ECE-320: Linear Control Systems
Homework 4

Due: Tuesday April 12 at 10 AM

<u>Exam Friday April 15</u>

1) For a system with plant

$$G_p(s) = \frac{s+3}{s(s-1)}$$

show that the quadratic optimal closed loop transfer function for $q = 100$ is

$$G_0(s) = \frac{10(s+3)}{s^2 + 12.7s + 30}$$

and the controller is

$$G_c(s) = \frac{10(s-1)}{s+2.7}$$

Show that $e_p = 0$ and $e_v = 0.09$.

2) For a system with plant

$$G_p(s) = \frac{s-1}{s(s-2)}$$

show that the quadratic optimal closed loop transfer function for $q = 100$ is

$$G_0(s) = \frac{-10(s-1)}{s^2 + 11.1s + 10}$$

and the controller is

$$G_c(s) = \frac{-10s + 20}{s + 21.14}$$

Show that $e_p = 0$ and $e_v = 2.11$.

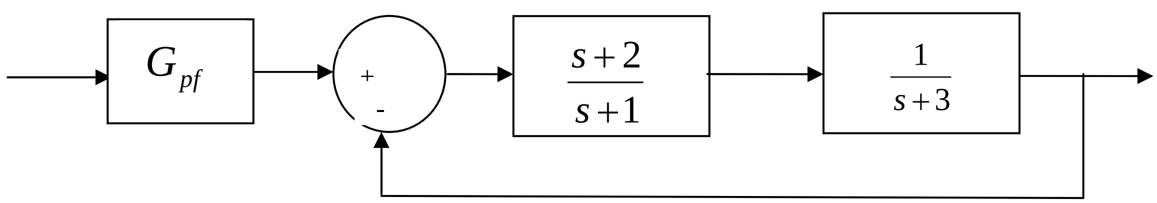
3) For the following systems

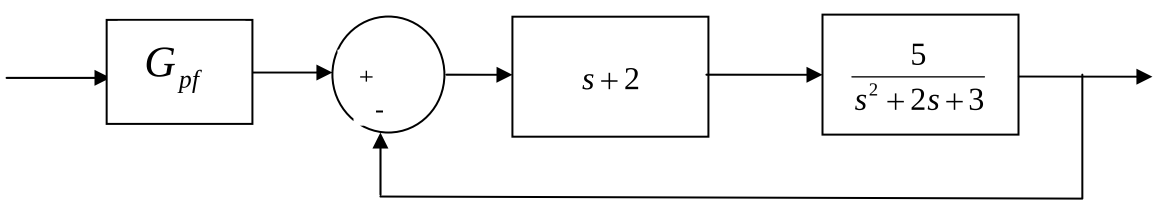
a) Determine the system type (0, 1, 2, ...)

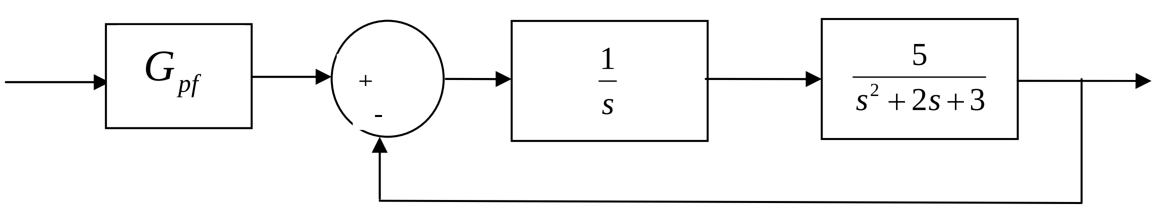
b) If the system is type 0 assume $G_{pf} = 1$ and determine the position error constant K_p and the position error e_p. Then determine the value of G_{pf} to make the position error zero.

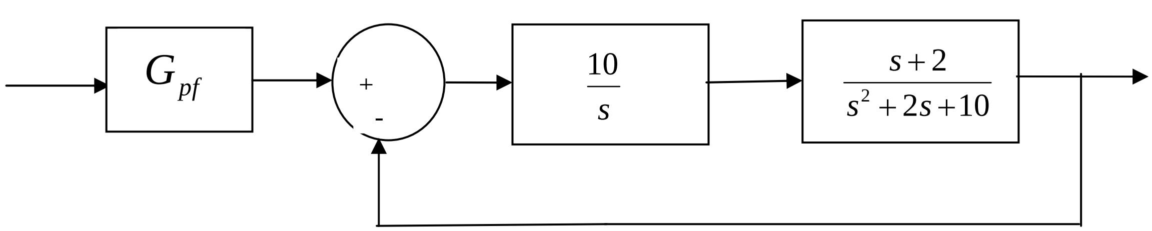
c) If the system is type 1, assume $G_{pf} = 1$ and determine the position error, the velocity error constant K_v, and the velocity error e_v. Is there any constant value of G_{pf} that can change the velocity error?

Ans. $e_p = \frac{3}{5}$ and $G_{pf} = \frac{5}{2}$, $e_p = \frac{3}{13}$ and $G_{pf} = \frac{13}{10}$, $e_v = \frac{3}{5}$, $e_v = \frac{1}{2}$, G_{pf} has no effect on e_v



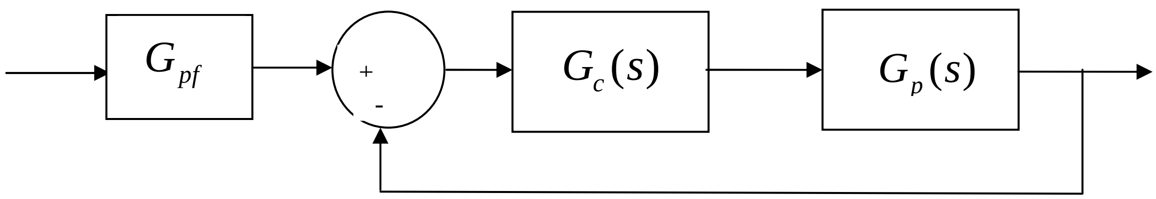






4) *(Note: You may want to do problems 6 and 7 before this problem. Your plots from problem 6 are acceptable for this problem. However, you must compute the closed loop transfer functions, controllers, and prefilter gains by hand in this problem.)*

In this problem we explore a slight change to the standard quadratic optimal controller. We assume we have the following system, with prefilter G_{pf}, plant $G_p(s)$ and controller $G_c(s)$



a) Assume the plant is

$$G_p(s) = \frac{200}{s+10}$$

and we want to design a quadratic optimal controller with $q = 0.001$. Show that

$$G_0(s) = \frac{3.381}{s+11.832} \quad G_c(s) = \frac{0.0169(s+10)}{s+8.451} \quad G_{pf} = 3.5$$

for zero position error.

Simulate the system with a unit step input. Your results should look like that in Figure 1. Turn in this plot.

b) Assume we utilize the quadratic optimal algorithm as above to determine, $G_0(s)$ but rather than utilizing a prefilter, we scale the closed loop transfer function so that $G_0(0) = 1$. Show that we then get

$$G_c(s) = \frac{0.05916(s+10)}{s} \text{ and } G_{pf} = 1$$

Simulate the system using this controller and prefilter with a unit step input. Your results should look like that in Figure 1.

Note that we are not guaranteed we will be able to get an implementable transfer function if we scale $G_0(s)$ so that $G_0(0) = 1$, but if we can, we get a type 1 system and there is no scaling outside the of the feedback loop ($G_{pf} = 1$)! Why do we care? See part c, d, and e below.

c) Assume that despite your best efforts, you did not get the model for the plant exactly correct, and instead, the correct transfer function for the plant is

$$G_p(s) = \frac{190}{s+12}$$

Simulate this system with a unit step input using the controller and prefilter from part (a) using this plant instead. Your results should look like those in Figure 2.

d) Now simulate this system (plant from part c) with a unit step input using the controller and prefilter from part (b). You should get results like those in Figure 3. Turn in your graph.

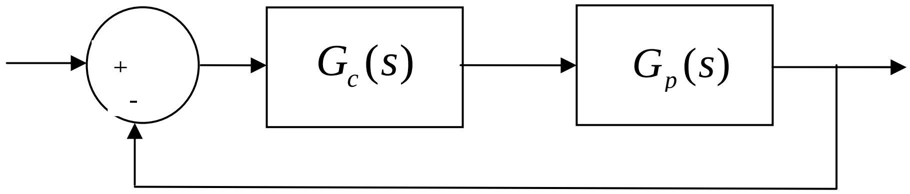
e) Assume that your fool of a lab partner copied down the numbers incorrectly, and the plant is more accurately modeled as

$$G_p(s) = \frac{100}{s+20}$$

Show (analytically) that the position error for the original system for a step input with amplitude A is 0.68A, while the position error for the type 1 system is still 0. Simulate this plant using the original control system (part a) and the modified control system (part b). You should get results like those in Figures 4 and 5.. Turn in your graphs ***and your Matlab code.***

The moral of this problem is that, if we can, we are usually better off if we do not depend on anything outside the feedback loop! Type 1 system can help alot with modeling errors if we want a zero position error. However, it may take a long time to reach the zero position error.

5) Consider the following simple feedback system, with which we want to use model matching to determine the controller $G_c(s)$.



However, the plant $G_p(s)$ has zeros in the right half plane, and we cannot cancel these poles or we will have an unstable controller. Let's denote the plant

$$G_p(s) = \frac{N_L(s)N_R(s)}{D(s)}$$

where we have partitioned the numerator polynomial of the plant into $N_L(s)$ which contains the zeros of the plant in the open left half plane, and $N_R(s)$ which contains the zeros of the plant in the closed right half plane. Let's assume then that the desired closed loop transfer function is written as

$$G_o(s) = \frac{N_o(s)}{D_o(s)} = \frac{N_{oo}(s)N_R(s)}{D_o(s)}$$

a) Show that the controller for this system is given by

$$G_c(s) = \frac{N_{oo}(s)D(s)}{N_L(s)[D_o(s) - N_{oo}(s)N_R(s)]}$$

b) Insert the expression for the plant and the expression for the controller from part (a) into the block diagram, cancel where appropriate, and show by simplifying the block diagram that

$$G_o(s) = \frac{N_{oo}(s)N_R(s)}{D_o(s)}$$

Did the controller cancel the right half plane zeros of the plant? Did the controller cancel the left half plane zeros of the plant? Did the controller cancel the poles of the plant?

Preparation for Lab 5.

6) In this problem we will modify the Matlab code **closedloop_driver.m** to determine the quadratic optimal controller for a given plant and value of penalty *q*. You should comment out those parts of the code with are not being used, do not delete them since they will be utilized later!

a) Download the file **solve_quadratic.m** from the class website. The input arguments to this function are (1) the transfer function of the plant Gp, and (2) the value of *q*. The routine returns the optimal closed loop transfer function Go(s). To use this function, in **closedloop_driver.m** type something like

```
q = 0.001;
Go = solve_quadratic(Gp,q);
```

Be sure all of your files are in the same directory!

b) Modify **closedloop_driver.m** to verify the results from problem 4a. Turn in a plot for the unit step response. Be sure to use the command **minreal** when manipulating transfer functions, such as finding Gc or finding the new closed loop transfer function to determine Gpre.

c) Modify **closedloop_driver.m** to verify the result of problem 4b (we are just looking to get the correct controller here). This should be done *automatically*, you should not hard code any numbers. Look at your results from Lab 1 for determining the prefilter gain.

d) Modify **closedloop_driver.m** to verify the results of problems 4c,d, and e. You should put in the ``new plant'' just before the simulation so it doesn't change anything else. Turn in your plots and your Malab code.

7) Modify **closedloop_driver.m** to utilize the1 dof state variable model one the web (for lab 1) Plot the unit step response for *q=0.001*, *q = 0.01*, and *q = 0.1* How dose changing *q* affect the response and the maximum control effort? Be sure your code works both with the original value of Go(s) (in which case the prefilter is probably not 1) and for the modified Go(s). Be sure to turn in all three plots.

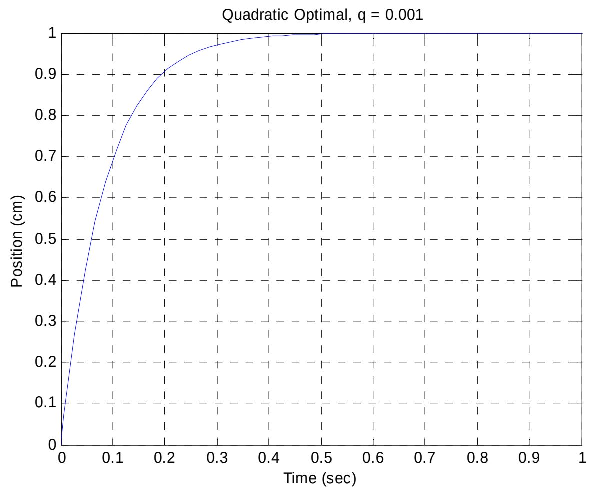


Figure1. The step response for the correct plant and original controller (Problem 4, part a)

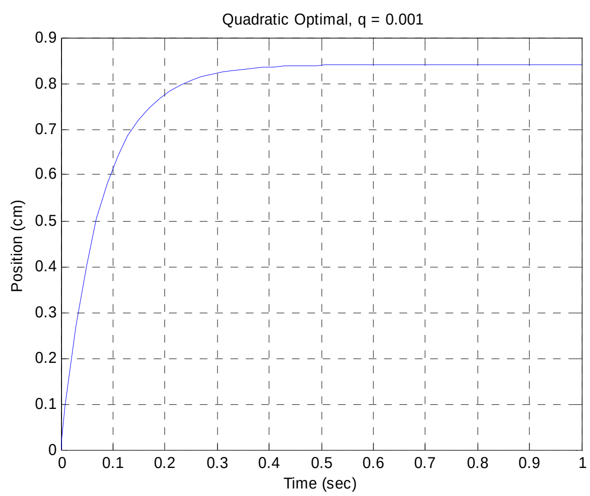


Figure 2. The step response for the new plant and the original controller (Problem 4, part c)

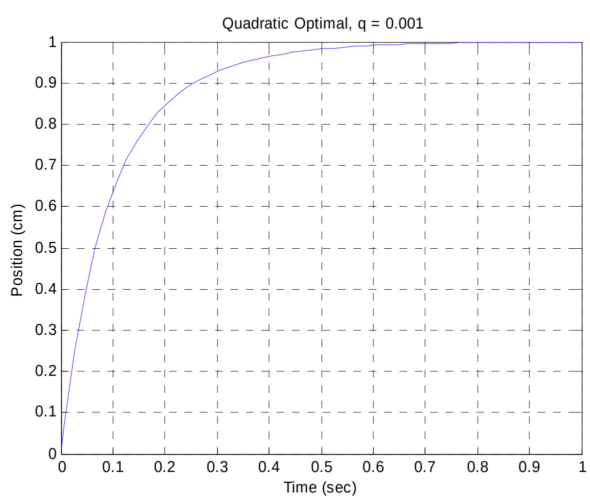


Figure 3. Step response of the new plant with the type 1 controller (Problem 4, part d)

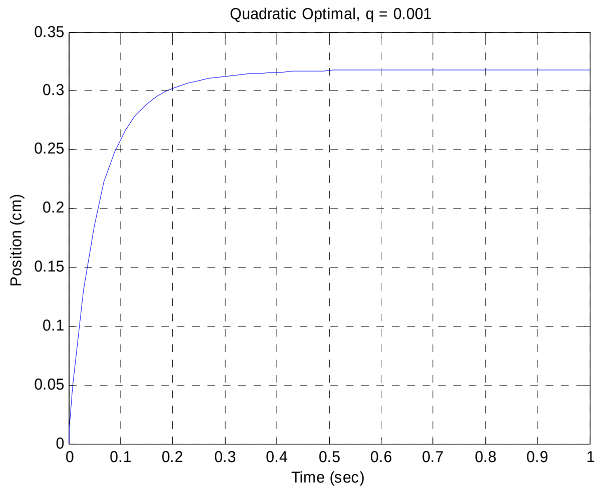


Figure 4. Step response of the new plant with the original controller (Problem 4, part e)

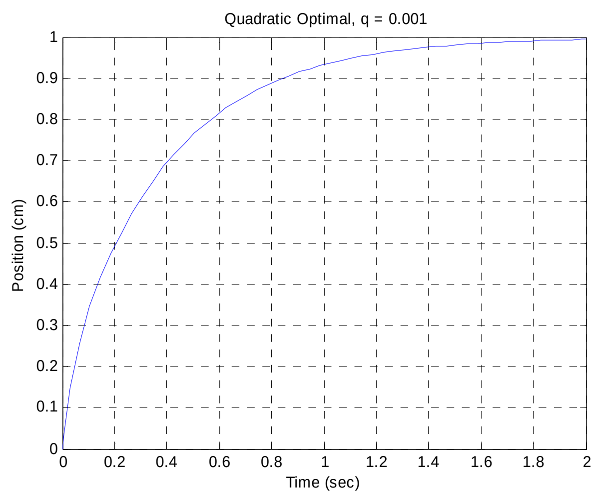


Figure 5. Step response of the new plant with the type 1 controller (Problem 4, part e)
Note that we trade off zero position error for a longer settling time.