Sample Write Up For Lab 15s

Instructions: ALL EXAMPLES BELOW USED A NEGATIVE IN H(s).

The below write up contains the CORRECT figures and equations. I included these so you can know whether or not you have the correct answer. I did not include critical pieces of the Matlab code and other items; you have to figure these out for yourselves.

This is NOT a formal lab report. You simply need to include the required information discussed in the Lab 15s assignment packet. You do NOT have to use the Matlab notebook feature. You are welcome to do it, but you are also welcome to simply cut and paste from Matlab. For your code, please use the COURIER NEW font.

We were given a transfer function. We were required to use Matlab to plot its pole-zero response, its step response, its Bode plot, and find its inverse Laplace transform. Below is the code I used in Matlab to create the transfer function.

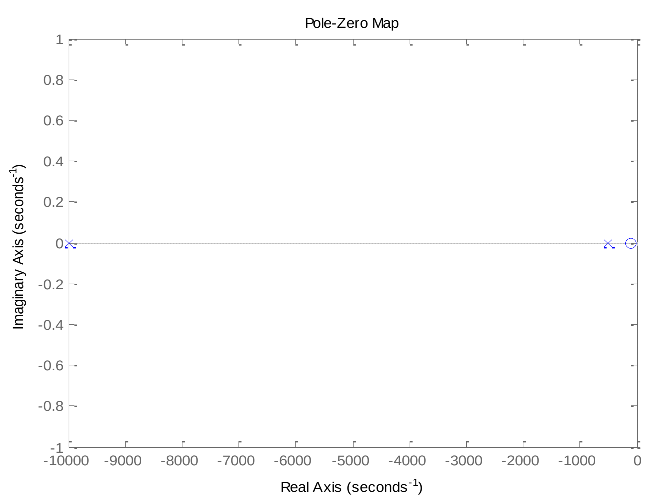
```
% Note the negatives in the numerator.  We were given the choice of plus or
% minus in the hand out
```

[redacted]

I then used the following code to create the pole-zero map.

[redacted]

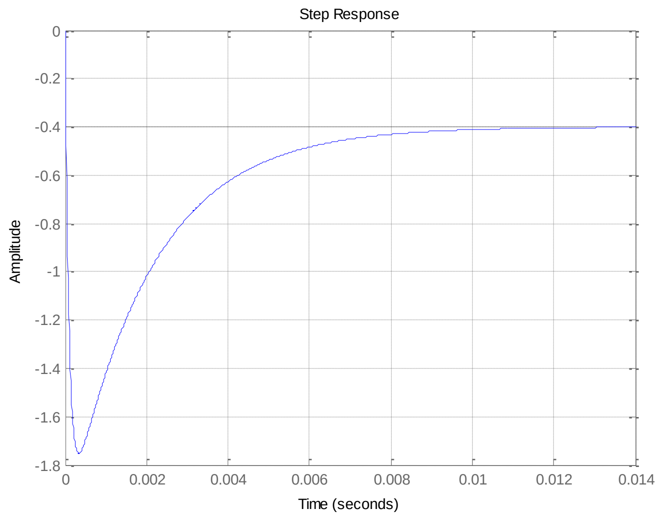
Below is the pole-zero plot.



I used the below code to plot the step response.

[redacted]

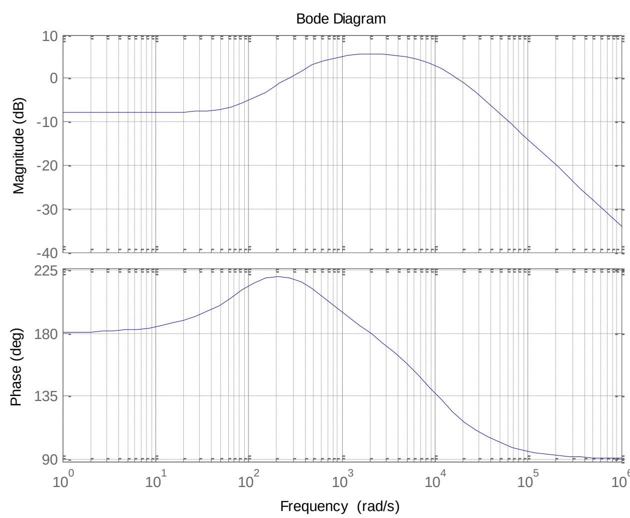
The step response is shown below.



I used the below code to create the Bode plot.

[redacted]

The bode plot is shown below.



Finally, I needed to find the inverse Laplace transform. Note I could not do this using the same code definitions when plotting the above functions. The above functions are used in conjunction with the `tf` command (see Lab 14m for details). The `ilaplace` command requires a symbolic function. Therefore, I used the following code to define my transfer function.

```
syms s t
```

[redacted]

The resulting transfer function, H(s), is shown below.

```
      20000 s + 2000000
(-1) ---------------------
    (s + 500) (s + 10000)
```

I used the `ilaplace` function to determine the inverse Laplace transform.

```
16000 exp(-500 t)   396000 exp(-10000 t)
----------------- - --------------------
       19                    19
```

Note the above is NOT the step response. I have just taken the inverse Laplace transform of the transfer function, H(s). To determine the equation for the step response, I can either integrate the above function or I can find G(s) and take the inverse Laplace transform. Note sG(s) = H(s). Therefore, I know G(s) is:

[redacted]

```
     20000 s + 2000000
- -----------------------
  s (s + 500) (s + 10000)
```

The inverse Laplace transform results in the equation for the step response.

```
   198 exp(-10000 t)   32 exp(-500 t)
----------------- - -------------- - 2/5
       95                 19
```

This concludes Part 1a of the Lab.

Next, I needed to use the above graphs to fill in the table below. I obtained the Matlab values by placing using the cursors and some visual estimating. Note the simulation results will be shown later in this report.

	Parameter	Matlab	Simulation	% Error
Step Response	t_s (ms) Final Value (V)	10.7 0.4 V		
Frequency Response	ω_{c1} ω_{c2} Gain	437 rad/s 10.92 krad/s 5.602 dB or 1.906		

This concludes Part 1b of the Lab.

I then created a circuit in Multisim to realize this transfer function. Below is the schematic I used to create the frequency response plots.

Below is the schematic I used to create my step-response plots. They are the same circuit, but have slightly different sources.

(Be very careful when doing your step response plots. I would recommend using a single pole, double throw (SPDT) switch, and manually switching it. You must ensure the circuit response goes to zero when there is no input; this can be tricky. If you are having problems, see me).

Note I have a buffer to isolate the source from the first stage. My first stage implements the first part of the bandpass filter. It is a HPF. The transfer function is:

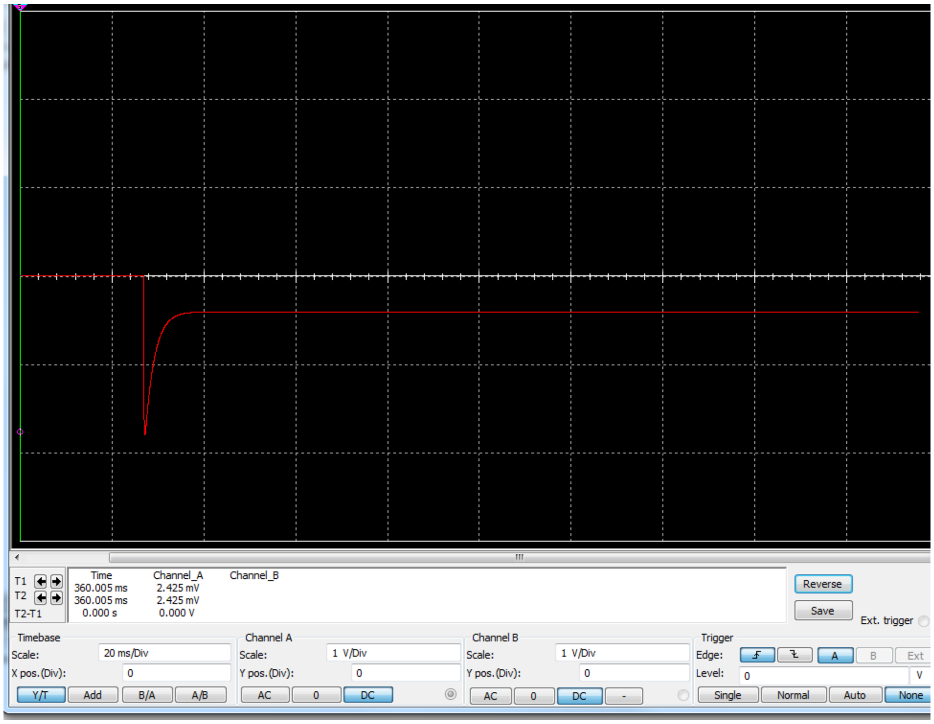
$$T_1(s) = -2\,(s + 98) / (s + 500)$$

The second stage implements the low pass filter portion of the bandpass response. It has a transfer function of:

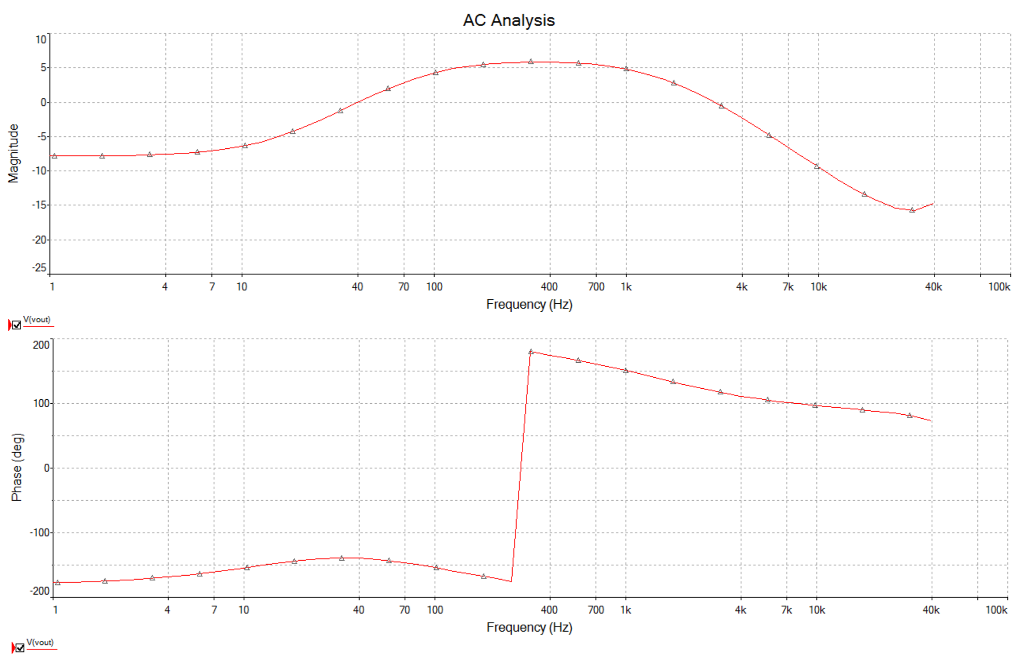
$$T_2(s) = 10000 / (s + 10000)$$

I used a buffer to isolate the small output resistance. Therefore, the transfer function is very close to the transfer function to which we designed.

The simulated circuit responses are shown below. First, the step response. Based on the scales, you can see the response is very similar to the Matlab predictions. Note the response doesn't occur at t = 0 because I was manually toggling the switch.



Next, I have the frequency response. Note the x-axis is in Hertz. The values compare favorably to the Matlab predictions. The phase looks different because Multisim only plots between -180^{o} and 180^{o}. When I do the overlay, you will see the phases are identical.



Although the step and frequency responses appear to be good overlays, I needed to plot both responses on the same graph to determine if they were a good match. To do so, I needed to export my data from Multisim into Matlab. I did this first with the step response data.

(To get the data, I first exported the data from the scope, saving as a .scp file. I then imported the data into Excel. You might have to delete a lot of the data, depending on how you saved it. You might also want to scale the time axis, so the response starts at t =0. I saved as a txt file, and then loaded into Matlab using the following commands:

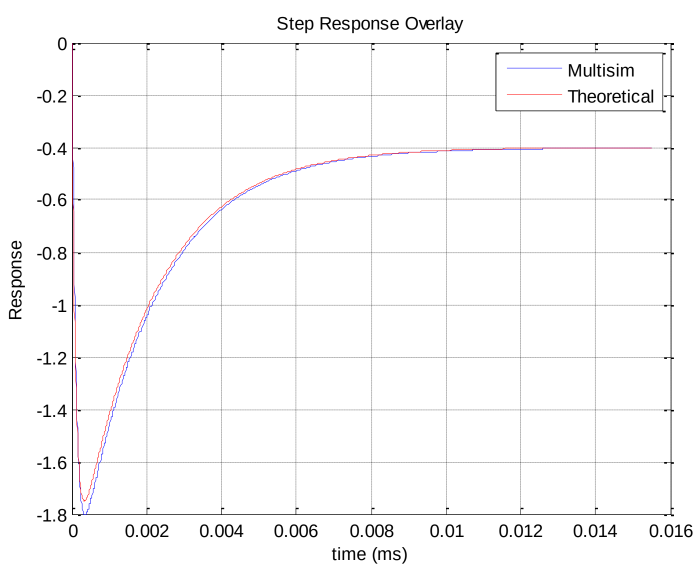
```
load Lab15sstepdata.txt
t = Lab15sstepdata(:,1);
vt = Lab15sstepdata(:,2);
```

This puts the time data into the "t" variable, and the response data into the "vt" variable.)

To plot the theoretical response on the same graph as the Matlab-predicted step response, I simply took the Matlab provided step response equation and used it to generate the theoretic response on the same graph.

[redacted]

The plot is shown below. Note they are very good overlays, with only a minor difference in the overshoot.



Next, I plotted the frequency response data on the same graph. As with the step response, it is necessary to output the Multsim data. ***(Note: There is a button in the graph menu that will export the data directly to an Excel file. This button is in the upper right-hand portion of the graph).***

I used the below Matlab commands to load the data and plot the data.

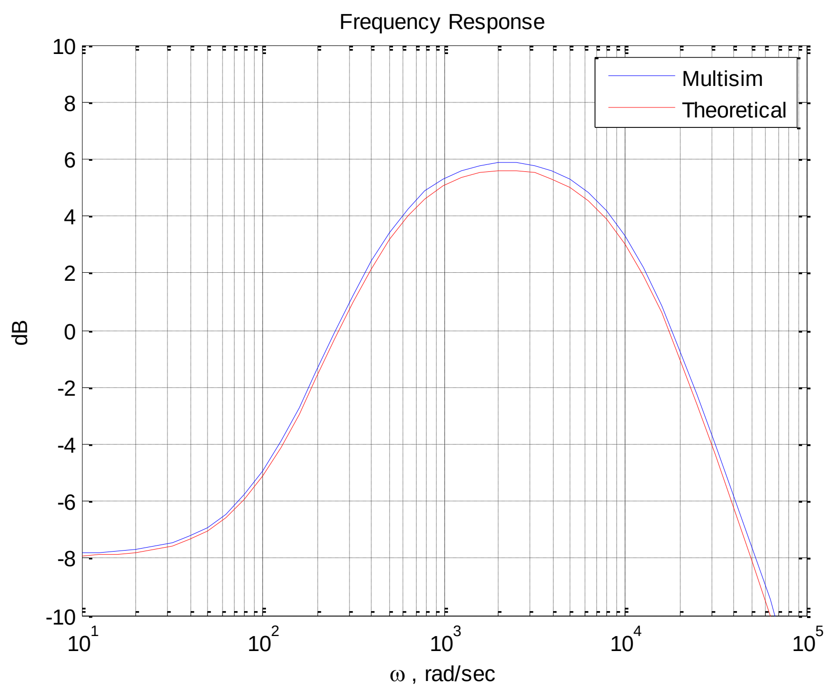
```
load Lab15sfreqresponse.txt
w = Lab15sfreqresponse(:,1)*2*pi;
H = Lab15sfreqresponse(:,2);
semilogx(w,20*log10(H))
```

I used the transfer function to generate the theoretical magnitude response:

[redacted]

```
xlabel('\omega , rad/sec')
ylabel('dB')
title('Frequency Response')
legend('Multisim','Theoretical')
```

The plot is shown below.



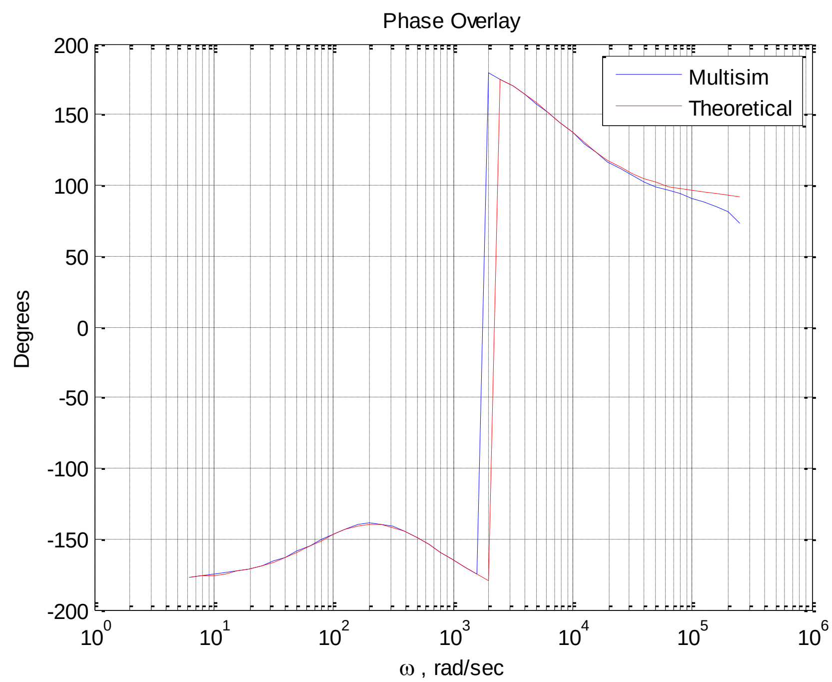
I then did the same for the phase response. The data from Multisim was exported exactly like it was done for the magnitude response.

```
load Lab15sphaseresponse.txt
```

```
w = Lab15sphaseresponse(:,1)*2*pi;
ph = Lab15sphaseresponse(:,2);
semilogx(w,ph)
```

[redacted]

```
legend('Multisim','Theoretical')
xlabel('\omega , rad/sec')
ylabel('Degrees')
```



Note there minor difference in the middle. This is not an error. Multisim calculates the phase as $+180^o$ while Matlab calculates the phase as -180^o. These are the same.

Finally, I used the Multisim graphs to complete the table and calculate the percent error.

	Parameter	Matlab	Simulation	% Error
Step Response	t_s (ms)	10.7	11.4	6.54
	Final Value (V)	0.4 V	0.402	0.5
Frequency Response	ω_{c1}	437 rad/s	437 rad/s	0
	ω_{c2}	10.92 krad/s	10.91 krad/s	-0.09
	Gain	5.602 dB or 1.906	5.85 dB or 1.96	2.83

(Finish the lab with some discussion and conclusions per the lab handout).