1. Keep working on your project!

2. Consider the system from HW#16:

\[ W(s) \quad + \quad T(s) \quad G_c(z) \quad G_{na}(z) \quad 10 \frac{s}{s+8} \quad Y(s) \]

Recall that we have already designed a lead compensator, \( G_c(z) \), to meet the following specifications:

\[ t_s \leq 0.4 \text{ sec and } M_p \leq 2\% \]

(see solution to HW#17 on the Web for one possible \( G_c(z) \)).

a) Now design a \( G_{na}(z) \) to be put in series with \( G_c(z) \) so that the compensated system meets the above specs plus the added spec that \( e_s \) due to a unit ramp \( \leq 1/50 \).

b) Sketch the compensated root locus.

c) Simulate (See Below or use Simulink) a step and ramp response for your original closed-loop system, your lead compensated design from HW#16 (see web solution), and your new lead/lag system using MATLAB and lsim(). Measure \( t_s \), \( M_p \), and \( e_s \) (both step and ramp) for all three cases. Does your design meet specs? (Hint: for the lead lag system, you may have to increase your time axis to measure \( e_s \)).

Turn this in on Wednesday since we didn’t get to the example

2. a) Now design a compensator for the system in part a) that meets all of the following specifications (Hint: Think PID):

\[ t_s \leq 0.65 \text{ sec} \]
\[ M_p \leq 10\% \]
\[ e_s \text{ due to a ramp} = 0 \]
\[ e_s \text{ due to a unit parabola} \leq 1/50 \]

b) Sketch the compensated root locus.

c) Simulate a step, ramp, and unit parabola responses for your lead/lag system using MATLAB. Measure \( t_s \), \( M_p \), and \( e_s \) (for step, ramp and parabola (a unit parabola is defined as \( 0.5t^2u(t) \)).

Here’s how to use Matlab to simulate the step response following closed-loop system (or, use Simulink):

Let’s assume in Matlab we already have \( G_{na}(s)G(s) = \text{num}/\text{den} \) and \( G_c(s) = \text{numc}/\text{denc} \)

(Note: the conv() command in Matlab multiplies two polynomials together!!! Also note that the commands cloop() and feedback() can also be used to find the closed - loop transfer function once you have found num_open_loop and den_open_loop )

```matlab
num_open_loop=conv(num,numc)  %Find the total open-loop numerator of G_cG_{na}G(s)
den_open_loop=conv(den,denc)  %Find the total open-loop denominator of G_cG_{na}G(s)
n_minus_m=length(den_open_loop)-length(num_open_loop)  %Find difference in order between numerator and denominator
num_closed_loop=num_open_loop  %The closed-loop and open-loop numerators are the same
num_same_order=[zeros(1,n_minus_m) num_closed_loop]  %Pad the closed-loop numerator with zeroes so we can add it
```
```
» den_closed_loop=num_same_order+den_open_loop
» Ts=0.01
» t=[0:100]*Ts;
» u=ones(1,length(t));
» y=lsim(num_closed_loop,den_closed_loop,u,t);
» plot(t,y)
% to the open-loop denominator to form the closed-loop denominator
% then select the sampling time to be 10 msec
% time scale is 1 second
% u (the input) is a unit step
% Use lsim to simulate the step response of the closed-loop
% system, Y(s)/W(s) and plot the results
```