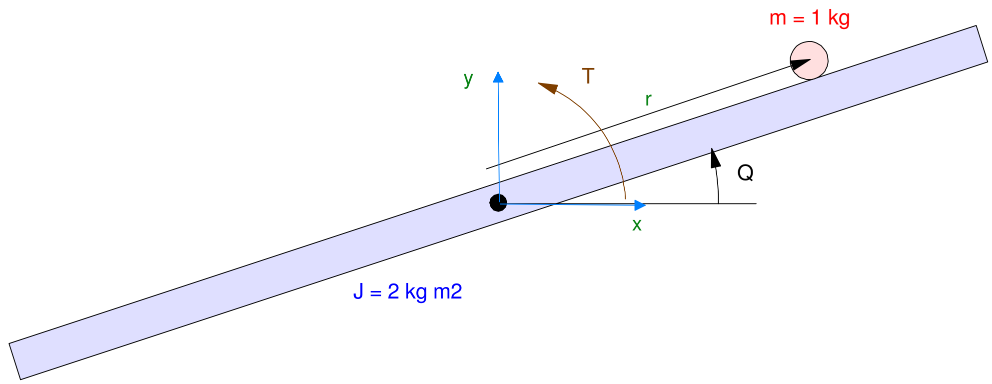
ECE 463/663: Solution to Homework #10

Kalman Filters. Due Monday, April 15th



Ball and Beam from homework #4

Use the dynamics for a ball and beam system from previous homework sets

- mass of ball = 1.0kg
- inertia of beam = 2 kg m2
- Closed-Loop System's 2% Settling Time = 6 seconds

Assume that only the beam angle and ball position are measured. Also assume each of these has noise:

$$y_r = r + N(0, 0.01m)$$ *Normal(mean, standard deviation)*

$$y_\theta = \theta + N(0, 0.002 \text{ radians})$$

Assume that the input (Torque) has a disturbance as well (due to the uncertain ball mass)

$$U_d = N(0, 0.1\ Nm)$$

1) Design a Kalman filter to estimate the four states of the plant using only position and angle information

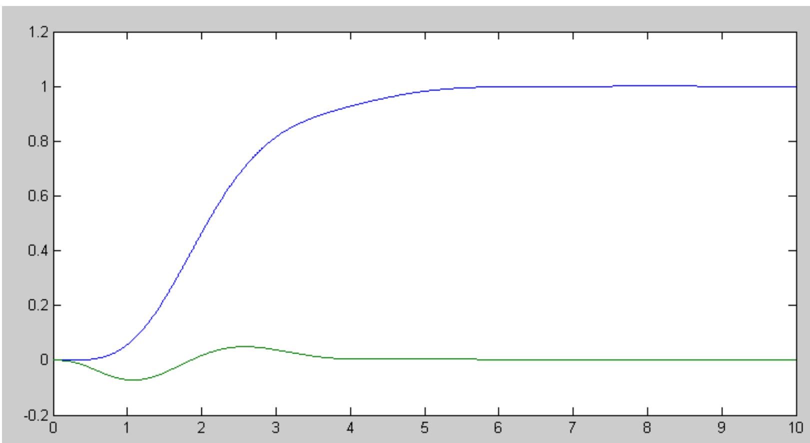
```
>> A = [0,0,1,0 ; 0,0,0,1 ; 0,-7,0,0 ; -3.27,0,0,0]

         0         0    1.0000         0
         0         0         0    1.0000
         0   -7.0000         0         0
   -3.2700         0         0         0

>> B = [0;0;0;0.33]

         0
         0
         0
    0.3300

>> C = [1,0,0,0;0,1,0,0]

     1     0     0     0
     0     1     0     0


W = [0.01,0 ; 0,0.002]

    0.0100         0
         0    0.0020

>> R = W'*W

>> Q = (0.1*B) * (0.1*B)'

         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0    0.0011

>> H = lqr(A', C', Q, R)'

    1.2418   -4.7668
   -0.1907    6.3053
    1.2255  -10.3848
   -1.0236   20.3326

>> eig(A-H*C)

  -2.7285 + 2.7285i
  -2.7285 - 2.7285i
  -1.0450 + 1.0450i
  -1.0450 - 1.0450i
```

2) Design a servo-compensator to track a constant set-point and reject a constant disturbance

```
A5 = [A, zeros(4,1) ; 1,0,0,0, 0]

         0         0    1.0000         0         0
         0         0         0    1.0000         0
         0   -7.0000         0         0         0
   -3.2700         0         0         0         0
    1.0000         0         0         0         0

B5 = [B ; 0]

         0
         0
         0
    0.3300
         0


C5 = [C, [0;0]]

     1     0     0     0     0
     0     1     0     0     0

D5 = [0;0];
B5r = [0;0;0;0;-1]

     0
     0
     0
     0
    -1

t = [0:0.01:10]';
X0 = zeros(5,1);
R = 0*t + 1;

K5 = lqr(A5, B5, diag([0,0,50,0,100]),1)
```

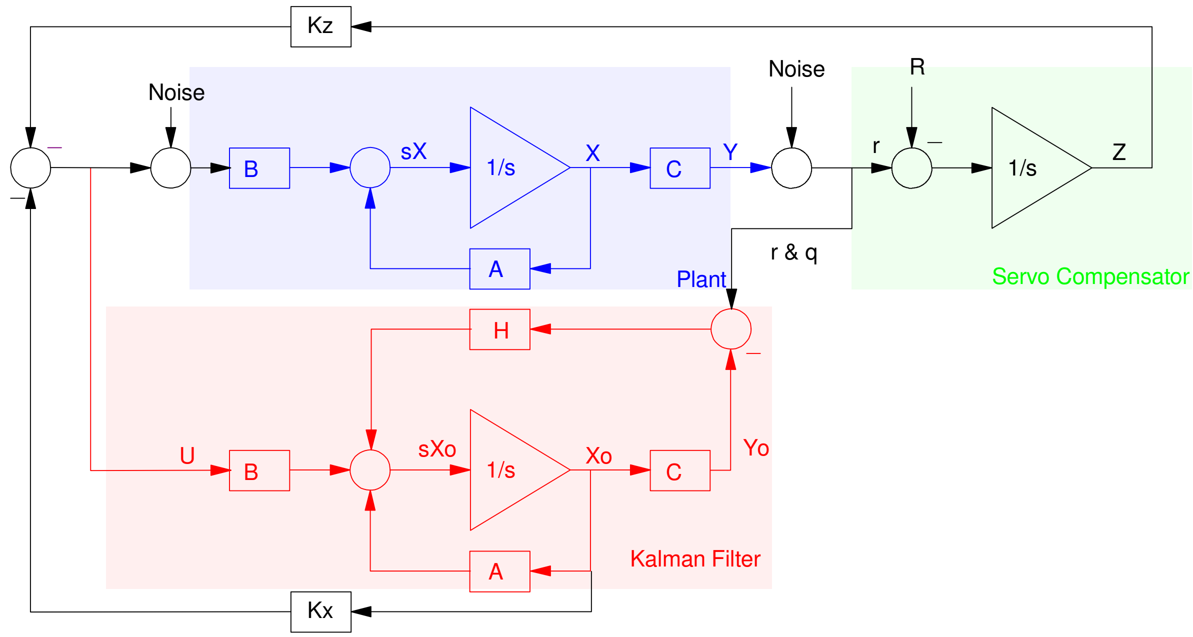
K5 = -32.3981   65.7222 -20.3782   19.9579 -10.0000

```
y = step3(A5-B5*K5, B5r, C5, D5, t, X0, R);
plot(t,y);
```



3) Simulate the step response of the plant, servo compensator, and observer in Matlab

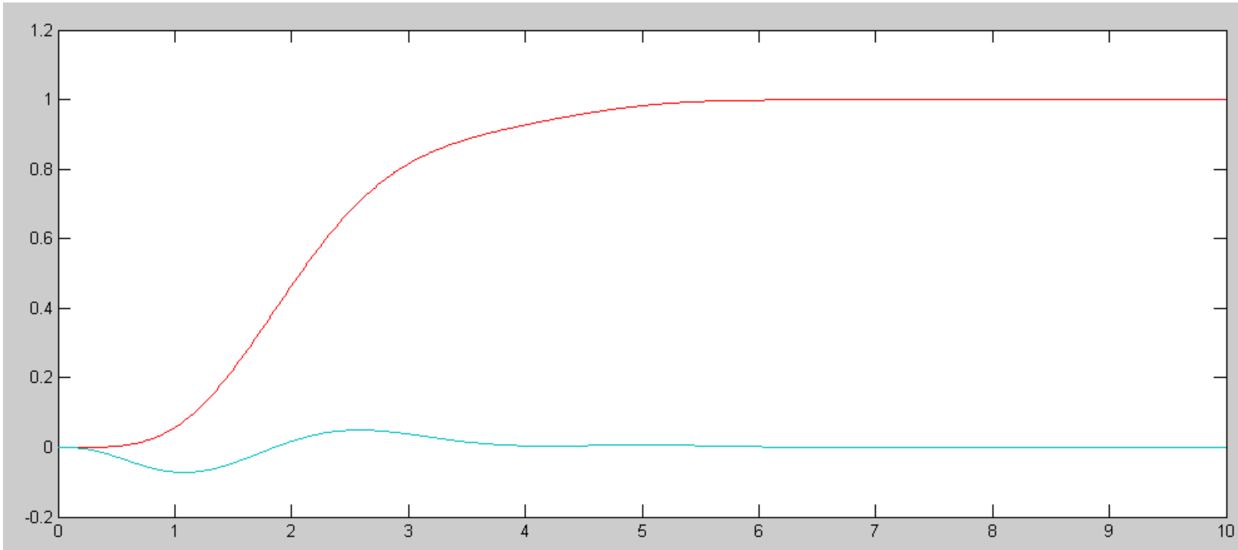
- With no noise, and
- With noise on the sensors and inputs



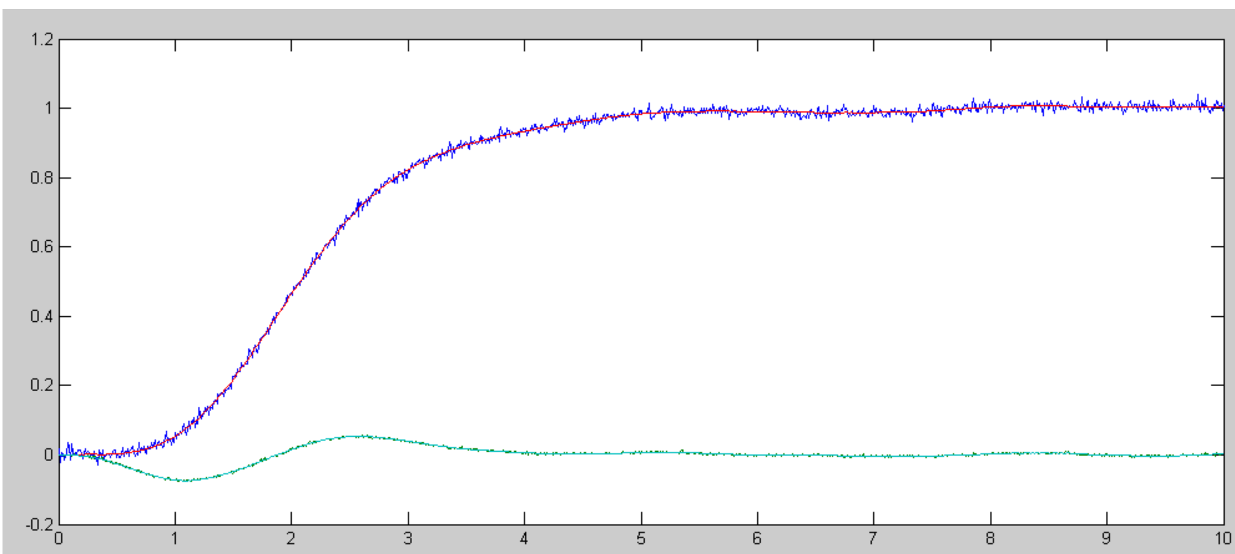
$$\begin{bmatrix} sX \\ sZ \\ sX_o \end{bmatrix} = \begin{bmatrix} A & -BK_z & -BK_x \\ C & 0 & 0 \\ HC & -BK_z & A - BK_x - HC \end{bmatrix} \begin{bmatrix} X \\ Z \\ X_o \end{bmatrix} + \begin{bmatrix} 0 & 0 & B \\ -1 & [10] & 0 \\ 0 & H & 0 \end{bmatrix} \begin{bmatrix} R \\ N_y \\ N_u \end{bmatrix}$$

```
>> Kz = K5(5)

Kz =

  -10.0000

>> Kx = K5(1:4)

Kx =

  -32.3981    65.7222   -20.3782    19.9579
```

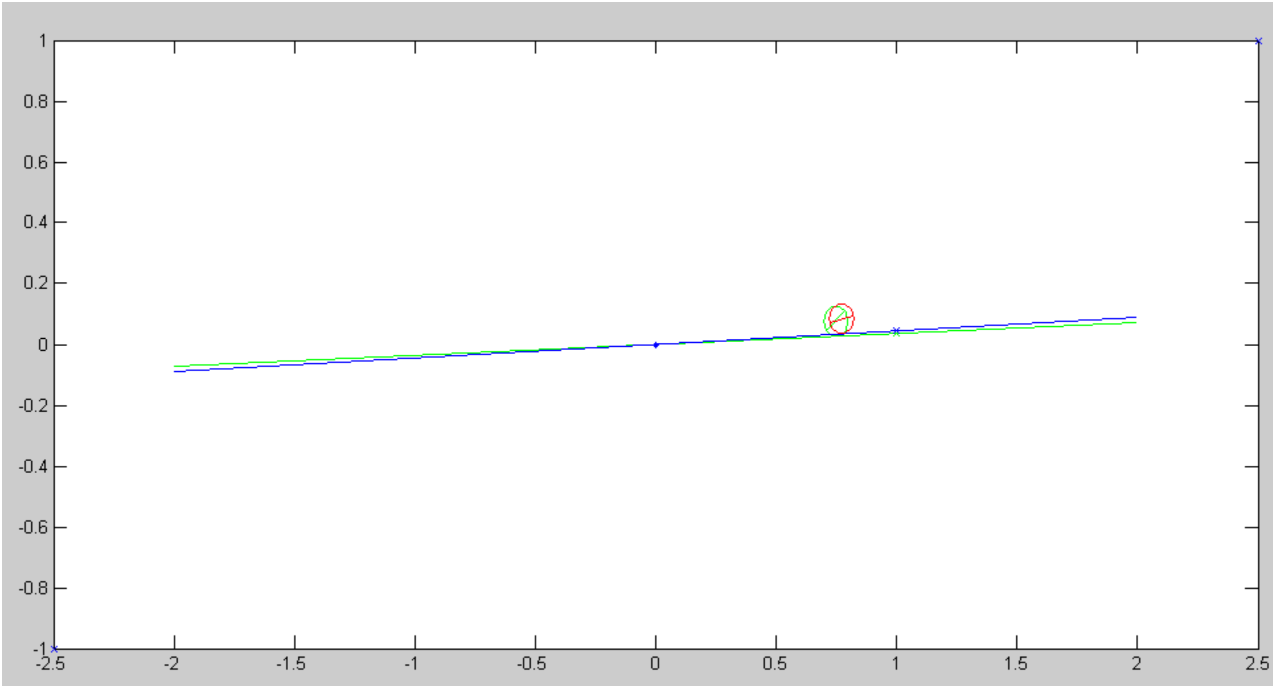
```
>> A9 = [A, -B*Kz, -B*Kx ; 1,0,0,0, 0, 0,0,0,0 ; H*C, -B*Kz, A-B*Kx-H*C]

A9 =

         0         0    1.0000         0         0         0         0         0         0
         0         0         0    1.0000         0         0         0         0         0
         0   -7.0000         0         0         0         0         0         0         0
   -3.2700         0         0         0    3.3000   10.6914  -21.6883    6.7248   -6.5861
    1.0000         0         0         0         0         0         0         0         0
    1.2418   -4.7668         0         0         0   -1.2418    4.7668    1.0000         0
   -0.1907    6.3053         0         0         0    0.1907   -6.3053         0    1.0000
    1.2255  -10.3848         0         0         0   -1.2255    3.3848         0         0
   -1.0236   20.3326         0         0    3.3000    8.4450  -42.0210    6.7248   -6.5861

>> eig(A9)

ans =

  -2.7285 + 2.7285i
  -2.7285 - 2.7285i
  -0.8172 + 2.3952i
  -0.8172 - 2.3952i
  -2.9024
  -1.0450 + 1.0450i
  -1.0450 - 1.0450i
  -1.0247 + 0.4390i
  -1.0247 - 0.4390i

>> B9 = [zeros(4,1),zeros(4,2),B ; -1, 1,0, 0 ; zeros(4,1), H, zeros(4,1)]


         0         0         0         0
         0         0         0         0
         0         0         0         0
         0         0         0    0.3300
   -1.0000    1.0000         0         0
         0    1.2418   -4.7668         0
         0   -0.1907    6.3053         0
         0    1.2255  -10.3848         0
         0   -1.0236   20.3326         0

>> >> R = 0*t + 1;
>> Nx = randn(size(t)) * 0.01;
>> Nq = randn(size(t)) * 0.002;
>> Nu = randn(size(t)) * 0.1;
>> X0 = zeros(9,1);
>> C9 = [1,0,0,0,0,0,0,0,0;0,1,0,0,0,0,0,0,0;0,0,0,0,0,1,0,0,0;0,0,0,0,0,0,1,0,0]

C9 =

     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     1     0     0
>> D9 = [0,1,0,0 ; 0,0,1,0 ; 0,0,0,0 ; 0,0,0,0]

D9 =

     0     1     0     0
     0     0     1     0
     0     0     0     0
     0     0     0     0
>> y = step3(A9, B9, C9, D9, t, X0, [R,0*Nx,0*Nq,0*Nu]);
>> plot(t,y)
```

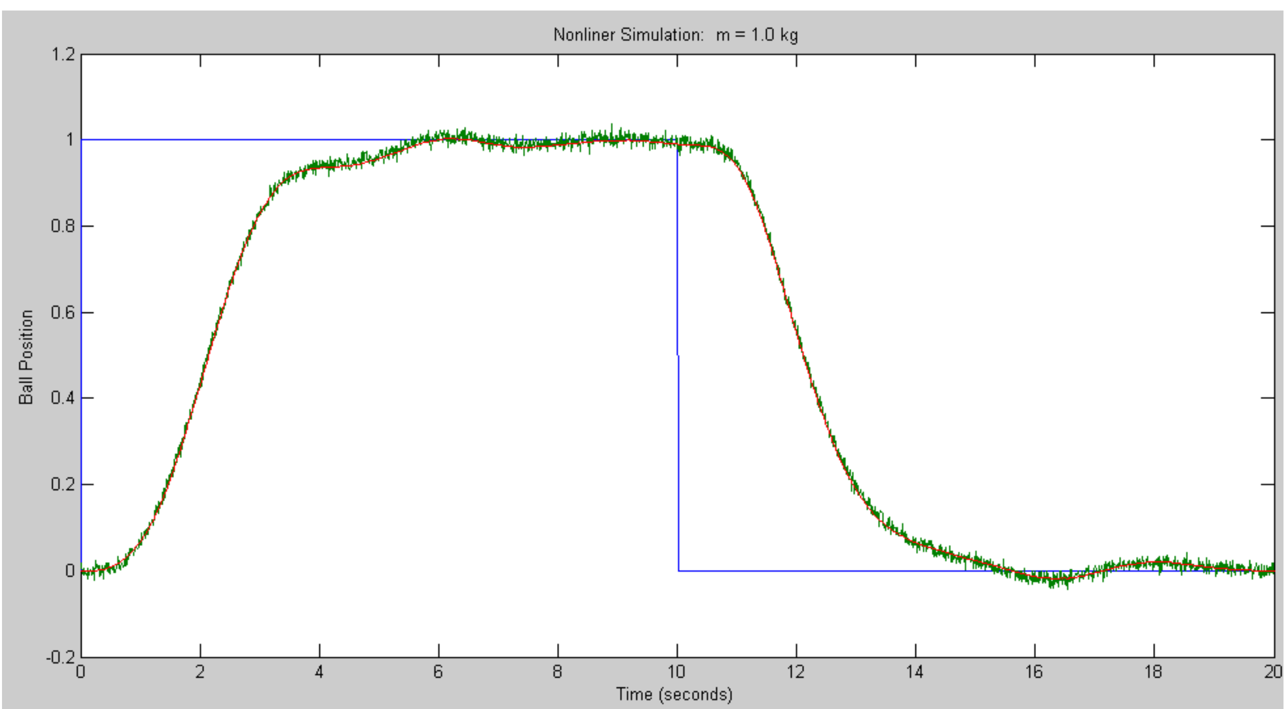


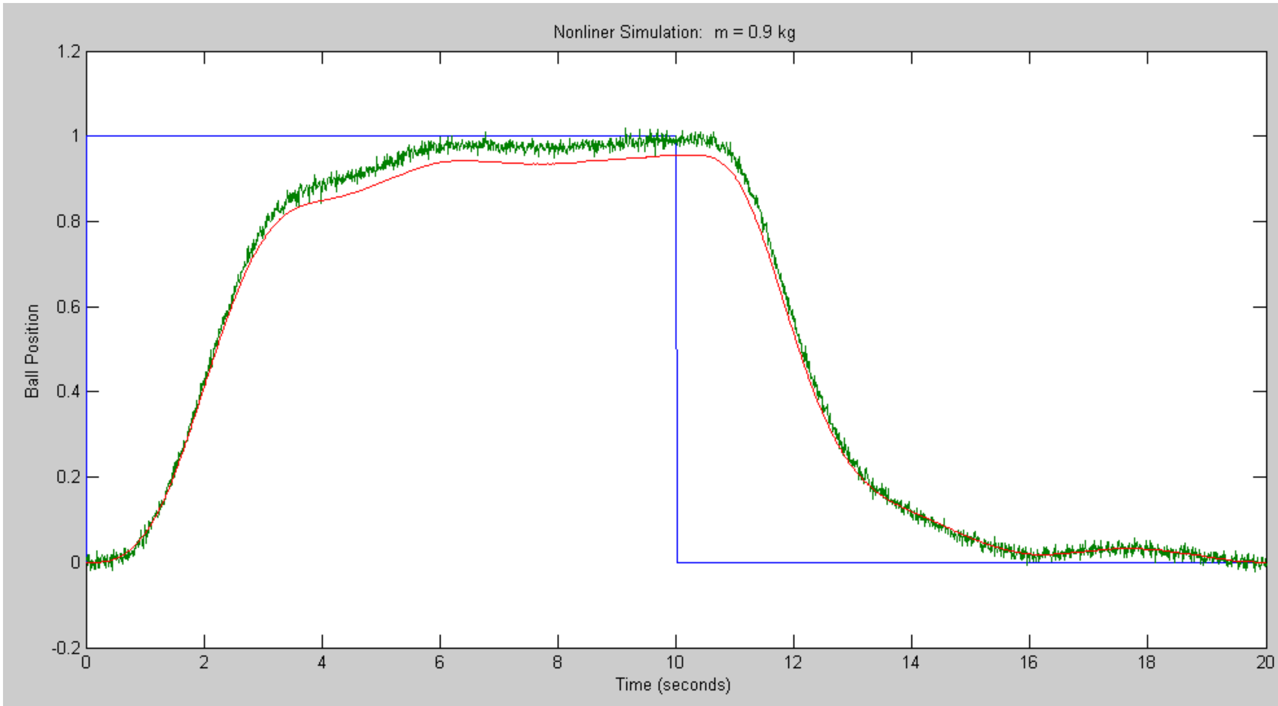

```
>> y = step3(A9, B9, C9, D9, t, X0, [R,Nx,Nq,Nu]);
>> plot(t,y)
```

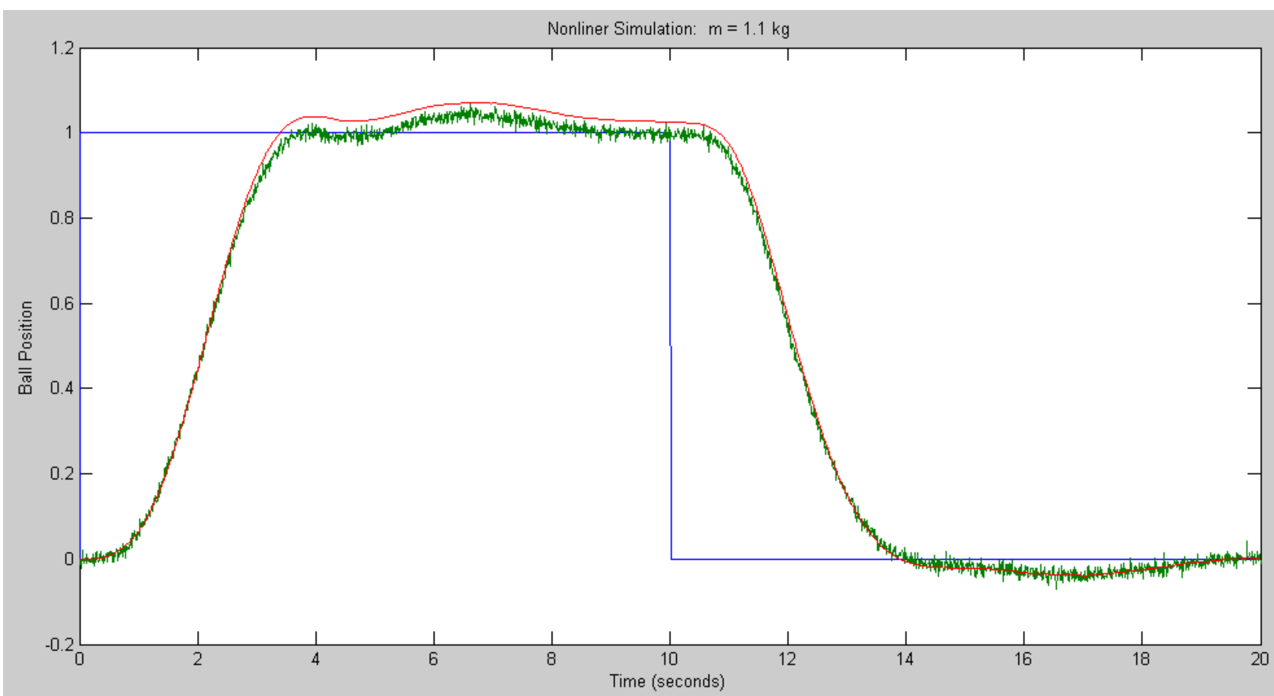


4) Check your full-order observer design with the nonlinear simulation of the ball and beam system.






Nonliner Simulation: m = 0.9 kg
Ball Position
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
2
4
6
8
10
12
14
16
18
20


Nonliner Simulation: m = 1.1 kg
Ball Position
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
2
4
6
8
10
12
14
16
18
20

```
% Ball & Beam System
% Sp 19 Version
% Homework #11 (Kalman Filter)
% m = 1kg
% J = 2 kg m^2

X = [0, 0, 0, 0]';
dt = 0.01;
t = 0;

K5 = lqr(A5, B5, diag([0,0,50,0,100]),1);

Kx = K5(1:4);
Kz = K5(5);
Z = 0;

% Observer

H = [    1.2418   -4.7668
   -0.1907    6.3053
    1.2255  -10.3848
   -1.0236   20.3326
];

A = [0,0,1,0 ; 0,0,0,1 ; 0,-7,0,0 ; -3.27,0,0,0];
B = [0;0;0;0.33];
C = [1,0,0,0;0,1,0,0];
Xo = zeros(4,1);


y = [];

while(t < 20)
 Ref = 1 * (sin(0.1*pi*t) > 0);

 U = -Kz*Z - Kx*Xo;

 % Noise
    Nu = 0.1*randn;
    Nx = 0.01*randn;
    Nq = 0.002*randn;

 dX = BeamDynamics(X, U + Nu);
 dZ = X(1) - Ref;
 dXo = A*Xo + B*U -H*(C*Xo  - ( C*X + [Nx ; Nq] ));

 X = X + dX * dt;
 Z = Z + dZ * dt;
 Xo = Xo + dXo * dt;
 t = t + dt;

 y = [y ; Ref, X(1) + Nx, Xo(1)];

 BeamDisplay(X, Xo, Ref);
 end

t = [1:length(y)]' * dt;


plot(t,y);
xlabel('Time (seconds)');
ylabel('Ball Position');
```