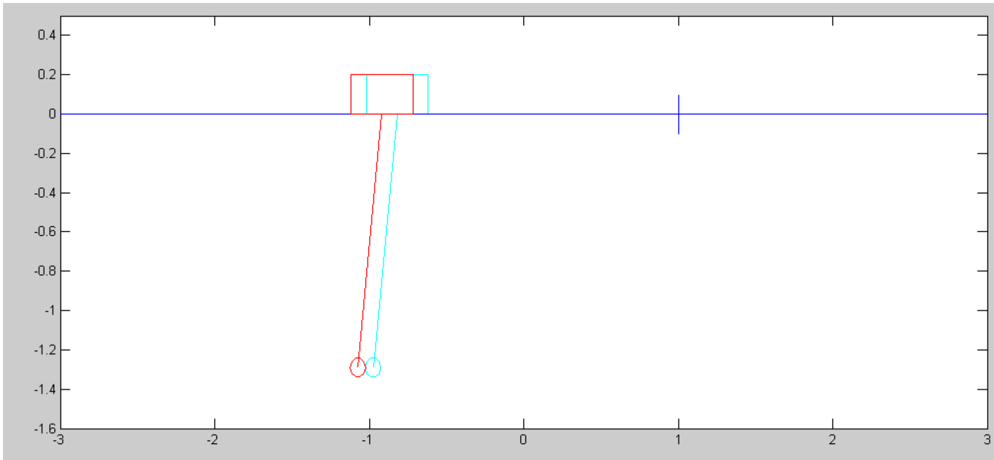
ECE 463/663 - Test #2: Name ________________

Due midnight Sunday, March 24th. Individual Effort Only (no working in groups)



The linearized dynamics for a gantry system (homework #4) are:

$$s\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2.45 & 0 & 0 \\ 0 & -9.42 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.25 \\ -0.1923 \end{bmatrix} (F + d)$$

C Level (max 80 points)

Design a feedback control law for the gantry system assuming

- All states are measured (no observer is needed)
- A sinusoidal set point (R(t) = sin(0.5t)), and
- A constant disturbance (d(t) = 1)

Input the dynamics into Matlab

```
>> A = [0,0,1,0;0,0,0,1;0,2.45,0,0;0,-9.42,0,0]

         0         0    1.0000         0
         0         0         0    1.0000
         0    2.4500         0         0
         0   -9.4200         0         0

>> B = [0;0;0.25;-0.1923]

         0
         0
    0.2500
   -0.1923

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

Add a servo compensator with poles at {0, +j0.5, -j0.5}

```
>> Az = [0,0.5,0;-0.5,0,0;0,0,0]

         0    0.5000         0
   -0.5000         0         0
         0         0         0

>> eig(Az)

        0 + 0.5000i
        0 - 0.5000i
        0

>> Bz = [1;1;1];
```

Put together the augmented system

$$\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} A & 0 \\ B_z C & A_z \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U$$

```
>> A7 = [A, zeros(4,3) ; Bz*C, Az]

         0         0    1.0000         0 |         0         0         0
         0         0         0    1.0000 |         0         0         0
         0    2.4500         0         0 |         0         0         0
         0   -9.4200         0         0 |         0         0         0
---------------------------------------------------------------------------
    1.0000         0         0         0 |         0    0.5000         0
    1.0000         0         0         0 |   -0.5000         0         0
    1.0000         0         0         0 |         0         0         0
```

```
>> B7u = [B; zeros(3,1)]

         0
         0
    0.2500
   -0.1923
         0
         0
         0
```

Find stabilizing feedback gains. Try {-1,-2,-3,-4,-5,-6,-7}

```
>> K7 = ppl(A7, B7u, [-1,-2,-3,-4,-5,-6,-7])

K7 =

  1.0e+004 *

    0.2720    0.1912    0.0164    0.0068    0.4193   -0.8320    1.0701
```

uff-da - too big. Try placing the poles closer to the open-loop poles, shifted left by 0.5:

```
>> P = eig(A7) - 0.5

  -0.5000
  -0.5000 + 3.0692i
  -0.5000 - 3.0692i
  -0.5000
  -0.5000 + 0.5000i
  -0.5000 - 0.5000i
  -0.5000

>> K7 = ppl(A7, B7u, P)

K7 =   13.3966   -9.8848   14.0146    0.0190    1.1557    2.6546    1.2833
```

much better. This defines Kx and Kz:

```
>> Kx = K7(1:4)
```

Kx =   13.3966   -9.8848   14.0146   0.0190

```
>> Kz = K7(5:7)
```

Kz =   1.1557   2.6546   1.2833

Validate your feedback control law on the linear system

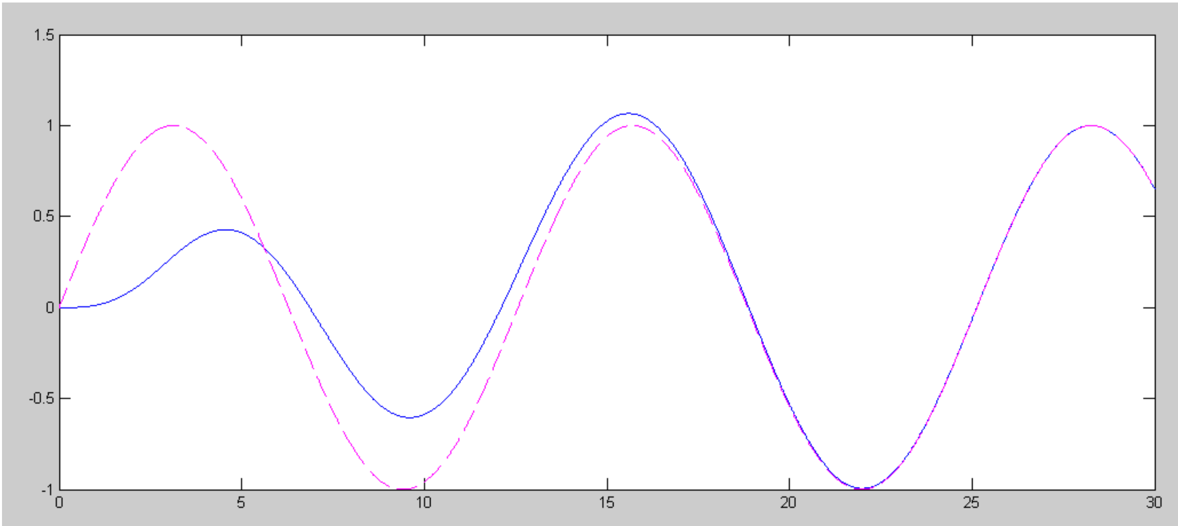
The closed-loop system is:

$$s\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} A - BK_x & -BK_z \\ B_z C & A_z \end{bmatrix}\begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} d + \begin{bmatrix} 0 \\ B_z \end{bmatrix} R$$

Plotting the responses in Matlab

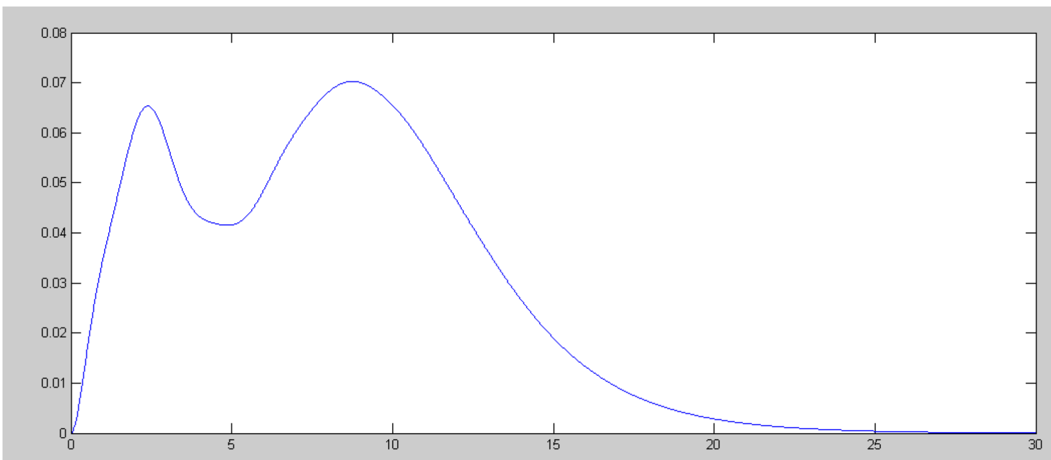
```
>> A7 = [A-B*Kx, -B*Kz ; Bz*C, Az]

         0         0    1.0000         0         0         0         0
         0         0         0    1.0000         0         0         0
   -3.3491    4.9212   -3.5036   -0.0047   -0.2889   -0.6637   -0.3208
    2.5762  -11.3209    2.6950    0.0036    0.2222    0.5105    0.2468
    1.0000         0         0         0         0    0.5000         0
    1.0000         0         0         0   -0.5000         0         0
    1.0000         0         0         0         0         0         0

>> B7d = [B ; 0*Bz]

         0
         0
    0.2500
   -0.1923
         0
         0
         0

>> B7r = [0*B; -Bz]

     0
     0
     0
     0
    -1
    -1
    -1

>> C7 = [1,0,0,0,0,0,0];
>> D7 = 0;
>> X0 = zeros(7,1);
>> t = [0:0.01:30]';
>> d = 0*t+1;
>> R = sin(0.5*t);
```

```
>> y = step3(A7,B7r,C7,D7,t,X0,R);
>> plot(t,y,'b',t,R,'m--')
```



The control law tracks a sinusoidal setpoint

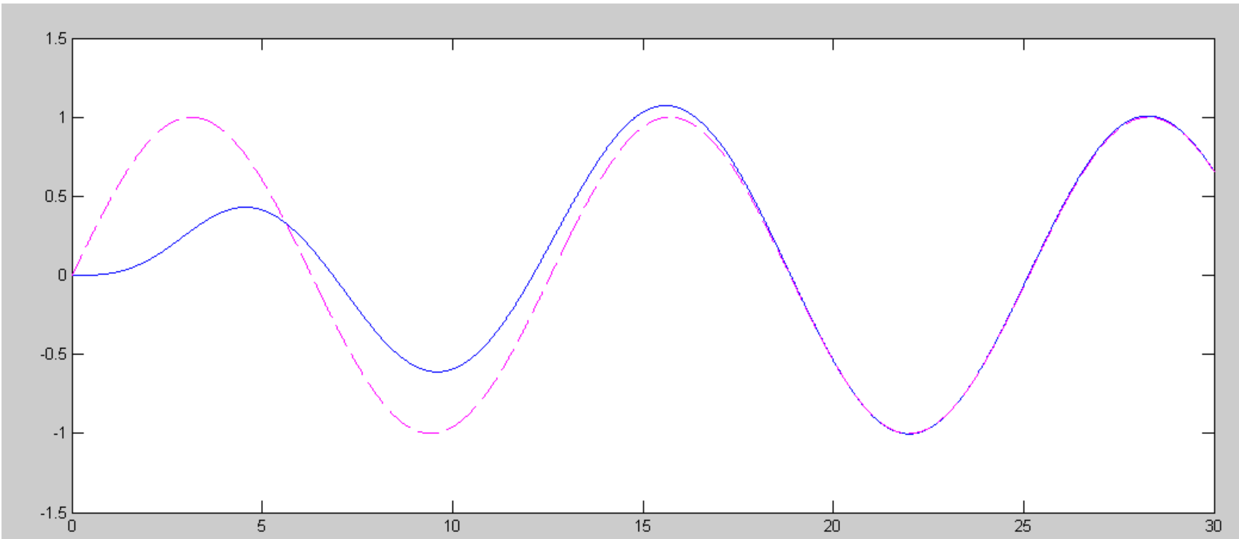
```
>> y = step3(A7,B7d,C7,D7,t,X0,d);
>> plot(t,y,'b')
```



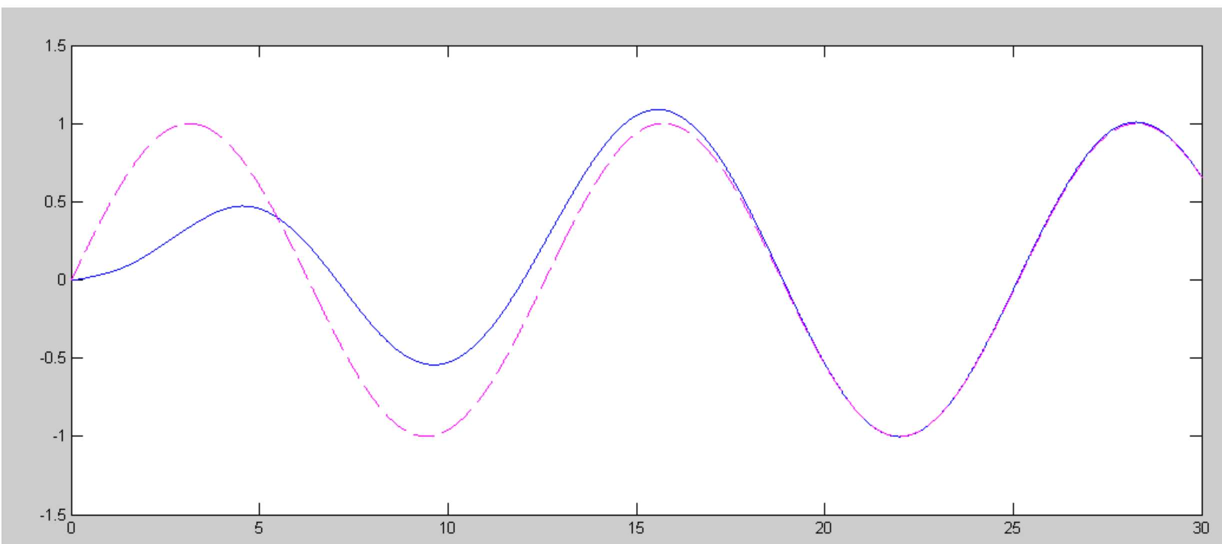
and it rejects a constant disturbance

Validate your feedback control law on the nonlinear system

- With d(t) = 0 and
- With d(t) = 1 (a cross-breeze pushes the gantry system to the right)



Tracks when d = 0



Tracks when d = 1

Code:

```
% Gantry System ( Sp24 version)
% m1 = 4.0kg
% m2 = 1.0kg
% L = 1.3m

X = [0;0;0;0];

dt = 0.01;
U = 0;
t = 0;

% Full State Feedback
Kx = [13.3966   -9.8848   14.0146    0.0190];
Kz = [1.1557    2.6546    1.2833];
% Servo Comp
Az = [0,0.5,0 ; -0.5,0,0 ; 0,0,0];
Bz = [1;1;1];
Z = zeros(3,1);

y = [];
d = 1;
n = 0;

while(t < 30)
 Ref = sin(0.5*t);
 U = -Kz*Z - Kx*X;

 dX = GantryDynamics(X, U + d);
 dZ = Az*Z + Bz*(X(1) - Ref);

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

 t = t + dt;

 n = mod(n+1, 10);
 if(n == 0)
     GantryDisplay(X, Xe, Ref);
     plot([Ref, Ref],[-0.1,0.1],'b');
     end

 y = [y ; X(1), Xe(1), Ref ];

end

hold off
t = [1:length(y)]' * dt;
plot(t,y(:,1),'r',t,y(:,2),'b', t, y(:,3), 'm--');
```

B Level (max 90 points)

1) Design a feedback control law for the gantry system assuming

- Only position and angle are measured, (observer is required)
- A sinusoidal set point (R(t) = sin(0.5t)), and
- No disturbance (d(t) = 0)

Same as C level but add a full-order observer

```
>> Co = [1,10,0,0]

Co =      1    10     0     0

>> H = ppl(A', Co', [-1,-2,-3,-4])'

    5.1826
    0.4817
    1.9327
    2.3647
```

2) Validate your feedback control law on the linear system

The plant & servo & observer (open-loop) are now...

$$s\begin{bmatrix} X \\ Z \\ X_e \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ B_z C & A_z & 0 \\ H C_o & 0 & A - H C_o \end{bmatrix} \begin{bmatrix} X \\ Z \\ X_e \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ B \end{bmatrix} U + \begin{bmatrix} 0 \\ -B_z \\ 0 \end{bmatrix} R$$

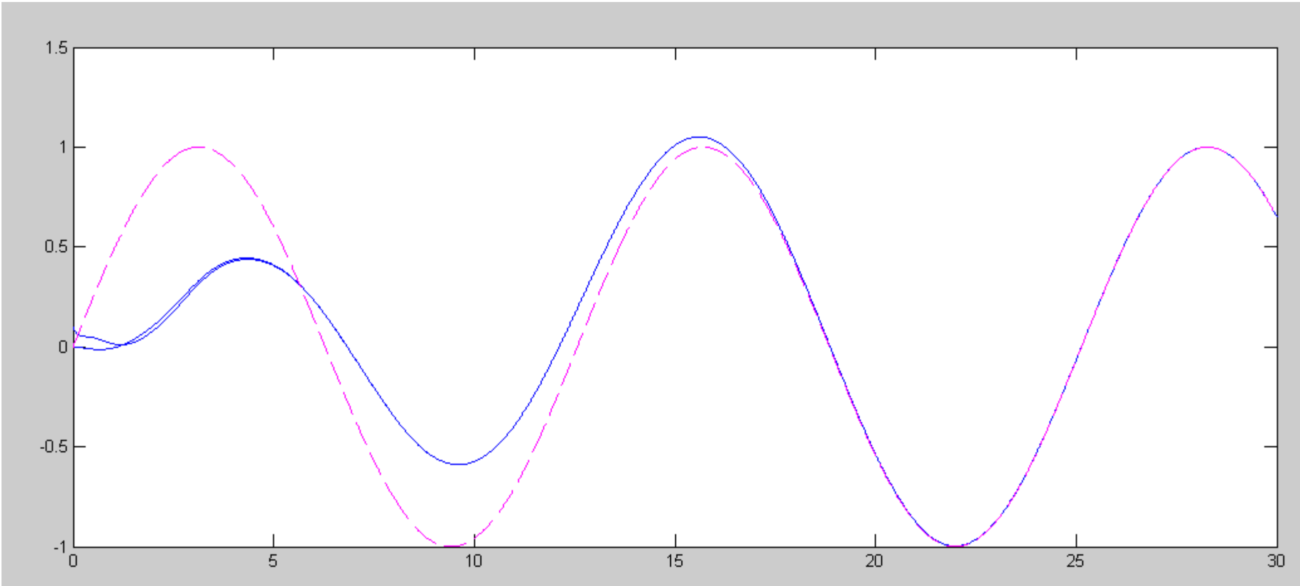
Closed-Loop:

$$s\begin{bmatrix} X \\ Z \\ X_e \end{bmatrix} = \begin{bmatrix} A & -BK_z & -BK_x \\ B_z C & A_z & 0 \\ H C_o & -BK_z & A - H C_o - BK_x \end{bmatrix} \begin{bmatrix} X \\ Z \\ X_e \end{bmatrix} + \begin{bmatrix} 0 \\ -B_z \\ 0 \end{bmatrix} R$$

Testing this in Matlab

```
>> A11 = [A, -B*Kz, -B*Kx ; Bz*C, Az, zeros(3,4) ; H*Co, -B*Kz, A-H*Co-B*Kx]

A11 =

         0         0    1.0000         0         0         0         0         0         0         0         0
         0         0         0    1.0000         0         0         0         0         0         0         0
         0    2.4500         0         0   -0.2889   -0.6636   -0.3208   -3.3491    2.4712   -3.5036   -0.0047
         0   -9.4200         0         0    0.2222    0.5105    0.2468    2.5762   -1.9008    2.6950    0.0037
    1.0000         0         0         0         0    0.5000         0         0         0         0         0
    1.0000         0         0         0   -0.5000         0         0         0         0         0         0
    1.0000         0         0         0         0         0         0         0         0         0         0
    5.1826   51.8256         0         0         0         0         0   -5.1826  -51.8256    1.0000         0
    0.4817    4.8174         0         0         0         0         0   -0.4817   -4.8174         0    1.0000
    1.9327   19.3274         0         0   -0.2889   -0.6636   -0.3208   -5.2819  -14.4062   -3.5036   -0.0047
    2.3647   23.6473         0         0    0.2222    0.5105    0.2468    0.2114  -34.9681    2.6950    0.0037

>> B11r = [0*B;-Bz;0*B];
>> C11 = [C,zeros(1,7);zeros(1,7),C]
>> D11 = [0;0];

>> X0 = zeros(11,1);
>> t = [0:0.01:30]';
>> R = sin(0.5*t);
>> X0(8) = 0.1;
>> y = step3(A11, B11r, C11, D11, t, X0, R);
```

```
>> plot(t,y,'b',t,R,'m--')
>>
```

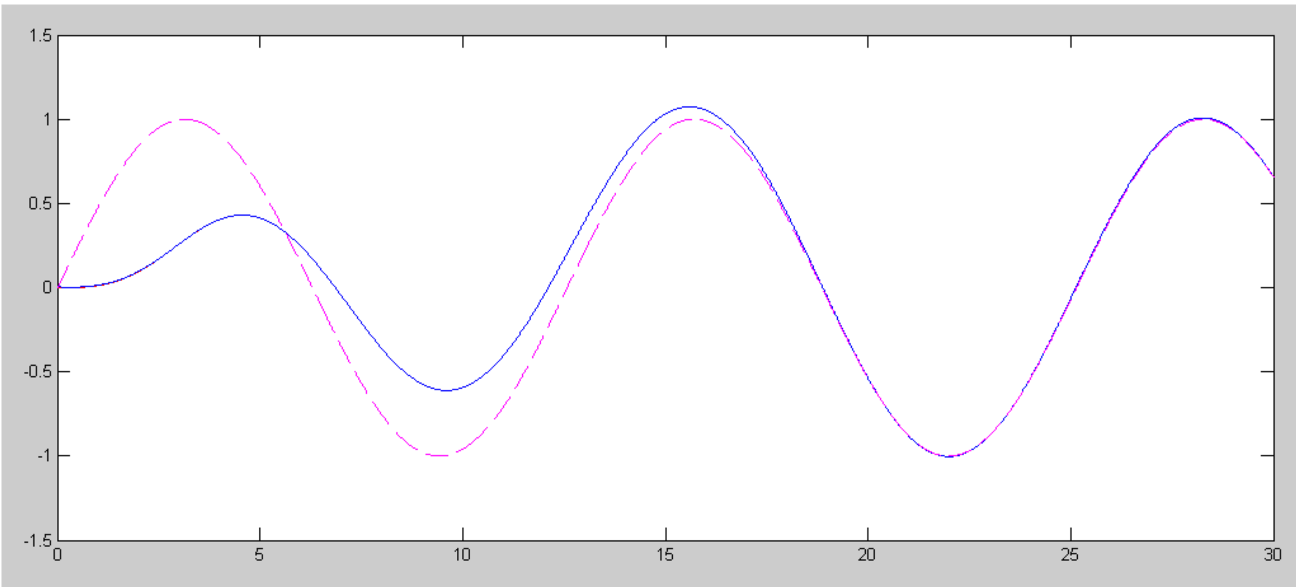


The observer converges and the net system tracks the set point

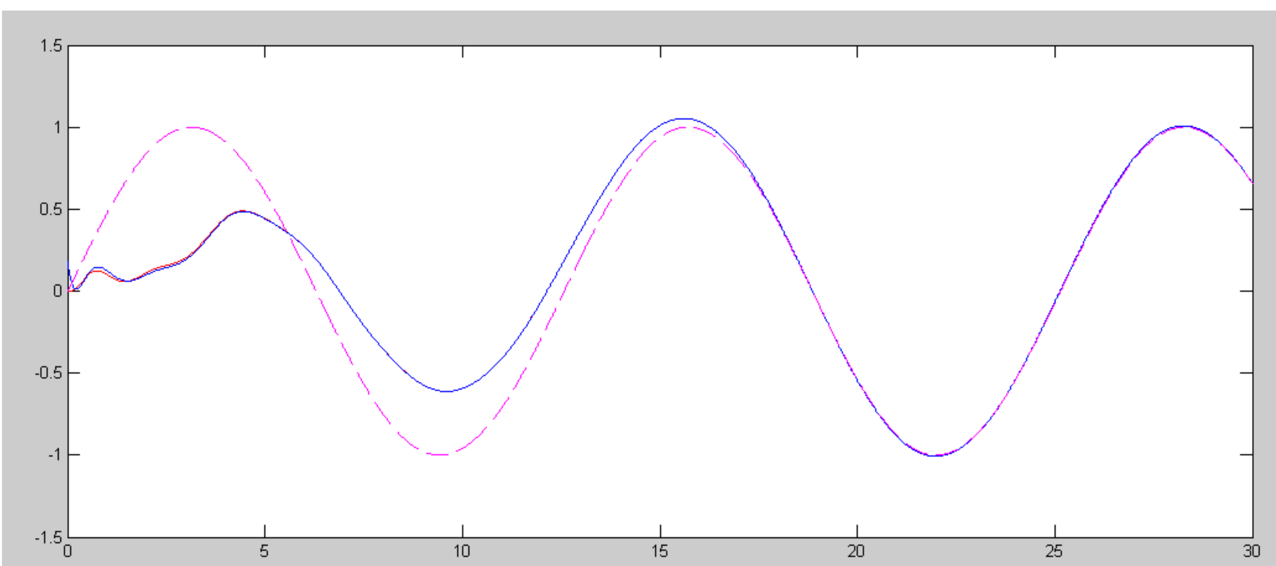
3) Validate your feedback control law on the nonlinear system

- Using the actual states for feedback (cheating)
```
d = 0
U = -Kz*Z - Kx*X
```
- Using the observer states
```
d = 0
U = -Kz*Z - Kx*Xe
```

Feeding back the actual states



Feeding back the state estimates



Code

```
% Gantry System ( Sp24 version)
% m1 = 4.0kg
% m2 = 1.0kg
% L = 1.3m

X = [0;0;0;0];

dt = 0.01;
U = 0;
t = 0;

% Full State Feedback
Kx = [13.3966   -9.8848   14.0146    0.0190];
Kz = [1.1557    2.6546    1.2833];
% Servo Comp
Az = [0,0.5,0 ; -0.5,0,0 ; 0,0,0];
Bz = [1;1;1];
Z = zeros(3,1);

% Full-Order Observer
A = [0,0,1,0;0,0,0,1;0,2.45,0,0;0,-9.42,0,0];
B = [0;0;0.25;-0.1923];
Co = [1,10,0,0];
H = ppl(A', Co', [-1,-2,-3,-4])';
Xe = [X] + 0.1*randn(4,1);

y = [];
d = 0;
n = 0;

while(t < 30)
 Ref = sin(0.5*t);
 U = -Kz*Z - Kx*Xe;

 dX = GantryDynamics(X, U + d);
 dXe = A*Xe + B*U + H*(Co*X - Co*Xe);
 dZ = Az*Z + Bz*(X(1) - Ref);

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

 t = t + dt;

 n = mod(n+1, 10);
 if(n == 0)
     GantryDisplay(X, Xe, Ref);
     plot([Ref, Ref],[-0.1,0.1],'b');
     end

 y = [y ; X(1), Xe(1), Ref ];

end

hold off
t = [1:length(y)]' * dt;
plot(t,y(:,1),'r',t,y(:,2),'b', t, y(:,3), 'm--');
```

A Level (max 100 points)

1) Design a feedback control law for the gantry system assuming

- Only position and angle are measured, (observer is requried)
- A sinusoidal set point (R(t) = sin(0.5t)), and
- A constant disturbance (d(t) = 1)

2) Validate your feedback control law on the linear system

3) Validate your feedback control law on the nonlinear system when d = 0

- Using the actual states for feedback (cheating)

```
d = 0
U = -Kz*Z - Kx*X
```

- Using the observer states

```
d = 0
U = -Kz*Z - Kx*Xe
```

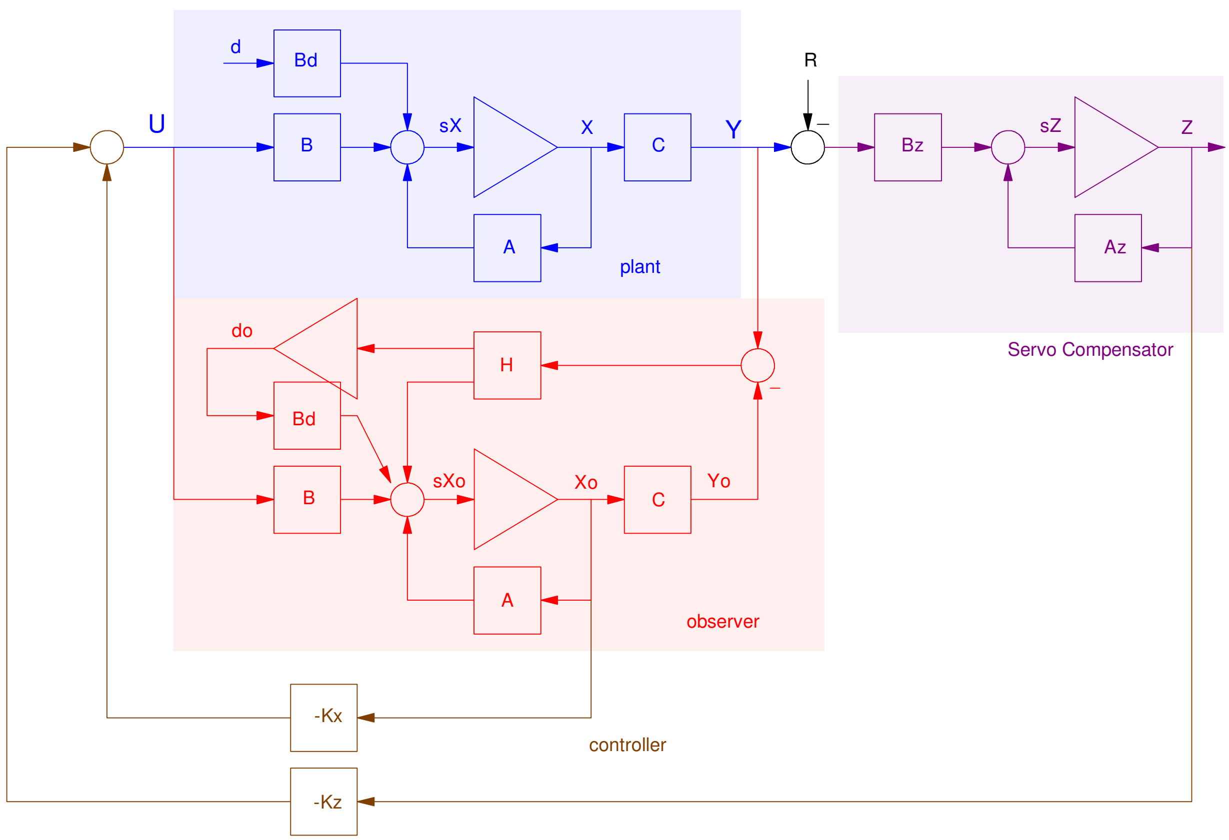
4) Validate your feedback control law on the nonlinear system when d = 1

- Using the actual states for feedback (cheating)

```
d = 1
U = -Kz*Z - Kx*X
```

- Using the observer states

```
d = 1
U = -Kz*Z - Kx*Xe
```



Block diagram for the Plant, Servo Compensator, Disturbance, Observer, and Full-State Feedback (A-Level)