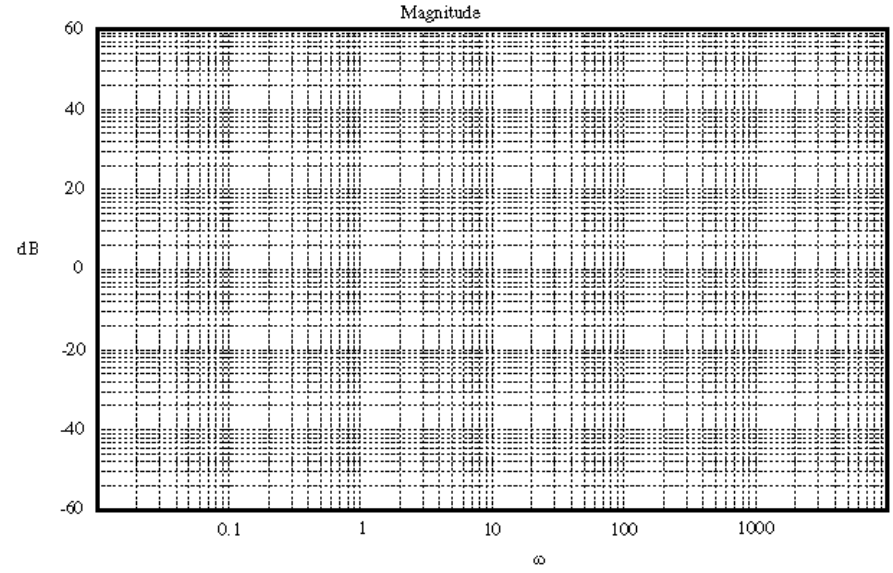
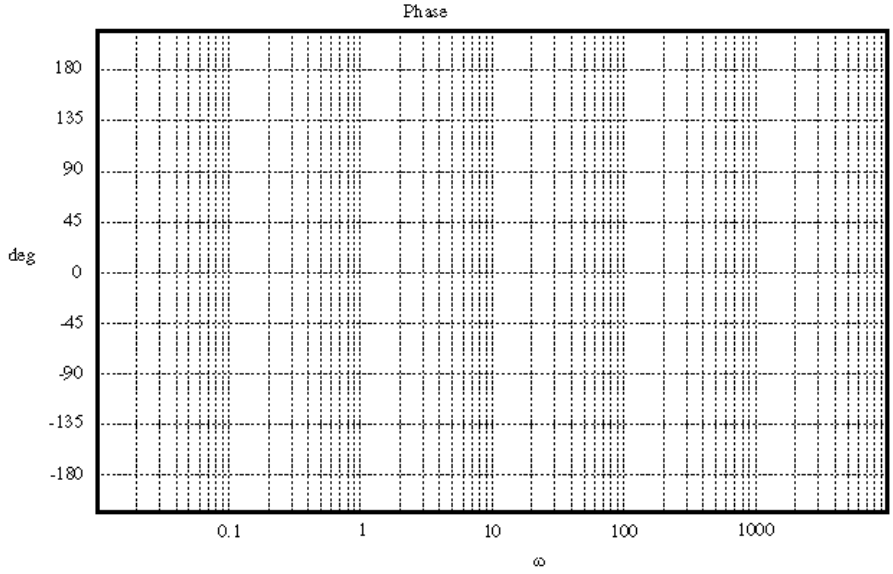
ESE 406 - SPRING 2012
HOMEWORK #9
DUE FRIDAY 30-Mar-2012 by 2PM (Monday, 2-April-2012 with late pass)

Problem 1. Submit your solutions to the following problems in the textbook. You should print this page out 5 times and use the "Bode Paper" below to attempt a hand sketch of each system before using MATLAB. Then use the Matlab `bode` command to check your answers. Submit a print-out of the Matlab result, with asymptotes drawn in by hand.

- Problem 6.3 (a)
- Problem 6.3 (d)
- Problem 6.3 (f)
- Problem 6.3 (i)
- Problem 6.5 (e)





Problem 2 (**ESE 505 & MEAM 513 Only**). In this problem, we will repeat the key ideas from last week's homework, designing a proportional feedback control for an unknown system. This time, we will do a more sophisticated form of "system identification" to figure out the dynamics of the unknown system.

The unknown system has been implemented in SIMULINK for you (HW09_Problem2.mdl). As before, think that the transfer function of the system might be a simple first-order system plus some time delay: $G(s) = \dfrac{Ae^{-Ts}}{s+p}$, 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. But the noise would also make these estimates inaccurate.

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:

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

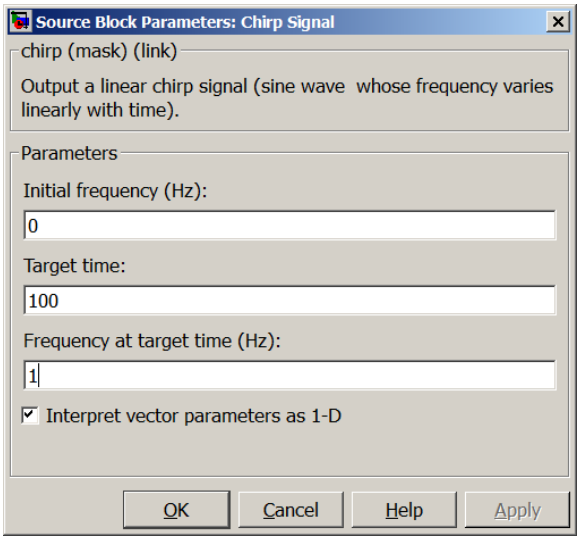
Next, we multiple this signal by the conjugate of the input, $U^*(j\omega)$, and do some sort of averaging procedure[1], which we denote as $\overline{(\ )}$. One way to think of the averaging is that we are replacing each term in the equation, at each ω,with some sort of weighted taking the average over frequencies near ω. Another way to think about it is that we repeat the experiment many times and average 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

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

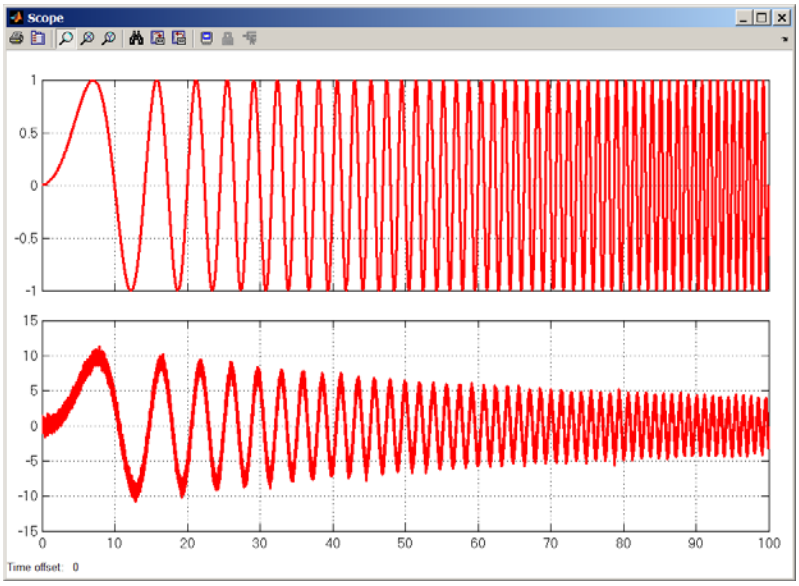
The beauty of this method is that noise is automatically removed by the averaging operation, leaving us with an unbiased estimate of the frequency response.

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.

[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.

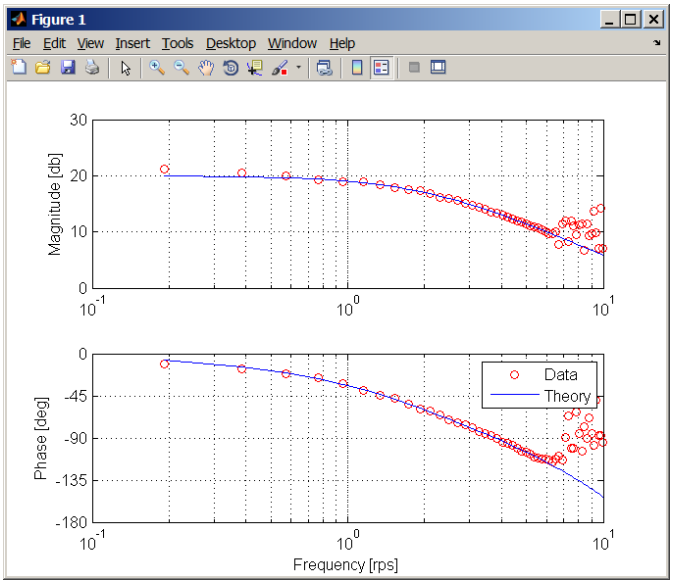


Change the simulation duration to 100 seconds and run the simulation. You should see something like this on the scope:



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.

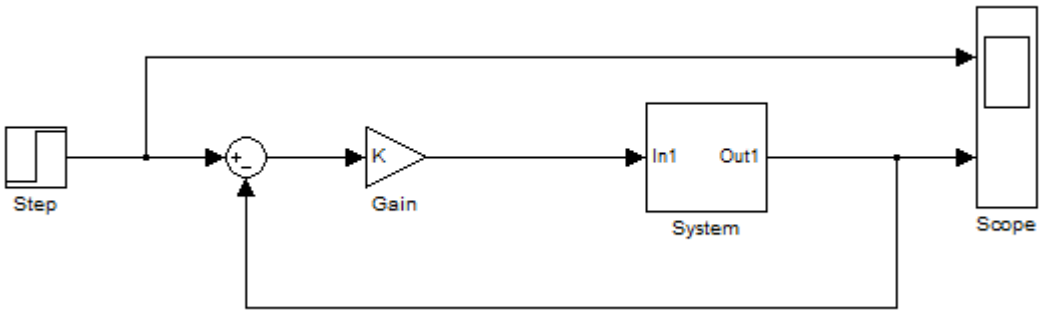
Now open and study the script `HW09.m`, . You should use `help tfestimate` to learn more about this command, which is the magic command that performs the frequency response estimation. Enter your guess at the system parameters, and run the script. When you have a good guess, you should see something 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. You should trim the frequency response data to eliminate frequencies where the data is no good.

a. SUBMIT your final frequency response graph (like the one above) and write your identified values of A, p, 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 for the closed-loop system.

Something has gone wrong at high frequency. Let's try to understand the dynamics to 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:

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

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.

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

d. SUBMIT a revised closed-loop step response, with a notch filter inserted in the loop to prevent the instability observed in response of part (b). Give the parameters of the notch on the time response.

Over the next two weeks, we will see how we could design the control system using only the experimentally determined frequency response. That is, we would never need to explicitly identify the transfer function, $G(s)$, if we have experimental measurements of $G(j\omega)$!