ECE-320: Linear Control Systems
Homework 4

Due: **Wednesday January 13** at the beginning of lab

1) Using long division determine the first four terms in the impulse response for the following two trasfer functions

$$G_a(z) = \frac{z^2+1}{z^3+2} \quad G_b(z) = \frac{z^2+1}{z^2+z+1}$$

2) For the z-transform $X(z) = \frac{3}{z-2}$

a) Show that, by multiplying and dividing by z and then using partial fractions, the corresponding discrete-time sequence is $x(n) = -\frac{3}{2}\delta(n) + \frac{3}{2}2^k u(n)$

b) By starting with the z-transform $G(z) = \frac{3z}{z-2}$ where $X(z) = z^{-1}G(z)$, determine $g(n)$ and use the delay property to show that $x(n) = 3 \times 2^{n-1} u(n-1)$

3) For impulse response $h(n) = \left(\frac{1}{3}\right)^{n-2} u(n-1)$ and input $x(n) = \left(\frac{1}{2}\right)^{n} u(n-1)$, use z-transforms of the input and impulse response to show the output is

$$y(n) = 9\left[\left(\frac{1}{2}\right)^{n-1} - \left(\frac{1}{3}\right)^{n-1}\right]u(n-2) = 9\left[\left(\frac{1}{2}\right)^{n-1} - \left(\frac{1}{3}\right)^{n-1}\right]u(n-1)$$

Hint: Assume $Y(z) = z^{-1}G(z)$, determine $g(n)$ and then $y(n)$

4) For impulse response $h(n) = \left(\frac{1}{2}\right)^{n-3} u(n-1)$ and input $x(n) = \left(\frac{1}{4}\right)^{n+1} u(n-2)$, use z-transforms of the input and impulse response to show the system output is $y(n) = \left[\left(\frac{1}{2}\right)^{n} - \left(\frac{1}{4}\right)^{n-1}\right]u(n-3)$

Hint: Assume $Y(z) = z^{-2}G(z)$,

5) At the end of this homework is a brief review of *sisotool*. We will be using *sisotool* in this problem to design discrete-time controllers. For a discrete-time controller, all of the closed loop poles must be inside the unit circle for stability and in general, the close to the origin the poles are the fast the system moves (again, the dominant poles are those furthest from the unit circle.)

The basic transfer function form of the components of a discrete-time PID controller are as follows:
Proportional (P) term : $C(z) = K_p E(z)$ Integral (I) term: $C(z) = \frac{K_i}{1-z^{-1}} = \frac{K_i z}{z-1}$

Derivative (D) term : $C(z) = K_d(1-z^{-1}) = \frac{K_d(z-1)}{z}$

PI Controller: To construct a PI controller, we add the P and I controllers together to get the overall transfer function:

$$C(z) = K_p + \frac{K_i z}{z-1} = \frac{(K_p + K_i)z - K_p}{z-1}$$

In *sisotool* this will be represented as $C(z) = \frac{K(z^2 + az)}{z(z-1)} = \frac{K(z+a)}{(z-1)}$

In order to get the coefficients we need out of the *sisotool* format we equate coefficients to get:

$$K_p = -Ka, \quad K_i = K - K_p$$

PID Controller: To construct a PID controller, we add the P, I, and D controllers together to get the overall transfer function:

$$C(z) = K_p + \frac{K_i z}{z-1} + \frac{K_d(z-1)}{z} = \frac{K_p z(z-1) + K_i z^2 - K_d(z-1)^2}{z(z-1)} = \frac{(K_p + K_i + K_d)z^2 + (-K_p - 2K_d)z + K_d}{z(z-1)}$$

In *sisotool* this will be represented as $C(z) = \frac{K(z^2 + az + b)}{z(z-1)}$

For the following parts you are to plot your step responses and include the controller you used (just write it on the plot) and turn it in. Be sure to make sure your controller forms match up to those shown above (you controller should also be in the zero/pole/gain form, as they are above.)

a) For a system with plant $G_p(z) = \frac{z}{z^2 + 0.5z + 0.1}$ and sampling interval $T_s = 0.1$

(enter this in Matlab as Gp = tf([1 0],[1 0.5 0.1],0.1)

i. Design an I controller with a settling time less than 0.9 seconds
ii. Design a PI controller with a settling time less than 0.4 seconds
iii. Design a PID controller with complex conjugate zeros with a settling time of less than 0.2 seconds

b) For a system with plant $G_p(z) = \frac{0.5z + 0.2}{z^2 + 0.1}$ and sampling interval $T_s = 0.1$

(enter this in Matlab as Gp = tf([0.5 0.2],[1 0 0.1],0.1)

i. Design an I controller with a settling time less than 0.6 seconds
ii. Design a PI controller with a settling time less than 0.5 seconds
iii. Design a PID controller with complex conjugate zeros with a settling time of less than 0.5 seconds

Sisotool (Brief) Summary

Getting Started

- Type **sisotool** in the command window
- Click **close** when the help window comes up
- Click on **View**, then **Design Plots Configuration**, and turn off all plots except the **Root Locus** plot (set the **Plot Type** to **Root Locus** for **Plot 1**, and set the **Plot Type** to **None** for all other Plots)

Loading the Transfer Function

- In the **SISO Design** window, Click on **file → import.**
- We will usually be assigning *Gp(z)* to block *G* (the plant). Under the **System** heading, click on the line that indicates **G**, then click on **Browse**.
- Choose the available Model that you want assigned to G (Click on the appropriate line) and then click on **Import**, and then on **Close**.
- Click **OK** on the System Data (Import Model) window
- Once the transfer function has been entered, the root locus is displayed. Make sure the poles and zeros of your plant are where you think they should be.

Generating the Step Response

- Click on **Analysis → Response to Step Command** (the system is unstable at this point)
- You will probably have two curves on your step response plot. To just get the output, type **Analysis → Other Loop Responses**. If you only want the output, then only *r* to *y* is checked, and then click **OK**. However, sometimes you will also want the *r* to *u* output, since it shows the control effort for P, I, and PI controllers.
- You can move the location of the pole in the root locus plot by putting the cursor over the pink button and holding the left mouse button down as you move the pole locations. You should note that the step response changes as the pole locations change.
- The bottom of the root locus window will show you the closed loop poles corresponding to the cursor location if you hold down the left mouse button. However, if you need all of the closed loop poles you have to look at all of the branches.

Entering a Compensator (controller):

- Click on **Designs**, then **Edit Compensators**.
- Right click in the **Dynamics** window to enter real poles and zeros. You will be able to changes these values very easily later.
- You can again see how the step response changes with the compensator by moving the locations of the zero (grab the pink dot and slide it) and moving the gain of the system (grab the squares and drag them). Remember we need all poles of the closed loop system to be inside the unit circle for stability!

Adding Constraints

- Right Click on the Root Locus plot, and choose **Design Requirements** then either **New** to add new constraints, or **Edit** to edit existing constraints.
- At this point you have a choice of various types of constraints.
- <u>*Remember these constraints are only exact for ideal second order systems!!!!!*</u>

Printing/Saving the Figures:

To save a figure **sisotool** has created, click **File → Print to Figure**

Odds and Ends :

You may want to fix the axes. To do this,

- Right click on the Root Locus Plot
- Choose **Properties**
- Choose **Limits**
- Set the limits and turn the **Auto Scale** off

You may also want to put on a grid, as another method of checking your answers. To do this, right click on the Root Locus plot, then choose **Grid**

It is easiest if you use the zero/pole/gain format for the compensators. To do this click on **Edit → SISO Tool Preferences → Options** and click on **zero/pole/gain**.