Due: Wednesday May 3 at 3:30 PM

1) Consider the plant

$$G_p(s) = \frac{\alpha_0}{s + \alpha_1} = \frac{3}{s + 0.5}$$

where 3 is the nominal value of α_0 and 0.5 is the nominal value of α_1. In this problem we will investigate the sensitivity of closed loop systems with various types of controllers to these two parameters. We will assume we want the settling time of our system to be 0.5 seconds and the position error for a unit step input to be less than 0.1.

a) (*ITAE Model Matching*) Since this is a first order system, we will use the first order ITAE model,

$$G_o(s) = \frac{\omega_o}{s + \omega_o}$$

i) For what value of ω_o will we meet the settling time requirements and the position error requirements?

ii) Determine the corresponding controller $G_c(s)$.

iii) Show that the closed loop transfer function (using the parameterized form of $G_p(s)$ and the controller from part ii) is

$$G_o(s) = \frac{\frac{8}{3}\alpha_0(s + 0.5)}{s(s + \alpha_1) + \frac{8}{3}\alpha_0(s + 0.5)}$$

iv) Show that the sensitivity of $G_o(s)$ to variations in α_0 is given by

$$S_{\alpha_0}^{G_0} = \frac{s}{s + 8}$$

v) Show that the sensitivity of $G_o(s)$ to variations in α_1 is given by

$$S_{\alpha_1}^{G_o} = \frac{-0.5s}{s^2 + 8.5s + 4}$$

b) (*Quadratic Optimal*) For $q = 7.083$, show that the quadratic optimal closed loop transfer function is approximately

$$G_o(s) = \frac{8}{s+8}$$

Hence the results of both of our model matching methods will be very nearly the same.

c) (*Proportional Control*) Consider a proportional controller, with $k_p = 2.5$.

i) Show that the closed loop transfer function is

$$G_o(s) = \frac{2.5\alpha_0}{s + \alpha_1 + 2.5\alpha_0}$$

ii) Show that the sensitivity of $G_o(s)$ to variations in α_0 is given by

$$S_{\alpha_0}^{G_0} = \frac{s+0.5}{s+8}$$

iii) Show that the sensitivity of $G_o(s)$ to variations in α_1 is given by

$$S_{\alpha_1}^{G_o} = \frac{-0.5}{s+8}$$

d) (*Proportional+Integral Control*) Consider a PI controller with $k_p = 4$ and $k_i = 40$.

i) Show that the closed loop transfer function is

$$G_o(s) = \frac{4\alpha_0(s+10)}{s(s+\alpha_1) + 4\alpha_0(s+10)}$$

ii) Show that the sensitivity of $G_o(s)$ to variations in α_0 is given by

$$S_{\alpha_0}^{G_0} = \frac{s(s+0.5)}{s^2 + 12.5s + 120}$$

iii) Show that the sensitivity of $G_o(s)$ to variations in α_1 is given by

$$S_{\alpha_1}^{G_o} = \frac{-0.5s}{s^2 + 12.5s + 120}$$

e) (*Diophantine Equations*) Consider a controller designed using the Diophantine equations. Assume we want a type 1 system and closed loop poles both at -8.

i) Show that the controller is given by

$$G_c(s) = \frac{21.333 + 5.1667s}{s}$$

ii) Show that the sensitivity of $G_o(s)$ to variations in α_0 is given by

$$S_{\alpha_0}^{G_0} = \frac{s(s+0.5)}{(s+8)^2}$$

iii) Show that the sensitivity of $G_o(s)$ to variations in α_1 is given by

$$S_{\alpha_1}^{G_o} = \frac{-0.5s}{(s+8)^2}$$

f) Using Matlab, simulate the unit step response of each type of controller (except the quadratic optimal). Plot all responses on one graph. Use different line types and a legend. Turn in your plot and code.

g) Using Matlab and subplot, plot the sensitivity to α_0 for each type of controller (except the quadratic optimal) on ***one graph*** at the top of the page, and the sensitivity to α_1 on one graph on the bottom of the page. Be sure to use different line types and a legend. Turn in your plot and code. Only plot up to about 8 Hz (50 rad/sec) using a semilog scale with the sensitivity in dB (see below). ***Do not*** *make separate graphs for each system!*

In particular, these results should show you that the model matching methods, which essentially try and cancel the plant, are generally more sensitive to getting the plant parameters correct that either the PI or the Diophantine methods for low frequencies. However, for higher frequencies the methods are all about the same. The P controller, the simplest of the four, is always the worst.

Hint: If $T(s) = \dfrac{2s}{s^2 + 2s + 10}$, *plot the magnitude of the frequency response using:*

```
T = tf([2 0],[1 2 10]);
w = logspace(-1,1.7,1000);
[M,P]= bode(T,w);
Mdb = 20*log10(M(:));
semilogx(w,Mdb); grid;
xlabel('Frequency (rad/sec)');
ylabel('dB');
```

2) Consider the following characteristic equations

$$\Delta(s) = s^2 + bs + 1$$
$$\Delta(s) = s^3 + bs^2 + cs + 1$$
$$\Delta(s) = s^4 + bs^3 + cs^2 + ds + 1$$

a) Show that for the 2nd order system we need $b > 0$ for no RHP poles

b) Show that for the 3rd order system we need $b > 0$, $c > 0$, and $bc - 1 > 0$ for no RHP poles

c) Show that for the 4th order system we need $b > 0$, $c > 0$, $d > 0$ and $bcd - d^2 - b^2 > 0$ for no RHP poles
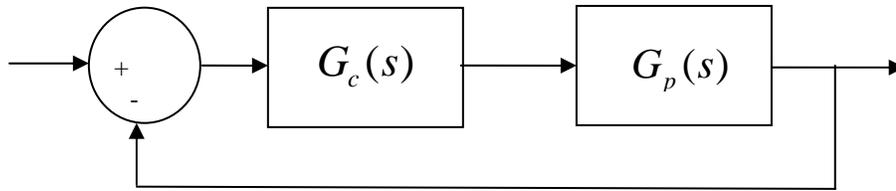
3) For $\Delta(s) = s^3 + s^2 + 2s + 2$,

a) determine if there are any poles in the RHP

b) if possible factor the characteristic equation and determine all of the poles

4) For $\Delta(s) = s^4 + 2s^3 + 4s^2 + 6s + 3$,

a) determine if there are any poles in the RHP

b) if possible factor the characteristic equation and determine all of the poles

5) Consider the following control system with plant $G_p(s) = \dfrac{1}{s^2 + s + 1}$

a) For the integral controller $G_c(s) = \dfrac{k}{s}$, use the Routh array to show that there are no poles in the RHP for $0 < k < 1$. Verify your results using Matlab (either sisotool or the rlocus command).

b) For the PI controller $G_c(s) = \dfrac{k(s+z)}{s}$, with $z > 0$, show that for no RHP poles we must have $z < 1$ and $k > 0$, or for $z > 1$ we must have $0 < k < \dfrac{1}{z-1}$. Determine the factors of $\Delta(s)$. Verify your results using Matlab (either *sisotool* or the *rlocus* command).

6) Consider the following control system with plant $G_p(s) = \dfrac{1}{s^2 + 2s + 1}$
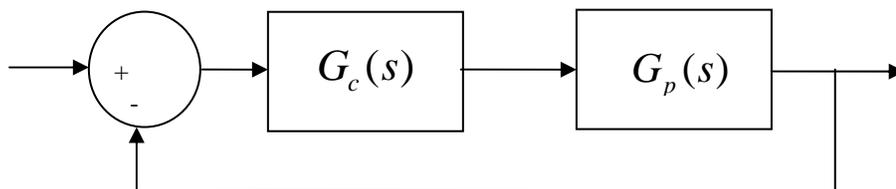
a) For the integral controller $G_c(s) = \dfrac{k}{s}$, use the Routh array to show that there are no poles in the RHP for $0 < k < 2$. Verify your results using Matlab (either sisotool or the rlocus command).

b) For the PI controller $G_c(s) = \dfrac{k(s+z)}{s}$, with $z > 0$, show that for no RHP poles we must have $0 < z < 2$ and $k > 0$, or for $z > 2$ we must have $0 < k < \dfrac{2}{z-2}$. Determine the factors of $\Delta(s)$. Verify your results using Matlab (either *sisotool* or the *rlocus* command).