

|  |  |
| --- | --- |
| Course Number and Name:  ME 747 | |
| Semester and Year:  Fall 2017 | Name of Lab Instructor:  Alireza Ebadi |
| Lab Section and Meeting Time:  Section 2B, Tuesday 2-5 pm | Report Type:  Internal Group Report |
| Title of Experiment:  Time and Frequency Response of RC and RLC Circuits | |
| Date Experiment Performed:  9/19/17 | Date Report Submitted:  10/3/17 |
| Names of Group Members:  Zhangxi Feng  Simon Popecki  Reilly Webb | Grader's Comments: |
| Grade: |

Table of Contents

[Objectives 3](#_Toc494825603)

[Executive Summary 4](#_Toc494825604)

[Theory and Experimental Methods 5](#_Toc494825605)

[Results and Discussion 12](#_Toc494825606)

[Conclusions 20](#_Toc494825607)

[References 21](#_Toc494825608)

[Appendices 22](#_Toc494825609)

List of Figures

List of Tables

# Objectives

A first order (RC) circuit and a second order (RLC) circuit were subjected to step and sinusoidal inputs and their outputs were measured to determine their responses. The system parameters of the circuits were derived and calculated from the outputs. The unknown inductance and capacitances of the circuits were calculated from the system parameters.

# Executive Summary

In part 1 of this lab, a first order (RC) circuit was investigated to determine its responses for two step inputs at different DC offsets and amplitudes, and a sinusoidal input with no DC offset. The differential equation and transfer function were found for the circuit, as well as the time constant, . The resistor value was known and the capacitance was calculated from the results, . The amplitude and phase shift of the frequency response- the bode plot, was created from measurements. The results were then compared to Matlab simulations. From the bode plot, and were calculated again to be and respectively.

In part 2 of this lab, a second order (RLC) circuit was investigated to determine its responses for the same step as well as sinusoidal inputs. The damping ratio () and natural frequencies () were determined from the step responses, 0.0620 and 313.1 Hz, and the bode plot, 0.0275 and 1959.7 Hz. The resistor value of the circuit was known, and the inductance and capacitance were calculated for the step response, 1.27 H and , and the frequency response, 0.568 H and . The results were compared between the responses and Matlab simulations.

In part 3, the same two circuits were connected to Labview via a Data Acquisition (DAQ) device to collect data for creating the bode plots. The system parameters and component values of each circuit were calculated from the bode plots and compared to the results from previous two parts. The frequency response of the circuit was tested via an input of simulated white noise. For the RC circuit, τ was found to be and . The RLC circuit returned a damping ratio of 0.0284 and an undamped natural frequency of 360 Hz. From these, L was found to be 0.5083 H and C was

# Theory and Experimental Methods

**Theory**

1. Step Response of a First Order System

Two methods were used to experimentally determine the time constant from the response data. The initial slope method locates τ by extrapolating the initial slope of the data. This slope is extended to the intersection with the steady-state voltage, and this point gives the time constant in seconds.

The second method is defined as follows

(1)

In short τ is the point where the output reaches 63.2% of the final steady state value. Kirchhoff’s Voltage Loop law can be applied to the RC circuit, yielding the following equations

(3)

(2)

Solving for an equation relating and through

(7)

(6)

(5)

(4)

To solve for capacitance

(8)

The measured resistance for the RC circuit was

Multiple data points and methods were used to find τ, the average calculated capacitance will be taken.

1. Frequency Response of a First Order System

To find the transfer function of the system, first take the Laplace transform of the differential equation.

(10)

(9)

Then rearrange to find

(11)

The time constant is related to the breakpoint frequency as follows

(12)

Experimentally the break frequency was found to be 250 Hz. With a measured resistance , the calculated capacitance is

1. Step Response of a Second Order System

The RLC circuit can be modeled using a 2nd order differential equation. Using Kirchhoff’s Voltage Loop law,

(14)

(13)

With the equivalent voltage output relating the equations

(18)

(17)

(16)

(15)

The damping ratio can then be found from the raw data using the log decrement method on any two successive peaks

(22)

(21)

(20)

(19)

where is the amplitude at time , is the period, and is the number of positive successive peaks. The undamped natural frequency can then be found as follows

(25)

(24)

(23)

and can then be found with the following equations derived from the general form of the system’s differential equation

1. Frequency Response of a Second Order System

The transfer function of the system can be derived by taking the Laplace transform of the differential equation

(27)

(26)

The damping ratio and undamped natural frequency were then determined from a bode plot of the transfer function, where max peak is in dB and correlates to this peak location.

(28)

and can be found using the same equations (24) and (25)

3. Frequency Response Using LabView

The time constant and capacitance was found from the 1st order LabView results using a bode plot and the following relationships

(28)

The damping ratio for the 2nd order system was found using the same max peak equation, equation (28). The natural frequency also correlates to the location of this peak.

and can be found using the same equations (24) and (25)

**Experimental Methods**

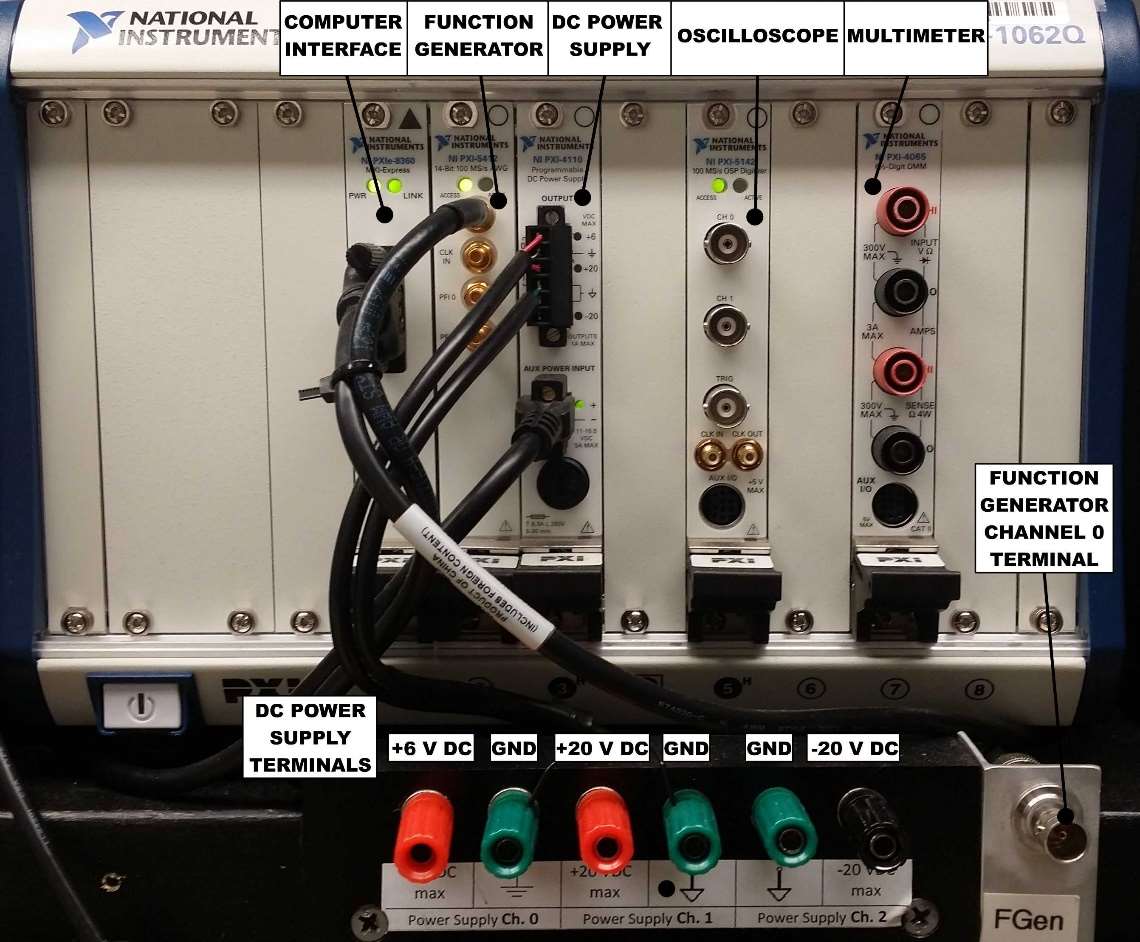
1. First Order System

**A first order RC circuit was set up as shown in FigureNumber. A T-junction BNC connector was used to connect a function generator to both a digital scope and the circuit input, ei(t).**



**The resistance, R (14.1 Ω), was measured using a digital multimeter. A square wave with an amplitude of 2 volts was sent through the circuit and the input and output curves were recorded using a digital oscilloscope. FigureNumber**

**Next a 4 volt DC offset was applied so that the square wave amplitude was from -3.0 V to +5.0 V, and the step response was recorded and compared.**

**The function generator was then set to generate a sine wave with a 2 volt amplitude and no dc offset. The output amplitude and phase delay was recorded using the scope cursor for a range of input frequencies. These input frequencies ranged from 10-10000 Hz, with a higher density of points near the break frequency (250 Hz).**

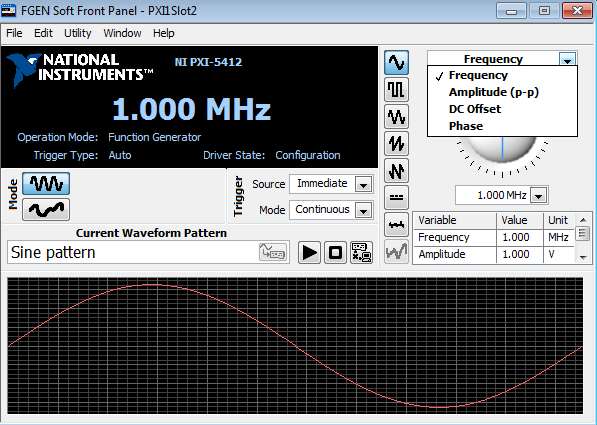
2. Second Order System

**A second order RLC circuit was set up as shown in FigureNumber. A T-junction BNC connector was used to connect a function generator to both a digital scope and the circuit input, ei(t).**



**The resistance, R (20 kΩ), was measured using a digital multimeter. A square wave with an amplitude of 2 volts was sent through the circuit input, and the input and output curves were recorded for a complete step response.**

**A 4 volt DC offset was applied so that the square wave amplitude is from -3.0 V to +5.0 V, and once again the input and output curves for a full step response were recorded using the scope.**

**The function generator FigureNumber was then set to sinusoidal input with an amplitude of 2 V and no DC offset. The break frequency (317 Hz) was estimated by sweeping the frequency of the** **sin wave from 10 Hz to 10 kHz. The output amplitude and phase shift was recorded for input frequencies of 1/20, 1/4, 1/2, 3/2, 2 and 4 times the break frequency.**

3. Frequency Response Using LabView

**LabView was used as an alternative method for determining the frequency response for both systems. Broadband noise was sent through the input of the circuit to test the whole spectrum of frequency inputs.**

**The following procedure was performed on both the RC and RLC circuits.**

**The filter input was connected to the NI DAQ FigureNumber output terminals. The RC circuit output was connected to scope Ch.1 and the input to scope Ch. 0.**

**Select the “Data View” tab, then press the “Run” button. Once the curve has converged, hit the “Stop” button. Export the bode plot and use the zoom tools in LV-SE to estimate the break frequencies.**

**Reference “Using LabView for Frequency Responses” by M.H. deLeon [1] for a more in depth procedure.**



# Results and Discussion

1.1

Two methods were used for calculating the time constants for both the unadjusted square wave input and the 1 V DC offset square wave input. It is clear from **TableNumber** that each method resulted in similar time constants, with only a 5% difference between the furthest values. It is also apparent that adding a DC offset to the input did not affect the time constant of the system.

|  |  |  |
| --- | --- | --- |
| Method | Tau Adjusted (s)\* | Calculated Capacitance (nF) |
| No Offset Initial Slope | 6.68155e-4 | 60.19 |
| No Offset 63.2% | 6.54000e-4 | 58.92 |
| DC Offset Initial Slope | 6.35428e-4 | 57.25 |
| DC Offset 63.2% | 6.52000e-4 | 58.74 |
| Average | 6.52396e-4 | 58.77 |

\*Adjusted Tau is the value found taking into account that the data did not start at time = 0.

From **TableNumber** it can be seen that the calculated capacitance (averaged between all of the methods) is



The theoretical simulation curve and the experimental result curve match closely (see **FigureNumber**). Small deviations are caused by assumptions made to simplify theoretical calculations, such as ignoring resistance in the wire and temperature sensitive circuit elements.

The only noticeable difference between the initial step response and the offset step response is the actual voltage offset. The curve shape was unaffected. The time constant will not change from a DC offset.

1.2

The Bode plot of the first order system would be unaffected by a dc offset. The Bode a plot displays response behavior, and remains the same for a circuit regardless of input.

|  |  |  |  |
| --- | --- | --- | --- |
| [Hz] | V | φ | dB |
| 10 | 3.93 | -2.2 | -0.08795 |
| 30 | 3.9 | -6.8 | -0.15451 |
| 100 | 3.64 | -21.7 | -0.75378 |
| 200 | 3.05 | -38.5 | -2.28981 |
| 230 | 2.85 | -42.4 | -2.87891 |
| 240 | 2.78 | -43.8 | -3.09491 |
| 250 | 2.71 | -44.9 | -3.31642 |
| 260 | 2.64 | -46.1 | -3.54373 |
| 270 | 2.57 | -47.2 | -3.77714 |
| 500 | 1.75 | -63.2 | -7.11504 |
| 1000 | 0.956 | -75.6 | -12.3666 |
| 5000 | 0.197 | -86.4 | -26.0864 |
| 10000 | 0.099 | -87.5 | -32.0631 |

**TableNumber** displays selected data over a range of 3 decades of frequencies. The transfer function for the RC circuit was found to be

This transfer function was used to build the bode plot in **FigureNumber**, which also has an overlay of an experimental bode plot that was calculated form the experiment data. Visually, both bode plots look very similar, especially the points close to the breakpoint frequency.



The experimental time constant was calculated to be 6.366e-04 seconds, which resulted in a capacitor value of 5.735e-08 F. This is only a 2.3% difference from the capacitor value found from the step response analysis (5.87 e-08 F).

The decibel drop in amplitude ratio at the breakpoint frequency of 250 Hz was found to be -3.118 dB.

2.1

For the unaltered square wave input, the damping ratio was found to be 0.0620, with an undamped natural frequency of 313.1 Hz. The DC offset square wave resulted in a damping ratio of 0.0581, and a natural frequency of 301.7 Hz. Theoretically the offset should not cause a difference in these values. In reality, the inductor, capacitor, or resistor could be dependent on voltage level.

The inductance of the system (without the offset), L, was calculated to be 1.27 H. The capacitance of the system was found to be 2.07 e-07 F



**FigureNumber**  further demonstrates the inaccuracy of the experimental damping ratio. It is approximately twice what the simulated damping ratio is. However, the same general shape of the curves are consistent.

2.2

|  |  |  |  |
| --- | --- | --- | --- |
| [Hz] | V | φ | dB |
| 10 | 3.97 | -0.2 | 0 |
| 16 | 3.99 | -0.3 | 0.04364 |
| 80 | 4.18 | -1.5 | 0.44771 |
| 158 | 5.12 | -5.9 | 2.20958 |
| 317 | 11.1 | -89 | 8.93064 |
| 476 | 2.95 | -162 | -2.57934 |
| 500 | 2.57 | -164 | -3.77714 |
| 634 | 1.39 | -169 | -9.11551 |
| 1000 | 0.512 | -174 | -17.7904 |
| 1268 | 0.311 | -175 | -22.1206 |
| 5000 | 0.019 | -178 | -46.4007 |
| 7000 | 0.0087 | -178 | -53.1854 |

Sample measurements for amplitude ratio and phase angle can be found in **TableNumber**.

The transfer function for the 2nd order system was determined using the differential equation of the system.

From this, a bode plot can be constructed **FigureNumber**



Experimental data points are plotted to the same bode plot in **FigureNumber**. It is clear that the simulated damping ratio is much higher than experimentally determined damping ratio.

From the bode plot, the damping ratio is 0.0275 with an undamped natural frequency of 1959.7 Hz. These values are much different compared to the damping ratio of 0.0620 and of 313.1 Hz that was found in the step response. This discrepancy could be the result of not picking enough data points that were close to the break frequency to get an accurate cusp for analysis.

Using these measurements from the bode plot, the inductance was found to be 0.5679 H, and the capacitance was found to be 4.585 e-07 F. These differ greatly from the L and C that resulted from the step response analysis There is a 55.2% difference in inductance (1.27 H) and a 54.9% difference in capacitance (2.07 e-07F) of both methods. Again, this could be error that propagated from not having enough data points to get an accurate estimate of the system characteristics.

3.1



The experimental values obtained from LabView for the first order system match the theoretical values closely (**FigureNumber**).

Using LabView, the time constant was found to be 6.1244e-04 seconds. This is a 6.125% difference from the time constant calculated with the step response analysis (6.52396e-4 sec) and only a 3.8% difference for the frequency response analysis (6.366e-04 sec).

The capacitor C value was calculated to be 5.5175e-08 F in LabView. This is a 3.94% difference from the frequency response calculated capacitor value (5.735e-08 F), and is a 6.0% difference from the capacitor value found from the step response analysis (5.87 e-08 F).

3.2



The curves in **FigureNumber** are the same general shape, however there is a 30 Hz offset between the experimental and the theoretical values. This could be from each component of the 2nd order circuit not being “ideal”. For example, the capacitor and inductor are assumed to have no resistance, when realistically every element of the circuit has a resistance. Wires comprising the circuit may also have an impact – as they each have a resistive and inductive identity.

The experimental damping ratio was found in LabView to be 0.0284 with a natural frequency of 360 Hz. The results from the square wave response shows a 54.19% difference in damping ratio (0.0620), with a 13.0% difference in undamped natural frequency (313.1 Hz). The frequency response analysis yielded a damping ratio with 3.17% difference (0.0275) and a difference in undamped natural frequency of 81.6% (1959.7 Hz).

The characteristics determined by LabView resulted in an inductance of 0.5083 H. This is 10.5% different from the inductance calculated from the frequency response (0.5679 H), and is 59.98% different from the step response inductance (1.27 H)

The capacitance C was calculated to be 3.8491e-07 F using LabView. This is 16.1% different from the frequency response capacitance calculation (4.585 e-07 F), and is 46.2% different from the step response analysis.

# Conclusions

Zhangxi Feng:

The capacitor in the first order circuit was found to have a capacitance of 58.7 nF. The capacitor in the second order circuit was found to have a capacitance of 0.207 µF. The inductor in the second order circuit was found to have an inductance of 1.27 H. The time constant of the first order circuit was found to be 0.65 ms

# References

[1] DeLeon, M H. “Using LabView for Frequency Responses.” 3 Sept. 2014.

# Appendices

1. **Data Tables**

**1st order Frequency Response Table**

|  |  |  |  |
| --- | --- | --- | --- |
| [Hz] | V | φ | dB |
| 10 | 3.93 | -2.2 | -0.08795 |
| 30 | 3.9 | -6.8 | -0.15451 |
| 100 | 3.64 | -21.7 | -0.75378 |
| 200 | 3.05 | -38.5 | -2.28981 |
| 230 | 2.85 | -42.4 | -2.87891 |
| 240 | 2.78 | -43.8 | -3.09491 |
| 250 | 2.71 | -44.9 | -3.31642 |
| 260 | 2.64 | -46.1 | -3.54373 |
| 270 | 2.57 | -47.2 | -3.77714 |
| 500 | 1.75 | -63.2 | -7.11504 |
| 1000 | 0.956 | -75.6 | -12.3666 |
| 5000 | 0.197 | -86.4 | -26.0864 |
| 10000 | 0.099 | -87.5 | -32.0631 |

2nd Order Frequency Response Table

|  |  |  |  |
| --- | --- | --- | --- |
| [Hz] | V | φ | dB |
| 10 | 3.97 | -0.2 | 0 |
| 16 | 3.99 | -0.3 | 0.04364 |
| 80 | 4.18 | -1.5 | 0.44771 |
| 158 | 5.12 | -5.9 | 2.20958 |
| 317 | 11.1 | -89 | 8.93064 |
| 476 | 2.95 | -162 | -2.57934 |
| 500 | 2.57 | -164 | -3.77714 |
| 634 | 1.39 | -169 | -9.11551 |
| 1000 | 0.512 | -174 | -17.7904 |
| 1268 | 0.311 | -175 | -22.1206 |
| 5000 | 0.019 | -178 | -46.4007 |
| 7000 | 0.0087 | -178 | -53.1854 |

1. **Sample Calculations**

Matlab

**One.m**

clear all;

close all;

%% Reading lvm data for first output

filename1 = '1.1.5 Data.lvm';

output1 = importdata(filename1,'\t',33);

responsedata1 = output1.data(:,3:4);

%% a)

% Output 1 from 1.1.5

% Observed from data, t = -3.13e-7 is the threshold

% Takes first 5 values to find the slope.

% Second method

threshold1 = -3.13e-7; % this is the first observable starting point of the response curve

responseFinalValue1 = mean(responsedata1(4950:end,2));

responseInitialValue1 = mean(responsedata1(1:350,2));

[onetauX1,onetauX2,onetauY1,onetauY2,onetauPer1,onetauPer2,p1] = ...

FindTaus(responsedata1,threshold1,responseFinalValue1,responseInitialValue1);

onetauTime1 = onetauX1 - threshold1;

onetauTime2 = onetauX2 - threshold1;

% The initial slope line

time1 = linspace(0,onetauTime1,100);

out1 = time1\*p1(1) + p1(2);

% Plot the results

figure(1);

hold on;

grid on;

xlim([-1 5]);

ylim([-2.5 2.5]);

% plot the original data with initial time offset adjusted

plot((responsedata1(:,1)-threshold1)\*1e3,responsedata1(:,2),'-.');

% plot the line

plot(time1\*1e3,out1);

% plot the tau location on the curve from the initial slope method

plot(onetauTime1\*1e3,onetauY1,'O');

% plot the tau location on the curve from the second method

plot(onetauTime2\*1e3,onetauY2,'X','markers',12);

% plot a drop down from the tip of the line to the tau location

plot([onetauTime1\*1e3 onetauTime1\*1e3],[onetauY1 out1(end)],'--');

legend('Location','best','Original response data','Initial slope line',...

'Tau point from initial slope method','Tau point from 63.2% method',...

'Drop down indicator line for initial slope method');

title('First Order Step Response 0 Offset Analysis','FontSize',14);

xlabel('Time (ms)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

%% Reading lvm data for second output

filename2 = '1.1.6 Data.lvm';

output2 = importdata(filename2,'\t',33);

responsedata2 = output2.data(:,3:4);

%% a)

% Output 1 from 1.1.6

% observed from data, t = 2.717e-7 is the threshold

% Takes first 5 values to find the slope.

threshold2 = 2.717e-7; % this is the first value higher than the initial noise

responseFinalValue2 = mean(responsedata2(4950:end,2));

responseInitialValue2 = mean(responsedata2(1:350,2));

[twotauX1,twotauX2,twotauY1,twotauY2,twotauPer1,twotauPer2,p2] = ...

FindTaus(responsedata2,threshold2,responseFinalValue2,responseInitialValue2);

twotauTime1 = twotauX1 - threshold2;

twotauTime2 = twotauX2 - threshold2;

% The initial slope line

time2 = linspace(0,twotauX1,100);

out2 = time2\*p2(1) + p2(2);

% Plot the results

figure(2);

hold on;

grid on;

xlim([-1 5]);

ylim([-3.5 5.5]);

% plot the original data with initial time offset adjusted

plot((responsedata2(:,1)-threshold2)\*1e3,responsedata2(:,2),'-.');

% plot the line

plot(time2\*1e3,out2);

% plot the tau location on the curve from the initial slope method

plot(twotauTime1\*1e3,twotauY1,'O');

% plot the tau location on the curve from the second method

plot(twotauTime2\*1e3,twotauY2,'X','markers',12);

% plot a drop down from the tip of the line to the tau location

plot([twotauTime1\*1e3 twotauTime1\*1e3],[twotauY1 out2(end)],'--');

legend('Location','best','Original response data','Initial slope line',...

'Tau point from initial slope method','Tau point from 63.2% method',...

'Drop down indicator line for initial slope method');

title('First Order Step Response 1V Offset Analysis','FontSize',14);

xlabel('Time (ms)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

%% e) Theoretical Simulation

R = 11.1e3;

C = 58.77e-9;

sys = tf(1,[R\*C 1]); % transfer function

% creates the step offset conditions

opt1 = stepDataOptions('InputOffset',-2,'StepAmplitude',4);

% step response

[V1,t1] = step(sys,opt1);

% Find tau

for i = 1:length(V1)

if (V1(i) >= 0.632\*(V1(end)-V1(1))+V1(1))

tauX = t1(i);

tauY = V1(i);

break;

end

end

figure(3);

hold on;

grid on;

xlim([-1 5]);

ylim([-2.5 2.5]);

plot(t1\*1e3,V1);

plot(tauX\*1e3,tauY,'\*','markers',14);

plot([-1 5],[tauY tauY],'k--');

title('Simulated Response of the First Order Circuit','FontSize',14);

xlabel('Time (ms)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

text(2,-0.5,['Tau = ',num2str(tauX\*1e3),' ms']);

text(2,-0.8,['Tau Voltage = ',num2str(tauY),' V']);

figure(4);

hold on;

grid on;

xlim([-1 5]);

ylim([-2.5 2.5]);

plot(t1\*1e3,V1);

plot((responsedata1(:,1)-threshold1)\*1e3,responsedata1(:,2),'-.');

legend('Location','best','Theoretical Simulation','Experimental Result');

title('Comparing Theoretical and Experimental Step Response','FontSize',14);

xlabel('Time (ms)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

%% 1.2 b) Bode Plots

% source data

omega\_i = [10,30,100,200,230,240,250,260,270,500,1000,5000,10000];

dB = [-0.087959128,-0.15451799,-0.753782462,-2.289813348,-2.878912935,...

-3.094914217,-3.316424318,-3.543731598,-3.777147669,-7.115049162,...

-12.36665229,-26.08648561,-32.06310624];

phi = [-2.2,-6.8,-21.7,-38.5,-42.4,-43.8,-44.9,-46.1,-47.2,-63.2,-75.6,-86.4,-87.5];

% retrieve bode plot data from bode(sys)

[mag,phase,wout] = bode(sys);

phase = squeeze(phase); % reduce the 3D matrix into 2D

mag = squeeze(mag); % reduce the 3D matrix into 2D

wout = wout/(2\*pi); % convert from rad/s to Hz

% magr = (dB/max(dB)).^2; % Power of the ratio

% dB3 = interp1(magr,[omega\_i' phi' dB'],0.5,'spline'); % interpret for Wb

% convert mag1 to decibels

magdb = 20\*log10(mag);

figure(5);

subplot(2,1,1);

semilogx(wout,magdb,'kO');

hold on;

grid on;

semilogx(omega\_i,dB,'b');

xlim([10 10000]);

ylabel('Amplitude Ratio (dB)','FontSize',12);

title('Simulated and Experimental Bode Response Plot','FontSize',14);

legend('Location','best','Simulated Bode Plot','Experimental Bode Plot');

subplot(2,1,2);

semilogx(wout,phase,'kO');

hold on;

grid on;

semilogx(omega\_i,phi,'b');

xlim([10 10000]);

ylim([-90 0]);

xlabel('Frequency (Hz)','FontSize',12);

ylabel('Phase Shift (deg)','FontSize',12);

legend('Location','best','Simulated Bode Plot','Experimental Bode Plot');

%% 1.2 c)

breakfrequency=250; %Hz

tauC=1/(2\*pi\*breakfrequency);

CapC=tauC/R;

%% 1.2 e)

[magE,phaseE,woutE]=bode(sys,250\*2\*pi);

magdbE = 20\*log10(magE);

**Two.m**

clear all;

close all;

%% 2.1 b) Finding Experimental damping ratio and undamped natural frequency

filename1 = '2.1.4 Data.lvm';

output1 = importdata(filename1,'\t',33);

responsedata1 = output1.data(:,3:4);

% find peaks

[maxtab,mintab] = peakdet(responsedata1(:,2),0.05);

% check peaks

figure(1);

grid on;

hold on;

plot(responsedata1(:,1),responsedata1(:,2));

plot(responsedata1(maxtab(:,1),1),maxtab(:,2),'O');

title('Experimental 0V Offset Step Response Data','FontSize',14);

xlabel('Time (s)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

legend('Location','best','Data','Peaks');

% Log decrement method

n = 2; % next peak is 2 periods away

sigma = (1/(n-1))\*log(maxtab(1,2)/maxtab(2,2));

zeta = sigma/sqrt(4\*pi^2+(sigma)^2); % the damping ratio

T = diff(responsedata1(maxtab(:,1),1));

T = T(1); % The first damped period

wd = 2\*pi/T; % The damped frequency in rad/s -> 2pi rad/cyc div s/cyc

wn = wd/(sqrt(1-zeta^2)); % Natural frequency in rad/s

wnHz = wn/(2\*pi); % Natural frequency in Hz -> rad/s div 2pi rad/cyc

%% c)

% From derived differential equation, LC = 1/wn^2, L/R = 2\*zeta/wn

R = 20200; % R = 20.2k ohms measured in lab

L = R\*2\*zeta/wn; % result = 1.3 mH

C = 1/wn^2/L; % result = 179.21 muF

sys = tf(1,[L\*C,L/R,1]);

%% d) Step Responses

% creates the step offset conditions

opt1 = stepDataOptions('InputOffset',-2,'StepAmplitude',4);

% step response

[V1,t1] = step(sys,opt1);

figure(2);

hold on;

grid on;

xlim([-0.0025 0.035]);

ylim([-2 6]);

plot(t1,V1,'--');

plot(responsedata1(:,1),responsedata1(:,2));

title('Simulated 0V Offset Step Response of the Second Order Circuit','FontSize',14);

xlabel('Time (ms)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

legend('Location','best','Simulated Response','Experimental Response');

%% d) second offset

filename2 = '2.1.5 Data.lvm';

output2 = importdata(filename2,'\t',33);

responsedata2 = output2.data(:,3:4);

% creates the step offset conditions

opt2 = stepDataOptions('InputOffset',-3,'StepAmplitude',8);

% step response

[V2,t2] = step(sys,opt2);

figure(3);

hold on;

grid on;

xlim([-0.0025 0.035]);

ylim([-4 12]);

plot(t2,V2,'--');

plot(responsedata2(:,1),responsedata2(:,2));

title('Simulated 1V Offset Step Response of the Second Order Circuit','FontSize',14);

xlabel('Time (ms)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

legend('Location','best','Simulated Response','Experimental Response');

% find peaks

[maxtab2,mintab2] = peakdet(responsedata2(:,2),0.05);

% check peaks

figure(4);

grid on;

hold on;

plot(responsedata2(:,1),responsedata2(:,2));

plot(responsedata2(maxtab2(:,1),1),maxtab2(:,2),'O');

title('Experimental 1V Offset Step Response Data','FontSize',14);

xlabel('Time (s)','FontSize',12);

ylabel('Voltage (V)','FontSize',12);

legend('Location','best','Data','Peaks');

% Log decrement method

n = 2; % next peak is 2 periods away

sigma2 = (1/(n-1))\*log(maxtab2(1,2)/maxtab2(2,2));

zeta2 = sigma2/sqrt(4\*pi^2+(sigma2)^2); % the damping ratio

T2 = diff(responsedata2(maxtab2(:,1),1));

T2 = T2(1); % The first damped period

wd2 = 2\*pi/T2; % The damped frequency in rad/s -> 2pi rad/cyc div s/cyc

wn2 = wd2/(sqrt(1-zeta2^2)); % Natural frequency in rad/s

wnHz2 = wn2/(2\*pi); % Natural frequency in Hz -> rad/s div 2pi rad/cyc

%% 2.2 a) - done in excel sheet

%% b)

omega\_i = [10,16,80,158,317,476,500,634,1000,1268,5000,7000]; % Hz

dB = [0,0.043647778,0.4477155,2.209589084,8.93064944,-2.579369816,... % dB

-3.777147669,-9.11551413,-17.79041092,-22.12060235,-46.40073812,-53.18542508];

phi = [-0.2,-0.3,-1.5,-5.9,-89,-162,-164,-169,-174,-175,-178,-178]; % deg

% retrieve bode plot data from bode(sys)

[mag,phase,wout] = bode(sys);

phase = squeeze(phase); % reduce the 3D matrix into 2D

mag = squeeze(mag); % reduce the 3D matrix into 2D

wout = wout/(2\*pi); % convert from rad/s to Hz

% convert mag1 to decibels

magdb = 20\*log10(mag);

figure(5);

subplot(2,1,1);

semilogx(wout,magdb,'kO');

hold on;

grid on;

semilogx(omega\_i,dB,'b');

xlim([10 10000]);

ylim([-40 25]);

ylabel('Amplitude Ratio (dB)','FontSize',12);

title('Simulated and Experimental Bode Response Plot','FontSize',14);

legend('Location','best','Simulated Bode Plot','Experimental Bode Plot');

subplot(2,1,2);

semilogx(wout,phase,'kO');

hold on;

grid on;

semilogx(omega\_i,phi,'b');

xlim([10 10000]);

ylim([-180 0]);

xlabel('Frequency (Hz)','FontSize',12);

ylabel('Phase Shift (deg)','FontSize',12);

legend('Location','best','Simulated Bode Plot','Experimental Bode Plot');

%% c)

[maxPeak,maxInd] = max(magdb); % maximum point on the measured bode plot

wnFromBode = wout(maxInd)\*2\*pi; % frequency where maxPeak happened

zetaExp = 1/(2\*maxPeak); % Approximate height = 1/2zeta

%% d)

% From derived differential equation, LC = 1/wn^2, L/R = 2\*zeta/wn

LFromBode = R\*2\*zetaExp/wnFromBode; %

CFromBode = 1/wnFromBode^2/LFromBode; %

**Three.m**

%This script is responsible for Part 3 of the analysis

%Author: Simon Popecki

%Date: 30 September 2017

%Class: ME 747

%Part 3 concerns the response of the first and second order systems to a

%random noise input. The experimental values are compared to the

%theoretical predictions.

%% Data Importation

clear all;

close all;

load Part3Data.mat

%DATA FORMAT:

%For amplitude response plots, the X-value from the raw data is frequency (Hz)

%The Y-value is the corresponding magnitude (dB)

%For phase response plots, the X-vlue from the raw data is frequency (Hz)

%The Y-value is the corresponding phase angle (degrees)

%Signal Conditioning

%The raw bode plots from LabView have a lot of noise past 10 kHz (1st

%order) and 1.4 kHz (2nd order)

%Trimming the 1st order bode plot at ~10 kHz

cutoff1 = 1000; %frequency index cutoff number, first order

frequencyAR1 = frequencyAR1(1:cutoff1);

magnitudeAR1 = magnitudeAR1(1:cutoff1);

frequencyPS1 = frequencyPS1(1:cutoff1);

phaseAnglePS1 = phaseAnglePS1(1:cutoff1);

%Trimming the 2nd order bode plot at 1.4 kHz

cutoff2 = 500; %frequency index cutoff number, second order

frequencyAR2 = frequencyAR2(1:cutoff2);

magnitudeAR2 = magnitudeAR2(1:cutoff2);

frequencyPS2 = frequencyPS2(1:cutoff2);

phaseAnglePS2 = phaseAnglePS2(1:cutoff2);

%% Freqeuncy Response of a First Order System Using LabView

%First order known values (R and C get overwritten)

R = 11.1e3; %Ohms, the resistance of the resistor

C = 58.77e-9; %Farrads, the capacitance of the capacitor

sys1 = tf(1,[R\*C 1]); %The transfer function of the 1st order system

[mag,theoreticalPhaseAngle1,theoreticalFrequency1] = bode(sys1); %Extracting data points from system object, sys1

theoreticalPhaseAngle1 = squeeze(theoreticalPhaseAngle1); %force array

mag1 = squeeze(mag); %force array

TheoreticalMagnitudeAR1 = 20\*log10(mag1); %converting from absolute magnitude to decibels

theoreticalFrequency1 = theoreticalFrequency1/(2\*pi); %convert from rad/s to Hz

%1/Tau (the time constant) occurs at the point where phase is 45 degrees

%omega is ~250 Hz

locationMap = find(phaseAnglePS1<=-45); %finding the indices of phase angles that meet the condition <= -45 degrees

freqInd = locationMap(1); %finding the index of the first value meeting the <= -45 degrees condition

omega1 = frequencyPS1(freqInd); %Hz, 1/tau - the natural frequency of the first order system

tau1 = 1/(omega1\*6.28); %seconds, the time constant of the first order system (multiplied by 6.28 to convert to radians per second

disp('Circuit 1 time constant (seconds):')

disp(tau1)

C1 = 1/(R\*omega1\*6.28); %Farrads, the calculated capacitance of the first order system using data from LabView

disp('Circuit 1 capacitance (Farrads):')

disp(C1)

figure(1)

subplot(2,1,1)

semilogx(frequencyAR1,magnitudeAR1,theoreticalFrequency1,TheoreticalMagnitudeAR1,'o')

% title('Theoretical and Experimental Frequency Response')

xlabel('Frequency (Hz)')

ylabel('Magnitude (dB)')

legend('Experimental Values','Theoretical Values','location','southwest')

grid on

xmin = 0;

xmax = 100000;

ymin = -40;

ymax = 5;

axis ([xmin xmax ymin ymax])

subplot(2,1,2)

semilogx(frequencyPS1,phaseAnglePS1,theoreticalFrequency1,theoreticalPhaseAngle1,'o')

xlabel('Frequency (Hz)')

ylabel('Phase Angle (∞)')

legend('Experimental Values','Theoretical Values','location','southwest')

grid on

xmin = 0;

xmax = 100000;

ymin = -100;

ymax = 10;

axis ([xmin xmax ymin ymax])

%% Frequency Response of a Second Order System Using LabView

%Second order known values

L = 1.272690763057150; %H, inductance

R = 20200; %Ohms, the resistance of the resistor

C = 2.030234051171926e-07; %Farrads, the capacitance of the capacitor

sys2 = tf(1,[L\*C L/R 1]); %The transfer function of the 1st order system

[mag,theoreticalPhaseAngle2,theoreticalFrequency2] = bode(sys2); %Extracting data points from system object, sys1

theoreticalPhaseAngle2 = squeeze(theoreticalPhaseAngle2); %force array

mag2 = squeeze(mag); %force array

TheoreticalMagnitudeAR2 = 20\*log10(mag2); %converting from absolute magnitude to decibels

theoreticalFrequency2 = theoreticalFrequency2/(2\*pi); %convert from rad/s to Hz

%Finding Experimental Damping Ratio

peakdB = max(magnitudeAR2); %finding the magnitude of the peak

zeta = 1/(peakdB\*2); %calculating the damping ratio given the peak magnitude

disp('Circuit 2 damping ratio:')

disp(zeta)

%Finding Experimental Natural Frequency

[~,indexMaxMag] = max(magnitudeAR2); %finding the index of the max amplitude

natFrequency = frequencyAR2(indexMaxMag); %Hz, finding the frequency at the break point

disp('Circuit 2 natural frequency (Hz):')

disp(natFrequency)

%Calculating L and C values from LabView experimental results

LC = 1/((natFrequency\*6.28)^2);

L = R\*(2\*zeta/(natFrequency\*6.28)); %H, calculating inductance

C = LC/L;

disp('Circuit 2 Inductance (H):')

disp(L)

disp('Circuit 2 Resistance (Ohms):')

disp(R)

disp('Circuit 2 Capacitance (F):')

disp(C)

figure(2)

subplot(2,1,1)

semilogx(frequencyAR2,magnitudeAR2,theoreticalFrequency2,TheoreticalMagnitudeAR2,'o')

xlabel('Frequency (Hz)')

ylabel('Magnitude (dB)')

legend('Experimental Values','Theoretical Values','location','southwest')

grid on

xmin = 0;

xmax = 10000;

ymin = -60;

ymax = 25;

axis ([xmin xmax ymin ymax])

subplot(2,1,2)

semilogx(frequencyPS2,phaseAnglePS2,theoreticalFrequency2,theoreticalPhaseAngle2,'o')

xlabel('Frequency (Hz)')

ylabel('Phase Angle (∞)')

legend('Experimental Values','Theoretical Values','location','southwest')

grid on

xmin = 0;

xmax = 10000;

ymin = -200;

ymax = 10;

axis ([xmin xmax ymin ymax])

**peakdet.m**

function [maxtab, mintab]=peakdet(v, delta, x)

%PEAKDET Detect peaks in a vector

% [MAXTAB, MINTAB] = PEAKDET(V, DELTA) finds the local

% maxima and minima ("peaks") in the vector V.

% MAXTAB and MINTAB consists of two columns. Column 1

% contains indices in V, and column 2 the found values.

%

% With [MAXTAB, MINTAB] = PEAKDET(V, DELTA, X) the indices

% in MAXTAB and MINTAB are replaced with the corresponding

% X-values.

%

% A point is considered a maximum peak if it has the maximal

% value, and was preceded (to the left) by a value lower by

% DELTA.

% Eli Billauer, 3.4.05 (Explicitly not copyrighted).

% This function is released to the public domain; Any use is allowed.

maxtab = [];

mintab = [];

v = v(:); % Just in case this wasn't a proper vector

if nargin < 3

x = (1:length(v))';

else

x = x(:);

if length(v)~= length(x)

error('Input vectors v and x must have same length');

end

end

if (length(delta(:)))>1

error('Input argument DELTA must be a scalar');

end

if delta <= 0

error('Input argument DELTA must be positive');

end

mn = Inf; mx = -Inf;

mnpos = NaN; mxpos = NaN;

lookformax = 1;

for i=1:length(v)

this = v(i);

if this > mx, mx = this; mxpos = x(i); end

if this < mn, mn = this; mnpos = x(i); end

if lookformax

if this < mx-delta

maxtab = [maxtab ; mxpos mx];

mn = this; mnpos = x(i);

lookformax = 0;

end

else

if this > mn+delta

mintab = [mintab ; mnpos mn];

mx = this; mxpos = x(i);

lookformax = 1;

end

end

end

**findtaus.m**

function [tauX1,tauX2,tauY1,tauY2,tauPer1,tauPer2,p] = ...

FindTaus(responsedata,threshold,responseFinalValue,responseInitialValue)

%% Calculate tau using initial slope method

% initialize variables

indSt = 1;

% cut off the initial flat data

for i = 1:size(responsedata,1)

% This condition will be met after the initial index, therefore if this

% condition is met, the loop will exit

if (responsedata(i,1) > threshold)

indSt = i;

break;

end

end

indEnd = indSt + 4;

% polyfit the initial slope segment data to find the equation of a line

% then use the equation values and final value to find the intersection

p = polyfit(responsedata(indSt:indEnd,1),responsedata(indSt:indEnd,2),1);

tauX1 = (responseFinalValue - p(2))/p(1); % x = (y - y(0))/m

for i = indEnd:size(responsedata,1) % we know the tau must be beyond indEnd

if (responsedata(i,1) >= tauX1)

tauY1 = responsedata(i,2);

break;

end

end

% check for the tau percentage

tauPer1 = (tauY1-responseInitialValue)/(responseFinalValue-responseInitialValue);

%% Calculate tau using the 63.2% method

% Finds the 63.2% of final value point, then scan the data until that

% point is reached.

tauY2 = (responseFinalValue-responseInitialValue)\*0.632 + responseInitialValue;

for i = indEnd:size(responsedata,1)

if (responsedata(i,2) >= tauY2)

tauX2 = responsedata(i,1);

break;

end

end

% check for the tau percentage

tauPer2 = (tauY2-responseInitialValue)/(responseFinalValue-responseInitialValue);

end % end of function

1. **Equipment List**
2. **Shared Peer percentage**