Duke University
Edmund T. Pratt, Jr. School of Engineering

ECE 141 Spring 2007

Test II

Michael R. Gustafson II

Name (please print)____

In keeping with the Community Standard, I have neither provided nor received any assistance on this test. I understand if it is later determined that I gave or received assistance, I will be brought before the Undergraduate Judicial Board and, if found responsible for academic dishonesty or academic contempt, fail the class. I also understand that I am not allowed to speak to anyone except the instructor about any aspect of this test until the instructor announces it is allowed. I understand if it is later determined that I did speak to another person about the test before the instructor said it was allowed, I will be brought before the Undergraduate Judicial Board and, if found responsible for academic dishonesty or academic contempt, fail the class.

Signature:____

Instructions for Paper-Based Sections

Be sure to put your name on **each page** of the test of any scratch paper you are turning in. Clearly indicate where each part is solved. You will be turning the test in as individual problems. This cover sheet should be stapled along with Problem I - every other part will consist of the page from the test and any extra pieces of paper used. For this reason, it is critical that you have no more than one problem's work on any given page. To turn in the test, you will check to make sure your name is on every page, then staple together any relevant scratch work, and finally you will turn in the six piles to the folders in the front of the room.

Instructions for Computer-Based Sections

All your files for this test will be placed in a directory on your OIT account. There are two scripts you will be running to set everything up. The first - `StartTest2` - will create a folder called `ECE141TEST2` in your account and then set the permissions such that I can look at the files. You *must* make sure all your scripts, worksheets, and graphs end up in this folder. The second - `EndTest2` - will send me a snapshot of the directory contents and lock the directory from further changes. After I receive the e-mail, I will copy the directory contents to another location and delete the originals.

Be sure to use the stated names for files in those problems requiring computational solutions. Also, if you use Matlab or Maple to generate answers for problems, you should write a script or worksheet called `ScratchProblemN.whatever` where N is the problem number and `whatever` is the appropriate extension for the program (generally `.m`, `.mw`, or `.mws`).

To run the `StartTest` script, log into your UNIX account and type:

```
~mrg/public/ECE141S07/StartTest2
```

Similarly, when you are finished and ready to lock your directory, type:

```
~mrg/public/ECE141S07/EndTest2
```

Name (please print):
Community Standard (print ACPUB ID):

Problem I: [8 pts.] System Stability I

The overall transfer function for a system is determined to be:

$$T(s) = \frac{s^2 + 2}{s^5 + 4s^4 + 2s^3 + 8s^2 + 4s + 1}$$

(a) Clearly, and by hand, generate a Routh array for this system.

(b) Based on the array, where are the poles for this system?

(c) Write a script in Matlab called `Problem1.m` to do the following:

(1) Generate an object to represent the system above.
(2) Create a pole-zero plot of the system above.

Add the following lines to the end of your code to generate and save the graph:

```
title('Pole Zero Plot for NETID')
print -deps PoleZeroPlot
```

where `NETID` is *your* NET ID.

Name (please print):
Community Standard (print ACPUB ID):

Problem II: [8 pts.] System Stability II

The overall transfer function for a system with two controllable gains is determined to be:

$$T(s) = \frac{1}{s^3 + (K_1 + 1)s^2 + 2s + K_2 + 3}$$

(a) Clearly, and by hand, generate a Routh array for this system.

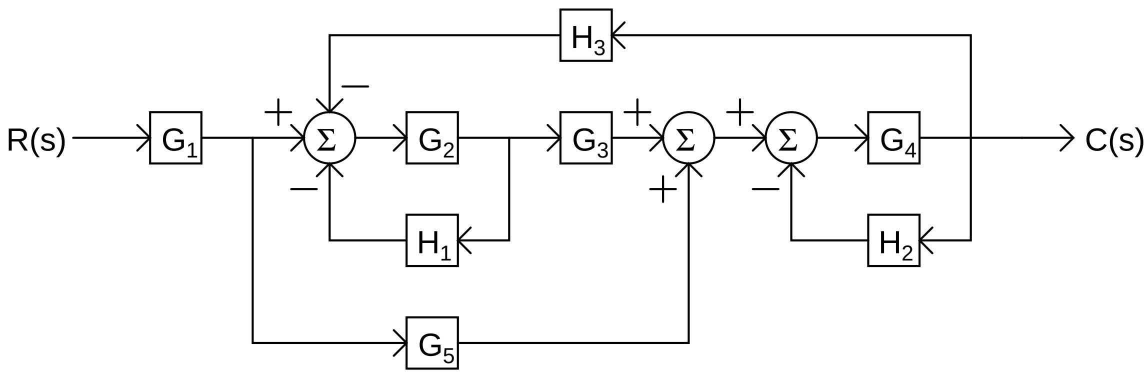
(b) Determine all the relational equations for K_1 and K_2 that must be satisfied to produce a stable system.

(c) Sketch a graph with K_1 on the x-axis and K_2 on the y-axis and shade in the region of stability as a function of the two values. That is, sketch the region that represents the intersection of the relational requirements determined above.

Name (please print):
Community Standard (print ACPUB ID):

Problem III: [12 pts.] System Simplification

Given the system below:



(a) Clearly draw a signal flow diagram for the system. Be sure to indicate where each node is on the system.

(b) Use Mason's Rule to determine the overall transfer function $T(s) = C(s)/R(s)$.

Name (please print):
Community Standard (print ACPUB ID):

Problem IV: [24 pts.] System Design I

A unity feedback system such as shown in Figure 7.3(b) on page 372 of the Nise book has a forward transfer function of:

$$G(s) = \frac{1}{(s+2)(s+4)}$$

A gain block with gain K is placed in the forward path between the summation block and $G(s)$.

(a) What type of system is this?

(b) Over what range of K is the system stable?

(c) What is the value of the appropriate finite static error constant with respect to K?

(d) What is the value of the steady state position error for a unit input with respect to K?

(e) Is there a value of K that will yield a settling time of 0.5 sec? If so, what is it? If not, why do you believe that?

(f) Is there a value of K that will yield an underdamped response such that the %OS is 15? If so, what is it? If not, why do you believe that?

(g) Is there a value of K that will produce a steady state position error that is 1% assuming a unit input? If so, what is it? If not, why do you believe that?

(h) Model the system in Simulink and call the model `SimModel.mdl`. Represent K and G in separate blocks and make sure those blocks are large enough that you can see their contents on the screen. Use an input function of:

$$r(t) = \left(2 + 4e^{-3t}\right) u(t)$$

where $u(t)$ is the unit step function, and simulate the system for 10 seconds using the value of K that produces a critically damped system. Next, write a script called `SimPlotter.m` that will produce a graph, that shows the input $r(t)$ as a solid black line (that is, 'k-' in the `plot` command), the output $c(t)$ as a dashed blue line ('b−−'), and the error $r(t) - c(t)$ as a dotted red line ('r:'); each should be a function of time. Add the following lines to the end of your code to generate and save the graph and model plot:

```
title('Simulink Output for NETID')
print -deps SimPlotterPlot
print -sSimModel -deps SimModelPlot
```

Name (please print):
Community Standard (print ACPUB ID):

Problem V: [24 pts.] System Design II

The device for this problem comes from Problem 7.50 on page 243 of Nise. The transfer function given is for an *open-loop* controller - that is, no feedback. Assume we have put a prefilter block between the input voltage and the summation block such that that the input voltage will actually be used to set an angular *position* instead of angular velocity.

When the system is used with no feedback, the transfer function between the angle of the robot arm and the voltage applied to the motor is thus:

$$T_{OL}(s) = \frac{\Theta(s)}{V_i(s)} = \frac{K}{(s+10)(s^2+4s+10)}$$

For this system:

(a) Find the poles and zeros of the transfer function.

(b) Estimate $\%OS$, T_s, T_p, and T_r. How accurate do you believe these values are, any why?

(c) Write a script in Matlab called `OLA.m` that will set up an object representing this transfer function (with K=1) and then use Matlab to determine the response to a unit step input. Use this to show what the simulation gives for $\%OS$, T_s, T_p, and T_r. To print the graph, once you are finished adding the necessary information, type the following at the Matlab command line:

```
title('Step Response for NETID')
print -deps StepResponsePlot
```

where `NETID` is *your* NET ID.

(d) What type of system does this represent? (be careful! and complete...)

(e) Determine the appropriate finite static error constant and steady state error assuming unit input.

Name (please print):
Community Standard (print ACPUB ID):

Problem VI: [24 pts.] System Design III

Now, we are going to add a simple feedback loop to the system, again looking at the *angle* as a function of voltage. This means that instead of the overall transfer function, we know the formula for the forward transfer function of a unity feedback system with gain control:

$$G(s) = \frac{K}{(s+10)(s^2+4s+10)}$$

For this system:

(a) Determine the range of K that will keep the system stable.

(b) What type of system does this represent?

(c) Determine the appropriate finite static error constant and steady state error as functions of gain assuming unit input.

(d) Determine the value of K that will produce a steady state position error of 25% relative to a unit input. Where are the poles for this system?

(e) Determine the value of K that will produce a marginally stable system as well as the frequency of oscillation of the system given that gain.