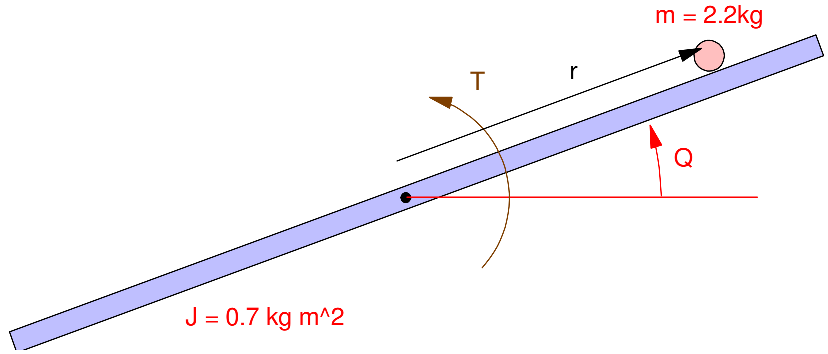
ECE 463/663 - Test #2: Name ________________

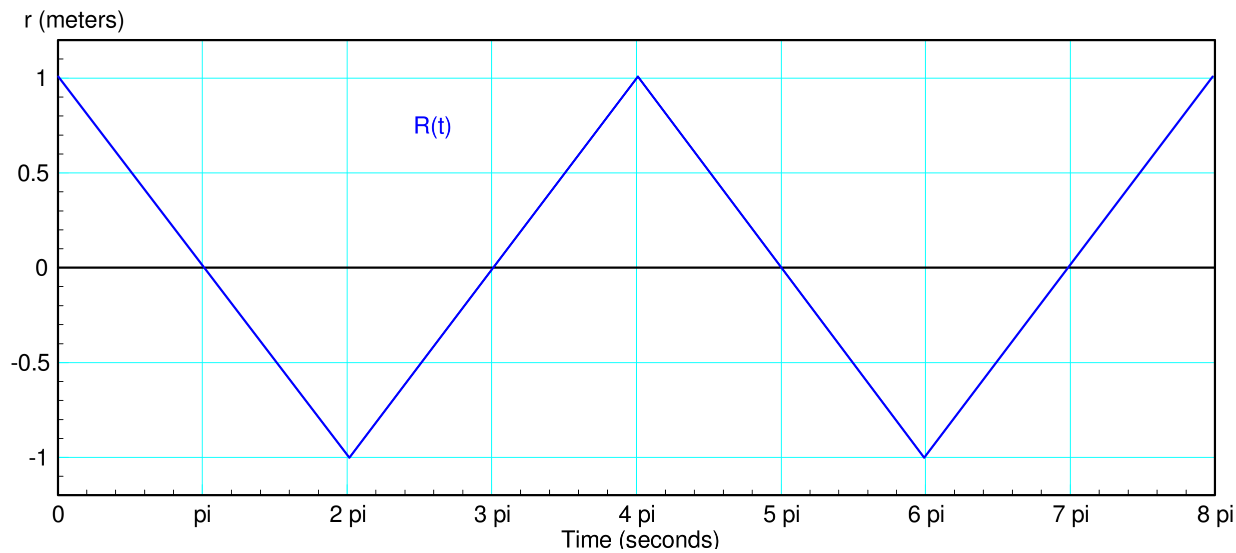
Due midnight Sunday, March 30th. Individual Effort Only (no working in groups)



The linearized dynamics for a ball and beam system (homework #4) are:

$$s\begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -7 & 0 & 0 \\ -7.434 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.345 \end{bmatrix} T$$

The ball is to move back and forth following R(t) from -1m to +1m and back, repeating every 4π seconds:



You can approximate R(t) using the first-three terms in its Fourier series expansion:

$$R(t) = \left(\frac{8}{\pi^2}\right)\cos(0.5t) + \left(\frac{8}{9\pi^2}\right)\cos(1.5t) + \left(\frac{8}{25\pi^2}\right)\cos(2.5t)$$

Problem 1) Controller Design (50 points)

Design a feedback controller to track R(t) (pick one method)

- (30 points partial credit) Using full-state feedback (no servo compensator)

  $$U = K_r R - K_x X$$

- (40 points partial credit) Using a servo compensator that tracks the 0.5 rad/sec term for R(t)

  $$U = -K_z Z - K_x X$$

- (50 points full credit) Using a servo compensator that tracks the {0.5, 1.5, 2.5} rad/sec terms for R(t)

  $$U = -K_z Z - K_x X$$

Provide in your solutions

- A block diagram of your plant and controller
- Calculations (Matlab code and results) for computing your feedback control gains
- The response of the linear system to R(t),
- The response of the nonlinear simulation to R(t) with your control law with
  - m = 2.2kg (nominal case)
  - m = 2.5kg (disturbance)
- The Matlab code for the main routine in the nonlinear simulation, including your control law.

Start with a servo-compensator with poles at {+/-0.5i, +/- 1.5i, +/- 2.5i}

$$sZ = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & -1.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.5 \\ 0 & 0 & 0 & 0 & -2.5 & 0 \end{bmatrix} Z + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} (R - y)$$

Use full-state feedback to stabilize the system

poles = $\{-0.3, -2.68, -0.3 \pm j2.68, -0.3 \pm j0.5, -0.3 \pm j1.5, -0.3 \pm j2.5\}$

```
% ECE 463/663 Test2
% Problem 1

% Plant
A = [0,0,1,0;0,0,0,1;0,-7,0,0;-7.434,0,0,0];
B = [0;0;0;0.345];
C = [1,0,0,0];

Az = [0,0.5,0,0,0,0;-0.5,0,0,0,0,0];
Az = [Az;0,0,0,1.5,0,0;0,0,-1.5,0,0,0];
Az = [Az;0,0,0,0,0,2.5;0,0,0,0,-2.5,0];

Bz = ones(6,1);
```

Create the augmented system (plant & servo compensator)

```
A10 = [A, zeros(4,6) ; Bz*C,Az]
B10u = [B; 0*Bz]
B10r = [0*B; -Bz]
C10 = [1,0,0,0,0,0,0,0,0,0];

        0         0    1.0000         0  |         0         0         0         0         0         0
        0         0         0    1.0000  |         0         0         0         0         0         0
        0   -7.0000         0         0  |         0         0         0         0         0         0
  -7.4340         0         0         0  |         0         0         0         0         0         0
----------------------------------------------------------------------------------------------------------
   1.0000         0         0         0  |         0    0.5000         0         0         0         0
   1.0000         0         0         0  |   -0.5000         0         0         0         0         0
   1.0000         0         0         0  |         0         0         0    1.5000         0         0
   1.0000         0         0         0  |         0         0   -1.5000         0         0         0
   1.0000         0         0         0  |         0         0         0         0         0    2.5000
   1.0000         0         0         0  |         0         0         0         0   -2.5000         0
```

Find the feedback gains using pole-placement

```
P = eig(Az) - 0.3;
P = [P ; -0.3; -2.68; -0.3+2.68i; -0.3-2.68i];

K10 = ppl(A10, B10u, P)'
Kx = K10(1:4)
```

`Kx =  -36.1290   51.1838  -16.5768   15.5942`

```
Kz = K10(5:10)
```

`Kz =  0.8001   -2.0479    3.8109   -2.7128    3.1046      0.4488`

Check: is the closed-loop system stable?

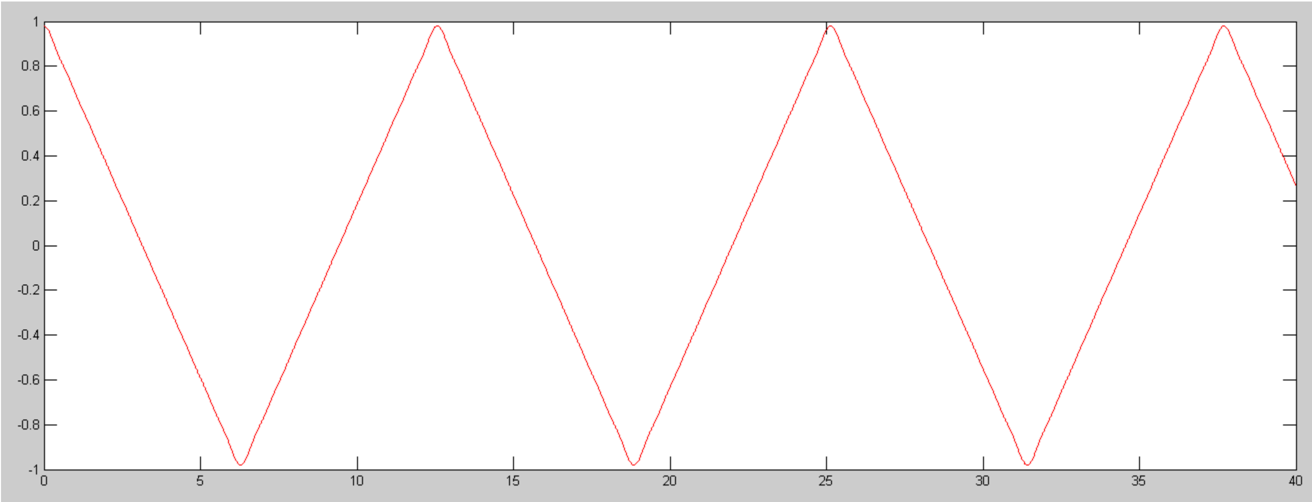
```
eig(A10 - B10u*K10)

  -2.6800
  -0.3000 + 2.6800i
  -0.3000 - 2.6800i
  -0.3000 + 2.5000i
  -0.3000 - 2.5000i
  -0.3000 + 1.5000i
  -0.3000 - 1.5000i
  -0.3000
  -0.3000 + 0.5000i
  -0.3000 - 0.5000i
```

Simulate the response to r(t). I had problems putting r(t) into Matlab, so I used the Fourier series expansion out to ten terms:

$$r(t) = \sum_{\text{n odd}} \left(\frac{8}{n^2\pi^2}\right) \cos(0.5nt)$$

```
t = [0:0.01:40]';
dt = 0.01;
t = [0:dt:40]';
R = 0*t;
R(1) = 1;
dR = 1/pi * dt;
for i=2:length(t)
    R(i) = R(i-1) + dR;
    if(abs(R(i)) > 1) dR = -dR; end
end
```

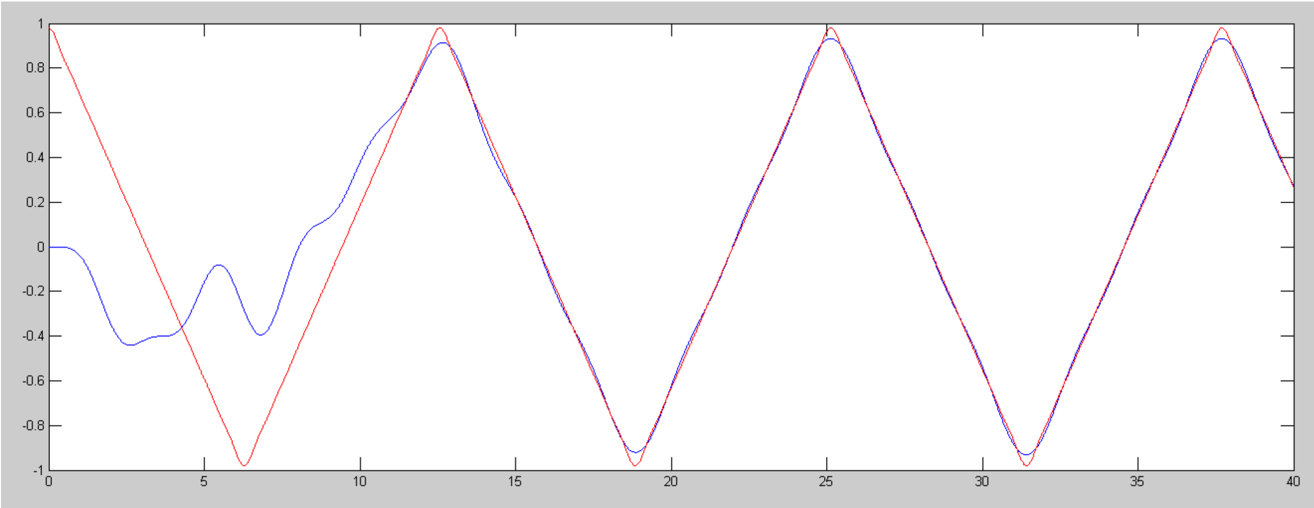
```
plot(t,R)
```



r(t): a triangle wave going from -1 to +1

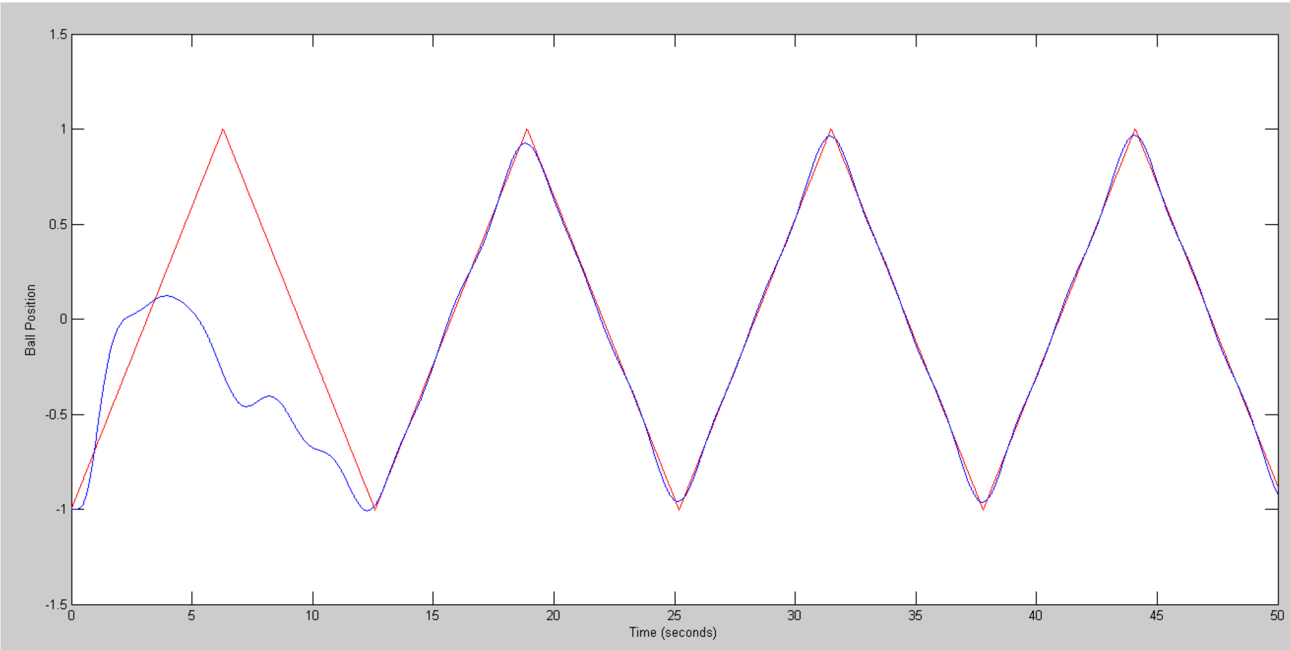
Simulate the response of the plant to r(t)

```
X0 = zeros(10,1);
y = step3(A10-B10u*K10, B10r, C10, 0, t, X0, R);
plot(t,y,'b',t,R,'r')
```



not perfect, but as good as you can do with only the first three harmonics

Nonlinear System Response:



Code:

```
% Ball & Beam System
% ECE 463/663 Test #2 (Sp25)

X = [-1,0,0,0]';
dt = 0.01;
t = 0;
n = 0;
y = [];

% Plant
A = [0,0,1,0;0,0,0,1;0,-7,0,0;-7.434,0,0,0];
B = [0;0;0;0.345];
C = [1,0,0,0];

% Servo Compensator
Az = [0,0.5,0,0,0,0;-0.5,0,0,0,0,0];
Az = [Az;0,0,0,1.5,0,0;0,0,-1.5,0,0,0];
Az = [Az;0,0,0,0,0,2.5;0,0,0,0,-2.5,0];
Bz = ones(6,1);

% Control Law
A10 = [A, zeros(4,6) ; Bz*C,Az]
B10u = [B; 0*Bz]
B10r = [0*B; -Bz]
C10 = [1,0,0,0,0,0,0,0,0,0];

P = eig(Az) - 0.3;
P = [P ; -0.3; -2.68; -0.3+2.68i; -0.3-2.68i];

K10 = ppl(A10, B10u, P)
Kx = K10(1:4);
Kz = K10(5:10);
Z = zeros(6,1);
```

```
dR = dt/pi;
Ref = -1;

while(t < 49)
 Ref = Ref + dR;
 if(Ref > 1) dR = -dR; end
 if(Ref < -1) dR = -dR; end

 U = -Kz*Z - Kx*X;
 Y = X(1);

 dX = BeamDynamics(X, U);
 dZ = Az*Z + Bz*(Y - Ref);

 X = X + dX * dt;
 Z = Z + dZ * dt;
 Xe = X;
 t = t + dt;

 y = [y ; Ref, X(1)];
 n = mod(n+1,5);
 if(n == 0)
    BeamDisplay3(X, Xe, Ref);
 end
 end

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


plot(t,y(:,1),'r',t,y(:,2),'b');
xlabel('Time (seconds)');
ylabel('Ball Position');
```

Problem 2: Observer Design (50 points)

Assume you can measure position (r) and angle (θ). Design a full-order observer assuming (pick one)

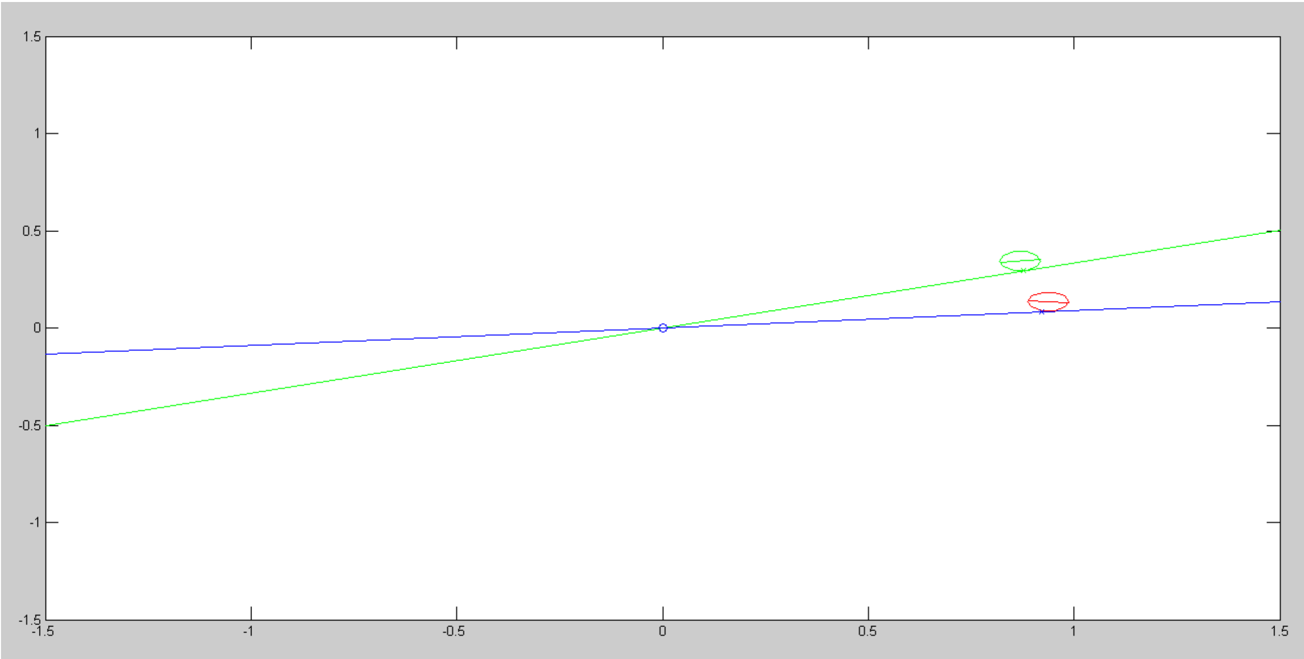
- (30 points partial credit) No disturbance
- (40 points partial credit) An input disturbance at 0.5 rad/sec,
- (50 points full credit) An input disturbance at {0.5, 1.5, 2.5} rad/sec

Start with an observer with poles at {-3, -4, -5, -6}. This tracks when m = 2.2kg. When m = 2.5kg, however, the observer no longer tracks due to the disturbance:

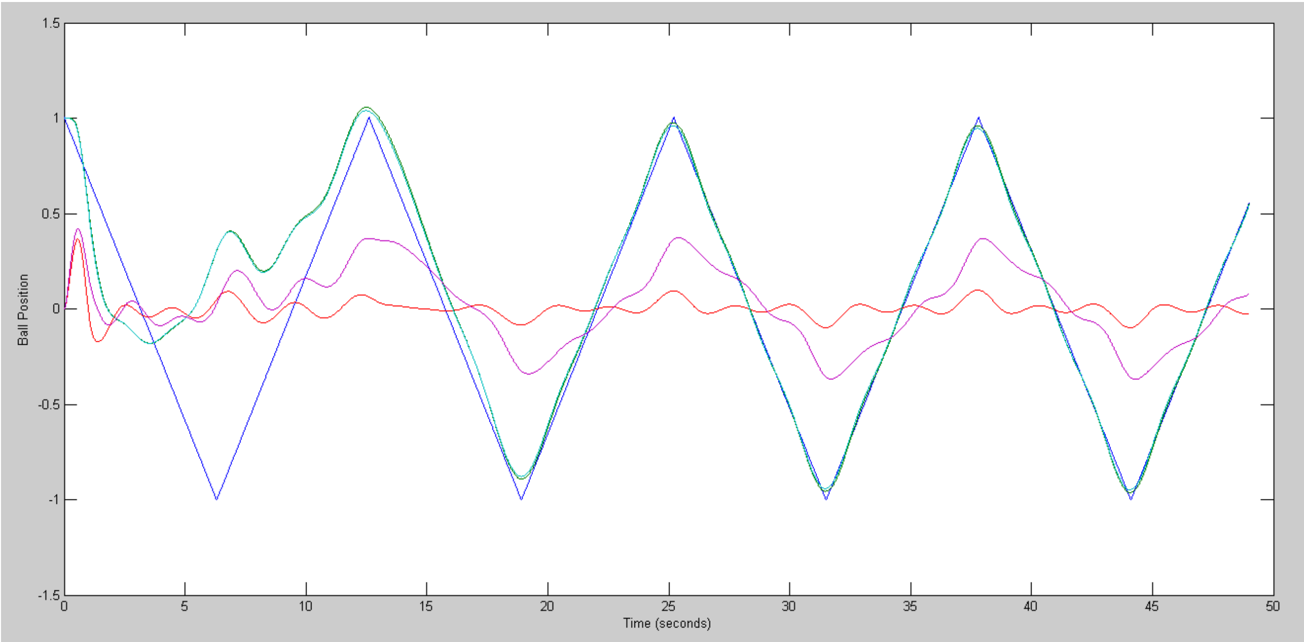
Observer Dynamics (full-order observer)

```
>> Ae

         0         0    1.0000         0
         0         0         0    1.0000
         0   -7.0000         0         0
   -7.4340         0         0         0

>> Be

         0
         0
         0
    0.3450

>> Ce

Ce =    1     0     0     0

>> H

   18.0000
  -48.8571
  119.0000
  -58.8626

>> eig(Ae - H*Ce)

   -6.0000
   -5.0000
   -4.0000
   -3.0000
```

Response of the nonlinear simulation:



Observer with m = 2.5kg (no disturbance modeled)

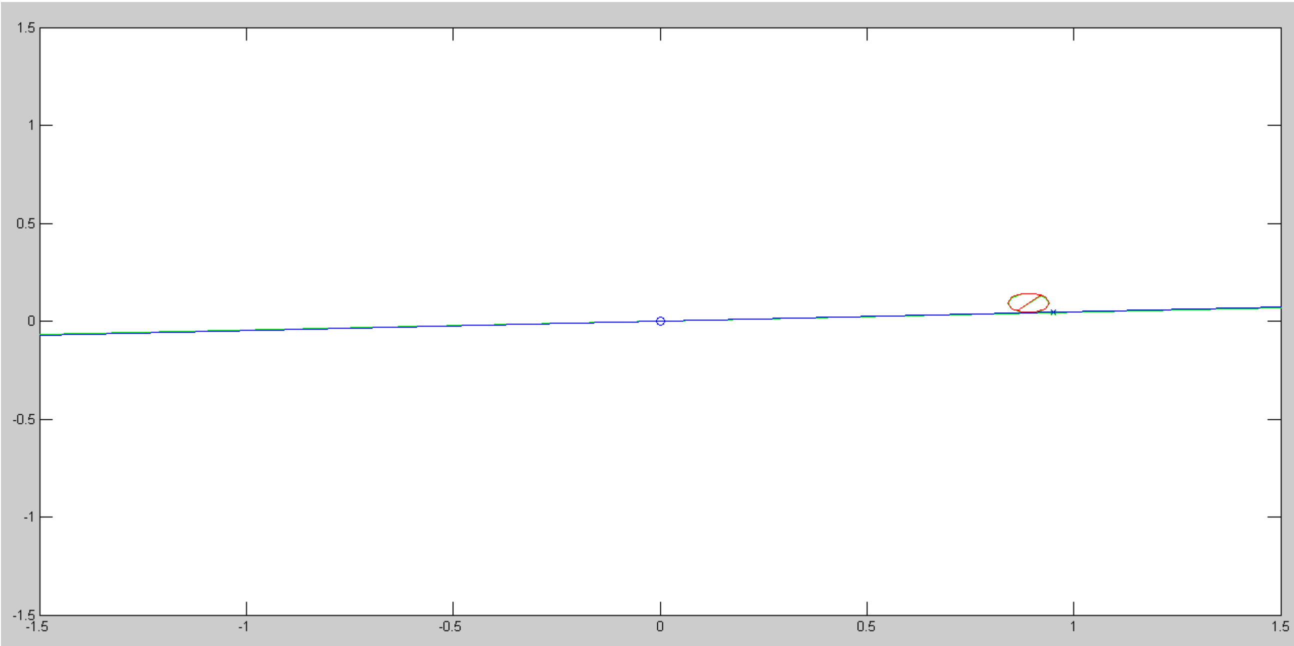


Nonlinear Response when m = 2.5kg
Blue = Ref, Green = Position, Teal = Estimated Position
Red = Angle, Magenta = Estimated Position

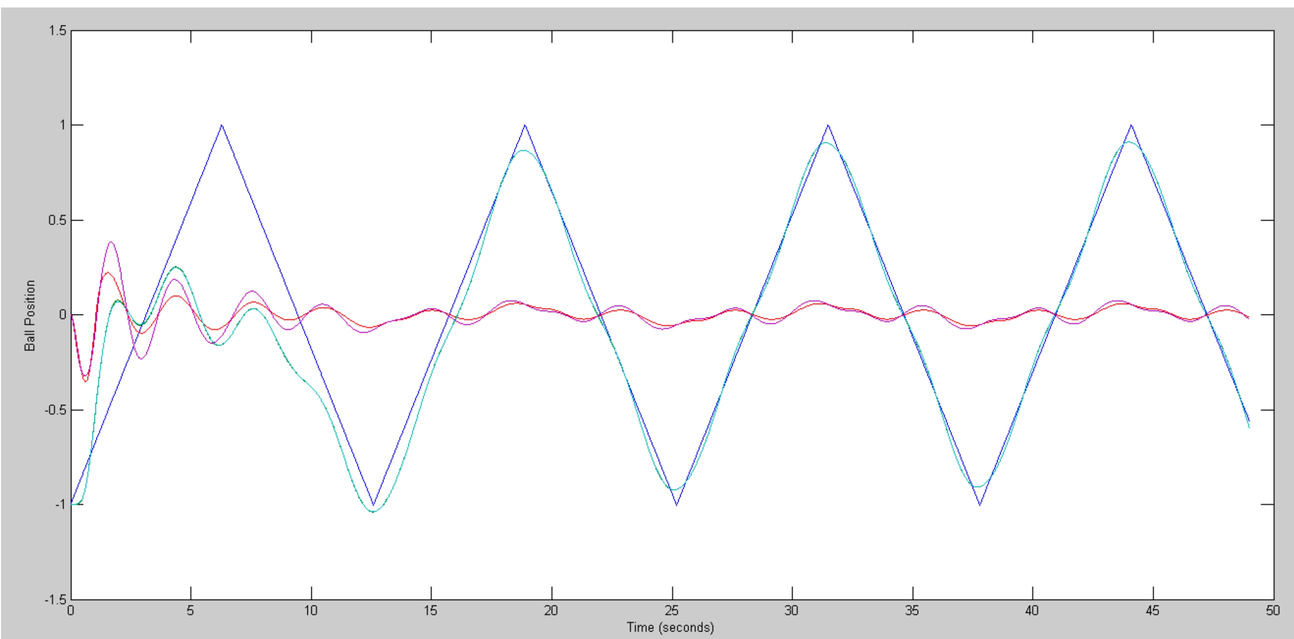
Trying again: add a disturbance at 0.5 rad/sec to the observer

```
>> Ae

         0         0    1.0000         0         0         0
         0         0         0    1.0000         0         0
         0   -7.0000         0         0         0         0
   -7.4340         0         0         0    0.3450         0
         0         0         0         0         0    0.5000
         0         0         0         0   -0.5000         0

>> Be

         0
         0
         0
    0.3450
         0
         0

>> Ce

     1     0     0     0     0     0

>> H

  1.0e+003 *

    0.0210
   -0.1211
    0.1832
   -0.3188
   -1.1924
   -1.0279

>> eig(Ae-H*Ce)

   -4.0000
   -3.8000
   -3.6000
   -3.4000
   -3.2000
   -3.0000

>>
```

Nonlinear System Response



With the disturbance at 0.5 rad/sec modeled, the observer tracks much better

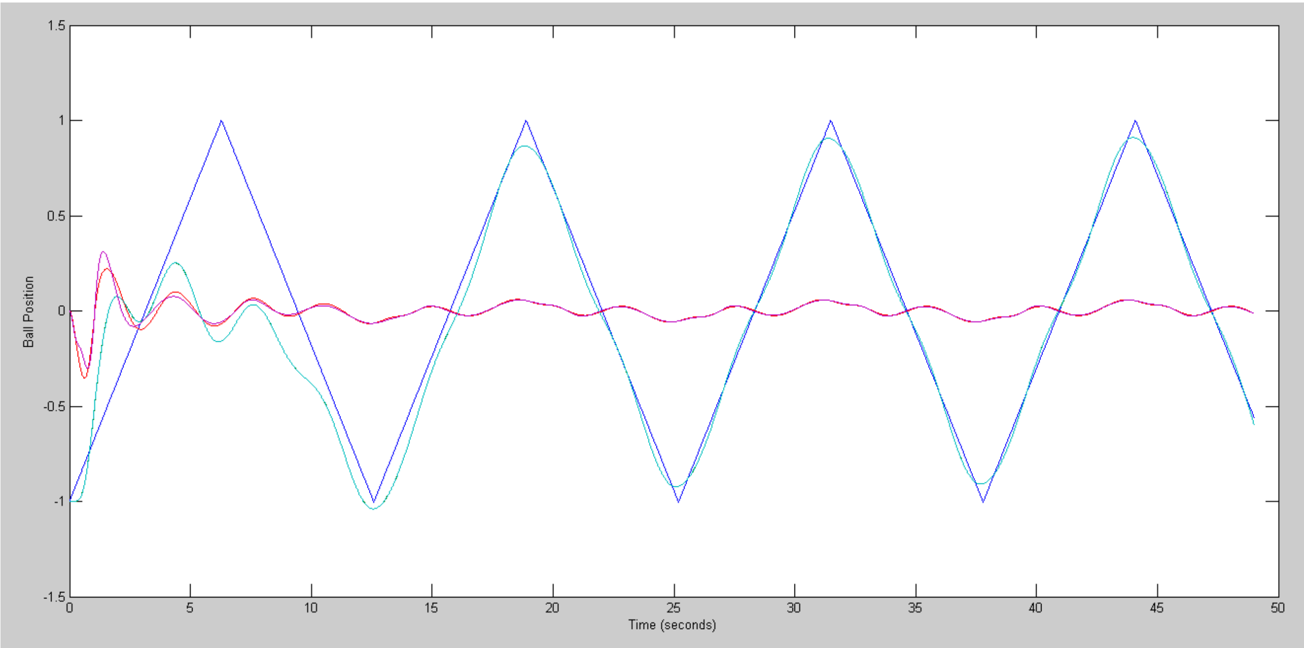


Nonlinear Response when m = 2.5kg
Blue = Ref, Green = Position, Teal = Estimated Position
Red = Angle, Magenta = Estimated Position

Adding disturbances at {0.5 rad/sec, 1.5 rad/sec} to the observer

```
>> Ae

         0         0    1.0000         0         0         0         0         0
         0         0         0    1.0000         0         0         0         0
         0   -7.0000         0         0         0         0         0         0
   -7.4340         0         0         0    0.3450         0    0.3450         0
         0         0         0         0         0    0.5000         0         0
         0         0         0         0   -0.5000         0         0         0
         0         0         0         0         0         0         0    1.5000
         0         0         0         0         0         0   -1.5000         0

>> Be

         0
         0
         0
    0.3450
         0
         0
         0
         0

>> Ce

     1     0     0     0     0     0     0     0

>> H

  1.0e+004 *

    0.0030
   -0.0392
    0.0380
   -0.1721
   -1.3079
   -0.6806
    0.0197
   -0.8578

>> eig(Ae-H*Ce)

   -4.4000
   -4.2000
   -4.0000
   -3.8000
   -3.6000
   -3.4000
   -3.2000
   -3.0000
```

Nonlinear System Response



Response when m = 2.5kg
Blue = Ref, Green = Position, Teal = Estimated Position
Red = Angle, Magenta = Estimated Position

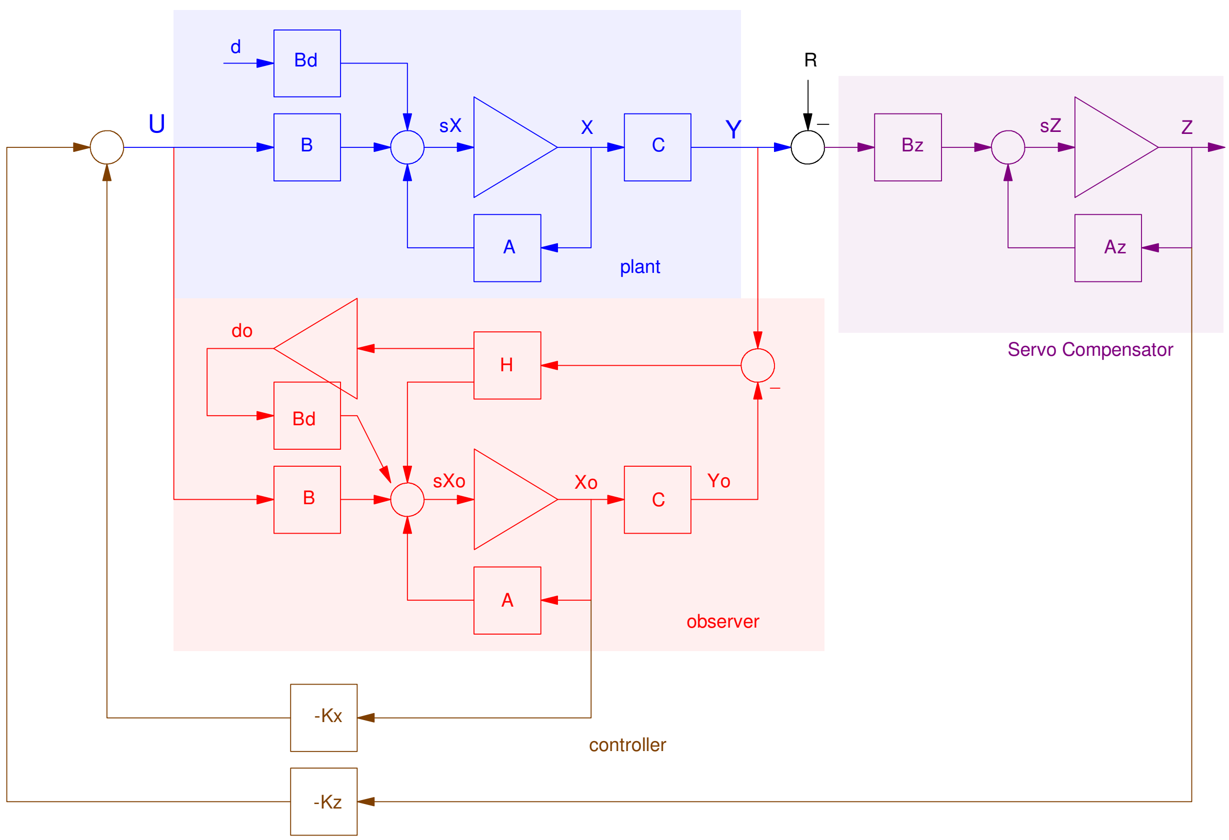
The final system is 18th-order (!)

- 4 poles for the plant
- 6 poles for the servo compensator
- 4 poles for the full-order observer
- 4 poles for the disturabance

Final Code:

```
% Ball & Beam System
% ECE 463/663 Test #2 (Sp25)

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

% Plant
A = [0,0,1,0;0,0,0,1;0,-7,0,0;-7.434,0,0,0];
B = [0;0;0;0.345];
C = [1,0,0,0];

% Servo Compensator
Az = [0,0.5,0,0,0,0;-0.5,0,0,0,0,0];
Az = [Az;0,0,0,1.5,0,0;0,0,-1.5,0,0,0];
Az = [Az;0,0,0,0,0,2.5;0,0,0,0,-2.5,0];

Bz = ones(6,1);

% Control Law
A10 = [A, zeros(4,6) ; Bz*C,Az]
B10u = [B; 0*Bz]
B10r = [0*B; -Bz]
C10 = [1,0,0,0,0,0,0,0,0,0];

P = eig(Az) - 0.5;
P = [P ; -0.5; -2.68; -0.5+2.68i; -0.5-2.68i];

K10 = ppl(A10, B10u, P);
Kx = K10(1:4);
Kz = K10(5:10);
Z = zeros(6,1);

% Full-Order Observer
Ad = [0,0.5,0,0;-0.5,0,0,0;0,0,0,1.5;0,0,-1.5,0];
Cd = [1,0,1,0];

% Input disturbance at 0.5 and 1.5 rad/sec
Ae = [A, B*Cd; zeros(4,4), Ad];
Be = [B; zeros(4,1)];
Ce = [C, 0*Cd];
H = ppl(Ae', Ce', [-3, -3.2, -3.4, -3.6, -3.8, -4, -4.2, -4.4])';
Xe = [X ; zeros(4,1)];

dR = dt/pi;
Ref = 0;

% main loop

while((t < 49) & (abs(X(1))<2))
 Ref = Ref + dR;
 if(Ref > 1) dR = -dR; end
 if(Ref < -1) dR = -dR; end

 U = -Kz*Z - Kx*X;

 Y = X(1);

 dX = BeamDynamics(X, U);
 dZ = Az*Z + Bz*(Y - Ref);
 dXe = Ae*Xe + Be*U + H*(C*X - Ce*Xe);

 X = X + dX * dt;
```

```
    Z = Z + dZ * dt;
    Xe = Xe + dXe * dt;
    t = t + dt;

    y = [y ; Ref, X(1), X(2), Xe(1), Xe(2)];
    n = mod(n+1,5);
    if(n == 0)
       BeamDisplay3(X, Xe, Ref);
    end
    end

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


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



Block diagram for the Plant, Servo Compensator, Disturbance, Observer, and Full-State Feedback

notes:

- For the nonlinear system, Kx feeds back the actual states
- The system is unstable if you use the estimated states due to the nonlinearities in the plant
  - beam inertia varies from 0.2 to 21.7 kg m2 as the ball goes from r=0 to r=1
- If you change the beam inertia to 7 and redo the entire design, then the observer states could be used
  - beam inertia now varies from 7 to 28.5 kg m2 as the ball goes from r=0 to r=1
  - feedback is able to handle this

Starting Code

```
% Ball & Beam System
% Test #2 (Spring 2025)

X = [-1,0,0,0]';
dt = 0.01;
t = 0;
n = 0;
y = [];

% Plant
A = [0,0,1,0;0,0,0,1;0,-7,0,0;-7.434,0,0,0];
B = [0;0;0;0.345];
C = [1,0,0,0];

% Control Law  (needs to change)
Kx = [-31   101   -21    29];
Kr = -21;

% Full-Order Observer (needs to change)
Ae = A;
Be = B;
Ce = C;
H = [0 0 0 0]';
Xe = X;

% Ramp Input
dR = dt/pi;
Ref = -1;

while(t < 40)
 Ref = Ref + dR;
 if(abs(Ref) > 1) dR = -dR; end

 U = Kr*Ref - Kx*X;

 dX = BeamDynamics(X, U);
 dXe = Ae*Xe + Be*U + H*(C*X - Ce*Xe);

 X = X + dX * dt;
% Observer (cheating for now)
 Xe = X + [0.1,0,0,0]';
 t = t + dt;

 y = [y ; Ref, X(1)];
 n = mod(n+1,5);
 if(n == 0)
    BeamDisplay3(X, Xe, Ref);
 end
 end

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


plot(t,y(:,1),'r',t,y(:,2),'b');
xlabel('Time (seconds)');
ylabel('Ball Position');
```