ECE 463/663 - Homework #5

Full State Feedback. Due Monday, February 24th

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
```

Check:

```
>> A = rand(4,4);
>> B = rand(4,1);
>> Kx = ppl(A, B, [-1,-2,-3,-4])

Kx =    9.0220 -289.7033  -41.4816  318.4603

>> eig(A - B*Kx)

   -4.0000
   -1.0000
   -3.0000
   -2.0000
```

Problems 2-4) Assume the following dynamic system:

$$sX = \begin{bmatrix} -10 & 5 & 0 & 0 & 0 \\ 5 & -10 & 5 & 0 & 0 \\ 0 & 5 & -10 & 5 & 0 \\ 0 & 0 & 5 & -10 & 5 \\ 0 & 0 & 0 & 5 & -5 \end{bmatrix} X + \begin{bmatrix} 5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

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

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 {-1, -2, -3, -4, -5}

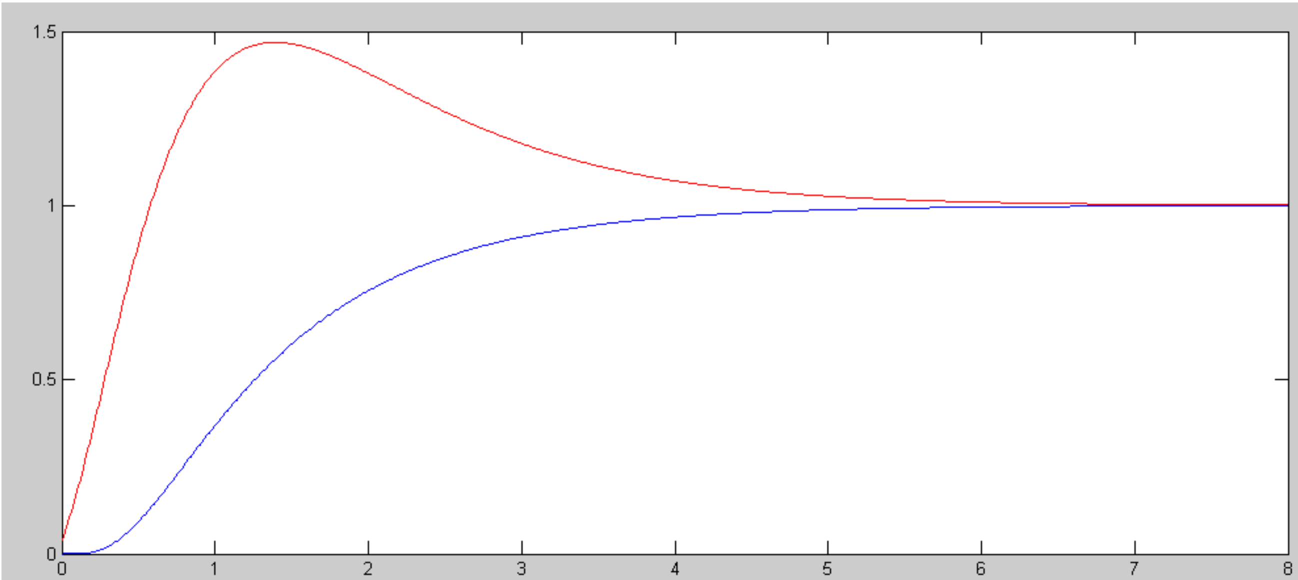
Plot

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

Matlab Code

```
>> A = [-10,5,0,0,0;5,-10,5,0,0;0,5,-10,5,0;0,0,5,-10,5;0,0,0,5,-5]

   -10     5     0     0     0
     5   -10     5     0     0
     0     5   -10     5     0
     0     0     5   -10     5
     0     0     0     5    -5

>> B = [5;0;0;0;0]

     5
     0
     0
     0
     0

>> Kx = ppl(A, B, [-1,-2,-3,-4,-5])

Kx =   -6.0000   17.4000  -30.2000   31.6384  -13.8000

>> C = [0,0,0,0,1];
>> DC = -C*inv(A-B*Kx)*B

DC =   26.0417

>> Kr = 1/DC

Kr =    0.0384
```

```
>> Gcl = ss(A-B*Kx, B*Kr, C, 0);
>> t = [0:0.01:8]';
>> y = step(Gcl, t);

>> Gu = ss(A-B*Kx, B*Kr, -Kx, Kr);
>> U = step(Gu, t);
>> plot(t, y, 'b', t, U, 'r')
>>
```



Step response to y (blue) and u (red)

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 = -1 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)

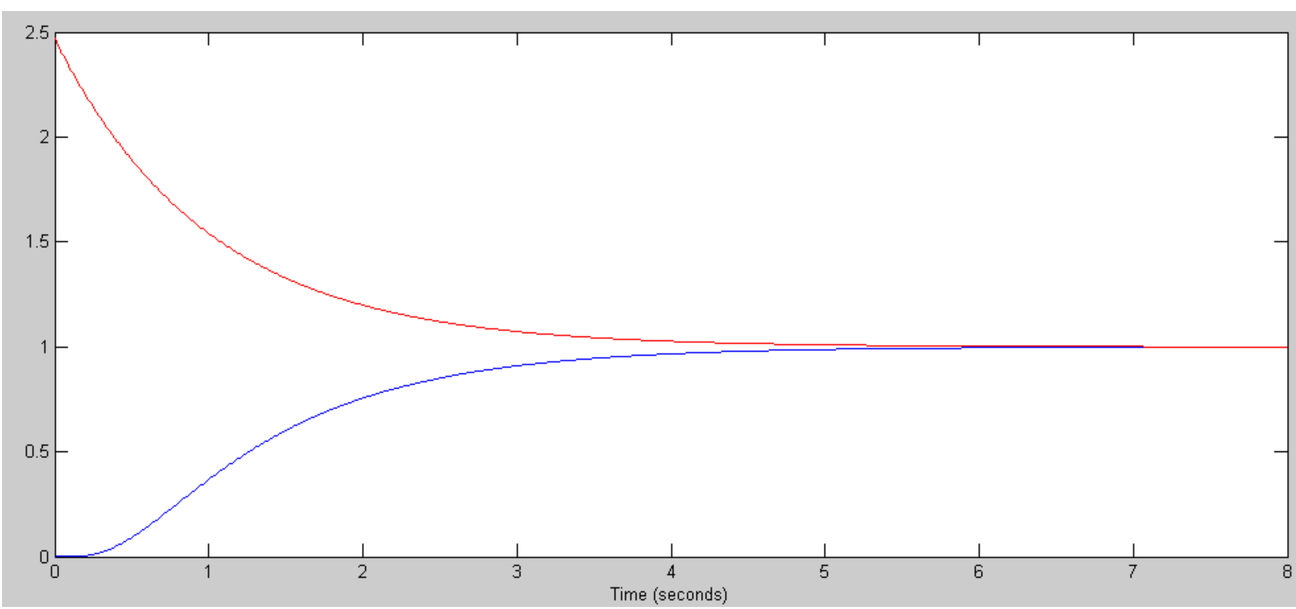
Plot

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

```
>> P = eig(A)

  -18.4125
  -14.1542
   -8.5769
   -3.4514
   -0.4051

>> Kx = ppl(A, B, [-1, P(1), P(2), P(3), P(4)])

Kx =     0.1190     0.2283     0.3192     0.3842     0.4180
```

note: The feedback gains a much smaller than before. It doesn't take as much energy to move a single pole as it does to move all five poles

```
>> DC = -C*inv(A-B*Kx)*B
DC =     0.4051

>> Kr = 1/DC
Kr =     2.4687

>> 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, 'b', t, U, 'r')
>> xlabel('Time (seconds)')
>>
```



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

First, determine where the closed-loop dominant pole belongs

2% settling time = 2 seconds

- real(s) = -2

10% overshoot

- damping ratio = 0.5911
- s = -2 + j2.7288

In Matlab, start with where to place the closed-loop poles:

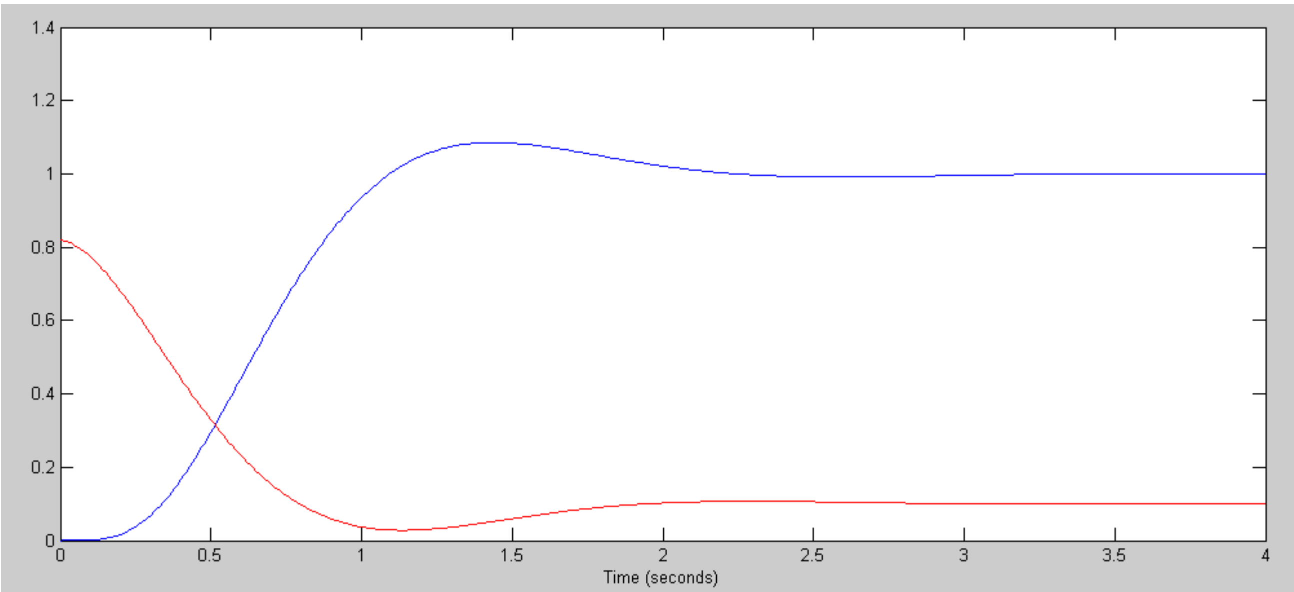
```
>> P = eig(A)

  -18.4125
  -14.1542
   -8.5769
   -3.4514
   -0.4051


>> P(4) = -2 + 2.7288i;
>> P(5) = conj(P(4));
>> P

 -18.4125
 -14.1542
  -8.5769
  -2.0000 + 2.7288i
  -2.0000 - 2.7288i
```

Find the feedback gains, Kx and Kr

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

Kx =    0.0287    0.4372    1.3107    2.3515    3.0592

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

DC =    0.1221

>> Kr = 1/DC

Kr =    8.1873
```

Plot the closed-loop response:

```
>> t = [0:0.01:4]';
>> 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, 'b', t, U, 'r')
>> xlabel('Time (seconds)')
```



Step response to y (blue) and U/10 (red)