ECE 3510   homework   RL5

These problems should be done using MATLAB or some other program that creates root-locus plots. Refer to a help sheet attaced to this homework to get started. You will need to compare shapes of root-locus (RL) plots and speeds of step responses. Unless you manually set your plot limits, Matlab will automatically scale on your plots. This can make it very hard to compare them or to determine the "45°" line.

1. a) Homework RL2 problem 1c. $\mathbf{G}(s) := \dfrac{1}{s \cdot (s+2) \cdot (s+4)}$ Experiment with moving the pole at -4.

   i) Describe what happens when you move it to the left.
   ii) Describe what happens when you move it to the right, especially when you move it right of -2.
       Put the pole back at -4.

   b) Describe what happens when you add a fourth pole left of the jω axis. Try different positions. Erase the added pole.

   c) Describe what happens when you add a zero left of the jω axis. Try different positions.

For the remaining problems:

   Print one or more plots for **each** problem. Each plot should be labeled clearly on the same page as the plot. If you added poles and/or zeros (a compensator) make sure it is clear which one(s) were added or what your $\mathbf{C}(s)$ is.   Plot of the root locus should always show the point(s) on the root locus you determined to be the "best" position for the CL pole(s) with squares (like Matlab does). Indicate the gain (k) used.
   "Best" = Fastest step response to get to and stay within 4% of the desired final response (usually 1 for step response). Limit overshoot to about 4%. To determine speed information, refer to the step response curve. Prints of those curves are nice, but not required. Be careful, if Matlab changes the scales on your plots it can be very hard to compare them.

2. a) Homework RL2 problem 1c (see above for $\mathbf{G}(s)$). Add a compensator to your system. This compensator will add one pole and one zero to the open-loop (OL) transfer function. The new pole must lie somewhere between -10 and +2, you choose where. Same goes for the zero. Choose the locations of your added OL pole and OL zero and gain (k) to get the "best" response.

   b) Homework RL2 problem 1d. $\mathbf{G}(s) := \dfrac{s+7}{s \cdot (s+2) \cdot (s+4)}$ Just find the "best" gain and print an RL plot.

   c) Homework RL2 problem 1d. Repeat part a) above for this system, only this time your added pole and zero are limited to -16 to +2.

3. a) Homework RL3 problem 1a $\mathbf{G}(s) = \dfrac{s \cdot (s+1)}{(s+2)^2 \cdot (s+3)}$ Just print an RL plot. Note that the step response goes to zero, why?

   b) Homework RL3 problem 1b. $\mathbf{G}(s) = \dfrac{(s+3)}{s \cdot (s+9)^3}$ Just find the "best" gain and print an RL plot.

   c) Homework RL3 problem 1b. Add a compensator. Your compensator may have up to 2 poles (0, 1, or 2) and they may be complex. Same for the zeros. All must lie between -20 and +2 and -12j and +12j. Try to choose the best possible poles and/or zeros, find the best gain and plot like you did for 1b. Don't spend too much time here, quickly decide between poles and zero, add 1 or 2, and tweak your gain. (2 zeros and ∞ gain looks good but isn't really practical.)

4. Homework RL4
   problem 1. $\mathbf{G}(s) = \dfrac{s+1}{s^2 - 2 \cdot s + 2}$ Confirm the departure angles (With a small gain, place a CL pole very close to $1 + 1j$ and figure out the angle from its location.). Find the true Im-axis crossing point and the gain at that point. Find the break in point.

5. a) Enter in the plant $\mathbf{G}(s)$ of the crude servo: $\mathbf{G}(s) = \dfrac{1643}{s \cdot (s + 16.64) \cdot (s + 53.78)}$
      Add a PI compensator (a pole at 0 and a zero at -0.1).
      Find the gain for a 0.707 damping factor. Record the time it takes the step response to reach 1 the first time.
      You don't need any prints or plots.

   b) Add a zero at -16.64 and repeat, no prints or plot..

   c) Adjust the zero location further to the right while pushing the gain and watching the step response to find an even better combination. Again record the time it takes the step response to reach 1 the first time and save.
      Print the Root locus **and** the step response for this case.

   d) You will be using this device to build widgets. The time per widget is 10 times the time you recorded in each case. Your profit per widget is 1 cent. Your fixed costs per 8-hour day is $250. Describe the results to your company of each of the 3 compensators above.

6. Create the most interesting root locus plot that you can with no more than 10 poles and 10 zeros. Have some fun with this. You can go over 10 poles and zeros if you want, but the fewer the better to make the figure you want.

Answers

1. a) i) If the pole at -4 is moved further to the left:
   Curved part of RL straightens out because the centroid moves to the left.
   The gain required to reach any given damping factor goes up.
   The breakaway point moves closed to -1

   ii) If the pole at -4 is moved to the right the opposite effects occur, plus, if moved right of zero at -2 effects get worse, fast.

   iii) Adding a pole makes matters worse. It makes the curves bend down and makes a new curve bend off to the left.

   iv) Adding a zero makes matters better. Especially if placed on top of the pole at -2. Further to the right of that and it tends to make a dip in the step response. If that can be tolerated, then increased speeds are possible with increased gain.

2. a)
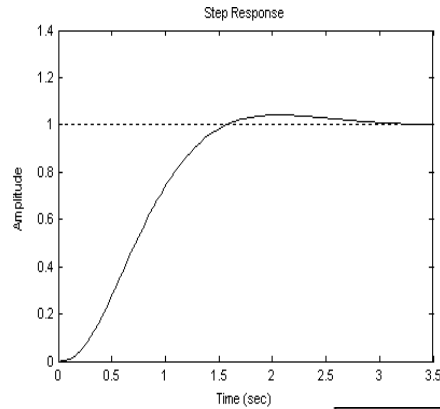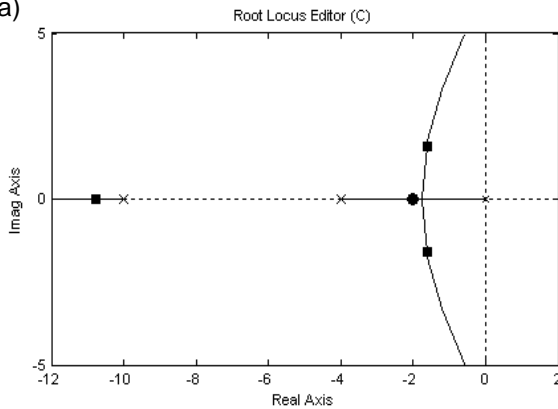
$$C(s) := \frac{s+2}{s+10}$$

Gain is about 56

Your answer may be different, especially, your zero may be to the right of -2.

b) See answers for RL2, gain is about 0.92   c) you're on your own

3. a) See answers for RL3   System has a zero at the origin.

   b) See answers for RL3

   Gain of 346 puts the CL poles on the 45º line, BUT, because of the zero at about -1.3, a gain of 1000 is a almost twice as fast.

   A gain of 800 with a feedforward gain of 1.04 will make it 4 times faster, but will result in a long-term +4% error in the output. If only we could limit the feedforward gain to the first part of the response. Ahhh... maybe that explains the weird feedback / feedforward scheme in lab 5b.

3b step response w/gain = 1000

3b step response w/gain = 800 and 1.04 feedforward

c)

gain = 253
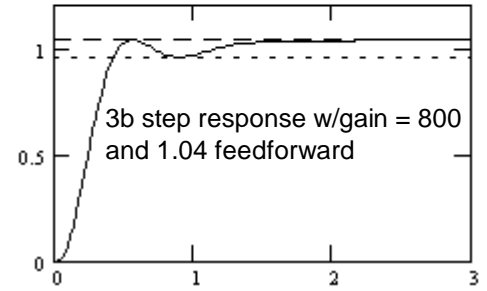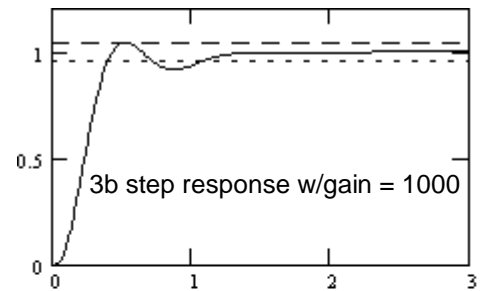
$$C(s) := \frac{s+9}{1}$$

gain = 253

$$C(s) := \frac{s+9}{1}$$ Your answer may be different

With gain = 253, overshoot is too high, gain = 215 meets requirements.
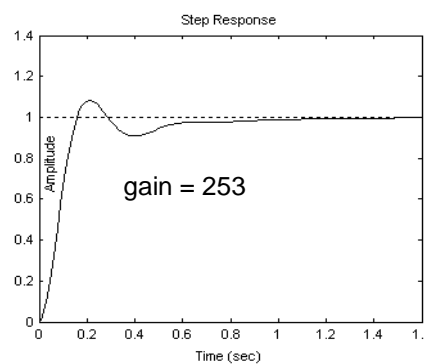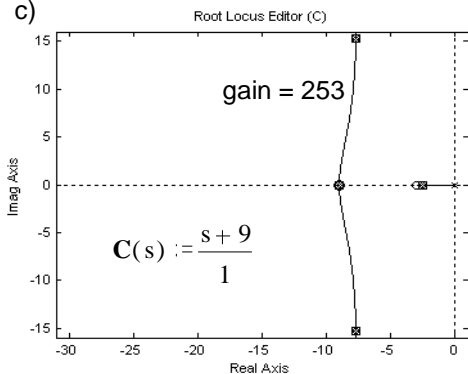
4. My close point: $s_1 := .945 + 1.1 \cdot j$ $\arg\left[s_1 - (1+j) \right] = 118.811 \cdot \deg$ close to 117 deg

   Place CL pole at crossing, gain = 2
   Move CL pole to break in point, location shows at -3.23, at a gain of 8.47.

5.d) a. You go broke at $168/day.   b. You make $101/day.
   c. You make $161/day. Your answers may be quite different

Matlab's SISO tool (single-input/single-output tool)
The version I used was R2016b, but this may still be of some help for other versions.

The SISO tool can be used to draw and manipulate rool-locus plots of single-input / single-output systems. It is part of the Matlab Control System Toolbox.

To use the SISO tool, you first need to create the open-loop transfer function "object" in Matlab. There are several ways to do this, but I recommend this way:

1. Define the variable "s" as a special TF model `s = tf('s');`
    This only needs to be done once, after that any other expression of s will automatically be interpreted by Matlab as a transfer function.
2. Enter your transfer function as a rational expression in s For example,   `G = s/(s^2 + 2*s +10);`
    Now G is a "transfer function object" of the transfer function

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

Now type: `sisotool(G)`

3. You can close the Bode plot views by: View -> Float . Now you can close the two Bode windows, leaving the Root Locus and Step Response windows. View -> Left/Right will return to larger windows.
4. The little red squares (or circles) on the root locus (RL) plot show the locations of the closed-loop (CL) poles. You can grab any of these and move then to a new location. Note how the other(s) move as well. Note the effects on the step response. Also note the information given at the bottom of the window as you grab the CL pole.
5. If you choose the ROOT LOCUS EDITOR tab, you can add a pole or zero to the real axis by: click X or O button –> click on plot where you want the pole or zero. You can later drag it left and right. You can erase it with the eraser tool.   You can add complex poles or zeros using the buttons that look like x/x or o/o fractions.
6. If you right-click on the root locus plot, you can open "Edit Compensator" and see the gain as well as add, delete or change compensator poles or zeros. (Right-click within the "Dynamics" window to get options.) Note: Each time you add a pole or zero it adds it at -1. Then you have to select it and change the position. The new pole will not automatically be selected. (Big pain in posterior.)
7. If you've added any poles or zeroes, then you may have noticed that the format of the compensator is a bit weird. They are in a (1 - s/p) or (1 - s/z) form. To fix this and get gain numbers that will match mine, Choose the CONTROL SYSTEM tab –> Preferences -> Options, click the Zero/pole/gain option to change the format of the poles and zeros from (1 + s/p) to (s + p).
8. If you move any compensator poles or zeros into the right-half plane the gain suddenly becomes negative. I don't know why or how to fix this, just keep it in mind. If you figure out how to eliminate this weirdness, let me know.
9. In order to effectively evaluate changes, you will need to inhibit Matlab's constant rescaling of the plot: right-click anywhere on root-locus plot area –> Properties –> Limits –> uncheck the Autoscale boxes and set limits to match the aspect ratio of your window.

You can't modify the Plant poles and zeros (at least as far as I know).

Play with this until you are ready to start the Homework.

Start Homework RL5. Read up through problem 1a and refer back to homework RL2. Back in the Matlab Command window, type: `G = 1` to create a very simple transfer function. That way all your poles and zeros can be manipulated in the SISO tool. type: `sisotool(G)`. Reset you preferences as above (items 3, 7, & 9). (SISOtool used to have a way to import a new G, i don't see that option anymore.)   Add poles at 0, -2, and -4 so that C(s) = 1 x 1/s(s+2)(s+4). You are now ready to work problem 1a.

More information
Google, sisotool help and Matlab help.