ECE 463/663 - Homework #5

Full State Feedback. Due Wednesday, February 21st
Please submit as a hard copy, email to jacob.glower@ndsu.edu, or submit on BlackBoard

1) Write a Matlab m-file which is passed

- The system dynamics (A, B),
- The desired pole locations (P)

and then returns the feedback gains, Kx, so that roots(A - B Kx) = P

```
function [ Kx ] = ppl( A, B, P0)

N = length(A);

T1 = [];
for i=1:N
   T1 = [T1, (A^(i-1))*B];
end


P = poly(eig(A));
T2 = [];
for i=1:N
    T2 = [T2; zeros(1,i-1), P(1:N-i+1)];
end

T3 = zeros(N,N);
for i=1:N
    T3(i, N+1-i) = 1;
end

T = T1*T2*T3;

Pd = poly(P0);

dP = Pd - P;

Flip = [N+1:-1:2]';
Kz = dP(Flip);
Kx = Kz*inv(T);

end
```

Problems 2-4) Assume the following dynamic system:

$$sX = \begin{bmatrix} -6.1 & 3 & 0 & 0 & 0 \\ 3 & -6.1 & 3 & 0 & 0 \\ 0 & 3 & -6.1 & 3 & 0 \\ 0 & 0 & 3 & -6.1 & 3 \\ 0 & 0 & 0 & 3 & -3.1 \end{bmatrix} X + \begin{bmatrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} X$$

2) (20 points) Find the feedback control law of the form

$$U = K_r R - K_x X$$

so that

- The DC gain is 1.000 and
- The closed-loop poles are at {-2, -10, -11, -12, -13}

Plot

- The resulting closed-loop step reponse, and
- The resulting input, U

Input {A, B, C, D}

```
>> A = [-6.1,3,0,0,0;3,-6.1,3,0,0;0,3,-6.1,3,0];
>> A = [A;0,0,3,-6.1,3;0,0,0,3,-3.1]

   -6.1000    3.0000         0         0         0
    3.0000   -6.1000    3.0000         0         0
         0    3.0000   -6.1000    3.0000         0
         0         0    3.0000   -6.1000    3.0000
         0         0         0    3.0000   -3.1000

>> B = [3;0;0;0;0]

     3
     0
     0
     0
     0

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

     0     0     0     0     1
```

Use the ppl() routine to find the feedback gains:

```
>> Kx = ppl(A,B,[-2,-10,-11,-12,-13])

Kx =      6.8333   20.1556   33.7952   36.5292   31.2093
```

Check: Are the closed-loop poles correct? (yes, they are)

```
>> eig(A - B*Kx)

   -13.0000
   -12.0000
   -11.0000
   -10.0000
    -2.0000
```

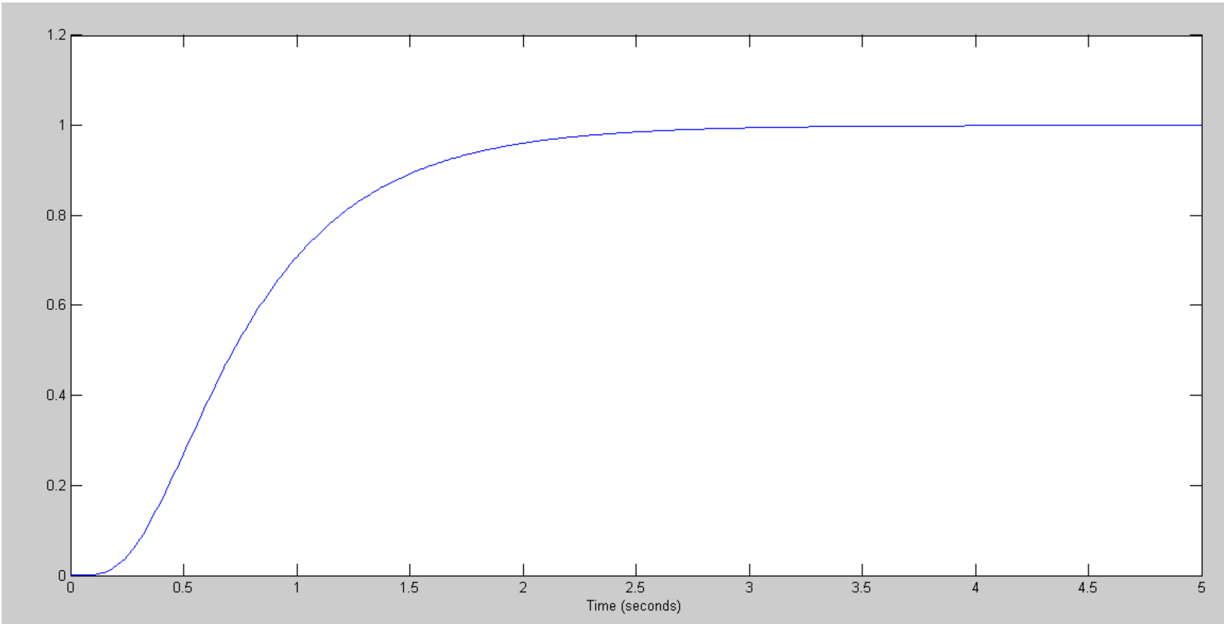
Find Kr to make the DC gain 1.0000

```
>> DC = -C*inv(A - B*Kx)*B

DC =     0.0071

>> Kr = 1/DC
```

Kr = 141.2346

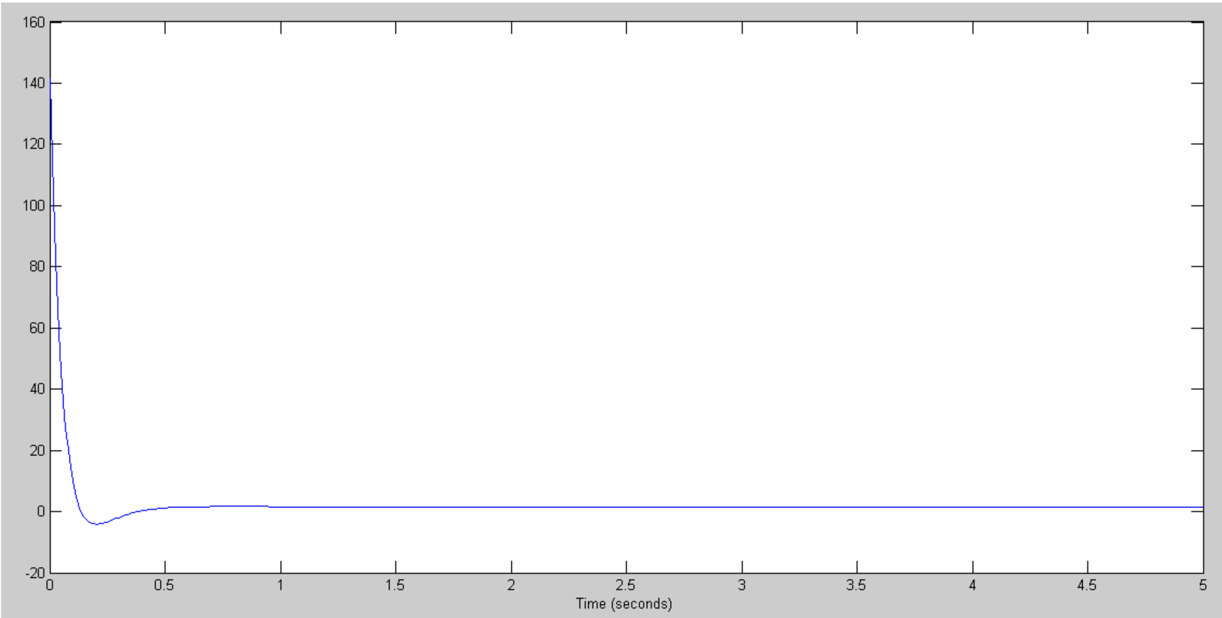
Plot the step response of the closed-loop system:

```
>> t = [0:0.01:5]';
>> Gcl = ss(A-B*Kx, B*Kr, C, 0);
>> zpk(Gcl)

              34320
---------------------------------
(s+13) (s+12) (s+11) (s+10) (s+2)

>> y = step(Gcl,t);
>> plot(t,y)
>> xlabel('Time (seconds)');
>>
```



Plotting the input (U)

```
>> Gu = ss(A-B*Kx, B*Kr, -Kx, Kr);
>> U = step(Gu, t);
>> plot(t,U)
>> xlabel('Time (seconds)');
```



Input, U(t)

3) (20 points) Repeat problem #2 but find Kx and Kr so that

- The DC gain is 1.000 and
- The closed-loop dominant pole is at s = -2 and the other four poles don't move (the are the same as the fast four poles of the open-loop system (eigenvalues of A)

Plot

- The resulting closed-loop step reponse, and
- The resulting input, U

First, determine where to place the closed-loop poles

```
>> P = eig(A)

  -11.1475
   -8.5925
   -5.2461
   -2.1708
   -0.3430

>> P(5) = -2

  -11.1475
   -8.5925
   -5.2461
   -2.1708
   -2.0000
```

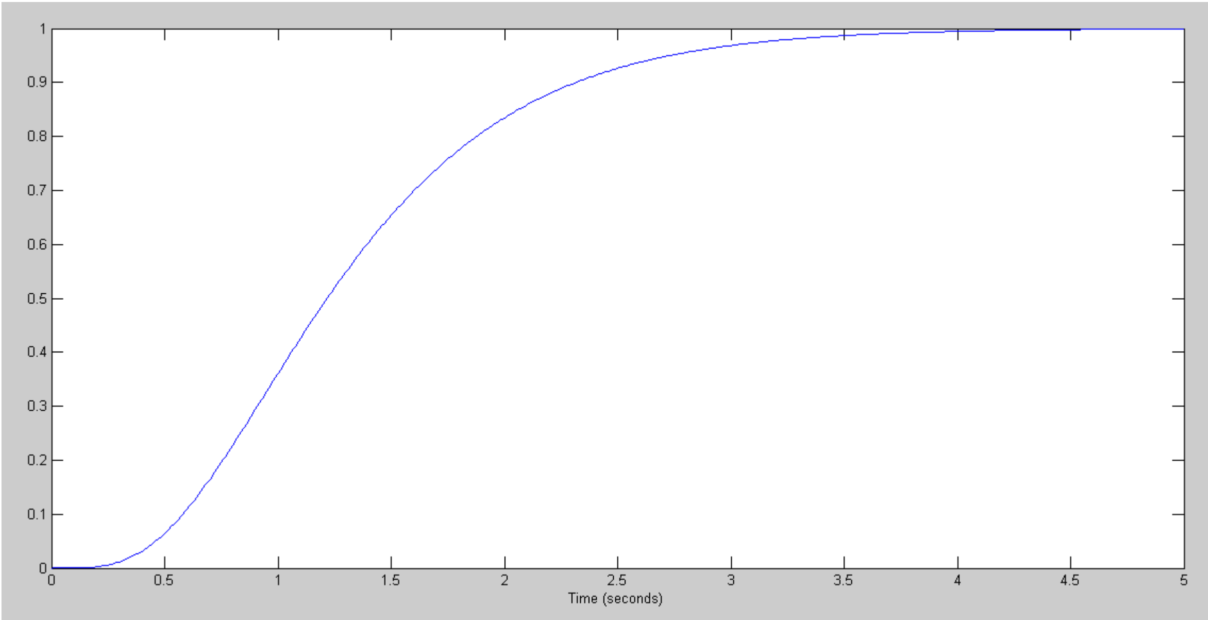
Find the feedback gains to place the closed-loop poles there:

```
>> Kx = ppl(A,B,P)

Kx =    0.5523    1.0599    1.4816    1.7833    1.9405

>> DC = -C*inv(A - B*Kx)*B

DC =    0.1114

>> Kr = 1/DC

Kr =    8.9781
```

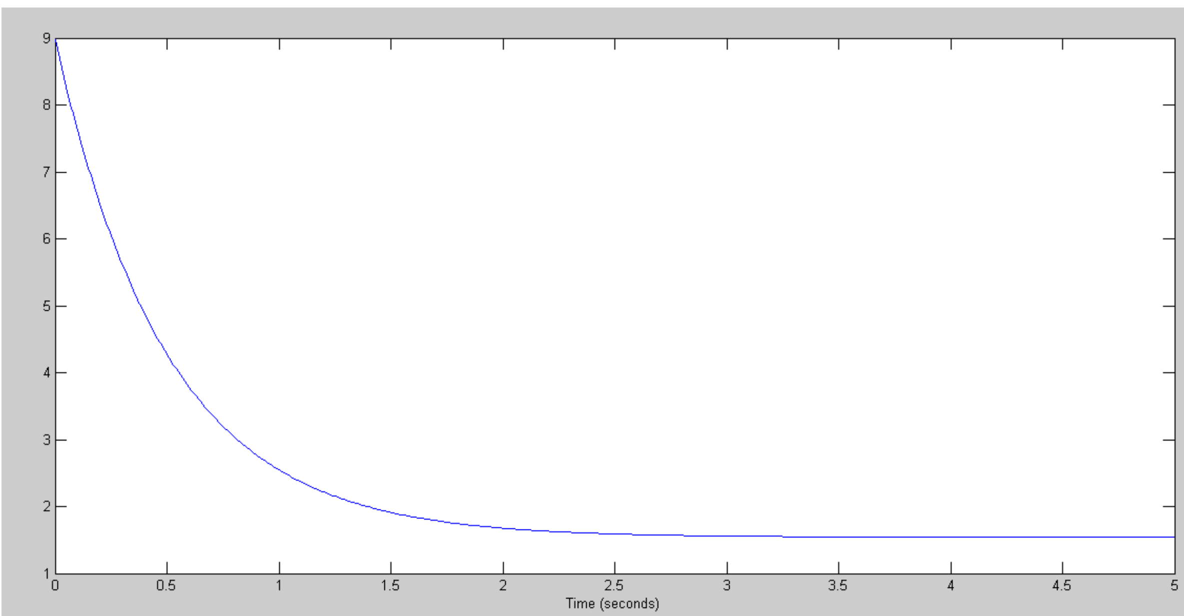
Note: Kx and Kr are *much* smaller than before. This should result in a similar response (same dominant pole) but smaller inputs

Plotting the closed-loop step responses:

```
>> Gy = ss(A-B*Kx,B*Kr,C,0);
>> y = step(Gy,t);
>> Gu = ss(A-B*Kx,B*Kr,-Kx,Kr);
>> U = step(Gu,t);
>> plot(t,y)
>> xlabel('Time (seconds)');
>> plot(t,U)
>> xlabel('Time (seconds)');
>>
```



Step response: y(t)



Step Response: U(t)

Note:

- y(t) is almost the same (same dominnat pole)
- u(t) is about 10x smaller

Some pole locations are better than others...

4) (20 points) Repeat problem #2 but find Kx and Kr so that

- The DC gain is 1.000
- The 2% settling time is 2 seconds, and
- There is 10% overshoot for a step input.

Plot

- The resulting closed-loop step reponse, and
- The resulting input, U

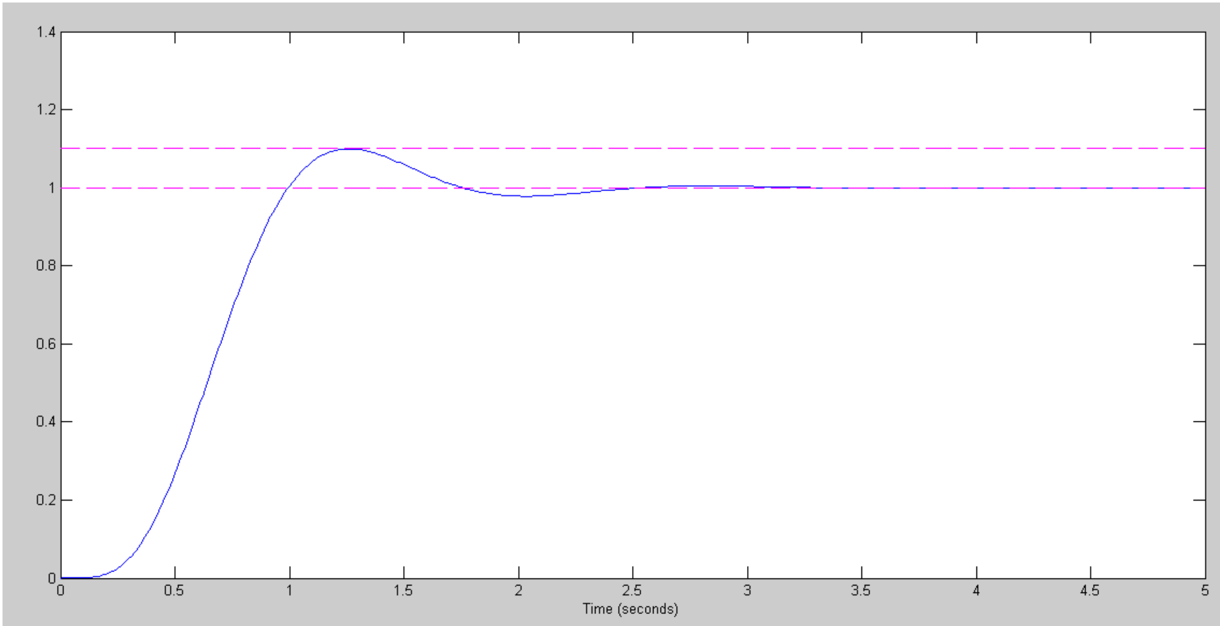
For 10% overshoot...

$$\zeta = 0.591$$

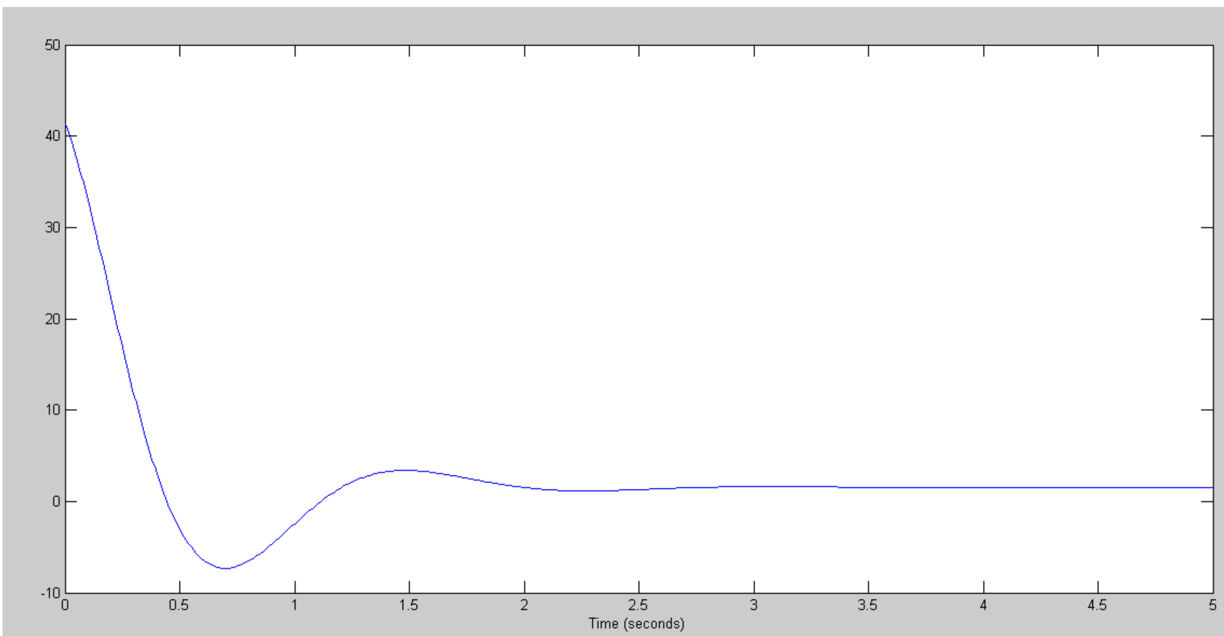
$$s = -2 + j2.73$$

This results in 6.9% overshoot (the three real poles reduce the overshoot). Adjust the complex part until you get 10% overshoot

```
>> P(5) = -2 + j*4;
>> P(4) = conj(P(5))

 -11.1475
  -8.5925
  -5.2461
  -2.0000 - 4.0000i
  -2.0000 + 4.0000i

>> Kx = ppl(A,B,P)

Kx =    0.4954    2.7316    7.0792   12.1274   15.5292

>> DC = -C*inv(A - B*Kx)*B;
>> Kr = 1/DC

Kr =   41.3579

>> Gy = ss(A-B*Kx,B*Kr,C,0);
>> y = step(Gy,t);
>> max(y)

ans =    1.0996

>> plot(t,y);
>> xlabel('Time (seconds)');
>> plot(t,y,t,0*y+1,'m--',t,0*y+1.1,'m--');
>> xlabel('Time (seconds)');
>>

>> Gu = ss(A-B*Kx,B*Kr,-Kx,Kr);
>> U = step(Gu,t);
>> plot(t,U);
>> xlabel('Time (seconds)');
```



Step response to y(t): 10% overshoot and a 2 second settling time



Step response to U(t): 10% overshoot can be achieved, but it takes more input