ENPM 667 Final Project

Controller Design for Double Pendulum Crane System

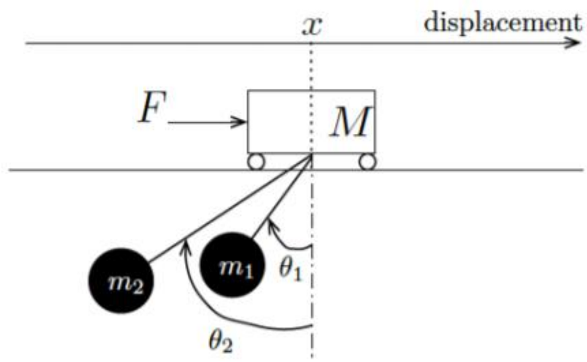
Pooja Kabra

117032444

December 2020

Table of Contents

System Model



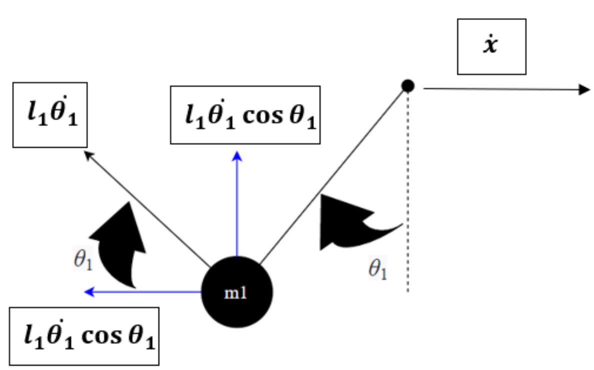
Kinetic Energy for Mass M

$$T_C = \frac{1}{2} M \dot{x}^2 \tag{0.1}$$

Kinetic Energy for Mass m_1

Angular velocity of m_1 is given by $l_1\dot{\theta}_1$ in the tangential direction.

$\dot{x}$ is the forward velocity of support.



$$\boldsymbol{v_1} = \left(\dot{x} - l_1\dot{\theta}_1 \cos\theta_1\right)\hat{\imath} + \left(l_1\dot{\theta}_1 \sin\theta_1\right)\hat{\jmath}$$

0.2

$$\boldsymbol{v_1}^2 = \left(\dot{x} - l_1\dot{\theta}_1 \cos\theta_1\right)^2 + \left(l_1\dot{\theta}_1 \sin\theta_1\right)^2$$

$$= \dot{x}^2 + l_1^{\,2}\,\dot{\theta}_1^{\,2} cos^2\theta_1 - 2\dot{x}\,l_1\dot{\theta}_1 \cos\theta_1 + l_1^{\,2}\,\dot{\theta}_1^{\,2} sin^2\theta_1$$

$$= \dot{x}^2 + l_1^{\,2}\,\dot{\theta}_1^{\,2} - 2\dot{x}\,l_1\dot{\theta}_1 \cos\theta_1$$

$$T_1 = \frac{1}{2} m_1 v_1^{\,2}$$

$$T_1 = \frac{1}{2} m_1\,\dot{x}^2 + \frac{1}{2} m_1 l_1^{\,2}\,\dot{\theta}_1^{\,2} - m_1\dot{x}\,l_1\dot{\theta}_1 \cos\theta_1$$

0.3

Kinetic Energy for Mass m_2

On the same lines as m_1, we obtain for m_2

$$T_2 = \frac{1}{2} m_2\,\dot{x}^2 + \frac{1}{2} m_2 l_2^{\,2}\,\dot{\theta}_2^{\,2} - m_2\dot{x}\,l_2\dot{\theta}_2 \cos\theta_2$$

0.4

Total Kinetic Energy of the System

$$T = T_C + T_1 + T_2$$

$$T = \frac{1}{2}(M + m_1 + m_2)\dot{x}^2 + \frac{1}{2}\,m_1 l_1^{\,2}\,\dot{\theta}_1^{\,2} + \frac{1}{2} m_2 l_2^{\,2}\,\dot{\theta}_2^{\,2} - \dot{x}(m_1\,l_1\dot{\theta}_1 \cos\theta_1 + m_2\,l_2\dot{\theta}_2 \cos\theta_2)$$

0.5

Potential Energy for Mass M

$$U_C = 0$$

0.6

Potential Energy for Mass m_1

We assume the cart height as reference for calculating potential energy.

$$U_1 = -m_1 g\ l_1 \cos\theta_1$$

Potential Energy for Mass m_2

$$U_2 = -m_2 g\ l_2 \cos\theta_2$$

0.7

Total Potential Energy of the System

$$U = U_C + U_1 + U_2 = -m_1 g\ l_1 \cos\theta_1 - m_2 g\ l_2 \cos\theta_2$$

0.8

Lagrangian Function

$$L = T - U$$

$$L = \frac{1}{2}\bar{M}\dot{x}^2 + \frac{1}{2} m_1 l_1{}^2\ \dot{\theta_1}{}^2 + \frac{1}{2} m_2 l_2{}^2\ \dot{\theta_2}{}^2 - \dot{x}(m_1\ l_1 \dot{\theta}_1 \cos\theta_1 + m_2\ l_2 \dot{\theta}_2 \cos\theta_2) - m_1 g\ l_1 \cos\theta_1 - m_2 g\ l_2 \cos\theta_2$$

$$where\ \bar{M} = M + m_1 + m_2$$

0.9

The generalized coordinates in our given system are x, θ_1, θ_2

The Euler - Lagrangian equations are given as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F$$

0.10

$$\frac{\partial L}{\partial \dot{x}} = \bar{M}\dot{x} - m_1\ l_1 \dot{\theta}_1 \cos\theta_1 - m_2\ l_2 \dot{\theta}_2 \cos\theta_2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = \bar{M}\ddot{x} - m_1\ l_1\ \ddot{\theta_1}\ \cos\theta_1 + m_1\ l_1 \dot{\theta_1}{}^2 \sin\theta_1 - m_2\ l_2 \ddot{\theta_2}\ \cos\theta_2 + m_2\ l_2 \dot{\theta_2}{}^2 \sin\theta_2$$

$$\frac{\partial L}{\partial x} = 0$$

$$\bar{M}\ddot{x} - m_1\ l_1\ \ddot{\theta_1}\ \cos\theta_1 + m_1\ l_1 \dot{\theta_1}{}^2 \sin\theta_1 - m_2\ l_2 \ddot{\theta_2}\ \cos\theta_2 + m_2\ l_2 \dot{\theta_2}{}^2 \sin\theta_2 = F$$

0.11

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) - \frac{\partial L}{\partial \theta_1} = 0$$

0.12

$$\frac{\partial L}{\partial \dot{\theta_1}} = m_1\, l_1{}^2\dot{\theta_1} - \dot{x}m_1 l_1 \cos\theta_1$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_1}}\right) = m_1\, l_1{}^2\ddot{\theta_1} - \ddot{x}m_1 l_1\dot{\theta_1}\cos\theta_1 + \dot{x}m_1 l_1\dot{\theta_1}\sin\theta_1$$

$$\frac{\partial L}{\partial \theta_1} = \dot{x}m_1 l_1\dot{\theta_1}\sin\theta_1 - m_1 g\, l_1 \sin\theta_1$$

$$m_1\, l_1{}^2\ddot{\theta_1} - \ddot{x}m_1 l_1\dot{\theta_1}\cos\theta_1 + m_1 g\, l_1 \sin\theta_1 = 0$$

0.13

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta_2}}\right) - \frac{\partial L}{\partial \theta_2} = 0$$

0.14

$$m_2\, l_2{}^2\ddot{\theta_2} - \ddot{x}m_2 l_2\dot{\theta_2}\cos\theta_2 + m_2 g\, l_2 \sin\theta_2 = 0$$

0.15

Solving ***1.11*** for $\ddot{x}$,

$$\ddot{x} = \frac{F + m_1\, l_1\, \ddot{\theta_1}\, \cos\theta_1 - m_1\, l_1\dot{\theta_1}{}^2 \sin\theta_1 + m_2\, l_2\ddot{\theta_2}\, \cos\theta_2 - m_2\, l_2\dot{\theta_2}{}^2 \sin\theta_2}{\bar{M}}$$

0.16

Solving ***1.13*** for $\ddot{\theta_1}$

$$\ddot{\theta_1} = \frac{\ddot{x}\cos\theta_1 - g\sin\theta_1}{l_1}$$

0.17

Solving ***1.15*** for $\ddot{\boldsymbol{\theta}}_2$,

$$\ddot{\theta}_2 = \frac{\ddot{x}\cos\theta_2 - g\sin\theta_2}{l_2} \tag{0.18}$$

State Space Representation

The state vector can be written as

$$\boldsymbol{X} = \begin{bmatrix} x \\ \dot{x} \\ \theta_1 \\ \dot{\theta_1} \\ \theta_2 \\ \dot{\theta_2} \end{bmatrix} \qquad \text{Hence, } \dot{\boldsymbol{X}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta_1} \\ \ddot{\theta_1} \\ \dot{\theta_2} \\ \ddot{\theta_2} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}$$

$\boldsymbol{f_1}$, $\boldsymbol{f_2}$, $\boldsymbol{f_3}$ are readily in terms of state variables.

Substituting for $\ddot{\theta_1}$ and $\ddot{\theta_2}$ in the equation for $\ddot{x}$, we can write $\ddot{x}$ in terms of state variables.

From ***1.16***, ***1.17*** and ***1.18***, we obtain

$$\ddot{x} = \frac{F - m_1 g \sin\theta_1 \cos\theta_1 - m_2 g \sin\theta_2 \cos\theta_2 - m_1 l_1 \dot{\theta_1}^2 \sin\theta_1 - m_2 l_2 \dot{\theta_2}^2 \sin\theta_2}{\bar{M} - m_1 cos^2\theta_1 - m_2 cos^2\theta_2}$$

$$= f_2 \tag{0.1}$$

We can compute functions for $\boldsymbol{f_4}$ and $\boldsymbol{f_6}$ by substituting $\ddot{x}$ obtained in ***1.19*** back in ***1.16*** and ***1.17***.

Below is the MATLAB code snippet to find $\boldsymbol{f_4} = \ddot{\theta_1}$ and $\boldsymbol{f_6} = \ddot{\theta_2}$

```
clc

syms M m1 m2 g l1 l2 x xd th1 th1d th1dd th2 th2d th2dd F
syms f1 f2 f3 f4 f5 f6

xdd_num = F- m1*g*sin(th1)*cos(th1)- m2*g*sin(th2)*cos(th2)-
 m1*l1*(th1d^2)*sin(th1)- m2*l2*(th2d^2)*sin(th2);
xdd_den = (M+m1+m2) - m1*(cos(th1))^2 - m2*(cos(th2))^2;
xdd = xdd_num/xdd_den;

th1dd= (xdd*cos(th1)-g*sin(th1))/l1;
disp('th1dd = ');
pretty(th1dd);

th2dd= (xdd*cos(th2)-g*sin(th2))/l2;
disp('th2dd = ');
pretty(th2dd);
```

Command Window

```
th1dd =
                                               2                      2
               cos(th1) (l1 m1 sin(th1) th1d  + l2 m2 sin(th2) th2d  - F + g m1 cos(th1) sin(th1) + g m2 cos(th2) sin(th2))
  g sin(th1) + ---------------------------------------------------------------------------------------------------------
                                                                2              2
                                                - m1 cos(th1)  - m2 cos(th2)  + M + m1 + m2
- ---------------------------------------------------------------------------------------------------------------------
                                                             l1

th2dd =
                                               2                      2
               cos(th2) (l1 m1 sin(th1) th1d  + l2 m2 sin(th2) th2d  - F + g m1 cos(th1) sin(th1) + g m2 cos(th2) sin(th2))
  g sin(th2) + ---------------------------------------------------------------------------------------------------------
                                                                2              2
                                                - m1 cos(th1)  - m2 cos(th2)  + M + m1 + m2
- ---------------------------------------------------------------------------------------------------------------------
                                                             l2
```

Linearization of the Non-Linear System

We now compute $\boldsymbol{A_f}$ matrix for the linearized system by taking the gradient of $\dot{X}$ with respect to the state variables

$$\boldsymbol{A_f} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} & \frac{\partial f_1}{\partial x_5} & \frac{\partial f_1}{\partial x_6} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} & \frac{\partial f_2}{\partial x_5} & \frac{\partial f_2}{\partial x_6} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} & \frac{\partial f_3}{\partial x_5} & \frac{\partial f_3}{\partial x_6} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} & \frac{\partial f_4}{\partial x_5} & \frac{\partial f_4}{\partial x_6} \\ \frac{\partial f_5}{\partial x_1} & \frac{\partial f_5}{\partial x_2} & \frac{\partial f_5}{\partial x_3} & \frac{\partial f_5}{\partial x_4} & \frac{\partial f_5}{\partial x_5} & \frac{\partial f_5}{\partial x_6} \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \frac{\partial f_6}{\partial x_3} & \frac{\partial f_6}{\partial x_4} & \frac{\partial f_6}{\partial x_5} & \frac{\partial f_6}{\partial x_6} \end{bmatrix}_{(\dot{x}=0,\dot{\theta}_1=0,\dot{\theta}_2\,0)}$$

and $\boldsymbol{B_f}$ matrix by taking the gradient of $\dot{X}$ with respect to input F

$$\boldsymbol{B_f} = \begin{bmatrix} \frac{\partial f_1}{\partial F} \\ \frac{\partial f_2}{\partial F} \\ \frac{\partial f_3}{\partial F} \\ \frac{\partial f_4}{\partial F} \\ \frac{\partial f_5}{\partial F} \\ \frac{\partial f_6}{\partial F} \end{bmatrix}_{(\dot{x}=0,\dot{\theta}_1=0,\dot{\theta}_2\,0)}$$

Below is the MATLAB code snippet to obtain $\boldsymbol{A_f}$ and $\boldsymbol{B_f}$.

```
f1= xd;
f2= xdd;
f3= th1d;
f4= th1dd;
f5= th2d;
f6= th2dd;

f = [f1 f2 f3 f4 f5 f6];
q = [x xd th1 th1d th2 th2d];
u = [F];

q_val = [0 0 0 0 0 0];

A_j = jacobian(f, q);
A_f = subs(A_j, q, q_val);
% A_f in symbolic form
disp('A_f = ');
disp(A_f);

B_j = jacobian(f, u);
B_f = subs(B_j, q, q_val);
% B_f in symbolic form
disp('B_f = ');
disp(B_f);
```

Command Window

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

B_f =
        0
      1/M
        0
 1/(M*l1)
        0
 1/(M*l2)
```

Controllability of the Linearized System

$$C = [B\ AB\ A^2B\ \dots\ A^{n-1}B]$$

A continuous time-invariant linear state-space model is controllable if and only if

$$rank(C) = n$$

where n is the number of state variables.

Here because we have only one input, our C matrix will be a square matrix of n x n dimension. A square matrix is full rank if and only if its determinant is nonzero

```
CO = [B_f A_f*B_f (A_f^2)*B_f (A_f^3)*B_f (A_f^4)*B_f (A_f^5)*B_f];
disp('det(CO) = ');
pretty(det(CO))
```

Command Window

```
det(CO) =
   6   2       6           6   2
  g  l1  - 2 g  l1 l2 + g  l2
- ----------------------------
            6   6   6
           M  l1  l2
```

It is easy to observe that

$$l_1{}^2 - 2l_1l_2 + l_2{}^2 \neq 0$$

$$(l_1 - l_2)^2 \neq 0$$

$$\therefore l_1 \neq l_2$$

Thus, for the system to be controllable lengths of cables should not be same for the two masses**.**

Below is the MATLAB code snippet to check controllability. As we have 6 state variables, Rank = 6 implies our system is controllable.

```
p = [M m1 m2 l1 l2 g];
p_val = [1000 100 100 20 10 10];

% A_f in double form
A_f = double(subs(A_f, p, p_val));

% B_f in double form
B_f = double(subs(B_f, p, p_val));

Co = ctrb(A_f,B_f);
disp('Co = ');
disp(Co);
disp('rank(Co) = ');
disp(rank(Co));
```

Command Window

```
Co =
   1.0e-03 *

         0    1.0000         0   -0.1500         0    0.1475
    1.0000         0   -0.1500         0    0.1475         0
         0    0.0500         0   -0.0325         0    0.0236
    0.0500         0   -0.0325         0    0.0236         0
         0    0.1000         0   -0.1150         0    0.1298
    0.1000         0   -0.1150         0    0.1298         0

rank(Co) =
     6
```

Linear Quadratic Regulator Design

The objective of LQR control design is to minimize the quadratic cost function given as

$$J = \int_0^{\infty} x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)\, d\tau$$

where x and u are the state and control of the LTI system.

$Q \succeq 0,\ R \succ 0;$

(A, B) are stabilizable

Regardless of the values of Q and R

matrices J has a unique minimum that can be obtained by solving the Algebraic Riccati Equation:

$$A^T X + XA - XBR^{-1}B^T X + Q = 0.$$

$$for\ u(t) = -kx(t),$$

$$K = R^{-1}B^T P$$

Q and R are our design parameters that we adjust depending on whether we want to penalize state variables or control variables. A large value of Q implies we want the system to stabilize with minimum changes in state variables. On the other hand, a small R means our input is inexpensive and we can afford to spend large(weighted) energy to stabilize our system.[1]

The eigen values of the linearized system matrix $\boldsymbol{A_f}$ before applying LQR are:

```
poles = eigs(A_f);
disp(poles);
```

```
0.0000 + 1.0531i
0.0000 - 1.0531i
0.0000 + 0.7356i
0.0000 - 0.7356i
0.0000 + 0.0000i
0.0000 + 0.0000i
```

All eigen values are on the imaginary axis. Thus, the system will produce sustained oscillations on the output.

MATLAB Code snippet and Simulation For Linearized System:

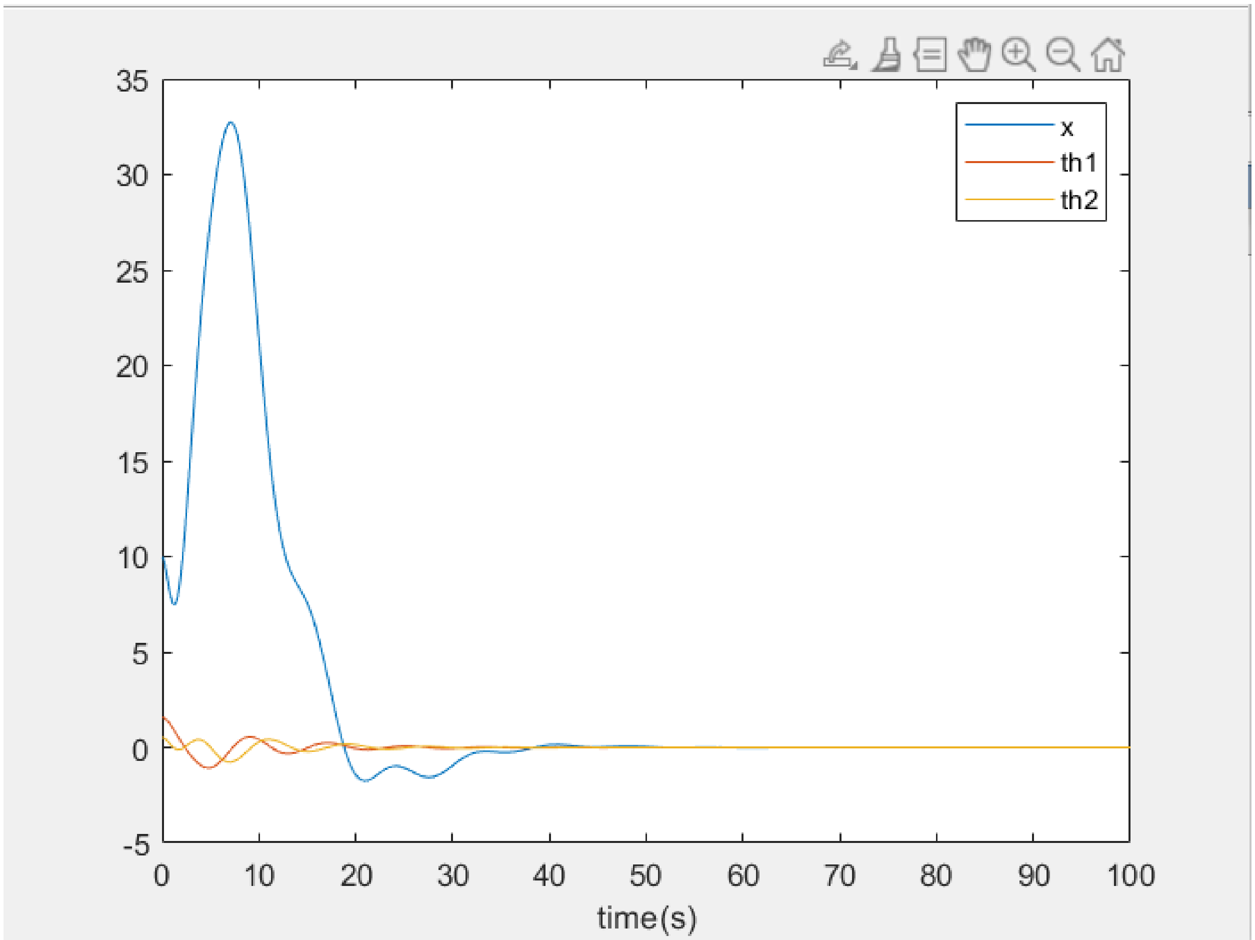
```
% Q matrix
Q = zeros(6);

Q(1,1) = 100;      % moderate penalty on x
Q(2,2) = 1;
Q(3,3) = 1000000; % high penalty on th1
Q(4,4) = 1;
Q(5,5) = 1000000; % high penalty on th2
Q(6,6) = 1;

% R matrix
R = 0.01;          % inexpensive F

% full state feedback gain
K = lqr(A_f,B_f,Q,R);

% closed loop system gain
Ac = (A_f - B_f*K);

% output variables: x,th1,th2
C = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
D = [0];
% Plant + LQR
LQR_sys = ss(Ac,B_f,C,D);
```

The eigen values of the linearized system matrix $\boldsymbol{A_f}$ before applying LQR are:

```
% initial conditions
X0 = [10 0 pi/2 0 pi/6 0]

time= 0:0.1:100
[y,t,x]=initial(LQR_sys,X0,time);
plot(t,y);

% Lyapunov's indirect method after LQR
poles_c = eigs(Ac);
disp(poles_c);
```

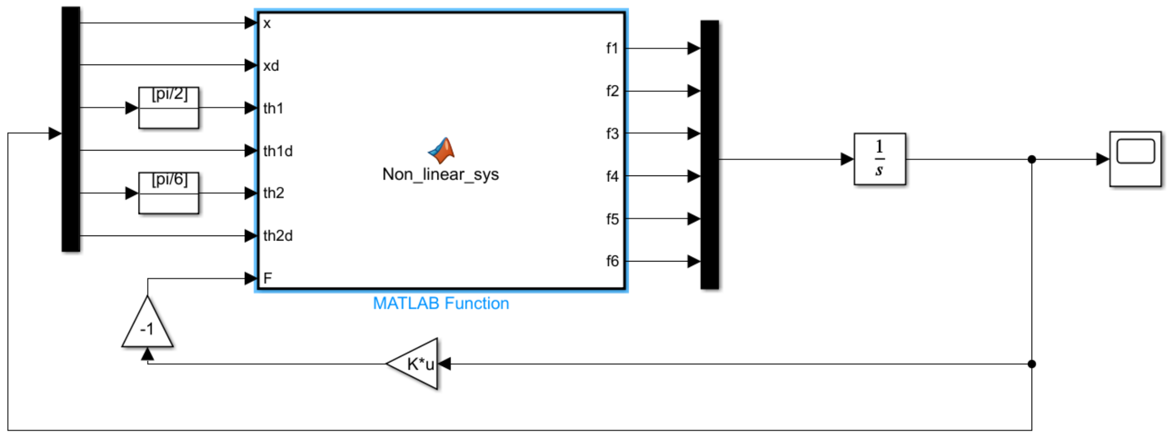
The eigen values of the linearized system matrix $\boldsymbol{A_f} - \boldsymbol{B_f K}$ after applying LQR are:

```
Command Window

  -0.4762 + 1.1385i
  -0.4762 - 1.1385i
  -0.1233 + 0.7700i
  -0.1233 - 0.7700i
  -0.1599 + 0.1686i
  -0.1599 - 0.1686i
```



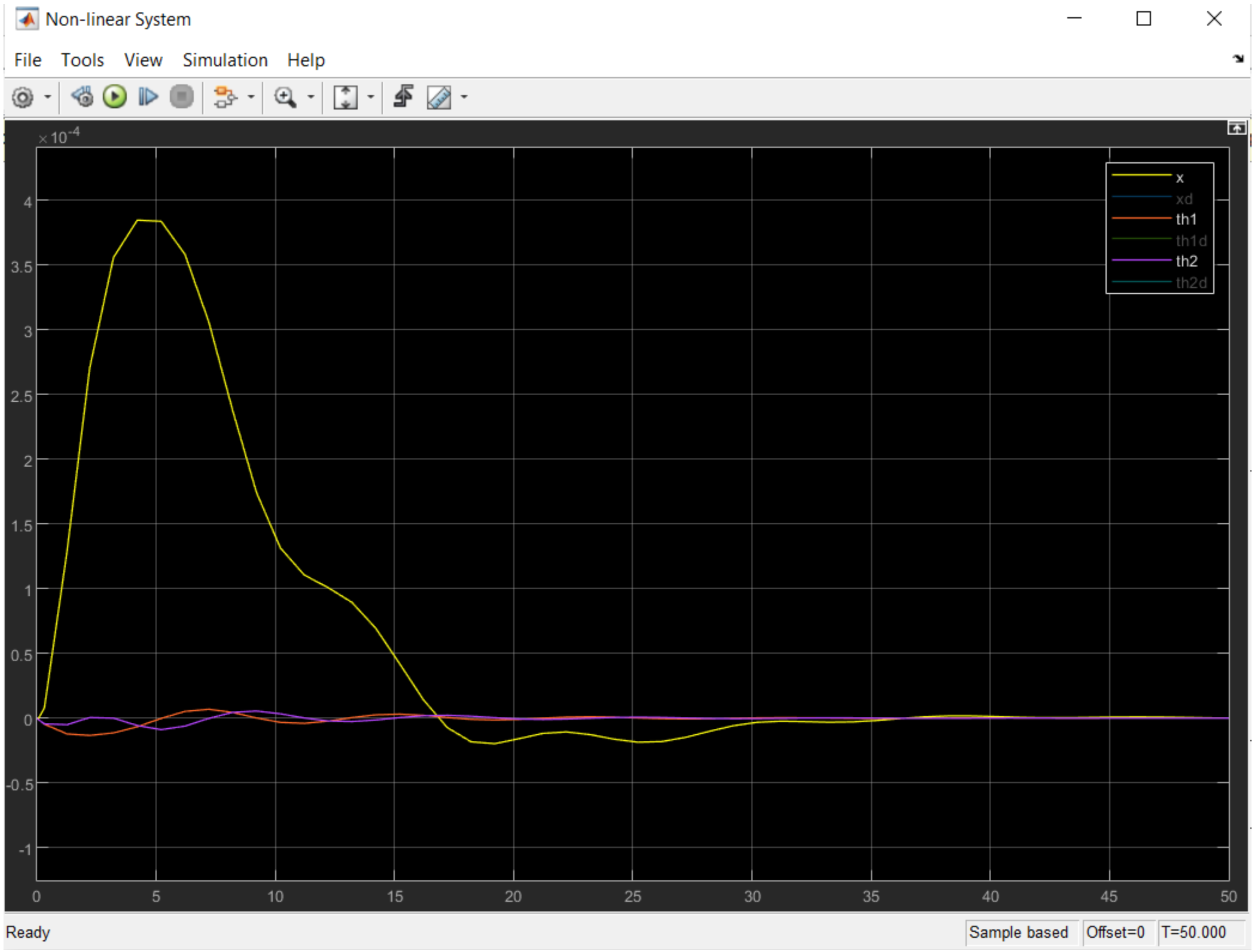
x
th1
th2
35
30
25
20
15
10
5
0
-5
0
10
20
30
40
50
60
70
80
90
100
time(s)

MATLAB Code snippet and Simulation For Non-Linear System:



MATLAB Function

```
function [f1,f2,f3,f4,f5,f6]  = Non_linear_sys(x,xd,th1,th1d,th2,th2d,F)

M = 1000; % mass of cart
m1 = 100; % mass 1
m2 = 100; % mass 2
l1 = 20;  % cable length of first pendulum 20
l2 = 10;  % cable length of second pendulum 10
g = 10;   % acceleration due to gravity


xdd_num = F- m1*g*sin(th1)*cos(th1)- m2*g*sin(th2)*cos(th2)- m1*l1*(th1d^2)*sin(th1)- m2*l2*(th2d^2)*sin(th2);
xdd_den = (M+m1+m2) - m1*(cos(th1))^2 - m2*(cos(th2))^2;
xdd = xdd_num/xdd_den;

th1dd= (xdd*cos(th1)-g*sin(th1))/l1;

th2dd= (xdd*cos(th2)-g*sin(th2))/l2;


f1= xd;
f2= xdd;
f3= th1d;
f4= th1dd;
f5= th2d;
f6= th2dd;
```



Non-linear System
File Tools View Simulation Help
× 10^-4
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0 5 10 15 20 25 30 35 40 45 50
x
xd
th1
th1d
th2
th2d
Ready
Sample based
Offset=0
T=50.000

Observability of the Linearized System

For time-invariant linear systems, if the row rank of the observability matrix, defined as

$$O = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA^2} \\ . \\ . \\ . \\ \boldsymbol{CA^{n-1}} \end{bmatrix}$$

is equal to $\boldsymbol{n}$, then the system is observable.

$$\boldsymbol{rank}(\boldsymbol{O}) = \boldsymbol{n}$$

where n is the number of state variables.

We have four cases of the output variables.

Case 1: $\boldsymbol{x}$

$$\boldsymbol{C_1} = [\mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0}]$$

Case 2: $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$

$$\boldsymbol{C_2} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Case 3: $\boldsymbol{x}, \boldsymbol{\theta_2}$

$$\boldsymbol{C_3} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Case 4: $\boldsymbol{x}, \boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$

$$\boldsymbol{C_4} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

```
C1 = [1 0 0 0 0 0];
C2 = [0 0 1 0 0 0; 0 0 0 0 1 0];
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0];
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];

O1 = obsv(A_f,C1);
disp('rank(O1) = ');
disp(rank(O1));
O2 = obsv(A_f,C2);
disp('rank(O2) = ');
disp(rank(O2));
O3 = obsv(A_f,C3);
disp('rank(O3) = ');
disp(rank(O3));
O4 = obsv(A_f,C4);
disp('rank(O4) = ');
disp(rank(O4));
```

Command Window

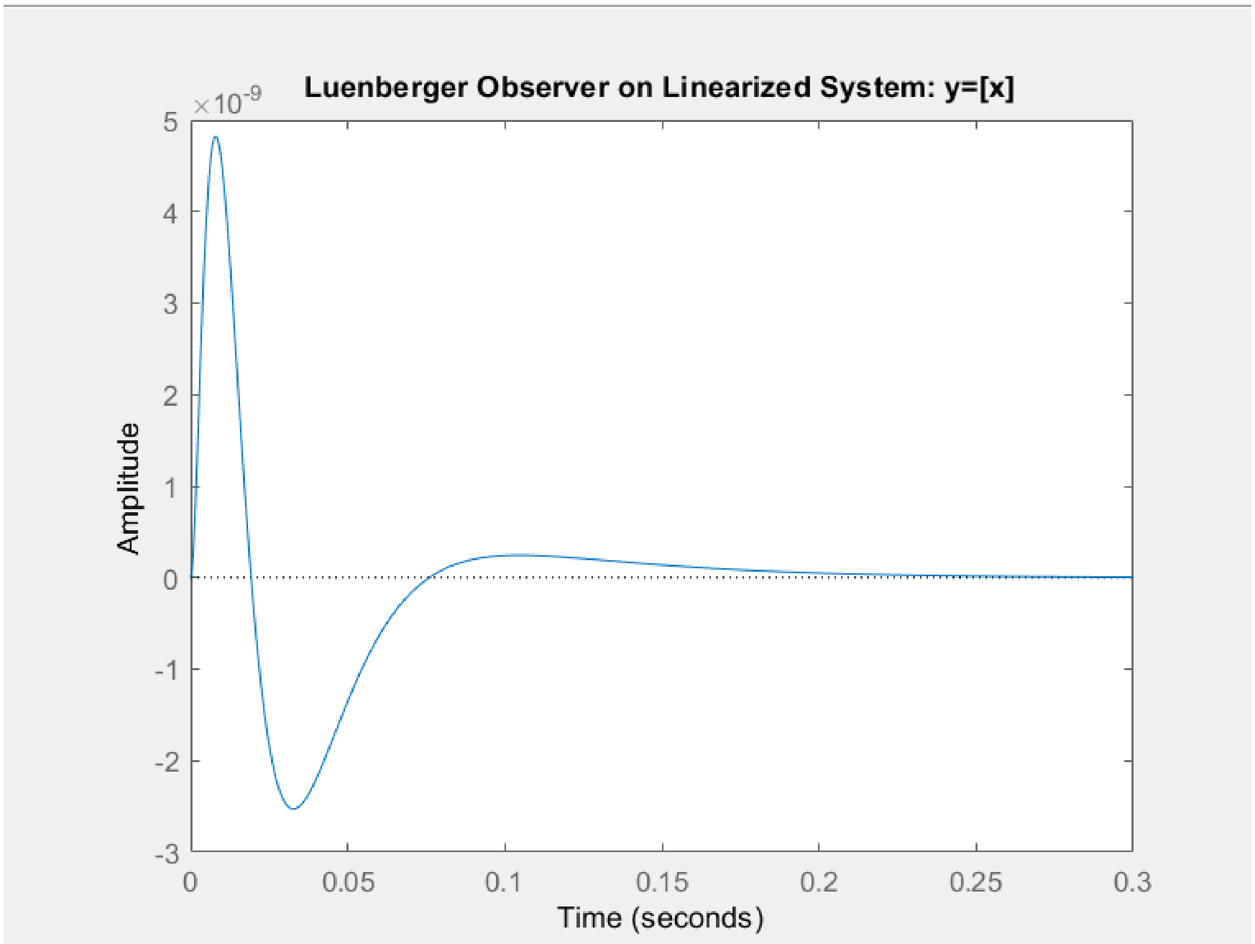
```
rank(O1) =
     6

rank(O2) =
     4

rank(O3) =
     6

rank(O4) =
     6
```

Thus, the system is not observable for output vector $[\boldsymbol{\theta_1}, \boldsymbol{\theta_2}]$.

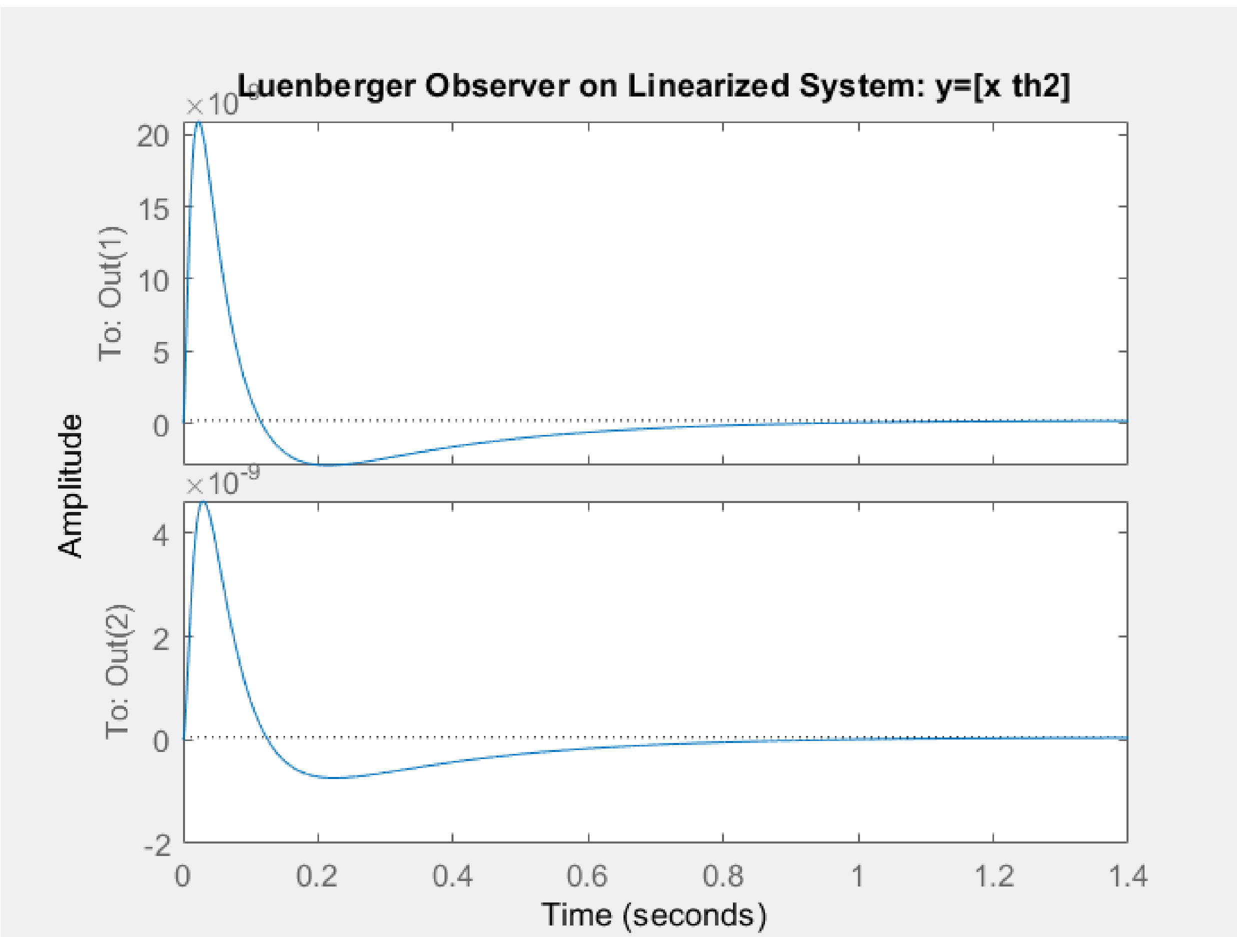
Luenberger Observer Design

MATLAB Code snippet and Simulation For Linearized System:

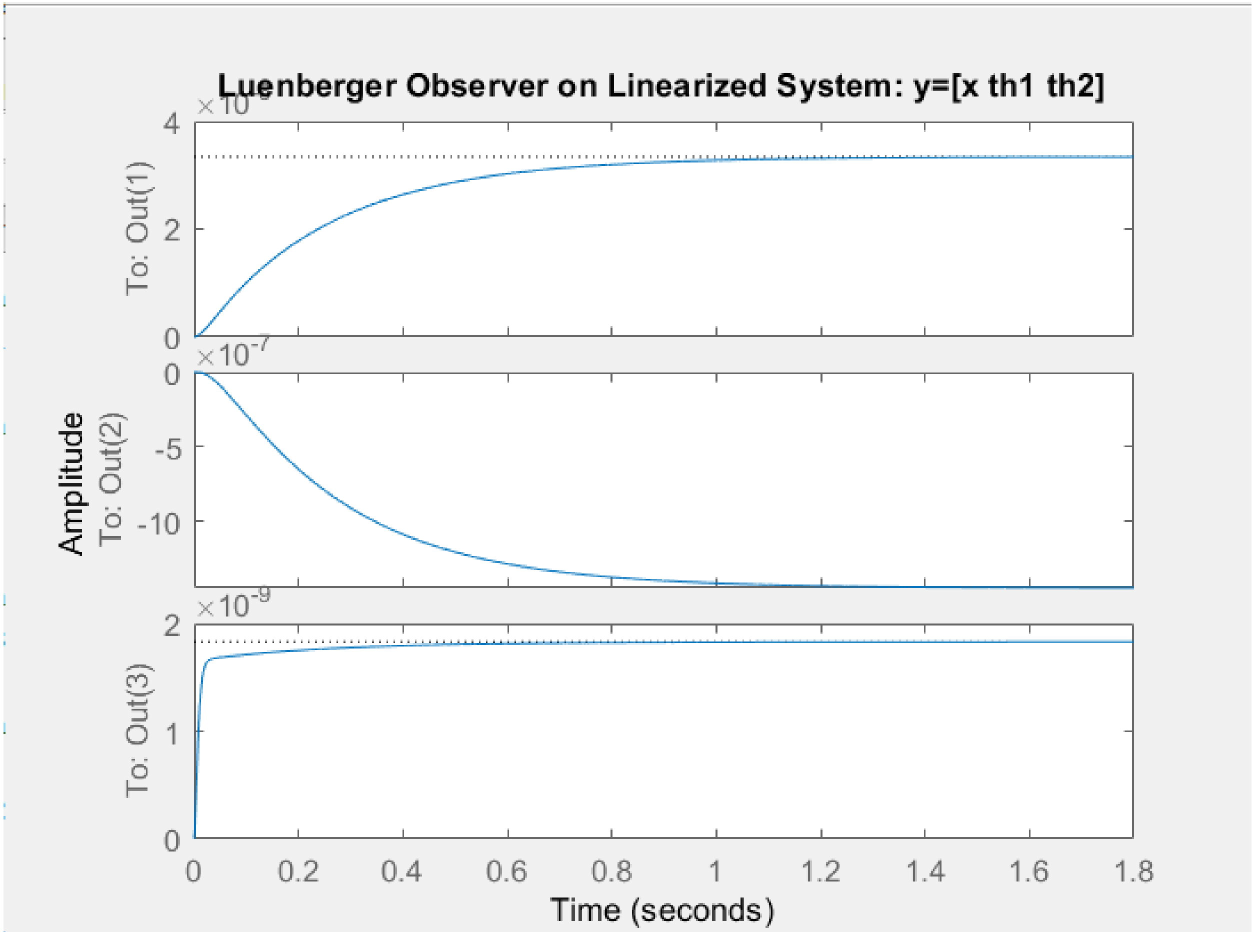
```
Poles_Luen = [-200 -300 -100 -60 -20 -4]';

% Output variables: x
C1 = [1 0 0 0 0 0];
L = place(A_f',C1',Poles_Luen)';
Luen_sys = ss(A_f-L*C1,B_f,C1,D);
% initial conditions
X0 = [10 0 pi/2 0 pi/6 0];

time= 0:0.1:100;
[y,~,x]=initial(Luen_sys,X0,time);
step(Luen_sys);
title('Luenberger Observer on Linearized System: y=[x]')
```



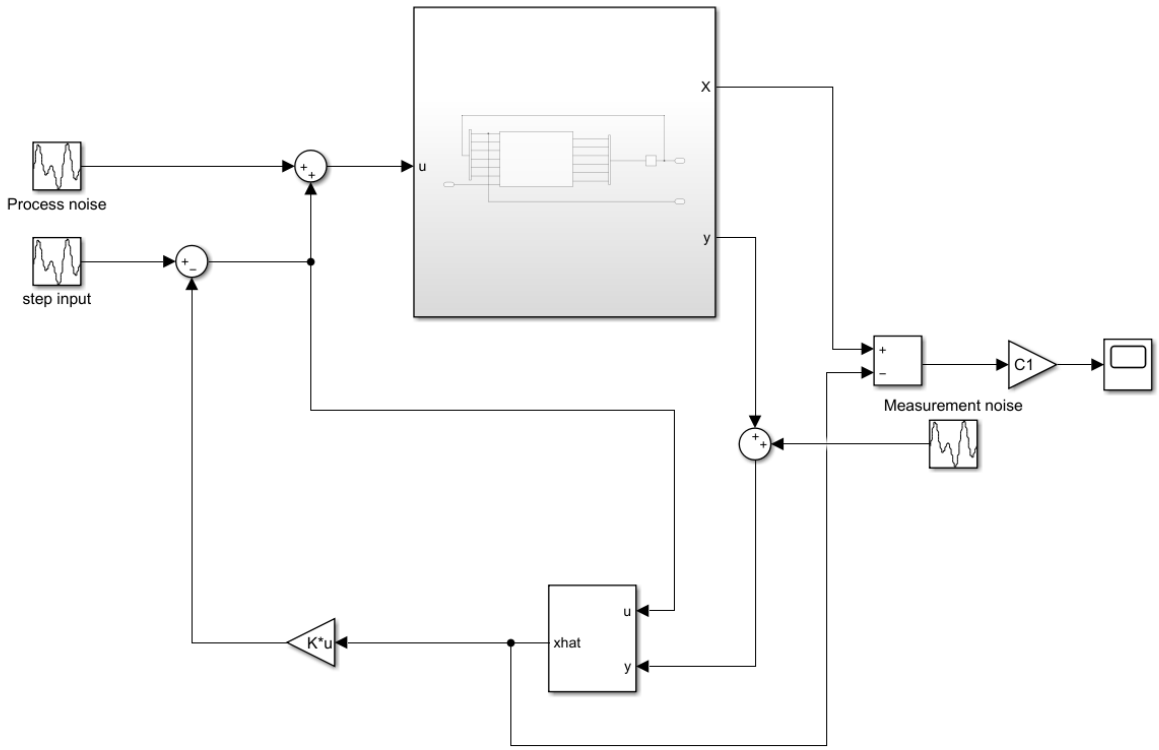
```
% Output variables: x, th2
C3 = [1 0 0 0 0 0; 0 0 0 0 1 0];
L = place(A_f',C3',Poles_Luen)';
Luen_sys = ss(A_f-L*C3,B_f,C3,D);
% initial conditions
X0 = [10 0 pi/2 0 pi/6 0];

time= 0:0.1:100;
[y,~,x]=initial(Luen_sys,X0,time);
step(Luen_sys);
title('Luenberger Observer on Linearized System: y=[x th2]')
```



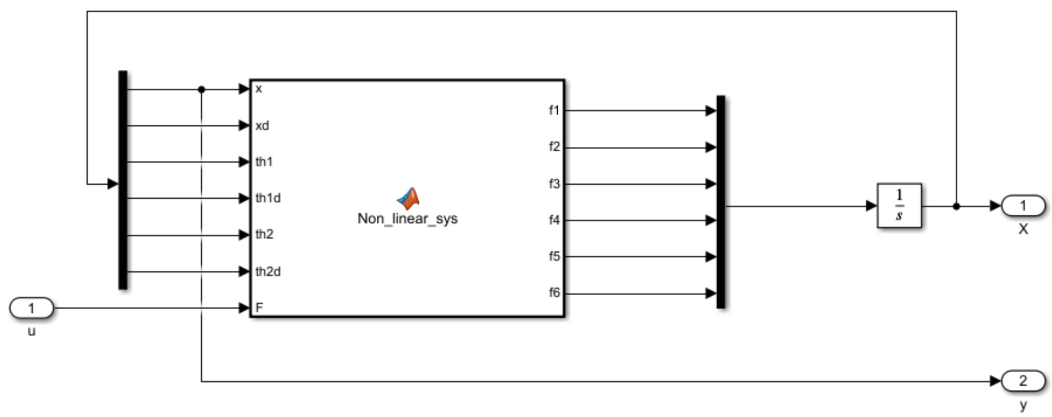
```
% Output variables: x, th1, th2
C4 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
L = place(A_f',C4',Poles_Luen)';
Luen_sys = ss(A_f-L*C4,B_f,C4,D);
% initial conditions
X0 = [10 0 pi/2 0 pi/6 0];

time= 0:0.1:100;
[y,~,x]=initial(Luen_sys,X0,time);
step(Luen_sys);
title('Luenberger Observer on Linearized System: y=[x th1 th2]')
```



Linear Quadratic Gaussian Controller Design



Inside the subsystem:



References

[1]https://www.researchgate.net/post/how_to_determine_the_values_of_the_control_matrices_Q_and_R_for_the_LQR_strategy_when_numerically_simulating_the_semi-active_TLCD#:~:text=The%20parameters%20Q%20and%20R,with%20less%20(weighted)%20energy