H2

Introduction

Below is a circuit was modelled to meet the specifications outlined in the lab deliverables. It contains one AC voltage supply, 3 resistors (RL is representative of the inductor's resistance), a capacitor, and an inductor. Also note that this circuit was designed using the actual values of physical components measured with the Hantek. This circuit was be solved for the currents, and non-trivial voltages across its components. Analytically, this was done using Kirchhoff's voltage and current laws, as well as mesh analysis. Digitally, the circuit was solved using the Tektronix virtual oscilloscope, as well as the single frequency AC mode in Multisim. Finally, the circuit was also solved for physically using a breadboard and the Hantek oscilloscope.

Problem Framing:

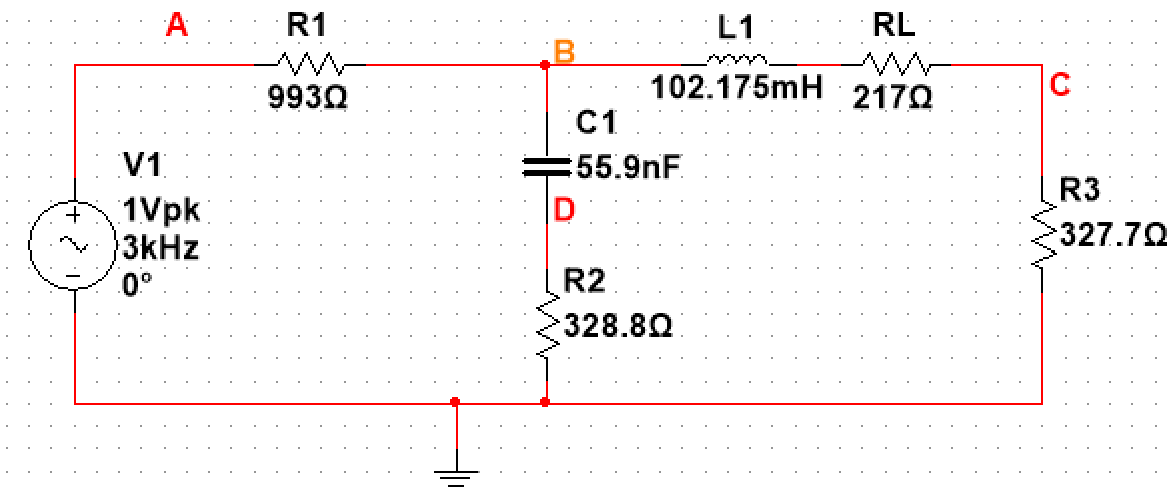


Figure 1: Circuit Diagram

The goal is to determine the voltage and current through components C1, L1, R2, and R3 analytically using 2 methods learned in class, digitally with Multisim single frequency AC sweep and Tektronix digital oscilloscope, and finally physically using the provided breadboard and Hantek oscilloscope.

Analytical Solution:

By using Kirchhoff's laws, I was able to solve for the 3 unknown currents in the circuit, I1, I2 and I3. With these currents and use of ohm's law (V = IR), I was able to solve for the voltages across each component. Note that for every calculation in this report, f = 3000 Hz (frequency), and w = 2*π*frequency (omega).

First, the circuit was split into two loops to which Kirchhoff's voltage law may be applied. One loop was composed of V1, R1, C1, and R2, while the other was composed of R2, C1, L1 (and RL), and R3. Then, we can apply Kirchhoff's current law to node B, where the current labelled I1, is equal to the sum of the current leaving labelled I2 and the current labelled I3.

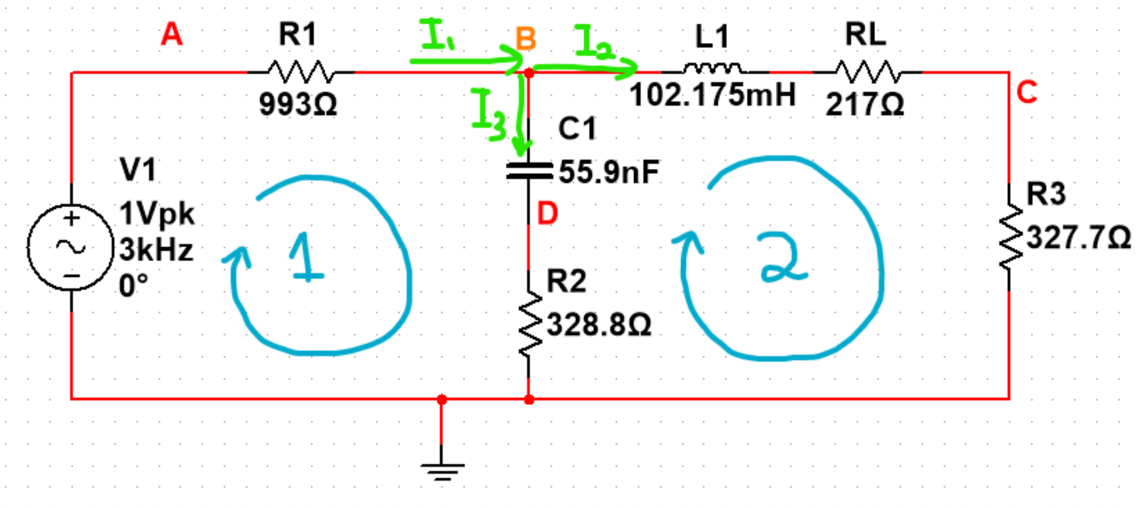


Figure 2: Circuit diagram with loops and currents labelled

With these relationships established, the following equations were derived.

0 = -I1*R1 - I3*R2 - I3*ZC1 + V1

0 = -I2*R3 - I2*RL - I2*ZL1 + I3*R2 + I3*ZC1

I1 = I2 + I3

These equations were used to derive the values of I1, I2, and I3 through the Maple CAS:

Maple input	**Maple Output**

f := 3000: omega:= 2*Pi*f:
R1 := 993; R2 := 328.8; R3 := 327.7; RL := 217; C1 := 0.559*10^(-7); L1 := 0.102175;
 	
evalf(solve([-I1*R1 - I3*R2 - I3*ZC1 + V1, -I2*R3 - I2*RL - I2*ZL1 + I3*R2 + I3*ZC1, I1 = I2 + I3])); assign(%);
 	
convert(I1, polar);
convert(I2, polar);
convert(I3, polar);
 	
VR1 := I1*R1; convert(%, polar);
VR2 := I3*R2; convert(%, polar);
VR3 := I2*R3; convert(%, polar);
VRL := I2*RL; convert(%, polar);
 	
VC1 := I3*ZC1; convert(%, polar);
VL1 := I2*ZL1; convert(%, polar);
	{I1 = 0.0003783157673 + 0.0001968871473 I,
	I2 = -0.000009103096655 - 0.0003267428168 I,
	I3 = 0.0003874188639 + 0.0005236299641 I}
	
	polar(0.0004264825536, 0.4798582753)
	polar(0.0003268695989, -1.598649250)
	polar(0.0006513691084, 0.9338090629)
	
	VR1 := 0.3756675569 + 0.1955089373 I
	polar(0.4234971758, 0.4798582754)
	VR2 := 0.1273833225 + 0.1721695322 I
	polar(0.2141701629, 0.9338090627)
	VR3 := -0.002983084774 - 0.1070736211 I
	polar(0.1071151676, -1.598649250)
	VRL := -0.001975371974 - 0.07090319125 I
	polar(0.07093070297, -1.598649250)
	VC1 := 0.4969491206 - 0.3676784694 I
	VL1 := 0.6292908998 - 0.01753212493 I
	polar(0.6295350760, -0.02785292300)

Table 1: Maple Kirchhoff's current law and voltage law calculations

As described above, the currents were found using KVL and KCL, and ohm's law was applied to find the voltage through each component.

However, while they are labelled as separate components, L1 and RL are in in reality a single component (since RL represents the actual resistance of the inductor). So, to find the voltage across the entire inductor, we can simply add the acquired voltages together in rectangular form:

VL = VL1 + VRL = (0.6292908998 - 0.01753212493*j) + (-0.001975371974 - 0.07090319125*j)

VL = (0.6273155278 - 0.0884353162*j)

And convert to polar form (using maple):

VL = polar(0.6335184106, -0.1400513041)

Then, these values are converted from rectangular to polar form, which shows the current and voltages in terms of their amplitudes and phase constants. Using this information, the values can be expressed in time-domain form as well as phasor form (values rounded to 3 significant digits):

	Amplitude	Phase Difference	Time Domain Form	Phasor Domain Form
I1	0.000426	0.480	0.000426*cos(wt + 0.480)	X = 0.000426e^(j*0.480)
I2	0.000327	-1.60	0.000327*cos(wt + (-1.60))	X = 0.000327e^(j*-1.60)
I3	0.000651	0.934	0.000651*cos(wt + 0.934)	X = 0.000651e^(j*0.934)
VR1	0.423	0.480	0.423*cos(wt +0.480)	X = 0.423e^(j*0.480)
VR2	0.214	0.934	0.214*cos(wt + 0.934)	X = 0.214e^(j*0.934)
VR3	0.107	-1.60	0.107*cos(wt + (-1.60))	X = 0.107e^(j*-1.60)
VC1	0.618	-0.637	0.618*cos(wt + (-0.637))	X = 0.618e^(j*-0.637)
VL	0.634	-0.140	0.633*cos(wt + (-0.140))	X = 0.633e^(j*-0.140)

Table 2: Kirchhoff's Laws analysis results

Note that source voltage (phase difference of 0) of lags current (I2, phase difference of -1.60) by approximately π/2 in the section of the circuit containing the capacitor. This value is expected, since we know that in theory, voltage tends to lag current by this amount in a circuit with a capacitor.

As for the mesh analysis, the same results were achieved as the Kirchhoff's laws solution. The following circuit demonstrates the meshes chosen for mesh analysis in the circuit.

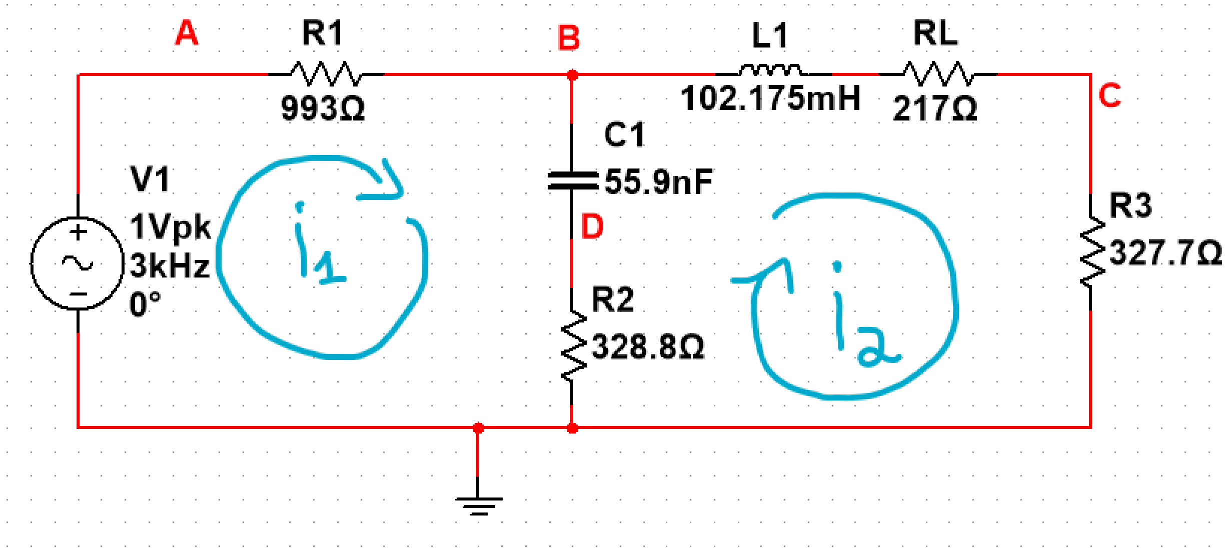


Figure 3: Circuit labelled for mesh analysis

The following equations were derived from mesh analysis:

0 = V1 - I1*R1 - ZC1*(I1 - I2) - R2*(I1 - I2)

0 = R2*(I1 - I2) + ZC1*(I1 - I2) - ZL1*I2 - RL*I2 - R3*I2

Maple input	Maple Output

evalf(solve([V1 - I1*R1 - ZC1*(I1 - I2) - R2*(I1 - I2), R2*(I1 - I2) + ZC1*(I1 - I2) - ZL1*I2 - RL*I2 - R3*I2]));
assign(%);
 	
convert(I1, polar);
convert(I2, polar);
 	
VR1 := I1*R1; convert(VR1, polar);
VR2 := R2*(I1 - I2); convert(VR2, polar);
VR3 := R3*I2; convert(VR3, polar);
VRL := RL*I2; convert(VRL, polar);
 	
VC1 := ZC1*(I1 - I2); convert(%, polar);
VL1 := ZL1*I2; convert(%, polar);
	{I1 = 0.0003783157673 + 0.0001968871473 I,
	I2 = -0.000009103096655 - 0.0003267428168 I}
	
	polar(0.0004264825536, 0.4798582753)
	polar(0.0003268695989, -1.598649250)
	
	VR1 := 0.3756675569 + 0.1955089373 I
	polar(0.4234971758, 0.4798582754)
	VR2 := 0.1273833225 + 0.1721695322 I
	polar(0.2141701629, 0.9338090627)
	VR3 := -0.002983084774 - 0.1070736211 I
	polar(0.1071151676, -1.598649250)
	VRL := -0.001975371974 - 0.07090319125 I
	polar(0.07093070297, -1.598649250)
	
	VC1 := 0.4969491206 - 0.3676784695 I
	polar(0.6181794929, -0.6369872641)
	VL1 := 0.6292908998 - 0.01753212493 I
	polar(0.6295350760, -0.02785292300)

Table 4: Maple code for Mesh analysis

Once again, the voltage across the inductor as a whole (L1 and RL) was calculated and results matched that of the Kirchhoff's analysis.

VL = VL1 + VRL = (0.6292908998 - 0.01753212493*j) + (-0.001975371974 - 0.07090319125*j)

VL = (0.6273155278 - 0.0884353162*j)

Polar form:

VL = polar(0.6335184106, -0.1400513041)

The current represented by I3 in the Kirchhoff's analysis also needs to be found, since all unknown currents in the circuit must be solved for. This can be expressed as the difference between I1 and I2 using Kirchhoff's current law at node B.

I1 = I2 + I3

I3 = I1 – I2

Solving in maple, we get:

I3 := 0.0003874188640 + 0.0005236299641*j

In polar form:

I3 := polar(0.0006513691085, 0.9338090628)

	Amplitude	Phase Difference	Time Domain Form	Phasor Domain Form	Match Kirchoff's?
I1	0.000427	0.480	0.000426*cos(wt + 0.480)	0.000426e^(j*0.480)	Yes
I2	0.000327	-1.60	0.000327*cos(wt + (-1.60))	0.000327e^(j*-1.60)	Yes
I3	0.000651	0.934	0.000651*cos(wt + 0.934)	0.000651e^(j*0.934)	Yes
VR1	0.423	0.480	0.423*cos(wt +0.480)	0.423e^(j*0.480)	Yes
VR2	0.214	0.934	0.214*cos(wt + 0.934)	0.214e^(j*0.934)	Yes
VR3	0.107	-1.60	0.107*cos(wt + (-1.60))	0.107e^(j*-1.60)	Yes
VC1	0.618	-0.637	0.618*cos(wt + (-0.637))	0.618e^(j*-0.637)	Yes
VL	0.634	-0.140	0.633*cos(wt + (-0.140))	0.633e^(j*-0.140)	Yes

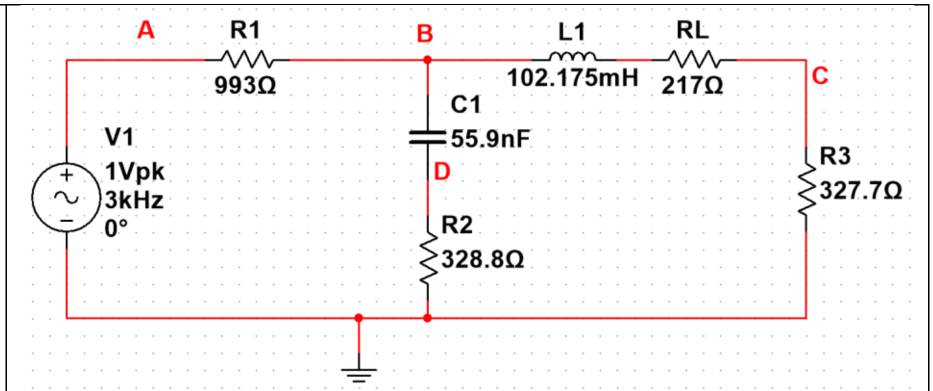
***rounded to 3 significant digits*

Table 4: Mesh analysis results and comparison to Kirchhoff's analysis

As seen in the table above, the mesh analysis results in phasor and time domain form are matching the Kirchhoff's analysis results.

Digital Solution Via Multisim:

Using the single frequency AC mode in Multisim, I was able to solve for the 3 unknown currents in the circuit, I1, I2 and I3. This was accomplished by acquiring the current through each component. The voltages of each component were also found by acquiring the voltage at each of the nodes, labelled on *Figure 1*, and solving for the voltage across each component as a difference between two of the node voltages. Note that both the magnitude/phase mode and complex numbers mode were used, with the mathematical operations being done using the rectangular form of the results.

<table>
<tr><td>Circuit (with nodes labelled)</td><td>
</td></tr>
<tr><td>Single Frequency Analysis (mag/phase)</td><td>
Single Frequency AC Analysis @ 3000 Hz
<table>
<tr><th></th><th>Variable</th><th>Magnitude</th><th>Phase (deg)</th></tr>
<tr><td>1</td><td>V(a)</td><td>1.00000</td><td>0.00000e+00</td></tr>
<tr><td>2</td><td>V(b)</td><td>654.22812 m</td><td>-17.38791</td></tr>
<tr><td>3</td><td>V(c)</td><td>107.11504 m</td><td>-91.59590</td></tr>
<tr><td>4</td><td>V(d)</td><td>214.17025 m</td><td>53.50327</td></tr>
<tr><td>5</td><td>I(C1)</td><td>651.36936 u</td><td>53.50327</td></tr>
<tr><td>6</td><td>I(R1)</td><td>426.48298 u</td><td>27.49387</td></tr>
<tr><td>7</td><td>I(R2)</td><td>651.36936 u</td><td>53.50327</td></tr>
<tr><td>8</td><td>I(R3)</td><td>326.86922 u</td><td>-91.59590</td></tr>
<tr><td>9</td><td>I(RL)</td><td>326.86922 u</td><td>-91.59590</td></tr>
<tr><td>10</td><td>I(L1)</td><td>326.86923 u</td><td>-91.59590</td></tr>
</table>
</td></tr>
</table>

Single Frequency Analysis (rectangular)

Single Frequency AC Analysis @ 3000 Hz

	Variable	Real	Imaginary
1	V(a)	1.00000	0.00000e+00
2	V(b)	624.33211 m	-195.50921 m
3	V(c)	-2.98316 m	-107.07350 m
4	V(d)	127.38351 m	172.16950 m
5	I(C1)	387.41944 u	523.62986 u
6	I(R1)	378.31610 u	196.88742 u
7	I(R2)	387.41944 u	523.62986 u
8	I(R3)	-9.10334 u	-326.74244 u
9	I(RL)	-9.10334 u	-326.74244 u
10	I(L1)	-9.10334 u	-326.74244 u

Table 5: Single frequency AC Analysis results in rectangular and polar form

With these results, the voltages across each component can now be calculated as a difference between node voltages. The following equations demonstrate this:

VR1 = Va – Vb

VR2 = Vd – 0

VR3 = Vc – 0

VC1 = Vb – Vd

VL = Vb – Vc

With these equations, we can solve for the voltages in maple, then convert to polar form for representation in phasor form.

Maple input	**Maple Output**
I1 := 0.00037831610 + 0.00019788742*I: I2 := -0.910334*10^(-5) + 0.00032674244*I: I3 := 0.00038741944 + 0.00052362986*I: VA := 1: VB := 0.62433211 - 0.19550921*I: VC := -0.00298316 - 0.10707350*I: VD := 0.12738351 + 0.17216950*I: VR1 := VA - VB; VR2 := VD - 0; VR3 := VC - 0; VL := VB - VC; VC1 := VB - VD;	VR1 := 0.37566789 + 0.19550921 I VR2 := 0.12738351 + 0.17216950 I VR3 := -0.00298316 - 0.10707350 I VL := 0.62731527 - 0.08843571 I VC1 := 0.49694860 - 0.36767871 I
VR1 := polar(VR1); VR2 := polar(VR2); VR3 := polar(VR3); VL := polar(VL);	VR1 := polar(0.4234975971, 0.4798584835) VR2 := polar(0.2141702485, 0.9338082695) VR3 := polar(0.1071150486, -1.598649983) VL := polar(0.6335182103, -0.1400519765)

VC1 := polar(VC1); I1 := polar(I1); I2 := polar(I2); I3 := polar(I3);	VC1 := polar(0.6181792174, -0.6369880777) I1 := polar(0.0004269455498, 0.4819361733) I2 := polar(0.0003268692290, 1.598650026) I3 := polar(0.0006513693674, 0.9338082568)

Table 5: Maple code for Mesh analysis

	Amplitude	Phase Difference	Time Domain Form	Phasor Domain Form
I1	0.000427	0.480	0.000426*cos(wt + 0.480)	0.000426e^(j*0.480)
I2	0.000327	-1.60	0.000327*cos(wt + (-1.60))	0.000327e^(j*-1.60)
I3	0.000651	0.934	0.000651*cos(wt + 0.934)	0.000651e^(j*0.934)
VR1	0.423	0.480	0.423*cos(wt +0.480)	0.423e^(j*0.480)
VR2	0.214	0.934	0.214*cos(wt + 0.934)	0.214e^(j*0.934)
VR3	0.107	-1.60	0.107*cos(wt + (-1.60))	0.107e^(j*-1.60)
VC1	0.618	-0.637	0.618*cos(wt + (-0.637))	0.618e^(j*-0.637)
VL	0.633	-0.140	0.633*cos(wt + (-0.140))	0.633e^(j*-0.140)

****rounded to 3 significant digits*

Table 6: Single frequency AC analysis results

Next, the circuit was solved for its node voltages and currents with the Tektronix virtual oscilloscope in Multisim. Since this method involves approximation due to the cursors, there will be uncertainty carried on the yielded values.

Using the multisim Tektronix oscilloscope, the cursor function was used to measure the amplitude of the waves (in volts) and time difference (in seconds), which can later be converted to cycle fractions for phasor form. The distance from the time axis (where voltage = 0) and the peak of the wave was measured to find amplitude, and the distance between the point where the source wave and the point where the component's wave intercepts with the time axis to find the time difference.

All measurements were taken using the following configuration:

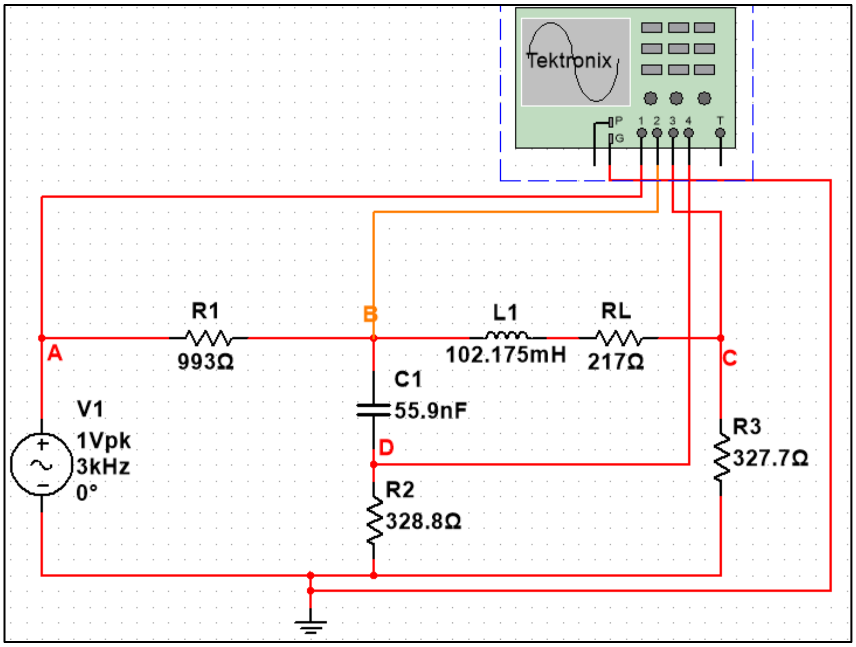
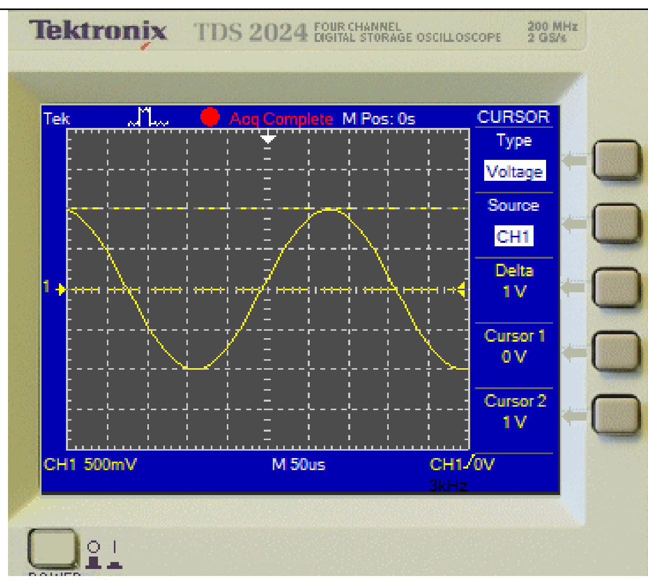
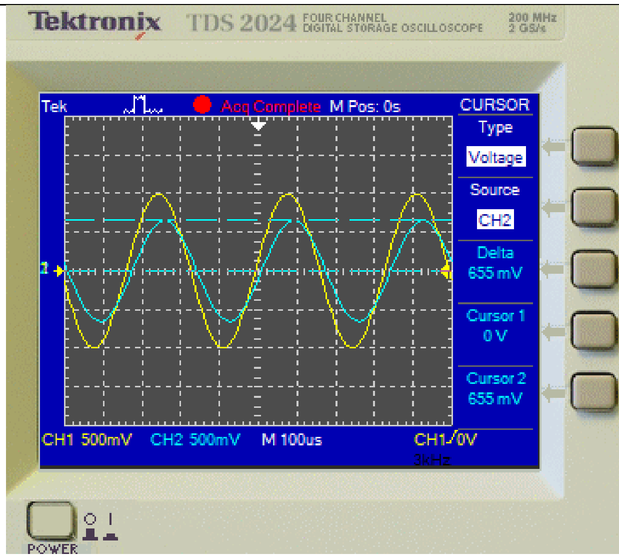


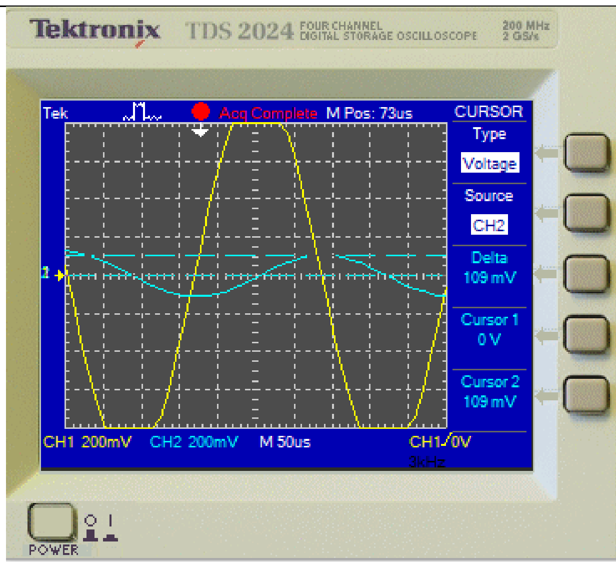
Figure 4: Circuit Diagram with Tektronix oscilloscope

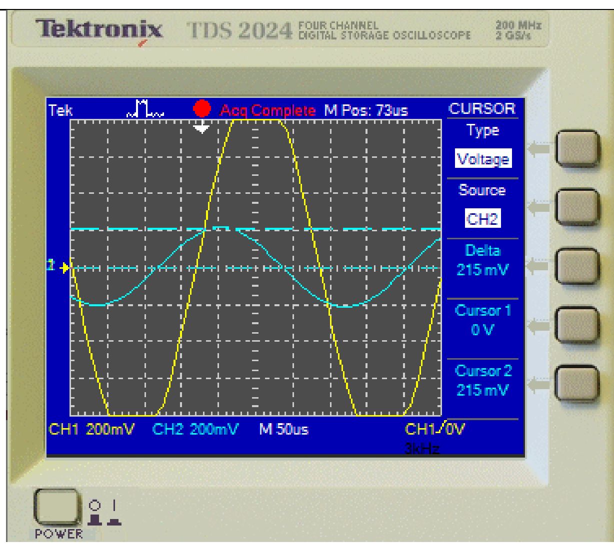
The following are images of the measurements taken with the Tektronix:

Node & measurement	**Image**	**Result (w/ uncertainty)**
Node A – Amplitude & time difference		VA = 1 +- 0.125 V Δt_A = 0 s

<table>
<tr>
<td>Node B -

Amplitude</td>
<td>
</td>
<td>VB = 0.655 +- 0.125 V</td>
</tr>
<tr>
<td>Node C -

Amplitude</td>
<td>
</td>
<td>VC = 0.109 +- 0.050 V</td>
</tr>
<tr>
<td>Node D -

Amplitude</td>
<td>
</td>
<td>VD = 0.215 +- 0.050 V</td>
</tr>
</table>

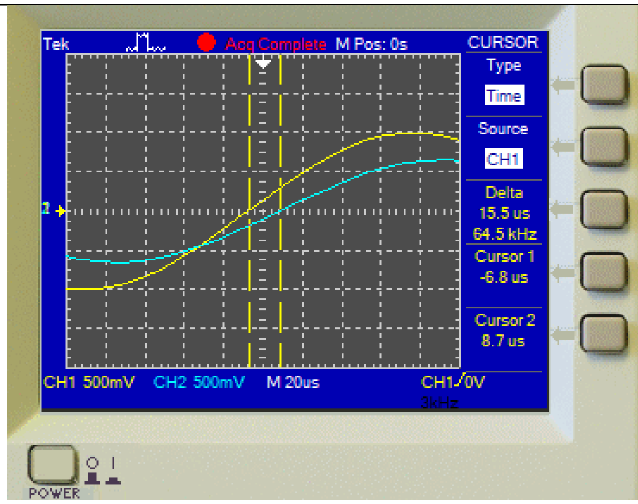
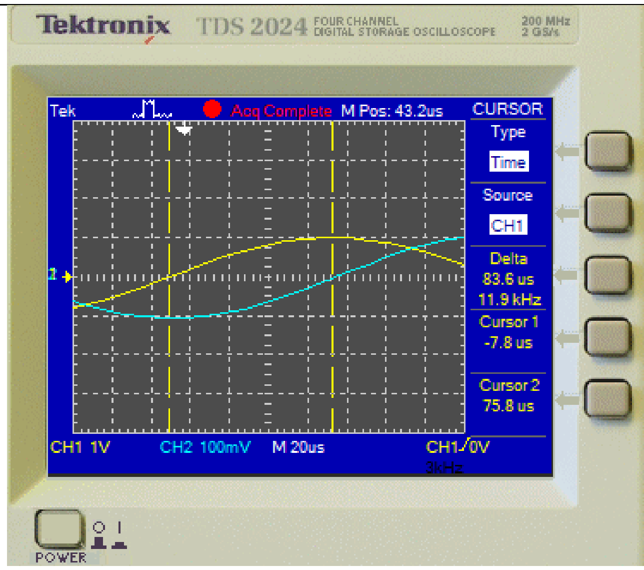
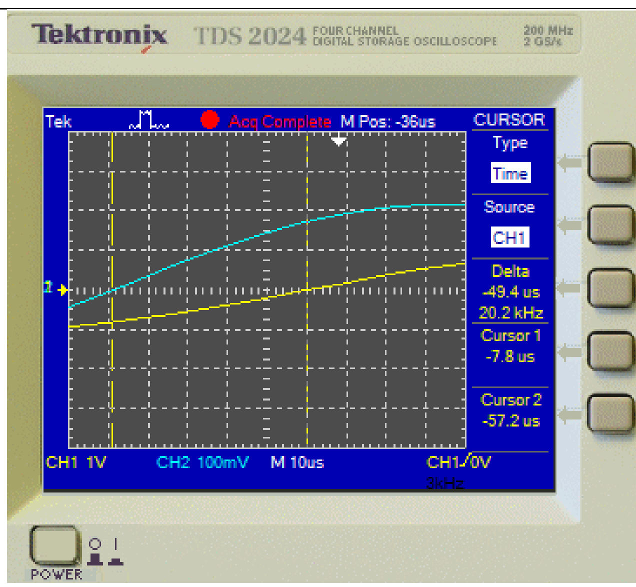
<table>
<tr><td>Node B - Time difference</td><td>
</td><td>Δt_B = -15.5 +- 4 µs</td></tr>
<tr><td>Node C - Time difference</td><td>
</td><td>Δt_C = -83.6 +- 4 µs</td></tr>
<tr><td>Node D - Time difference</td><td>
</td><td>Δt_D = 49.4 +- 2 µs</td></tr>
</table>

Table 7: Measured values with Tektronix oscilloscope

Please note thar the sign of the value in the rightmost column of the table above when measuring time is opposite of the value in the “Delta” box. This is because the sign of the “Delta” box value is dependent on which cursor is ahead of the other, and not on the actual direction of the phase shift. The values in the table were adjusted to reflect the phase shift direction.

The measured phase shifts at each node were converted to fractions of a cycle (phi).

Maple input	**Maple Output**
#Node B DTime := -15.5*10^(-6): DAmplitude := (0.655 &+- 0.125)*V: DphiCycleFractions := DTime*f; DphiRadians := 2*DphiCycleFractions*Pi;	DphiRadians := -0.2921681168
#Node C DTime := -83.6*10^(-6); DAmplitude := (0.109 &+- 0.050)*V; DphiCycleFractions := DTime*f; DphiRadians := 2*DphiCycleFractions*Pi;	DphiRadians := -1.575822875
#Node D DTime := 49.4*10^(-6); DAmplitude := (0.215 &+- 0.050)*V; DphiCycleFractions := DTime*f; DphiRadians := 2*DphiCycleFractions*Pi;	DphiRadians := 0.9311680626

Table 8: Maple code for conversion to cycle fractions

Also, the uncertainties (half of the smallest increment on the scope multiplied by 2 for two cursors) on each measured value were propagated through the conversion to cycle fractions.

Maple input	**Maple Output**
#Node B DTabsolute := 4.*10^(-6): DTrelative := DTabsolute/|DTime|; DphiRadiansAbsolute := |DphiRadians*DTrelative|;	DTrelative := 0.2580645161 DphiRadiansAbsolute := 0.07539822368
#Node C DTabsolute := 4.*10^(-6); DTrelative := DTabsolute/|DTime|; DphiRadiansAbsolute := |DphiRadians*DTrelative|	DTrelative := 0.04784688995 DphiRadiansAbsolute := 0.07539822368

#Node D DTabsolute := 2.*10^(-6); DTrelative := DTabsolute/|DTime|; DphiRadiansAbsolute := |DphiRadians*DTrelative|	DTrelative := 0.04048582996 DphiRadiansAbsolute := 0.03769911185

Table 9: Uncertainties on phase difference

The acquired values can be summarized (rounded to 3 significant digits) and converted to phasor form:

	Amplitude	Phase Difference (radians)	Phasor Domain Form
VA	1 +- 0.125	0	X = (1+-0.125)
VB	0.655 +- 0.125	-0.292 +- 0.0754	X = (0.655 +- 0.125)e^(j*(-0.292 +- 0.0754))
VC	0.109 +- 0.050	-1.58 +- 0.0754	X = (0.109 +- 0.050)e^(j*(-1.58 +- 0.0754))
VD	0.215 +- 0.050	0.931 +- 0.0377	X = (0.215 +- 0.050)e^(j*(0.931 +- 0.0377))

Table 10: Table of values for Tektronix measurements

Next, the voltages across each component were calculated by taking the difference between 2 nodes. Note that the brute force method of uncertainty propagation was used to calculate the uncertainty on the phasor domain form of each component.

With the brute force method, one obtains the upper and lower bounds of the number by making the calculation with the uncertainty added and repeating with the uncertainty subtracted. Then, the difference between the upper and lower bound is found. Half of this difference is the “new” uncertainty. The calculation is then done as if there is no uncertainty, then the recently acquired “new” uncertainty is added.

Maple input	**Maple Output**

restart;
VA = 1 +- 0.125:
VB = (0.655 +- 0.125)*exp((-(0.292 +- 0.0754))*I):
VC = (0.109 +- 0.050)*exp((-(1.58 +- 0.0754))*I):
VD = (0.215 +- 0.050)*exp((0.931 +- 0.0377)*I):
 	
#No error calculations
VA := 1;
VB := 0.655*exp((-0.292)*I);
VC := 0.109*exp((-1.58)*I);
VD := 0.215*exp(0.931*I);
 	
VR1 := VA - VB;
abs(VR1);
argument(VR1);
VR2 := VD + 0;
abs(VR2);
argument(VR2);
VR3 := VC + 0;
abs(VR3);
argument(VR3);
VL := VB - VC;
	VR1 := 0.3727261140 + 0.1885536315 I
	0.4177047139
	0.4683382829
	
	VR2 := 0.1283618937 + 0.1724767354 I
	0.2150000000
	0.9309999999
	
	VR3 := -0.001003186216 - 0.1089953835 I
	0.1090000000
	-1.580000000
	
	VL := 0.6282770722 - 0.0795582480 I
	0.1090000000
	-1.580000000
	
	VC1 := 0.4989119923 - 0.3610303669 I
	0.6158377237

abs(VR3);
argument(VR3);
VC1 := VB - VD;
abs(VC1);
argument(VC1);
	-0.6264131429

restart;
#Upper bound
VA := 1 + 0.125:
VB := (0.655 + 0.125)*exp((-0.292 + 0.0754)*I):
VC := (0.109 + 0.050)*exp((-1.58 + 0.0754)*I):
VD := (0.215 + 0.050)*exp((0.931 + 0.0377)*I):
 	
uVR1 := VA - VB;
uVR1_abs := abs(uVR1);
uVR1_arg := argument(uVR1);
 	
uVR2 := VD + 0;
uVR2_abs := abs(uVR2);
uVR2_arg := argument(uVR2);
 	
uVR3 := VC + 0;
uVR3_abs := abs(uVR3);
uVR3_arg := argument(uVR3);
 	
uVL := VB - VC;
uVL_abs := abs(uVL);
uVL_arg := argument(uVL);
 	
uVC1 := VB - VD;
uVC1_abs := abs(uVC1);
uVC1_arg := argument(uVC1);
	uVR1 := 0.3632256454 + 0.1676300471 I
	uVR1_abs := 0.4000408756
	uVR1_arg := 0.4323792720
	
	uVR2 := 0.1500884222 + 0.2183997837 I
	uVR2_abs := 0.2650000000
	uVR2_arg := 0.9687000001
	
	uVR3 := 0.01051753081 - 0.1586517619 I
	uVR3_abs := 0.1590000000
	uVR3_arg := -1.504600000
	
	uVL := 0.7512568238 - 0.0089782852 I
	uVL_abs := 0.7513104717
	uVL_arg := -0.01195045090
	
	uVC1 := 0.6116859324 - 0.3860298308 I
	uVC1_abs := 0.7233109360
	uVC1_arg := -0.5629677770

#Lower Bound
VA := 1 - 0.125:
VB := (0.655 - 0.125)*exp((-0.292 - 0.0754)*I):
VC := (0.109 - 0.050)*exp((-1.58 - 0.0754)*I):
VD := (0.215 - 0.050)*exp((0.931 - 0.0377)*I):
 	
lVR1 := VA - VB;
lVR1_abs := abs(lVR1);
lVR1_arg := argument(lVR1);
 	
lVR2 := VD + 0;
lVR2_abs := abs(lVR2);
lVR2_arg := argument(lVR2);
 	
lVR3 := VC + 0;
lVR3_abs := abs(lVR3);
lVR3_arg := argument(lVR3);
 	
lVL := VB - VC;
lVL_abs := abs(lVL);
lVL_arg := argument(lVL);
 	
lVC1 := VB - VD;
lVC1_abs := abs(lVC1);
lVC1_arg := argument(lVC1);
	lVR1 := 0.3803698715 + 0.1903707855 I
	lVR1_abs := 0.4253495916
	lVR1_arg := 0.4640384148
	
	lVR2 := 0.1034293041 + 0.1285588544 I
	lVR2_abs := 0.1650000000
	lVR2_arg := 0.8932999999
	
	lVR3 := -0.004985664033 - 0.05878897137 I
	lVR3_abs := 0.05900000000
	lVR3_arg := -1.655400000
	
	lVL := 0.4996157925 - 0.1315818141 I
	lVL_abs := 0.5166524111
	lVL_arg := -0.2575183344
	
	lVC1 := 0.3912008244 - 0.3189296399 I
	lVC1_abs := 0.5047318102
	lVC1_arg := -0.6839756620

#Final Absolute Amplitude Uncertainties Unc_Amp_VR1 := abs(lVR1_abs - uVR1_abs)/2; Unc_Amp_VR2 := abs(lVR2_abs - uVR2_abs)/2; Unc_Amp_VR3 := abs(lVR3_abs - uVR3_abs)/2; Unc_Amp_VL := abs(lVL_abs - uVL_abs)/2; Unc_Amp_VC1 := abs(lVC1_abs - uVC1_abs)/2;	Unc_Amp_VR1 := 0.01265435800 Unc_Amp_VR2 := 0.05000000000 Unc_Amp_VR3 := 0.05000000000 Unc_Amp_VL := 0.1173290303 Unc_Amp_VC1 := 0.1092895629

#Final Absolute Phase Uncertainties Unc_Phase_VR1 := abs(lVR1_arg - uVR1_arg)/2; Unc_Phase_VR2 := abs(lVR2_arg - uVR2_arg)/2; Unc_Phase_VR3 := abs(lVR3_arg - uVR3_arg)/2; Unc_Phase_VL := abs(lVL_arg - uVL_arg)/2; Unc_Phase_VC1 := abs(lVC1_arg - uVC1_arg)/2;	Unc_Phase_VR1 := 0.01582957140 Unc_Phase_VR2 := 0.03770000010 Unc_Phase_VR3 := 0.07540000000 Unc_Phase_VL := 0.1227839418 Unc_Phase_VC1 := 0.06050394250

Table 11: Voltages across each component calculations + uncertainty

From this (very long) set of maple calculations, we can derive the phasor forms of each voltage, with uncertainties:
VR1 = (0.418 +- 0.0126)e^(j*(0.468 +- 0.0158))
VR2 = (0.215 +- 0.0500)e^(j*(0.931 +- 0.0377))
VR3 = (0.109 +- 0.0500)e^(j*(-1.58 +- 0.0754))
VC1 = (0.615 +- 0.109)e^(j*(-0.626 +- 0.0605))
VL = (0.633 +- 0.117)e^(j*(-0.126 +- 0.123))

Finally, the currents throughout the circuit can be calculated using ohm's law.

V = IR I1 = VR1/R1
I = V/R I1 = [(0.418 +- 0.0126)e^(j*(0.468 +- 0.0158))] V / 993 ohm
I1 = [(0.418 +- 0.0126) / 993] * e^(j*(0.468 +- 0.0158))
I1 = [(0.418 +- 3.01%) / 993] * e^(j*(0.468 +- 0.0158))
I1 = (0.000421 +- 3.01%)*e^(j*(0.468 +- 0.0158))
I1 = (0.000421 +- 0.0000127) *e^(j*(0.468 +- 0.0158))

I2 = VR3/R3 = (0.000333 +- 0.000152)*e^(j*(-1.58 +- 0.0754))

I3 = VR2/R2 = (0.000654 +- 0.000152)e^(j*(0.931 +- 0.0377))

The results acquired from the multisim Tektronix oscilloscope are summarized below.

	Amplitude	Phase Difference	Time Domain Form	Phasor Domain Form
I1	0.000421 +- 0.0000127	0.468 +- 0.0158	(0.000421 +- 0.0000127)* cos(wt + (0.468 +- 0.0158))	(0.000421 +- 0.0000127) *e^(j*(0.468 +- 0.0158))
I2	0.000333 +- 0.000152	-1.58 +- 0.0754	(0.000333 +- 0.000152)* cos(wt + (-1.58 +- 0.0754))	(0.000333 +- 0.000152)*e^(j*(-1.58 +- 0.0754))
I3	0.000654 +- 0.000152	0.931 +- 0.0377	(0.000654 +- 0.000152)* cos(wt + (0.931 +- 0.0377))	(0.000654 +- 0.000152)* e^(j*(0.931 +- 0.0377))
VR1	0.418 +- 0.0126	0.468 +- 0.0158	(0.418 +- 0.0126)* cos(wt + (0.468 +- 0.0158))	(0.418 +- 0.0126)*e^(j*(0.468 +- 0.0158))
VR2	0.215 +- 0.0500	0.931 +- 0.0377	(0.215 +- 0.0500)* cos(wt + (0.931 +- 0.0377))	(0.215 +- 0.0500)*e^(j*(0.931 +- 0.0377))
VR3	0.109 +- 0.0500	-1.58 +- 0.0754	(0.109 +- 0.0500)* cos(wt + (-1.58 +- 0.0754))	(0.109 +- 0.0500)*e^(j*(-1.58 +- 0.0754))
VC1	0.615 +- 0.109	-0.626 +- 0.0605	(0.615 +- 0.109)* cos(wt + (-0.626 +- 0.0605))	(0.615 +- 0.109)*e^(j*(-0.626 +- 0.0605))
VL	0.633 +- 0.117	-0.126 +- 0.123	(0.633 +- 0.117)* cos(wt + (-0.126 +- 0.123))	VL = (0.633 +- 0.117)*e^(j*(-0.126 +- 0.123))

Table 12: Results from Tektronix Analysis

When comparing these results to those of the single frequency AC sweep (which almost exactly matched the results from the analytical solution), it is observed that one of the values, I1, is not within the range of uncertainty of the I1 acquired from the Tektronix. There are many possibilities for this discrepancy, but a likely cause is the cursor and display inaccuracy. The box outputting the difference between the cursors tended to round numbers to the nearest multiple of 5. For example, 813 mV would be rounded to 815 mV. However, this should be accounted for in the uncertainty calculations above. So, this error is most likely due to the accuracy of the method of measuring with cursors and the increments in which said cursors measure time and voltage.

	Tektronix	**Single Frequency AC**	Within Error?
I1	(0.000421 +- 0.0000127) *e^(j*(0.468 +- 0.0158))	0.000426e^(j*0.480)	No
I2	(0.000333 +- 0.000152)*e^(j*(-1.58 +- 0.0754))	0.000327e^(j*-1.60)	Yes
I3	(0.000654 +- 0.000152)* e^(j*(0.931 +- 0.0377))	0.000651e^(j*0.934)	Yes
VR1	(0.418 +- 0.0126)*e^(j*(0.468 +- 0.0158))	0.423e^(j*0.480)	Yes
VR2	(0.215 +- 0.0500)*e^(j*(0.931 +- 0.0377))	0.214e^(j*0.934)	Yes
VR3	(0.109 +- 0.0500)*e^(j*(-1.58 +- 0.0754))	0.107e^(j*-1.60)	Yes
VC1	(0.615 +- 0.109)*e^(j*(-0.626 +- 0.0605))	0.618e^(j*-0.637)	Yes
VL	VL = (0.633 +- 0.117)*e^(j*(-0.126 +- 0.123))	0.633e^(j*-0.140)	Yes

Table 13: Single Frequency AC Sweep vs Tektronix (3 sig. figs.)

Physical Solution Via Hantek and Breadboard:

The physical solution largely followed the same process as the Tektronix solution. Using the Hantek, the cursor and measure functions were used to measure the amplitude of the waves and time difference, which can later be converted to cycle fractions for phasor form. The values were measured in the same way as the Tektronix oscilloscope.

All measurements were taken using the following breadboard configuration:

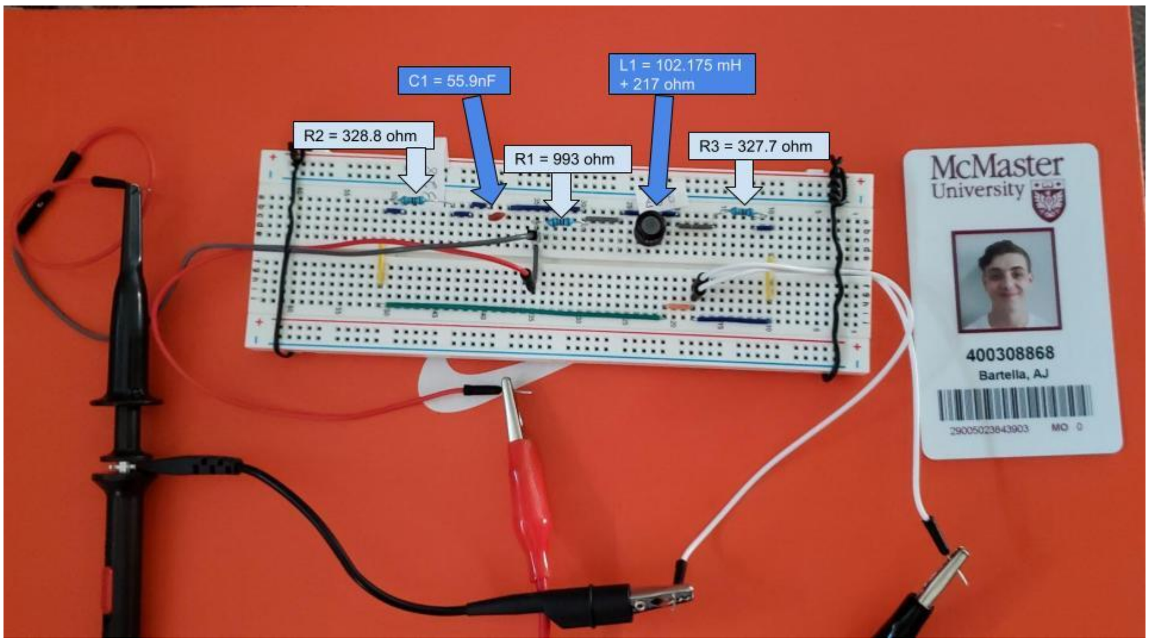
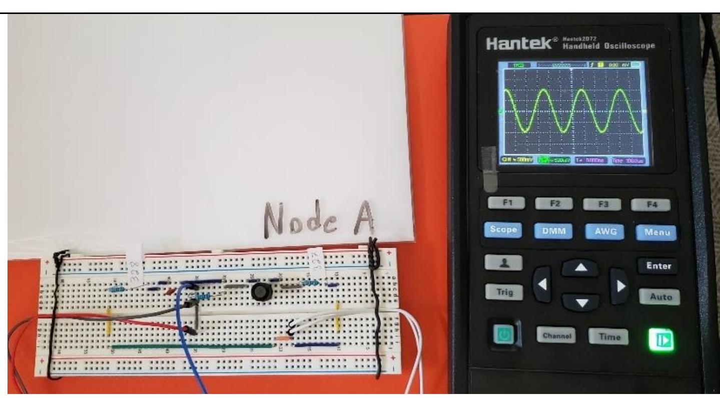
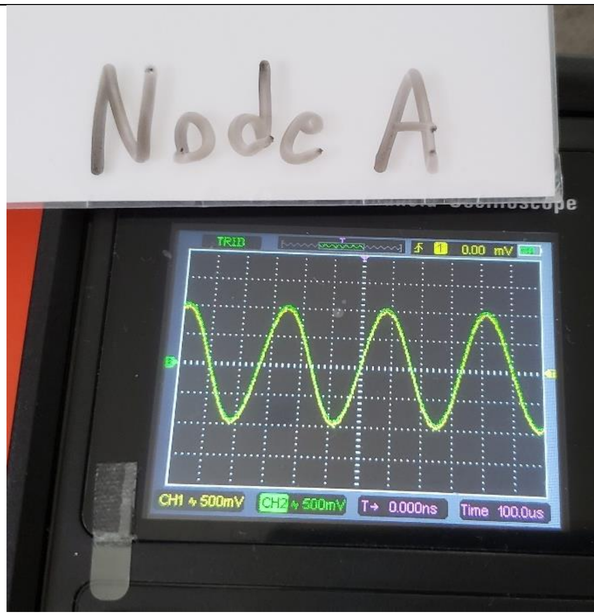
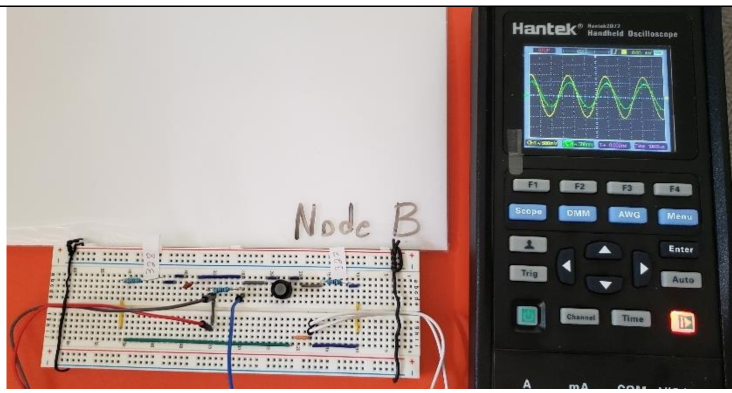
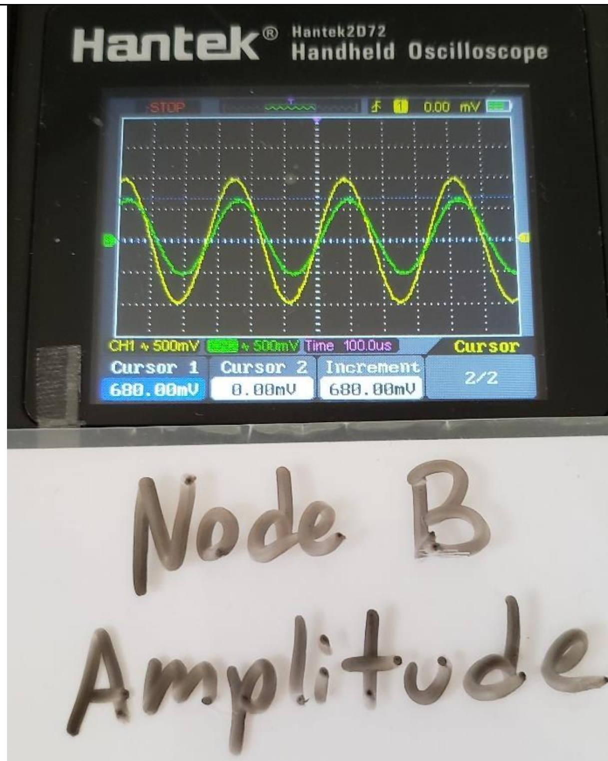


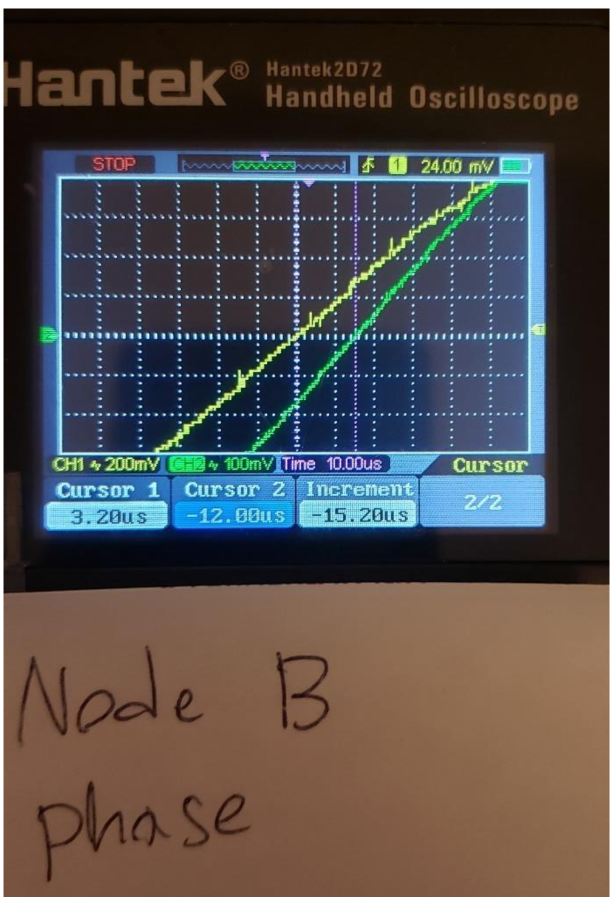
Figure 5: Breadboard configuration

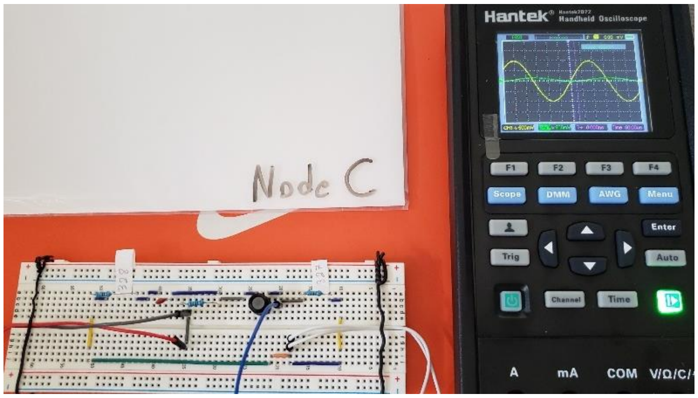
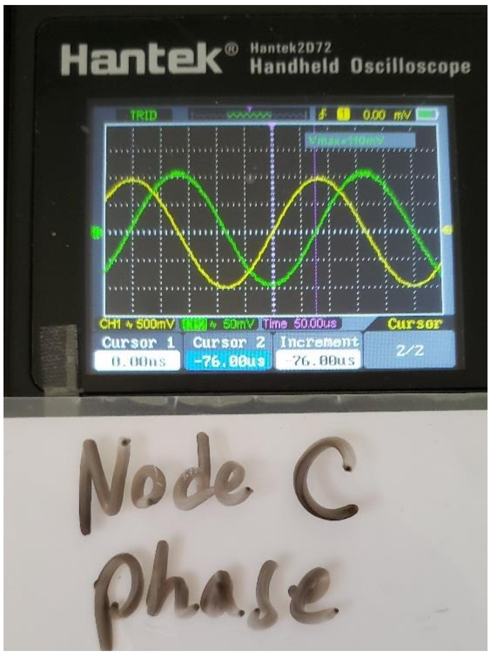
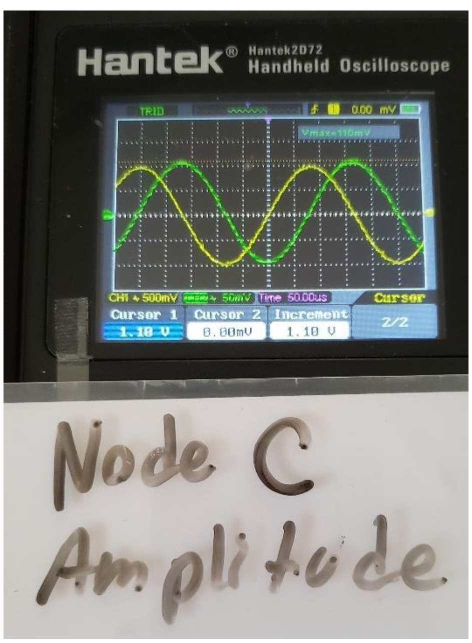
The following are images of the measurements taken with the Tektronix (the blue wire is connected to the Hantek's probe):

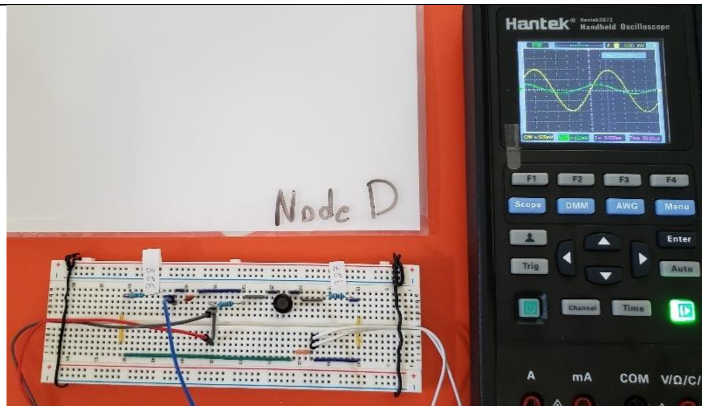
Node	Image	Result (w/ uncertainty)
Node A		VA = 1 +- 0.100 V Δt_A = 0 s

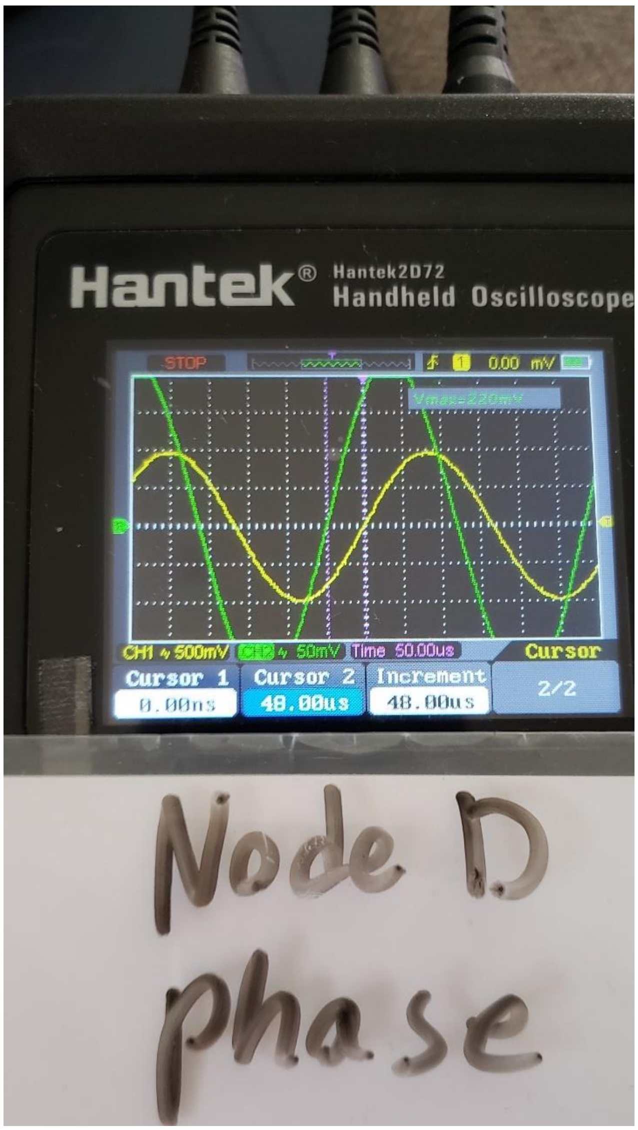
<table>
<tr>
<td></td>
<td>
</td>
<td></td>
</tr>
<tr>
<td>Node B</td>
<td>
</td>
<td>VB = 0.680 +- 0.100 V
Δt_B = -15.2e-6 +- 2e-6 s</td>
</tr>
</table>


Hantek®
Hantek2D72
Handheld Oscilloscope
STOP
0.00 mV
CH1 500mV
CH2 500mV
Time 100.0us
Cursor
Cursor 1
680.00mV
Cursor 2
0.00mV
Increment
680.00mV
2/2
Node B
Amplitude


Hantek®
Hantek2D72
Handheld Oscilloscope
STOP
24.00 mV
CH1 200mV
CH2 100mV
Time 10.00us
Cursor
Cursor 1
3.20us
Cursor 2
-12.00us
Increment
-15.20us
2/2
Node B
phase

Node C	   	VC = 0.110 +- 0.010 V Δt_C = -76.0e-6 +- 10e-6 s

<table>
<tr>
<td>Node D -
Amplitude</td>
<td>


</td>
<td>VD = 0.220 +- 0.020 V
Δt_D =48.0e-6 +- 10e-6 s</td>
</tr>
</table>

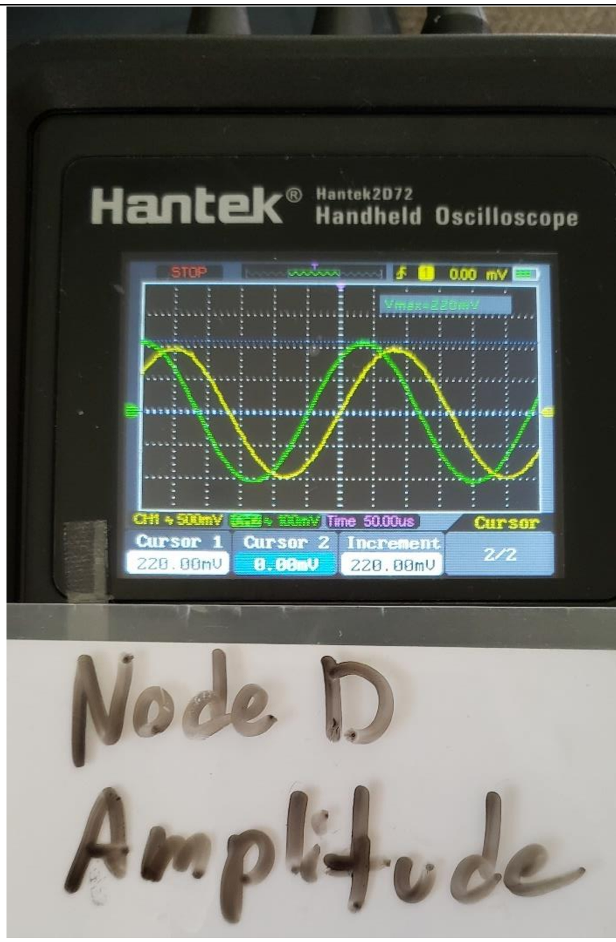
		

Table 14: Measured values with Hantek oscilloscope

Repeating the methods used when measuring from the Tektronix, the measured phase shifts at each node were converted to fractions of a cycle (phi).

Maple input	**Maple Output**
#Node B DTime := -15.2*10^(-6); DAmplitude := (0.680 &+- 0.100); DphiCycleFractions := DTime*f; DphiRadians := 2*DphiCycleFractions*Pi;	DphiRadians := -0.2865132500
#Node C DTime := -76.0*10^(-6); DAmplitude := (0.110 &+- 0.010)*V; DphiCycleFractions := DTime*f; DphiRadians := 2*DphiCycleFractions*Pi;	DphiRadians := -1.432566250

#Node D DTime := 48.0*10^(-6); DAmplitude := 0.220 &+- 0.020; DphiCycleFractions := DTime*f; DphiRadians := 2*DphiCycleFractions*Pi;	DphiRadians := 0.9047786844

Table 15: Maple code for conversion to cycle fractions

It can be observed that these values are similar to those acquired from the multisim Tektronix analysis.

Also, the uncertainties (half of the smallest increment on the scope multiplied by 2 for two cursors, plus 5% for the error of the Hantek) on each measured value were propagated through the conversion to cycle fractions.

Maple input	**Maple Output**
#Node B DTabsolute := 2.*10^(-6) + 0.05*DTime; DTrelative := DTabsolute/abs(DTime); DphiRadiansAbsolute := |DphiRadians*DTrelative|;	DTrelative := 0.08157894737 DphiRadiansAbsolute := 0.02337344934
#Node C DTabsolute := 10.*10^(-6) + 0.05*DTime DTrelative := DTabsolute/|DTime|; DphiRadiansAbsolute := |DphiRadians*DTrelative|	DTrelative := 0.08157894737 DphiRadiansAbsolute := 0.1168672467
#Node D DTimeAbsolute := 10.*10^(-6) + 0.05*DTime DTrelative := DTabsolute/|DTime|; DphiRadiansAbsolute := |DphiRadians*DTrelative|	DTimeRelative := 0.2583333333 DphiRadiansAbsolute := 0.2337344934

Table 16: Uncertainties on phase difference

The acquired values can be summarized (rounded to 3 significant digits) and converted to phasor form:

	Amplitude	Phase Difference (radians)	Phasor Domain Form
VA	1 +- 0.100	0	X = (1+-0.100)
VB	0.680 +- 0.100	-0.287 +- 0.0234	X = (0.680 +- 0.100)e^(j*(-0.287 +- 0.0234))
VC	0.110 +- 0.010	-1.432 +- 0.117	X = (0.110 +- 0.010)e^(j*(-1.432 +- 0.117))
VD	0.220 +- 0.020	0.905 +- 0.234	X = (0.220 +- 0.020)e^(j*(0.905 +- 0.234))

Table 17: Table of values for Hantek measurements

Next, the voltages across each component were calculated by taking the difference between 2 nodes, with uncertainty being carried once again by the brute force method.

Maple input	**Maple Output**

restart;
VA = 1 + (&+- 0.100);
VB = (0.680 + (&+- 0.100))*exp((-0.2865132500 + (&+- 0.02337344934))*I);
VC = (0.110 + (&+- 0.010))*exp((-1.432566250 + (&+- 0.1168672467))*I);
VD = (0.220 + (&+- 0.020))*exp((0.9047786844 + (&+- 0.2337344934))*I);
 	
"No error";
VA := 1;
VB := 0.608*exp((-0.2865132500)*I);
VC := 0.110*exp((-1.432566250)*I);
VD := 0.220*exp(0.9047786844*I);
VR1 := VA - VB;
abs(VR1);
argument(VR1);
 	
VR2 := VD + 0;
abs(VR2);
argument(VR2);
 	
VR3 := VC + 0;
abs(VR3);
argument(VR3);
 	
VL := VB - VC;
abs(VL);
argument(VL);
 	
VC1 := VB - VD;
abs(VC1);
argument(VC1);
	VR1 := 0.4167850638 + 0.1718264768 I
	0.4508149593
	0.3910359291
	
	VR2 := 0.1359291149 + 0.1729834551 I
	0.2200000000
	0.9047786842
	
	VR3 := 0.01515693198 - 0.1089507568 I
	0.1100000000
	-1.432566250
	
	VL := 0.5680580042 - 0.0628757200 I
	0.5715271230
	-0.1102366691
	
	VC1 := 0.4472858213 - 0.3448099319 I
	0.5647641057
	-0.6567396823

restart;
#Upper bound
VA = 1 + 0.100;
VB = (0.680 + 0.100)*exp((-0.2865132500 + 0.02337344934)*I);
VC = (0.110 + 0.010)*exp((-1.432566250 + 0.1168672467)*I);
VD = (0.220 + 0.020)*exp((0.9047786844 + 0.2337344934)*I);
 	
uVR1 := VA - VB;
uVR1_abs := abs(uVR1);
uVR1_arg := argument(uVR1);
 	
uVR2 := VD + 0;
uVR2_abs := abs(uVR2);
uVR2_arg := argument(uVR2);
 	
uVR3 := VC + 0;
uVR3_abs := abs(uVR3);
uVR3_arg := argument(uVR3);
 	
uVL := VB - VC;
uVL_abs := abs(uVL);
uVL_arg := argument(uVL);
 	
uVC1 := VB - VD;
uVC1_abs := abs(uVC1);
uVC1_arg := argument(uVC1);
	uVR1 := 0.3468491332 + 0.2028885702 I
	uVR1_abs := 0.4018309260
	uVR1_arg := 0.5292780338
	
	uVR2 := 0.1005468044 + 0.2179227848 I
	uVR2_abs := 0.2400000000
	uVR2_arg := 1.138513178
	
	uVR3 := 0.03028075009 - 0.1161166490 I
	uVR3_abs := 0.1200000000
	uVR3_arg := -1.315699003
	
	uVL := 0.7228701167 - 0.0867719212 I
	uVL_abs := 0.7280594563
	uVL_arg := -0.1194664384
	
	uVC1 := 0.6526040624 - 0.4208113550 I
	uVC1_abs := 0.7765141716
	uVC1_arg := -0.5727242588

#Lower Bound
VA = 1 - 0.100;
VB = (0.680 - 0.100)*exp((-0.2865132500 - 0.02337344934)*I);
VC = (0.110 - 0.010)*exp((-1.432566250 - 0.1168672467)*I);
VD = (0.220 - 0.020)*exp((0.9047786844 - 0.2337344934)*I);
 	
lVR1 := VA - VB;
lVR1_abs := abs(lVR1);
lVR1_arg := argument(lVR1);
lVR2 := VD + 0;
lVR2_abs := abs(lVR2);
lVR2_arg := argument(lVR2);
lVR3 := VC + 0;
lVR3_abs := abs(lVR3);
lVR3_arg := argument(lVR3);
lVL := VB - VC;
lVL_abs := abs(lVL);
lVL_arg := argument(lVL);
lVC1 := VB - VD;
lVC1_abs := abs(lVC1);
lVC1_arg := argument(lVC1);
	lVR1 := 0.3476264863 + 0.1768714260 I
	lVR1_abs := 0.3900354796
	lVR1_arg := 0.4706606744
	
	lVR2 := 0.1566345621 + 0.1243608216 I
	lVR2_abs := 0.2000000000
	lVR2_arg := 0.6710441912
	
	lVR3 := 0.002136120494 - 0.09997718234 I
	lVR3_abs := 0.1000000000
	lVR3_arg := -1.549433497
	
	lVL := 0.5502373932 - 0.07689424366 I
	lVL_abs := 0.5555842993
	lVL_arg := -0.1388481861
	
	lVC1 := 0.3957389516 - 0.3012322476 I
	lVC1_abs := 0.4973431258
	lVC1_arg := -0.6506238715

#Final Absolute Amplitude Uncertainties Unc_Amp_VR1 := abs(lVR1_abs - uVR1_abs)/2; Unc_Amp_VR2 := abs(lVR2_abs - uVR2_abs)/2; Unc_Amp_VR3 := abs(lVR3_abs - uVR3_abs)/2; Unc_Amp_VL := abs(lVL_abs - uVL_abs)/2; Unc_Amp_VC1 := abs(lVC1_abs - uVC1_abs)/2; #Final Absolute Phase Uncertainties Unc_Phase_VR1 := abs(lVR1_arg - uVR1_arg)/2; Unc_Phase_VR2 := abs(lVR2_arg - uVR2_arg)/2; Unc_Phase_VR3 := abs(lVR3_arg - uVR3_arg)/2; Unc_Phase_VL := abs(lVL_arg - uVL_arg)/2; Unc_Phase_VC1 := abs(lVC1_arg - uVC1_arg)/2;	Unc_Amp_VR1 := 0.005897723200 Unc_Amp_VR2 := 0.02000000000 Unc_Amp_VR3 := 0.01000000000 Unc_Amp_VL := 0.08623757850 Unc_Amp_VC1 := 0.1395855229 Unc_Phase_VR1 := 0.02930867970 Unc_Phase_VR2 := 0.2337344934 Unc_Phase_VR3 := 0.1168672470 Unc_Phase_VL := 0.009690873850 Unc_Phase_VC1 := 0.03894980635

Table 18: Voltages across each component calculations + uncertainty

From this, we can derive the phasor forms of each voltage (rounded):

VR1 = (0.451 +- 0.00590)e^(j*(0.391 +- 0.0293))
VR2 = (0.220 +- 0.0200)e^(j*(0.904 +- 0.234))
VR3 = (0.110 +- 0.0100)e^(j*(-1.43 +- 0.117))
VC1 = (0.565 +- 0.139)e^(j*(-0.657 +- 0.0389))
VL = (0.572 +- 0.086)e^(j*(-0.110 +- 0.00969))

Finally, the currents throughout the circuit can be calculated using ohm's law, just like in the Tektronix solution.

V = IR   I1 = VR1/R1
I = V/R   I1 = (0.000454 +- 0.00000595) *e^(j*(0.391 +- 0.0293))

I2 = VR3/R3 = (0.000336 +- 0.0000305)*e^(j*(-1.43 +- 0.117))

I3 = VR2/R2 = (0.000671 +- 0.0000610)e^(j*(0.904 +- 0.234))

The results acquired from physical measurements are summarized below.

	Amplitude	Phase Difference	Time Domain Form	Phasor Domain Form
I1	0.000454 +- 0.00000595	0.391 +- 0.0293	(0.000454 +- 0.00000595)* cos(wt + (0.391 +- 0.0293))	(0.000454 +- 0.00000595) *e^(j*(0.391 +- 0.0293))
I2	0.000333 +- 0.000152	-1.43 +- 0.117	(0.000336 +- 0.0000305)* cos(wt + (-1.43 +- 0.117))	(0.000336 +- 0.0000305)*e^(j*(-1.43 +- 0.117))
I3	0.000654 +- 0.000152	0.904 +- 0.234	(0.000671 +- 0.0000610)* cos(wt + (0.904 +- 0.234))	(0.000671 +- 0.0000610)e^(j*(0.904 +- 0.234))
VR1	0.451 +- 0.00590	0.391 +- 0.0293	(0.451 +- 0.00590)*cos(wt + (0.391 +- 0.0293))	(0.451 +- 0.00590)e^(j*(0.391 +- 0.0293))
VR2	0.220 +- 0.0200	0.904 +- 0.234	(0.220 +- 0.0200)*cos(wt + (0.904 +- 0.234))	(0.220 +- 0.0200)e^(j*(0.904 +- 0.234))
VR3	0.110 +- 0.0100	-1.43 +- 0.117	(0.110 +- 0.0100)*cos(wt + (-1.43 +- 0.117))	(0.110 +- 0.0100)e^(j*(-1.43 +- 0.117))
VC1	0.565 +- 0.139	-0.657 +- 0.0389	(0.565 +- 0.139)*cos(wt + (-0.657 +- 0.0389))	(0.565 +- 0.139)e^(j*(-0.657 +- 0.0389))
VL	0.572 +- 0.086	-0.110 +- 0.00969	(0.572 +- 0.086)*cos(wt + (-0.110 +- 0.00969))	(0.572 +- 0.086)e^(j*(-0.110 +- 0.00969))

Table 19: Results from Physical Analysis

When comparing the results from the physical analysis to the digital, it can be observed that all values are similar, but none are within error bars. A likely cause for this is the natural inaccuracy of the Hantek's oscilloscope. Although efforts were made to ensure good contact, poor contact with the board is possible. In addition, the output voltage of the AWG is estimated to have a larger amplitude than displayed. Another cause for error is the variance of physical conditions (as some of the measurements were taken at home and others on-campus). Because of this variance in location, differences in humidity and electromagnetic interference could have changed the measured output.

Another source of error is the nature of the capacitor, as Mr. Johnasson stated in a help session that the capacitor's actual capacitance value can vary if its position is changed (i.e. if it were bent slightly in transportation). Also, a source of error related to the inductor can stem from our inability to measure the inductance with the Hantek, as an approximation based on the values provided by Mr. Johnasson were used to determine the inductance.

	Physical	**Single Frequency AC**	Within Error?
I1	(0.000454 +- 0.00000595) *e^(j*(0.391 +- 0.0293))	0.000426e^(j*0.480)	No
I2	(0.000336 +- 0.0000305)*e^(j*(-1.43 +- 0.117))	0.000327e^(j*-1.60)	No
I3	(0.000671 +- 0.0000610)e^(j*(0.904 +- 0.234))	0.000651e^(j*0.934)	No
VR1	(0.451 +- 0.00590)e^(j*(0.391 +- 0.0293))	0.423e^(j*0.480)	No
VR2	(0.220 +- 0.0200)e^(j*(0.904 +- 0.234))	0.214e^(j*0.934)	No
VR3	(0.110 +- 0.0100)e^(j*(-1.43 +- 0.117))	0.107e^(j*-1.60)	No
VC1	(0.565 +- 0.139)e^(j*(-0.657 +- 0.0389))	0.618e^(j*-0.637)	No
VL	(0.572 +- 0.086)e^(j*(-0.110 +- 0.00969))	0.633e^(j*-0.140)	No

Table 20: Single Frequency AC Sweep vs Hantek

Conclusion

The task of the lab was to find the voltages across and current through 3 or more components in a circuit physically, mathematically, and digitally. Having solved using each of these methods, it was observed that the mathematical and single frequency AC sweep are the same, while the Tektronix measurements match (within error), and the physical measurements are similar, but not within error bars. This lab was a challenge for me: I was confused when I began, but as I worked through the lab I deepened my understanding of phasors and AC current (which were both very unfamiliar topics to me). Also, I learned about node voltages, and how measuring voltages at the nodes allows one to find voltages and currents through any component. In addition, I learned a lot about the physical build process, such as using the Hantek as a voltage source, using the “Measure” function on the Hantek for more accurate data, and how to use the cursors.

All in all, it was difficult for me to grasp the content of topic 2 and apply it, but as I worked on lab H2 I sharpened my skills and deepened my understanding of the course content. I can safely say I have a much stronger grasp on the content than earlier thanks to the lab.