ECE 463/663 - Homework #11

LQR Observers. Due Monday, April 24th
Please submit as a hard copy, email to jacob.glower@ndsu.edu, or submit on BlackBoard

Kalman Filters

Cart and Pendulum (HW #4): The dynamics for a cart and pendulum system with sensor and input noise is as follows

$$s\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -29.4 & 0 & 0 \\ 0 & 26.133 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -0.667 \end{bmatrix} (F + \eta_u)$$

$$y_1 = x + n_x$$

$$y_2 = \theta + n_\theta$$

where there is Gaussian noise at the input and output

$n_u \sim N(0, 0.2^2)$ *mean zero, standard deviation 0.2*

$n_x \sim N(0, 0.05^2)$ *mean zero, standard deviation 0.1*

$n_\theta \sim N(0, 0.01^2)$ *mean zero, standard deviation 0.01*

Problem 1) Use a servo-compensator to force the DC gain to one (i.e. use the servo compensator from homework set #10.

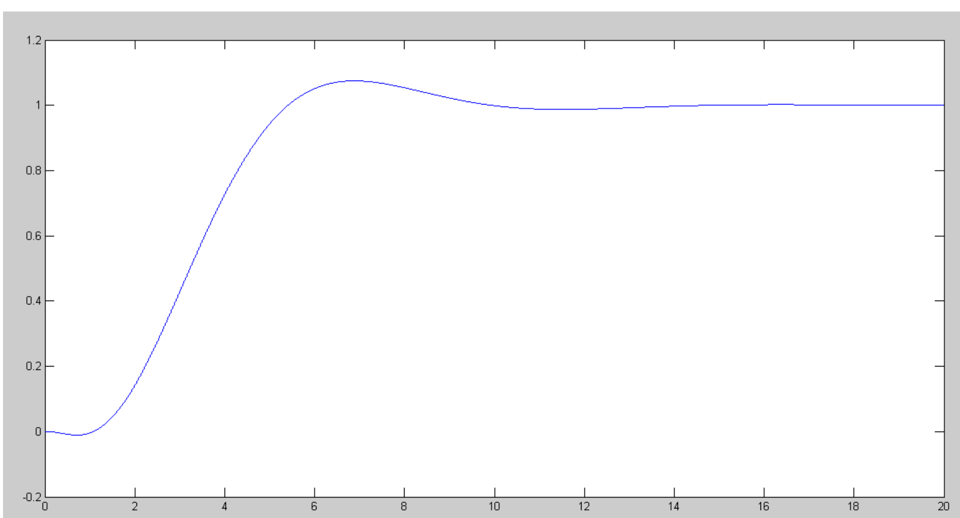
The plant + servo + noise is

$$\begin{bmatrix} sX \\ sZ \end{bmatrix} = \begin{bmatrix} A & 0 \\ C_x & 0 \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} F + \begin{bmatrix} 0 & B & 0 & 0 \\ -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R \\ n_u \\ n_x \\ n_q \end{bmatrix}$$

```
>> A = [0,0,1,0;0,0,0,1;0,-29.4,0,0;0,26.133,0,0]
>> B = [0;0;1;-0.667];
>> C = [1,0,0,0];
>> D = 0;

>> A5 = [A, zeros(4,1);C,0];
>> B5r = [0;0;0;0;-1];
>> B5u = [B;0];
>> B5x = [0;0;0;0;1];
>> B5q = [0;0;0;0;0];
>> C5 = [C,0];

>> K5 = lqr(A5, B5u, diag([0,0,0,0,3]),1);



>> X0 = zeros(5,1);

>> R = 0*t + 1;
>> nu = randn(size(t))*0.2;
>> nx = randn(size(t))*0.05;
>> nq = randn(size(t))*0.01;
```

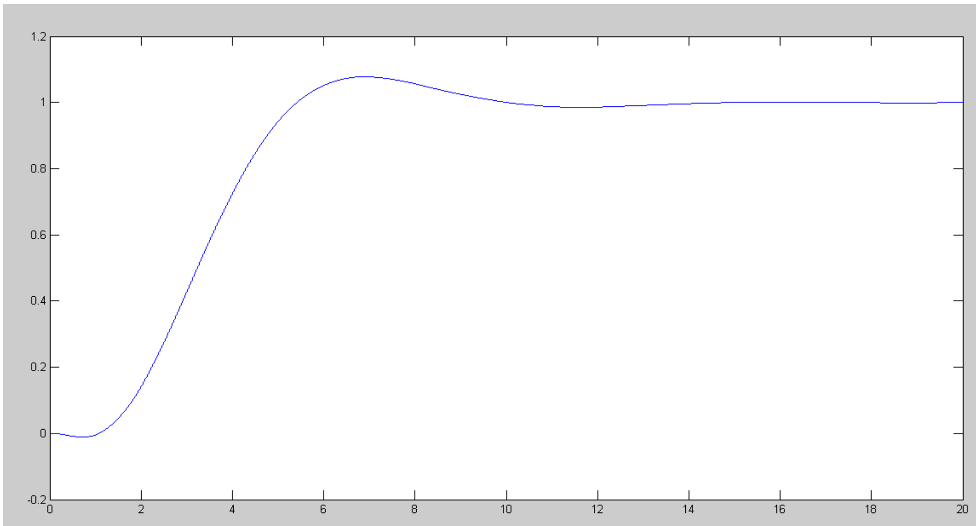
No Noise:

```
>> y = step3(A5-B5u*K5, [B5r,B5u,B5x,B5q],C5,D5,t,X0,[R,nu*0,nx*0,nq*0]);
>> plot(t,y)
```



With Noise

```
>> y = step3(A5-B5u*K5, [B5r,B5u,B5x,B5q],C5,D5,t,X0,[R,nu,nx,nq]);
>> plot(t,y)
```



Problem 2) Design a full-order observer using pole-placement to place the observer poles at

- {-3, -4, -5, -6}
- Simulate the response of the cart with noise added at the input and output.
- Plot the states of the plant and the observer with noise,.

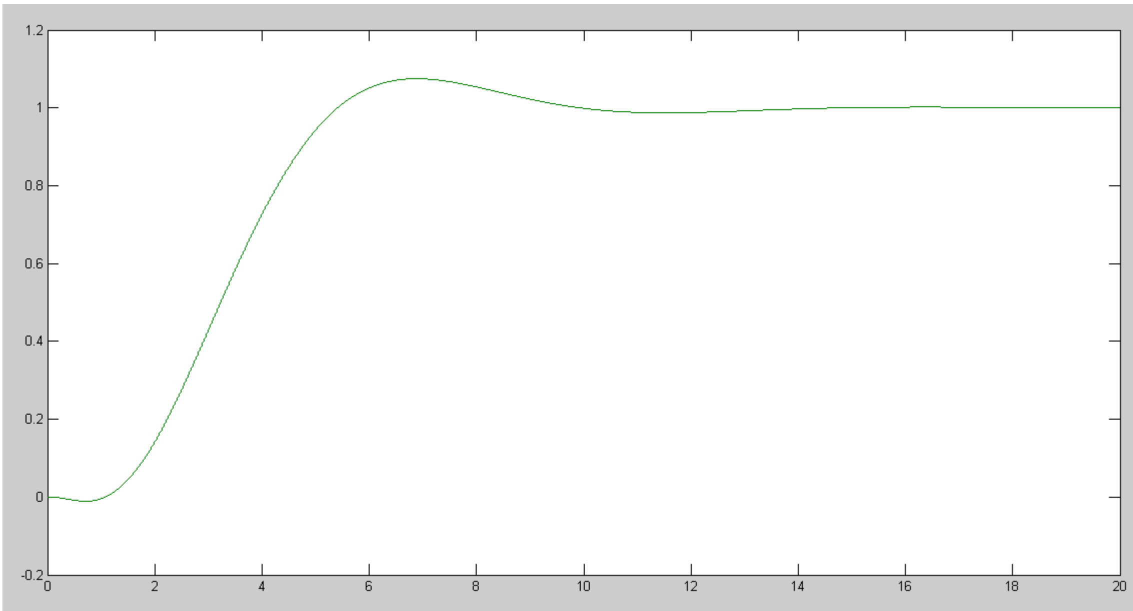
The plant + servo + observer is:

$$\begin{bmatrix} sX \\ sZ \\ sX_e \end{bmatrix} = \begin{bmatrix} A & -BK_z & -BK_x \\ C_x & 0 & 0 \\ HC & -BK_z & A-BK_x-HC \end{bmatrix} \begin{bmatrix} X \\ Z \\ X_e \end{bmatrix} + \begin{bmatrix} 0 & B & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & H_x & H_q \end{bmatrix} \begin{bmatrix} R \\ n_u \\ n_x \\ n_q \end{bmatrix}$$

```
>> Hx = ppl(A', C', [-3,-4,-5,-6])'

   18.0000
  -27.6324
  145.1330
 -141.2504

>> Hq = zeros(4,1)

     0
     0
     0
     0

>> C = [1,0,0,0;0,1,0,0];
>> H = [Hx, Hq];
>> Kx = K5(1:4);
>> Kz = K5(5);

>> A9 = [A, -B*Kz, -B*Kx ; Cx, 0, zeros(1,4) ; H*C, -B*Kz, A-B*Kx-H*C]

>> B9 = [0*B,B,0*B,0*B ; -1,0,0,0 ; zeros(4,1),zeros(4,1),Hx,Hq]

      Ref        nu        nx          nq
         0         0         0         0
         0         0         0         0
         0    1.0000         0         0
         0   -0.6670         0         0
   -1.0000         0         0         0
         0         0   18.0000         0
         0         0  -27.6324         0
         0         0  145.1330         0
         0         0 -141.2504         0


>> C9 = [1,0,0,0,0,0,0,0,0 ; 0,0,0,0,0,1,0,0,0];
>> D9 = [0,0,1,0 ; 0,0,0,0];
```

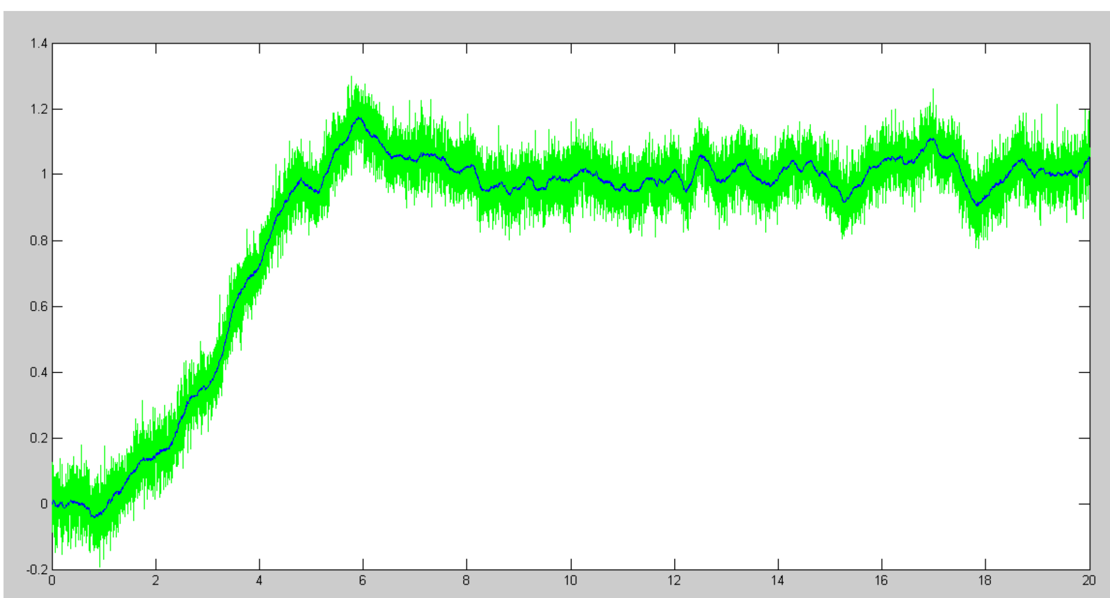
No Noise

```
>> y = step3(A9, B9, C9, D9, t, X0, [R, nu*0, nx*0, nq*0]);
>> plot(t,y)
```



With Noise

```
>> y = step3(A9, B9, C9, D9, t, X0, [R, nu, nx, nq]);
>> plot(t,y(:,1),'g',t,y(:,2),'b')
```



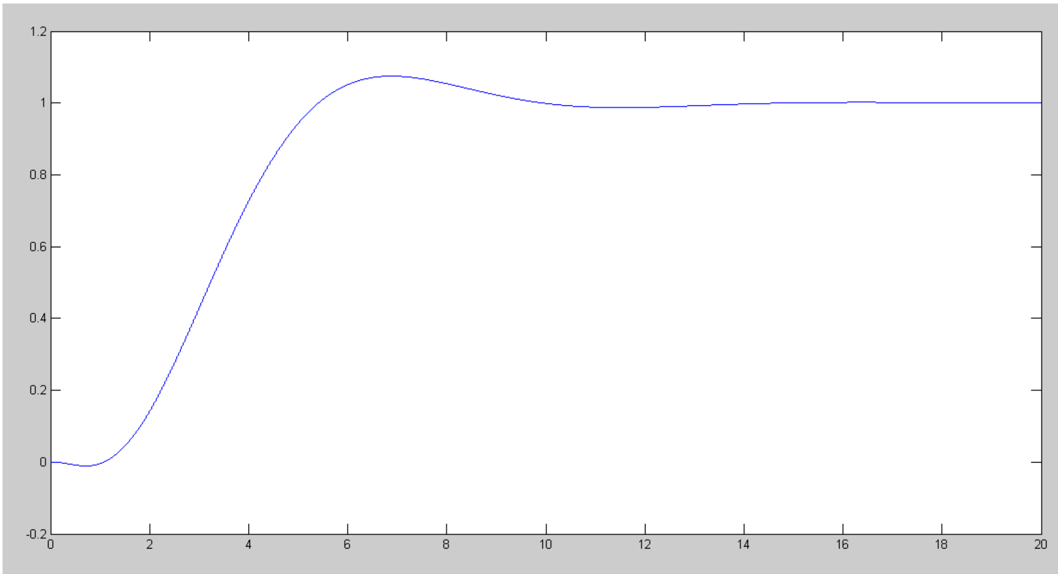
3) Design a Kalman filter (i.e. a full-order observer with a specific Q and R)

- Simulate the response of the cart with noise added at the input and output.
- Plot the states of the plant and the observer with noise,.

```
>> F = B

         0
         0
    1.0000
   -0.6670

>> Q = F*F' * 0.2^2;
>> R = diag([0.05^2, 0.01^2]);
>> H = lqr(A', C', Q, R)'

     Hx         Hq
    1.7591   -9.4592
   -0.3784   10.1757
    3.3367  -60.8009
   -2.0837   53.5619

>> Hx = H(:,1);
>> Hq = H(:,2);
>> eig(A - H*C)

  -0.7013 + 0.6284i
  -0.7013 - 0.6284i
  -5.2661 + 1.3023i
  -5.2661 - 1.3023i

>> A9 = [A, -B*Kz, -B*Kx ; Cx, 0, zeros(1,4) ; H*C, -B*Kz, A-B*Kx-H*C];
>> B9 = [0*B,B,0*B,0*B ; -1,0,0,0 ; zeros(4,1),zeros(4,1),Hx,Hq];
>> C9 = [1,0,0,0,0,0,0,0,0 ; 0,0,0,0,0,1,0,0,0];
>> D9 = [0,0,1,0 ; 0,0,0,0];

>> Ref = 0*t + 1;
```

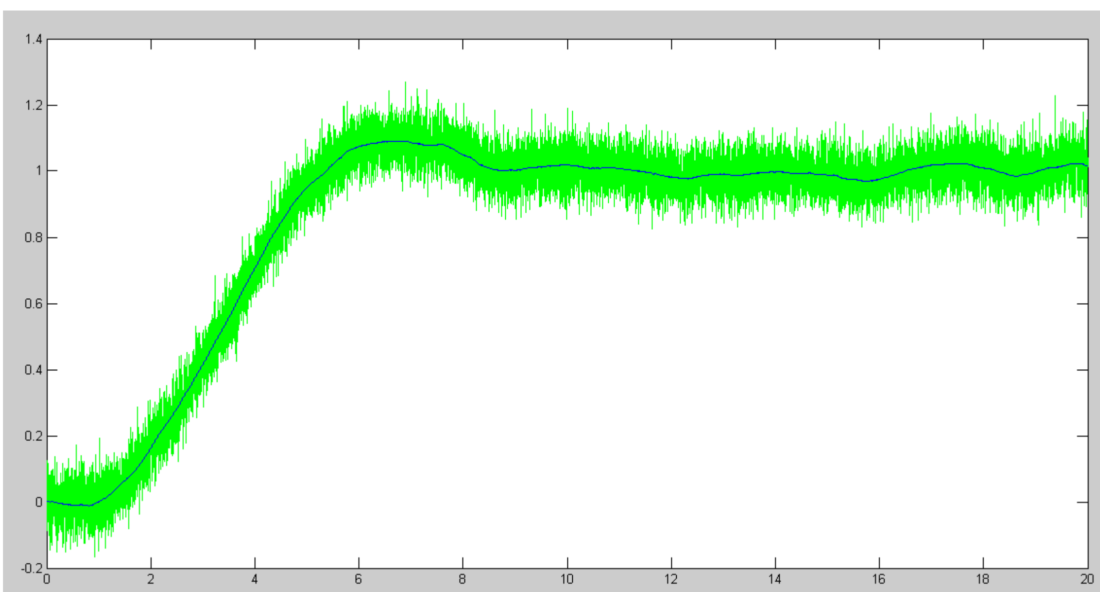
Plant + Servo + Kalman Filter: No Noise

```
>> y = step3(A9, B9, C9, D9, t, X0, [Ref, nu*0, nx*0, nq*0]);
>> plot(t,y(:,1),'g',t,y(:,2),'b')
```



With Noise:

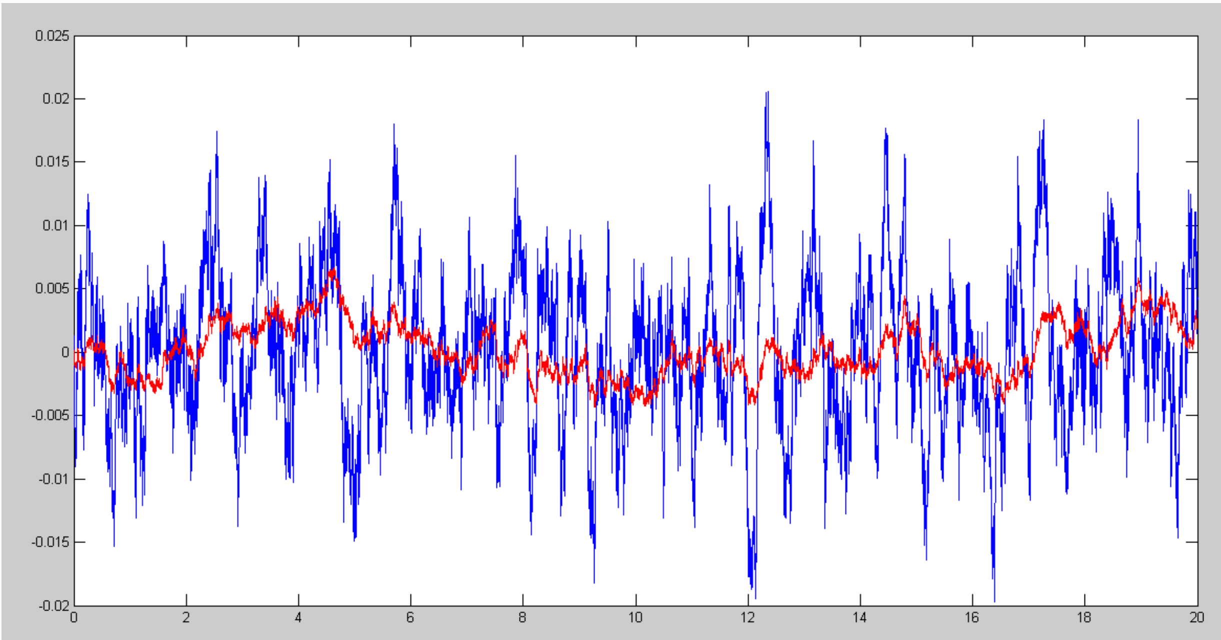
```
>> y = step3(A9, B9, C9, D9, t, X0, [Ref, nu, nx, nq]);
>> plot(t,y(:,1),'g',t,y(:,2),'b')
```



For comparison, if you set Ref = 0, the error in the estimated position with

- H using pole placement (problem #1 - shown in blue), and
- H using LQR (Kalman filter - shown in red)

is shown below. The Kalman filter does a better job estimating the position



Error in position (Xe - X) for H found using pole-placement (blue) and Kalman filter (red)

```
>> std(y0)*1000

    6.0580mm   << standard deviation of the error in Xe: pole placement for H

>> std(yk)*1000

    2.0849mm  << standard deviation in the errror in Xe: Kalman filter
```