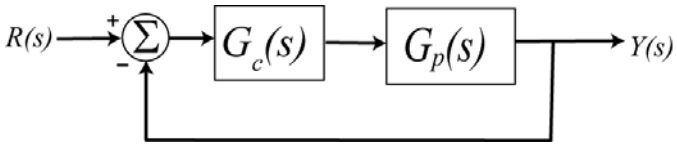
ECE-205 : Dynamical Systems

Homework #9

Due : **Tuesday** February 9 at the beginning of class

Exam 3, Thursday February 11

1) Consider the following simple feedback control block diagram. The plant is $G_p(s) = \frac{2}{s+4}$. The input is a unit step.



a) Determine the settling time and steady state error of the plant alone (assuming there is no feedback)

b) Assuming a proportional controller, $G_c(s) = k_p$, determine the closed loop transfer function, $G_0(s)$

c) Assuming a proportional controller, $G_c(s) = k_p$, determine the value of k_p so the steady state error for a unit step is 0.1, and the corresponding settling time for the system.

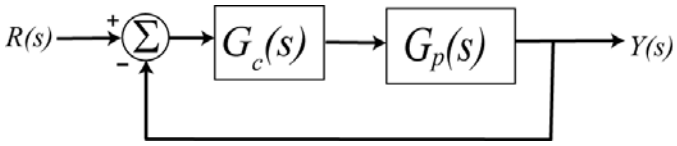
d) Assuming a proportional controller, $G_c(s) = k_p$,determine the value of k_p so the settling time is 0.5 seconds, and the corresponding steady state error.

e) Assuming an integral controller, $G_c(s) = k_i / s$, determine closed loop transfer function, $G_0(s)$

f) Assuming an integral controller, $G_c(s) = k_i / s$, determine the value of k_i so the steady state error for a unit step is less than 0.1 and the system is stable.

PartialAnswers: $T_s = 1, e_{ss} = 0.5, k_p = 18, k_p = 2, T_s = 0.1, e_{ss} = 0.5, k_i > 0$

2) (Model Matching) Consider the following closed loop system, with plant $G_p(s)$ and controller $G_c(s)$.

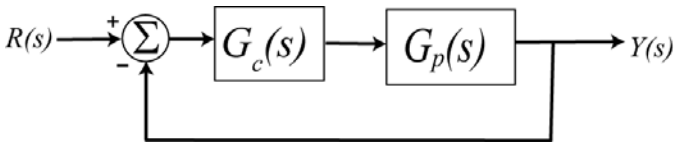


One way to choose the controller is to try and make your closed loop system match a transfer function that you choose (hence the name model matching). Let's assume that our **desired** closed loop transfer function $G_o(s)$, our plant can be written in terms of numerators and denominators as $G_o(s) = \frac{N_o(s)}{D_o(s)} \quad G_p(s) = \frac{N_p(s)}{D_p(s)}$

Show that our controller is then $G_c(s) = \frac{N_o(s)D_p(s)}{N_p(s)\left[D_o(s) - N_o(s)\right]}$

Note that there are some restrictions here, in that for implementation purposes the controller must be stable, and it must be proper.

3) For the following system, with plant $G_p(s) = \dfrac{1}{s+1}$, and controller $G_c(s)$



a) Using the results from problem 2, determine the controller so that the closed loop system matches a second order ITAE (**I**ntegral of **T**ime and **A**bsolute **E**rror) optimal system, i.e., so that the closed loop transfer function is

$$G_0(s) = \frac{\omega_0^2}{s^2 + 1.4\omega_0 s + \omega_0^2}$$

Ans. $G_c(s) = \dfrac{\omega_0^2(s+1)}{s(s+1.4\omega_0)}$, *note that there is a pole/zero cancellation between the controller and the plant and there is a pole at zero in the controller.*

b) Show that the damping ratio for this system is 0.7, the closed loop poles of this system are at $-0.7\omega_0 \pm j0.714\omega_0$. For faster response should ω_0 be large or small?

c) What is the steady state error for this system if the input is a unit step?

d) Determine the controller so that the closed loop system matches a third order **deadbeat** system, i.e., so that the closed loop transfer function is

$$G_0(s) = \frac{\omega_0^3}{s^3 + 1.90\omega_0 s^2 + 2.20\omega_0^2 s + \omega_0^3}$$

Ans. $G_c(s) = \dfrac{\omega_0^3(s+1)}{s(s^2 + 1.9\omega_0 s + 2.20\omega_0^2)}$, *note that there is a pole/zero cancellation between the controller and the plant and there is a pole at zero in the controller.*

e) What is the steady state error of this system for a unit step input?

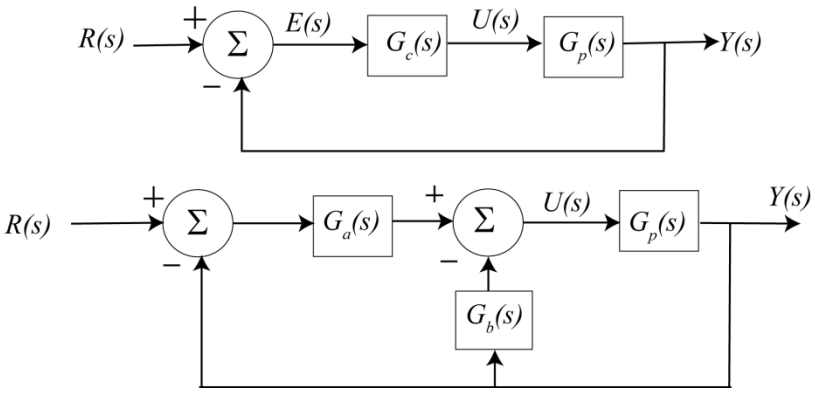
4) Assume we have the same control structure as in the previous problem where $G_p(s) = \dfrac{1}{s+2}$. We want to design a model matching controller so that

- the closed loop system is a second order
- the state error for a step input is zero
- the settling time of the closed loop system is 3 seconds
- the percent overshoot of the closed loop system is 20%
-

. You are to determine the closed loop transfer function and then the controller to meet these specifications.

Ans. $G_c(s) = \dfrac{8.55(s+2)}{s(s+2.66)}$

5) Consider the following two feedback control structures,



Let's assume we can break up the controller so that we can write the transfer functions as

$$G_p(s)=\frac{N_p(s)}{D_p(s)}, G_c(s)=\frac{N_c(s)}{D_c(s)}, G_a(s)=\frac{N_a(s)}{D_a(s)}, G_b(s)=\frac{N_b(s)}{D_b(s)}$$

a) Show that if we can partition the controller so that

$$G_a(s)+G_b(s)=G_c(s)$$

Then we can write the closed loop transfer functions as

$$G_o(s)=\frac{N_c(s)N_p(s)}{D_c(s)D_p(s)+N_c(s)N_p(s)}, G_o(s)=\frac{D_c(s)N_a(s)N_p(s)}{D_a(s)\left[D_c(s)D_p(s)+N_c(s)N_p(s)\right]}$$

Hence, if $D_a(s)$ is a constant, the two structures produce the same closed loop poles.

b) For the first structure, we showed in class that the initial control effort was given by

$$u(0^+)=\lim_{s\to\infty} sU(s)=\lim_{s\to\infty}\frac{sR(s)G_c(s)}{1+G_c(s)G_p(s)}$$

Show that the initial control effort for the second structure is given by

$$u(0^+)=\lim_{s\to\infty} sU(s)=\lim_{s\to\infty}\frac{sR(s)G_a(s)}{1+\left[G_a(s)+G_b(s)\right]G_p(s)}$$

c) Assume we have the plant $G_p(s)=\frac{3}{s+2}$, and assume we are trying to use a PD controller. So

$$G_c(s)=k_p+k_d s,\quad G_a(s)=k_p,\quad G_b(s)=k_d s$$

Show that if the input is a step of amplitude A, then for the two control structures we have

$$u(0^+)=\infty, e_{ss}=\frac{A2}{2+3k_p},\quad u(0^+)=\frac{Ak_p}{1+3k_d}, e_{ss}=\frac{A2}{2+3k_p}$$

This second control structure is often used with PID controllers, as I-PD and PI-D controllers.

6) (very short Matlab Problem) We have been using arrays in Matlab, but Matlab is also very good at matrices. So in this short problem, you will write a short m-file and do some (good) stuff.

To enter a matrix into Matlab, you enter the rows, and end each row with a semicolum. Hence to enter the matrix $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ into Matlab, type A = [1 2; 3 4; 5 6];

To access any element of the matrix, you use the usual matrix indexing. Hence A(1,1) = 1, A(2,2) =4, and A(3,2) = 6; You can also use this indexing to change the value of a matrix element. If you want to access an entire column or row, you use the colon command. Hence A(:,1) gets the first column, A(:,2) gets the second column, A(1,:) gets the first row, A(2,:) gets the second row, and A(3,:) gets the third row.

To find the inverse of the matrix A, you can use the command inv(A). However, if you want to solve the system of equations Ax = b, it is (usually) a (numerically) bad idea to type x = inv(A)*b. It is better in Matlab to type x=A\b, which is more robust numerically.

<u>For you to do</u>

a) Open an m-file (which you will e-mail to me). Do not put semicolons at the end of any line, so your results will print to the screen.

b) Enter the following matrices into Matlab

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

c) Compute AB, AB^{-1}, $A^{-1}B$

d) Solve the system of equations $Ax = b$ when b is the first, second, and third column of B (solve these individually)

e) Extract the second row of A using the semicolon notation