_{1ref}) + k_5(\theta_2(t) - \theta_{2ref}) + k_6(\dot{\theta}_2 - \dot{\theta}_{2ref})$$

The equation shows that the LQR for this system gives a PD type controller where the control action is dependent only on the variables and their derivatives only. Thus we need to add a integral to the state such that the control action then behaves like a PID controller and tracks the error between the variable and reference value to 0

Thus on adding additional state and considering all reference variables to be 0, State Space Model of the equivalent system becomes:

$$\dot{\overrightarrow{X(t)}} = A\overrightarrow{X(t)} + B\overrightarrow{U(t)}$$

$$Y(t) = C\overrightarrow{X(t)} + D\overrightarrow{U(t)}$$

$$\dot{\overrightarrow{X(t)}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -m1g/M & 0 & -m2g/M & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -(M+m1)g/Ml1 & 0 & -m2g/Ml1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -m1g/Ml2 & 0 & -(M+m2)g/Ml2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \\ \theta_1(t) \\ \dot{\theta}_1(t) \\ \theta_2(t) \\ \dot{\theta}_2(t) \\ Ei \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ 1/Ml1 \\ 0 \\ 1/Ml2 \\ 0 \end{bmatrix} [F(t)]$$

And suppose, we use this state as output i.e. input to observer.

$$\text{Than, } \vec{Y}(t) = C\overrightarrow{X}(t) + D\vec{U}(t)$$

Therefore the C matrix becomes:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Using A and B Matrix, one can check for controllability and using A and C Matrix we can test the observability of the system with the integral state.

- Test for Stability of closed loop system with addition of Integral state to the system

Using %Matlab Code:

```
clear;
clc;
%% Initialize Parameters
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.8;

%% Use A and Vmatrix from Q1(B)
A =[ 0, 1,                0,      0,                0, 0, 0;
     0, 0,        -(m1*g)/M,      0,       -(m2*g)/M, 0, 0;
     0, 0,                0,      1,                0, 0, 0;
     0, 0,  -(M+m1)*g/M/l1,      0,    -(m2*g)/M/l1, 0, 0;
     0, 0,                0,      0,                0, 1, 0;
     0, 0,    -(m1*g)/M/l2,      0, -(M+m2)*g/M/l2, 0, 0;
     1, 0,                0,      0,                0  0, 0];
B =[  0;
      1/M;
        0;
 1/(M*l1);
        0;
 1/(M*l2);
        0];

C_all = eye(7);

Cx_xi = [1 0 0 0 0 0 0;
         0 0 0 0 0 0 1];

D = zeros(7,1);

%% Set Values of R and Q to get K matrix using LQR()
R = 0.0001 ;
Q = [ 10   0   0   0   0   0 0;
      0   100   0   0   0  0 0;
      0   0  1000*l1  0   0  0 0;
      0   0   0  100*l1^2  0   0 0;
      0   0   0   0  1000*l2  0 0;
      0   0   0   0   0  100*l2^2 0;
      0   0   0   0   0    0    1];
[K,S,e] = lqr(A,B,Q,R)
pause;
%% Controlability Matrix
disp(['Controlability of State-Space model after addition of integral state'])
C_M = [B A*B A^2*B A^3*B A^4*B A^5*B A^6*B];
rank(C_M)
pause;
%% Observability Matrix
disp('Observability of State-Space model after addition of integral state:: [x(t)
Inegral State]')
```

```
O_M = [Cx_xi; Cx_xi*A; Cx_xi*A^2; Cx_xi*A^3; Cx_xi*A^4; Cx_xi*A^5]
disp(['rank = ' num2str(rank(O_M))])
pause;
```

```
%OUTPUT
Controllability of State-Space model after addition of integral state

C_M =

   1.0e-03 *

         0    1.0000         0   -0.1470         0    0.1417         0
    1.0000         0   -0.1470         0    0.1417         0   -0.1444
         0    0.0500         0   -0.0319         0    0.0227         0
    0.0500         0   -0.0319         0    0.0227         0   -0.0183
         0    0.1000         0   -0.1127         0    0.1246         0
    0.1000         0   -0.1127         0    0.1246         0   -0.1366
         0         0    1.0000         0   -0.1470         0    0.1417


rank(C_M) = 7

Observability of State-Space model after addition of integral state:: [x(t) Inegral
State]

O_M =

    1.0000         0         0         0         0         0         0
         0         0         0         0         0         0    1.0000
         0    1.0000         0         0         0         0         0
    1.0000         0         0         0         0         0         0
         0         0   -0.9800         0   -0.9800         0         0
         0    1.0000         0         0         0         0         0
         0         0         0   -0.9800         0   -0.9800         0
         0         0   -0.9800         0   -0.9800         0         0
         0         0    0.6243         0    1.1045         0         0
         0         0         0   -0.9800         0   -0.9800         0
         0         0         0    0.6243         0    1.1045         0
         0         0    0.6243         0    1.1045         0         0

rank(O_M) = 7
```

Hence, the system is controllable and observable with additional integrator term.

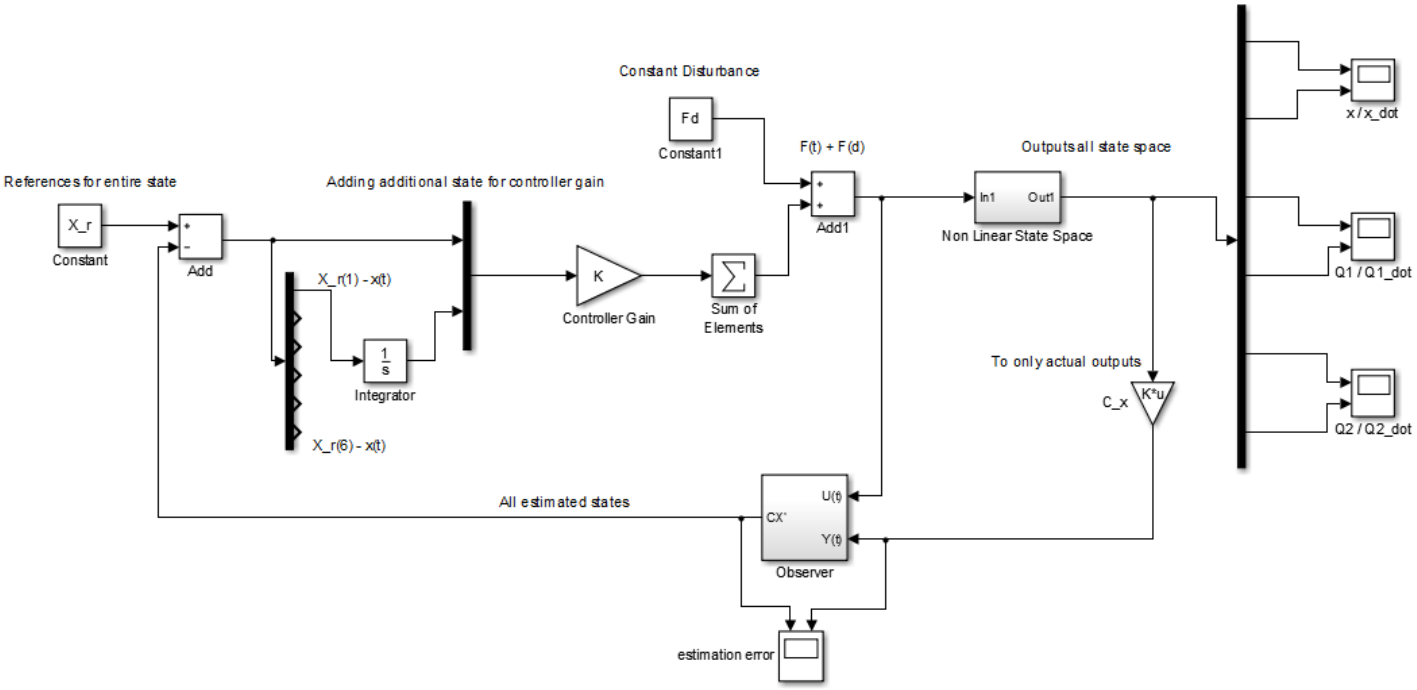
In physical realization of this system, since we only have measurement of x(t). Hence we can directly use this measurement to integrate the error between $x(t)$ and x_{ref} i.e. $Err = \int_0^\tau \left(x(t) - x_{ref}\right) \partial\tau$

Thus we only need to change the controller gain Matrix (K), such that the system defined in it has Err as a state.
Thus, modified open-loop matrix of the given system with additional integral state is :

```
A =[ 0, 1,                0,      0,               0, 0, 0;
     0, 0,       -(m1*g)/M,      0,       -(m2*g)/M, 0, 0;
     0, 0,                0,      1,               0, 0, 0;
     0, 0,  -(M+m1)*g/M/l1,      0,    -(m2*g)/M/l1, 0, 0;
     0, 0,                0,      0,               0, 1, 0;
     0, 0,    -(m1*g)/M/l2,      0, -(M+m2)*g/M/l2, 0, 0;
     1, 0,                0,      0,               0  0, 0];
```

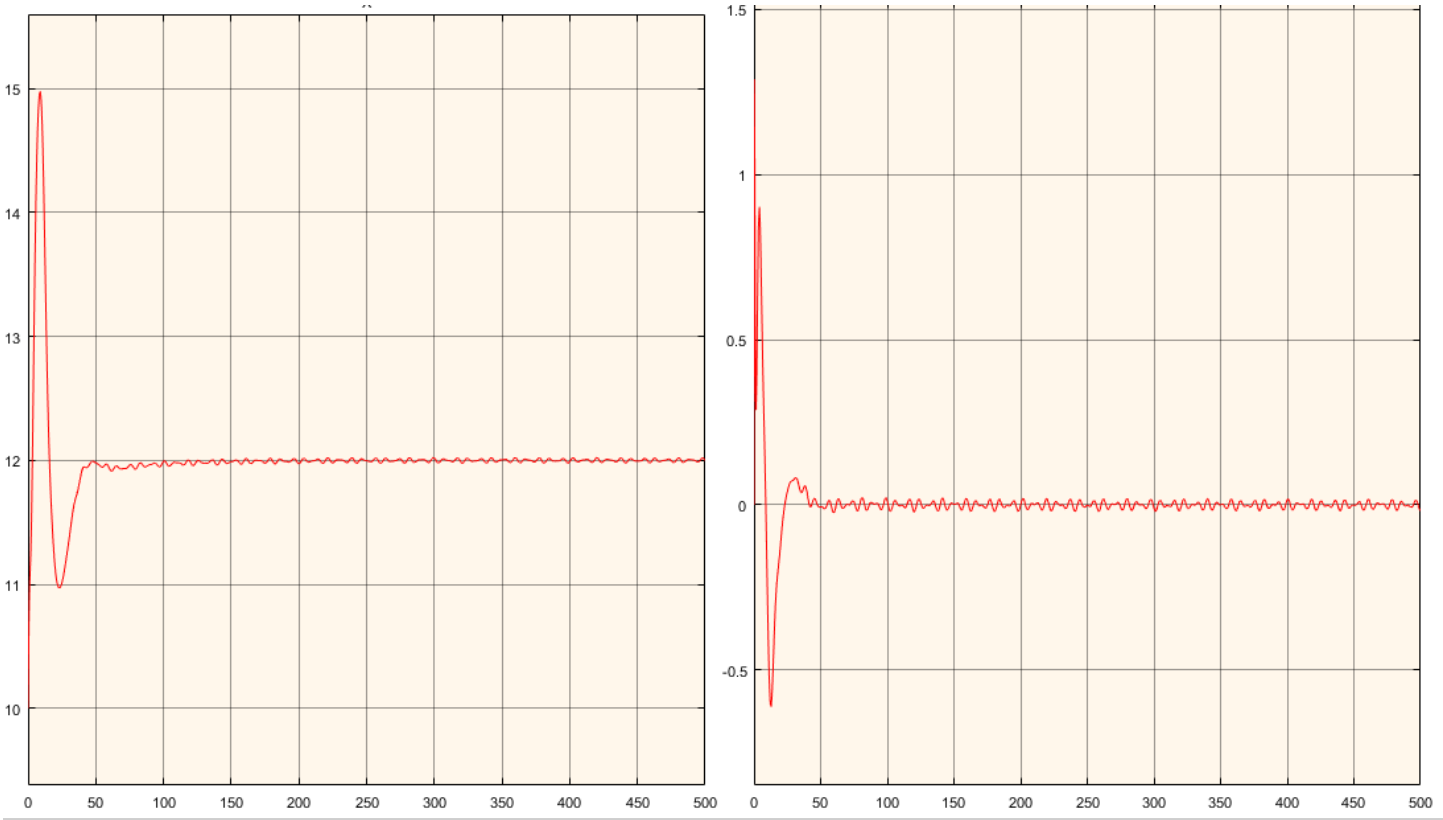
```
B =[   0;
       1/M;
         0;
 1/(M*l1);
         0;
 1/(M*l2);
         0];
```

** We preferred to use this element (integral state) only in the controller gain matrix for the non-linear system instead of adding it into the observer state-space too, because the observer will not contribute anything in estimating the integral term, since we know the x(t). The x(t) term along with it's integral will always be known.

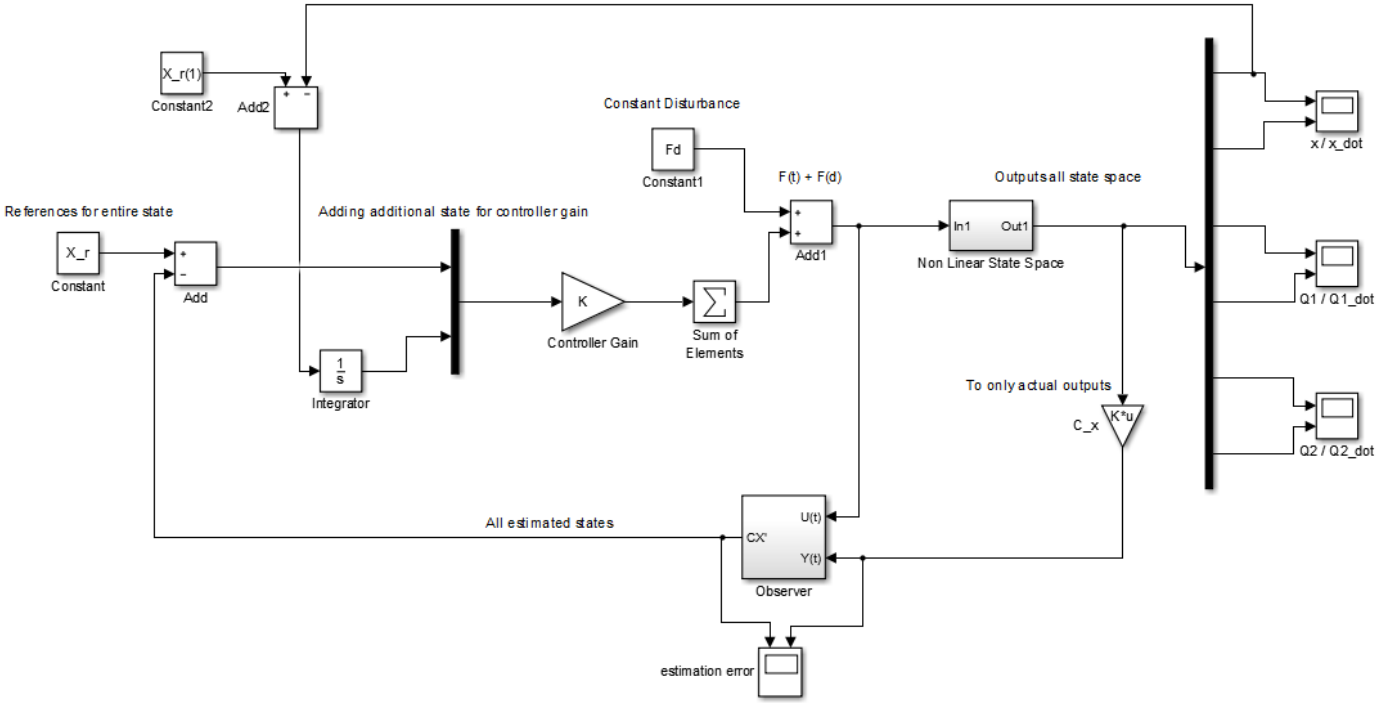


As we see, the observer will give us an estimate of the known state x(t). Hence ideally, has to be accurate i.e. without error. Thus we demux and take the estimated x, get the difference (x_ref - x(t)) and integrate it – after the observer stage. This calculated element is then feed to multiplexer such that the controller gain (K) perceives this as an additional state and computes the input to the system F(t) accordingly.

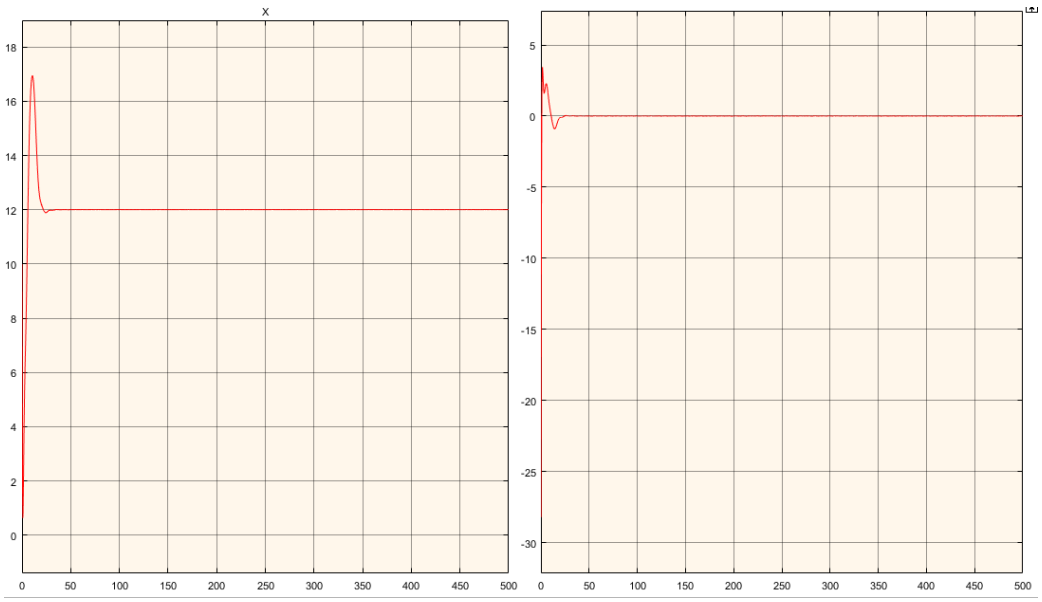
Since we were getting very low steady state error for unit step, we used Fd = 100 and with X_0 = [10,0,0,0,0,0].
We find that the x(t) converges to the given reference and oscillates around it with error of order 10^-3 meters.



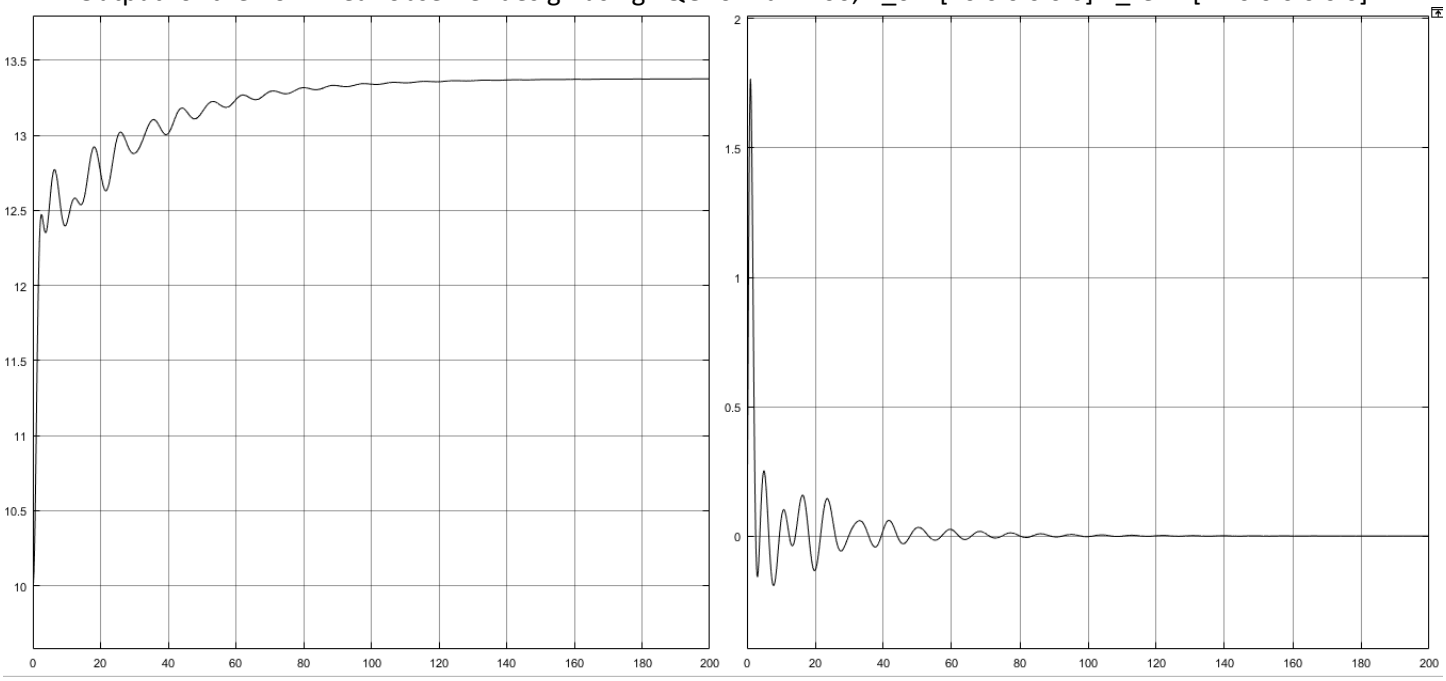
Now, we use the measure x(t) instead of estimated x(t)



We find that the oscillations are considerably reduced. This is because, now we have the initial condition of the x(t) and the estimation error for x(t) is removed completely. Only error that may occur is due to the presence of noise, but integrator takes care of the it as integral of uniformly distributed noise is zero over period of time.



Output for the non-linear observer design using LQG for Fd = 100; X_0 = [10 0 0 0 0 0] X_ref = [12 0 0 0 0 0 0]



As seen, the system settles at around 13.3 where as the integral system oscillates around 12 with oscillations of order 10^{-3}

** LQG method does not return L, or states or control gain K, hence we could not use our “addition of integral” on the optimized response

%MATLAB CODE

```
clear;
clc;
%% Initialize Parameters
M = 1000;
m1 = 100;
m2 = 100;
l1 = 20;
l2 = 10;
g = 9.8;

%% Use A and Vmatrix from Q1(B)
A =[ 0, 1,               0,     0,               0, 0;
     0, 0,       -(m1*g)/M,     0,       -(m2*g)/M, 0;
     0, 0,               0,     1,               0, 0;
     0, 0,  -(M+m1)*g/M/l1,     0,   -(m2*g)/M/l1, 0;
     0, 0,               0,     0,               0, 1;
     0, 0,    -(m1*g)/M/l2,     0, -(M+m2)*g/M/l2, 0];
B =[  0;
      1/M;
        0;
 1/(M*l1);
        0;
 1/(M*l2)];

C_all = eye(6);

CxQ1Q2 = [1 0 0 0 0 0;
          0 0 1 0 0 0;
          0 0 0 0 1 0];

D = zeros(6,1);

%% Set Values of R and Q to get K matrix using LQR()
Af =[ 0, 1,               0,     0,               0, 0, 0;
     0, 0,       -(m1*g)/M,     0,       -(m2*g)/M, 0, 0;
     0, 0,               0,     1,               0, 0, 0;
     0, 0,  -(M+m1)*g/M/l1,     0,   -(m2*g)/M/l1, 0, 0;
     0, 0,               0,     0,               0, 1, 0;
     0, 0,    -(m1*g)/M/l2,     0, -(M+m2)*g/M/l2, 0, 0;
     1, 0,               0,     0,               0  0, 0];
Bf =[  0;
      1/M;
        0;
 1/(M*l1);
        0;
 1/(M*l2);
        0];

Q=[ 10 0 0 0 0 0 0;
    0 600 0 0 0 0 0;
    0 0 50 0 0 0 0;
    0 0 0 700 0 0 0;
    0 0 0 0 50 0 0;
    0 0 0 0 0 150 0;
    0 0 0 0 0 0 10];
R=  0.004;
[K,S,e] = lqr(Af,Bf,Q,R);
```

```
%% Set Initial Condition X_0 in degrees --  X_0_1 is to convert X_0 in radians
X_0 = [10 0 0 0 0 0]';
X_0_1 = [X_0(1) 0 X_0(3)*pi/180 X_0(4)*pi/180 X_0(5)*pi/180 X_0(6)*pi/180]';

%%  Set Reference :  where the system should reach
X_r = [12 0 0 0 0 0]';
Fd = 10;
%% Observer StateSpace for States (x(t))
Cx = [1 0 0 0 0 0];
Lx = [ 1.5300;
   13.4278;
 -106.2389;
   10.9072;
   87.8797;
  -42.1509]; %% From pole placement method used in report
Ax_obs = [A-Lx*Cx];
Bx_obs = [B Lx];
Cx_obs = Cx;
Dx_obs = [0 0; 0 0; 0 0; 0 0; 0 0; 0 0];
X_0_x_obs = [0 0 0 0 0 0];
```