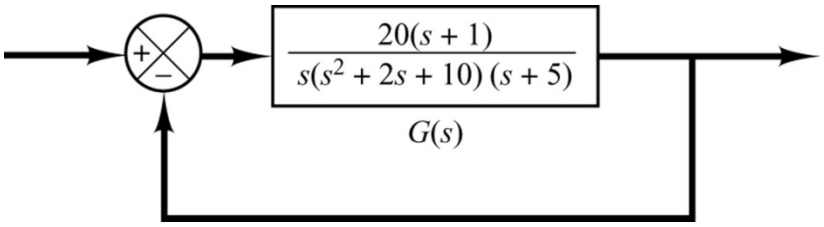
ESE 406/505 & MEAM 513 - SPRING 2013
HOMEWORK #9
DUE by Wednesday 2013-04-10 (Late Pass Monday 2013-04-15)

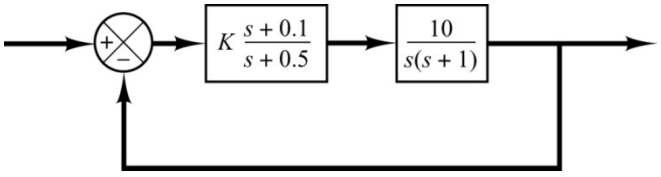
1. Solve the following problems in the textbook:

   a. B-7-25: Use MATLAB to make a bode plot of the loop gain for the system shown below. What are the gain and phase margins?



   Answers: The gain margin is 9.9 dB at 4.0 rps. The phase margin is 103 deg at 0.44 rps. This system is practically begging for lag compensation! We have plenty of phase at low frequency, so we could get much better tracking performance with lag shaping, pushing the crossover frequency up by a factor of about 2.

   b. B-7-27: What value of K yields a phase margin of 50 deg? What is the gain margin for this value of K? Submit a bode plot of the loop gain when K takes this value and label the margins on the plot.



   Hint: Use MATLAB to make a bode plot with K=1. Then figure out what gain is required to get a 0 dB crossover that will yield a phase margin of 50 deg.

   Answers: The required gain is K=0.266. There is no gain margin (the phase is always above -180 deg). You could make a root locus and confirm that it never passes into the right half-plane.

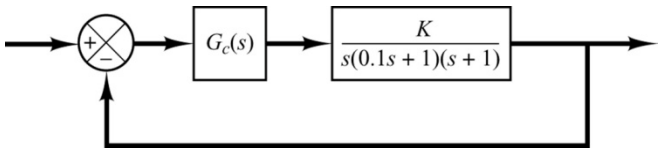
   c. B-7-32: For the system shown below, design a lead compensator, $G_C(s) = \frac{T_1 s + 1}{T_2 s + 1}$, and choose a plant gain, K, that will yield the following properties:

      i. Phase margin of 45 deg.

      ii. Gain margin of at least 8 dB

iii. Velocity error constant of at least 4.0 sec^{-1}. This means that at low frequency, the loop transfer function gain is at least $\frac{4}{\omega}$.

Submit a bode plot of the loop transfer function that includes your final design. Label the margins. Also include a closed-loop step response. You don't have to make a unit-ramp response.



Answers: With the given form of $G_C(s)$, the velocity error constant is simply K, so we know that we need $K \geq 4$. Try K=4 and then make a bode plot without the lead compensator. You will see that you need to add phase at the crossover frequency, which should guide your choice of T_1 and T_2. One pair of values that satisfy the requirements is $T_1 = 0.78\text{sec}$ and $T_2 = 0.16\text{sec}$, which is a lead ratio of about 5.

d. B-7-34: For the system shown below (the plant is our old friend from the last 3 homework assignments!), design a lead-lag compensator, $G_C(s) = K\frac{s+z}{s+p}\frac{T_1 s+1}{T_2 s+1}$, that will yield the following properties:

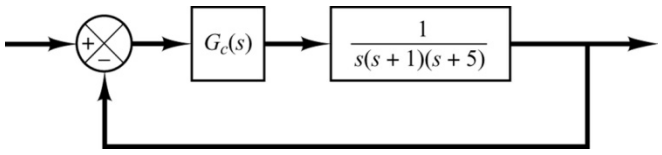
i. Phase margin of 60 deg.

ii. Gain margin of at least 8 dB.

iii. Crossover frequency between 2.0 and 3.0 rps.

iv. Velocity error constant of at least 20.0 sec^{-1}. This means that at low frequency, the loop transfer function gain is at least $\frac{20}{\omega}$.

Submit a bode plot of the loop transfer function that includes your final design. Label the margins. Also include a closed-loop step response. You don't have to make a unit-ramp response. Finally, make a closed-loop frequency response. What is the command response bandwidth?



Answers: There are lots of possible answers, e.g., $G_C(s) \approx \frac{s+0.2}{s+0.01}\frac{2.7s+1}{0.13s+1}$. Notice

that to achieve the very generous phase margin, we have to use a very high lead ratio of just over 20 (2.7/0.13 > 20). This much lead is often unrealistic, as it can cause actuator saturation for modest inputs and amplify noise too much. Practical limits on the maximum lead are very system dependent, but they are often the determining factor in the achievable closed-loop bandwidth.

2. (**ESE 505 & MEAM 513 Only**) In this problem, we do a more sophisticated form of "system identification" to figure out the dynamics of an unknown system. We will use the identified dynamics to design a feedback controller.

   The unknown system has been implemented in SIMULINK for you (`HW09_Problem2.mdl`). As before, we think that the transfer function of the system might be a simple first-order system plus some time delay, $G_P(s) = \dfrac{Ae^{-Ts}}{s+a}$, but the system also has some noise. You can do some simple experiments with the "step" input that is included in the model to get a sense of the magnitudes of the parameters, but you can see that the noise creates problems for this approach.

   Instead of using step inputs, we could excite the system with a sinusoidal input and try to measure the magnitude and phase of the output, as we did last week. But the noise would also make these estimates inaccurate, particularly at higher frequencies, where we think the amplitude of the response will be low.

   There is a better way to use experiments to estimate the frequency response of a system. We use what is called a "chirp" input, which is essentially a sinusoidal input whose frequency increases with time. Then we take the Fourier Transform of the input and output signals, which gives us the following relationship:

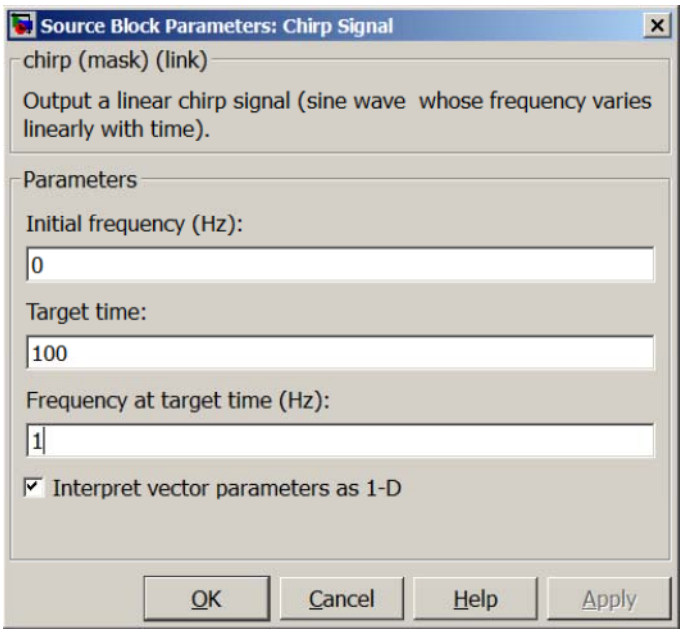
   $$Y(j\omega) = G_P(j\omega)U(j\omega) + N(j\omega)$$

   Next, we multiply the output signal by the conjugate of the input, $U^*(j\omega)$, and do some sort of averaging procedure[1], which we denote with an overbar: $\overline{(\,)}$. One way to think of the averaging is that we are replacing each term, at each ω, with some sort of weighted average over frequencies near ω. Another way to think about it is that we repeat the experiment many times and average the Fourier transforms over all these repetitions. In either case, if the noise is not correlated with the input, on average, then we find that the term $\overline{U^*(j\omega)N(j\omega)}$ is zero, leaving

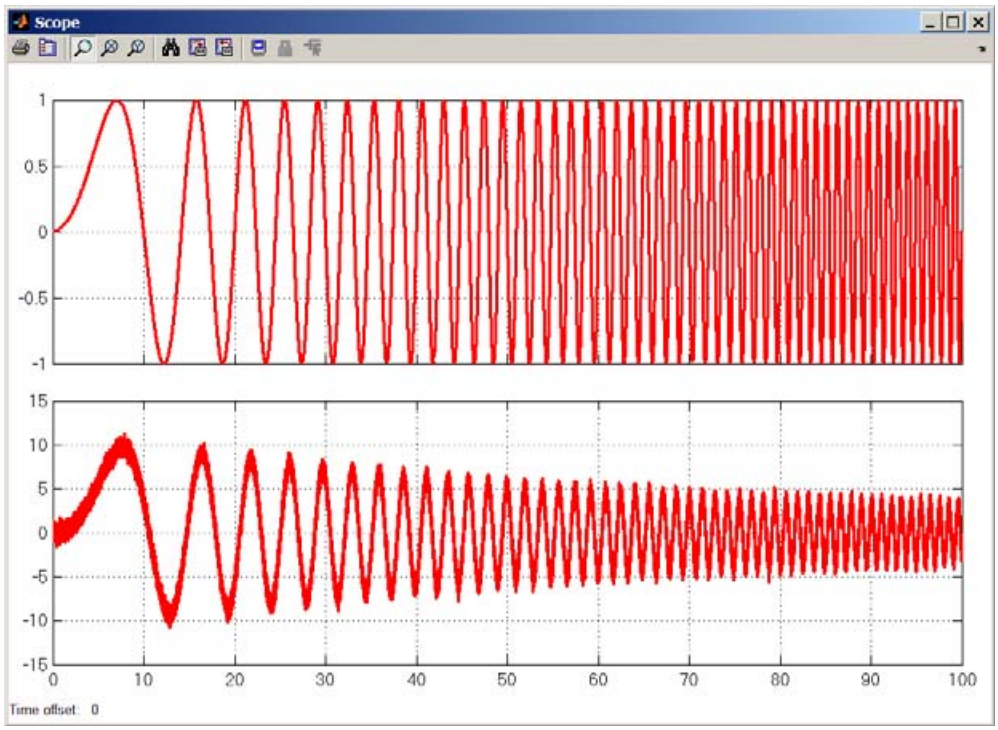
[1] The most common way of averaging is the method used by MATLAB's `tfestimate` command. It breaks the single measured time history of input and output into multiple pieces and computes the average over these pieces, as if they were distinct, independent measurements. There are some theoretical issues with this approach, but it is generally adequate and very widely used.

$$G_P(j\omega) \approx \frac{\left[\overline{U^*(j\omega)Y(j\omega)}\right]}{\left[\overline{U^*(j\omega)U(j\omega)}\right]}$$

The beauty of this method is that noise is automatically removed by the averaging operation, leaving us with an unbiased estimate of the frequency response. Fortunately for us, MATLAB & SIMULINK have all these tools available as built-in functions. Save the SIMULINK model under a new name and modify it by replacing the step input with a chirp node, which is found in the "Sources" library. Open the chirp node change the settings as shown in the figure at below.

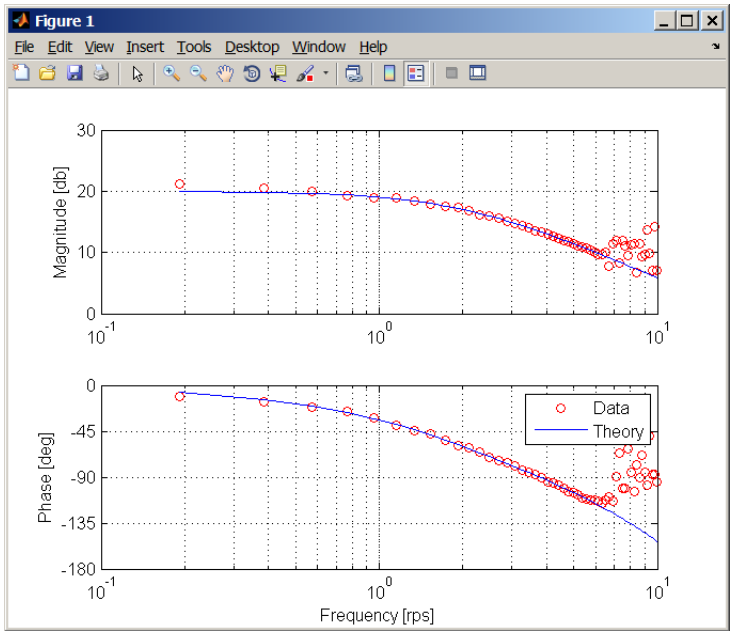


Change the simulation duration to 100 seconds and run the simulation. You should see something like this on the scope:



The top graph is the chirp input. The bottom graph is the output. Notice that the amplitude is dropping as the frequency increases, as we expect for a first-order system.

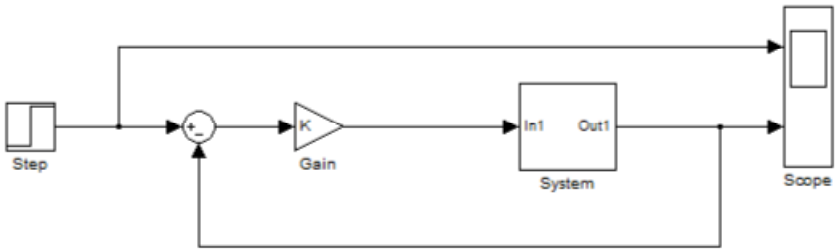
Now open and study the script `HW09.m`. You should use `help tfestimate` to learn more about this command, which is the magic command that performs the frequency response estimation, including the averaging discussed above. Enter your guess at the system parameters, and run the script. When you have a good guess, you should see something like the figure below:



Notice that the data gets very noisy for frequencies larger than about 6 rps. This is because we stopped the chirp at 1 Hz, so there isn't any information in the data at those frequencies, just noise. You should trim the frequency response data to eliminate frequencies where the data are no good.

a. SUBMIT your final frequency response graph (like the one above) and write your identified values of A, a, and T on the graph.

Next, use the `rlocus` and `rlocfind` commands, with a first-order Pade approximation, to design a proportional feedback control system with a closed-loop damping ratio of 0.5 (You should get K of about 0.3). Now modify the original block diagram so that it looks like the figure below. Run the simulation for 5 seconds and see what happens.



b. SUBMIT an image of the output of the scope showing the closed-loop step response.

Something has obviously gone wrong at high frequency! Let's try to understand the dynamics at higher frequency. Perhaps the system has a high-frequency elastic mode, like the one in the notes from last week. That is, perhaps the system dynamics are actually of the following form[2]:

$$G_P(s) = \frac{Ae^{-Ts}}{s+a}\left[\frac{(1-K_B)s^2 + 2\varsigma\omega_B s + \omega_B^2}{s^2 + 2\varsigma\omega_B s + \omega_B^2}\right]$$

Go back to the SIMULINK model with the chirp input and change the "frequency at target time" to 10 Hz and run the simulation again. Now update the script to add the additional system dynamics and again try to play with the parameters to get a good match to the data. Be sure to adjust the ranges of the axes on the graphs so you can see all the data. And again clean up the data by eliminating points where the data are obviously corrupted by insufficient input energy.

---

[2] In the notes, we considered a system with $(1+K_B)s^2 + \cdots$ in the numerator. For the system considered in this problem, the elastic modes are destabilizing with $(1-K_B)s^2 + \cdots$.

c. SUBMIT the frequency response graph, as in part (a), and write the identified values of the elastic mode constants on the graph.

Now, design a notch filter to prevent instability of the elastic mode using the same gain you chose above. You can do this using the model of the plant that you have extracted from the frequency response data, together with a notch-plus-gain compensator. Or, you could simply use the raw frequency response data (gain and phase) and add the gain and phase of your compensator at the same frequencies to get the loop bode plot. ***We can use experimental data to do control system design in the frequency domain without ever explicitly writing down a plant transfer function!***

d. SUBMIT a loop bode plot of your modified design, showing the gain and phase margins. Write the transfer function of your compensator on the graph.

One more thing to do:

e. SUBMIT a final closed-loop step response for the closed-loop system.