DC And AC Currents Through LCR Circuits

PHY224 Fall 2021

Fredrik Dahl Bråten, Pankaj Patil

November 30, 2021

Abstract

The aim of this experiment is to study the behavior of different circuit elements, like Resistors, Capacitors and Inductors in DC and AC circuits. We observed that a capacitor opposes increase in charge accumulation, while an Inductor opposes changes in the current passing through it. As a result, a Capacitor in a DC circuit is an open circuit, while an Inductor in a DC circuit is a short circuit. In this report we present these behaviors with the help of Oscilloscope measurements.

1 Introduction

A. Background Theory

In a RC circuit, the voltage of the capacitor is given by $V_0 = \frac{Q_0}{C}$, and as we know $I_0 = \frac{V_R}{R} \implies V_R = V_0 e^{-\frac{t}{\tau}}$, where $\tau = RC$. This is also the fitting equation for this case. Here the measured time is our independent variable, while V_R across the resistor is our dependent variable. The capacitance (C) and the the resistance (R) are auxiliary parameters.

In a LR circuit, we have $I_R(t) = I_0(1 - e^{\frac{t}{L/R}}) \implies V_R = V_0(1 - e^{\frac{t}{L/R}})$, where $I_0 = \frac{V_0}{R}$. Our fitting equation is the same, and again time is our independent variable, and voltage across the Resistor V_R, is the dependent variable. L and R are auxiliary parameters.

In a LC circuit, we have $v(t) = \frac{q(t)}{C} = \frac{Q_0}{C}\cos(\omega_r t) = V_0 \cos(\omega_r t)$, where $\omega_r = \frac{1}{\sqrt{LC}}$, and $v(t)$ is the transient voltage drop across capacitor. Here the time is independent variable and voltage drop across capacitor is dependent variable. L, C are auxiliary parameters. This is an idealization of LC circuit which is hard to achieve in practical circuits.

In a LCR circuit, the total impedance Z is defined as $Z = \sqrt{(\omega L - \frac{1}{\omega C})^2 + R^2}$. Z can be found by measuring the total Voltage V, compared to the voltage $V_R = RI \implies Z = \frac{V}{V_R}R$.

It is clear from the fitting equation for the RC circuit that we would observe exponential decay for the Voltage across the resistor, as can be seen in Figure 1. In the case of the LR circuit, the relationship is reversed, and we observe exponential increase in voltage across the Resistor, as seen in Figure 3.

B. Materials and Methods

In this experiment we used various circuit elements namely Resistors, Capacitors, and Inductors, in the setups provided in the lab manual [1]. The measurements were made by connecting the two wires of the channels of the Oscilloscope to each side of the circuit element(s) for which we wanted to measure voltage across over time.

We followed the steps provided in the lab manual [1] for this experiment. We also made adjustments based on TA's suggestion to the circuits as the AC circuits in the manual were applicable to old Oscilloscopes.

2 Results

A. Transient Decay (RC)

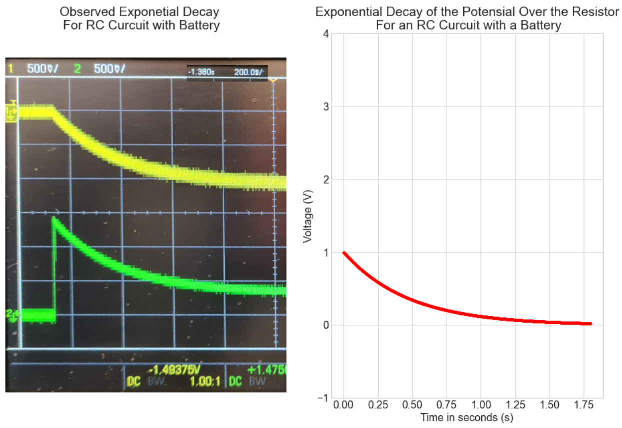


Figure 1: Transient Potential Decay in RC circuit (Measurement Vs. Simulation)

$$\tau \ (measured) = 0.4 \ s, \quad RC = 0.47 \ s$$

The estimated value of the time constant, estimated from our measurements, matches the theoretical value RC closely.

B. DC Square Wave

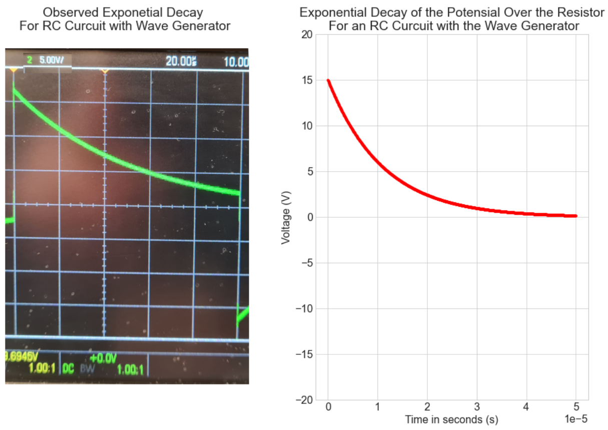


Figure 2: Exponential Potential Decay in RC circuit (Measurement Vs. Simulation)

$$\tau \ (measured) = 8 \times 10^{-5} \ s$$
$$RC = 1.1 \times 10^{-5} \ s$$

The estimated value of the time constant, estimated from our measurements, matches the theoretical value RC in order of magnitude.

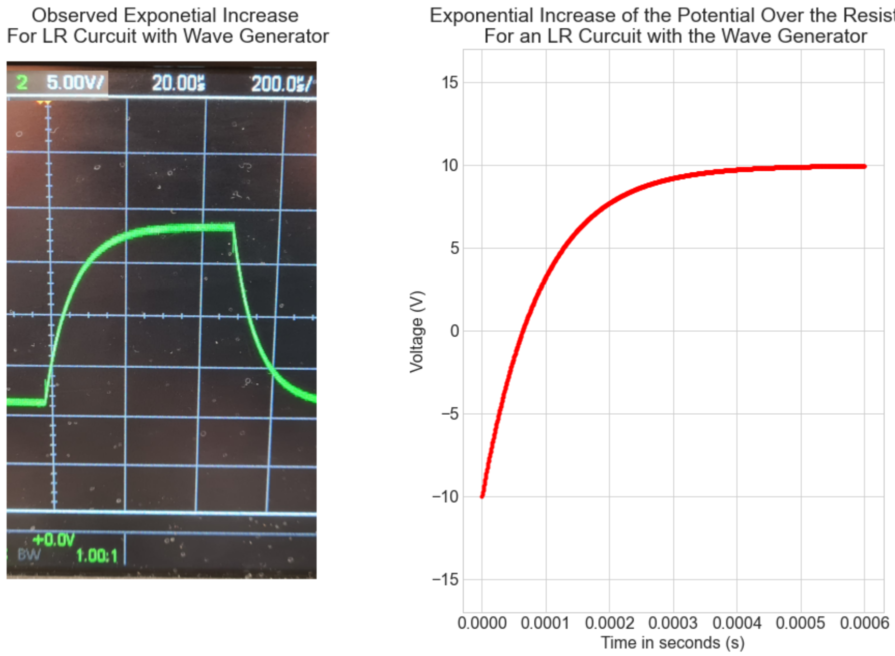
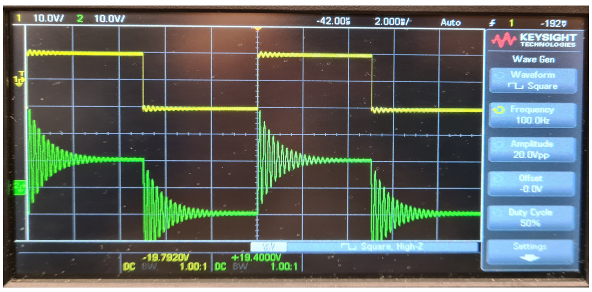


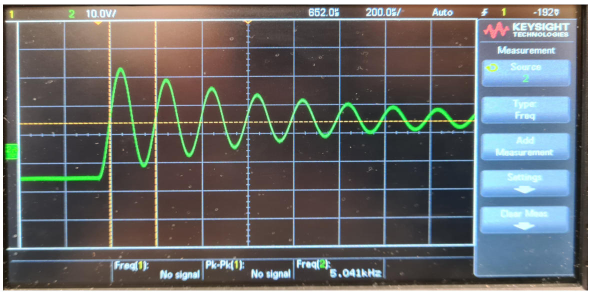
Figure 3: Exponential Potential Increase in LR circuit (Measurement Vs. Simulation)

$$\tau \ (measured) = 5.0 \times 10^{-5} \ s$$
$$L/R = 9.2 \times 10^{-5} \ s$$

The estimated value of the time constant, estimated from our measurements, matches the theoretical value L/R in order of magnitude. All these three theoretical curves matches our measured curves well, and as mentioned, gives time constants within an order of magnitude of the measured values.



(a) LC Circuit with Square Wave



(b) LC Circuit with Square Wave (Zoomed In)

Figure 4: LC Circuit

For the LC circuit, the observed behavior is different from the theoretically predicted one. This is because we can never make pure LC circuit, as there will always be some resistance in the circuit which will cause damping.

C. AC Sine Wave

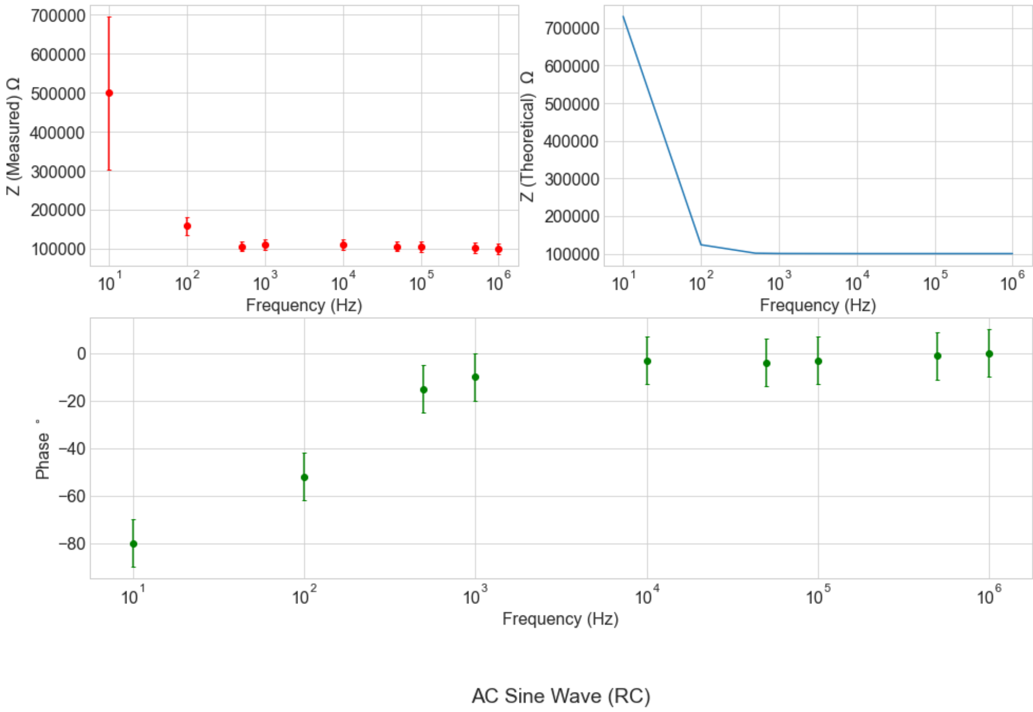


Figure 5: AC Sine Wave (RC): Z vs Frequency and Phase vs Frequency Plots

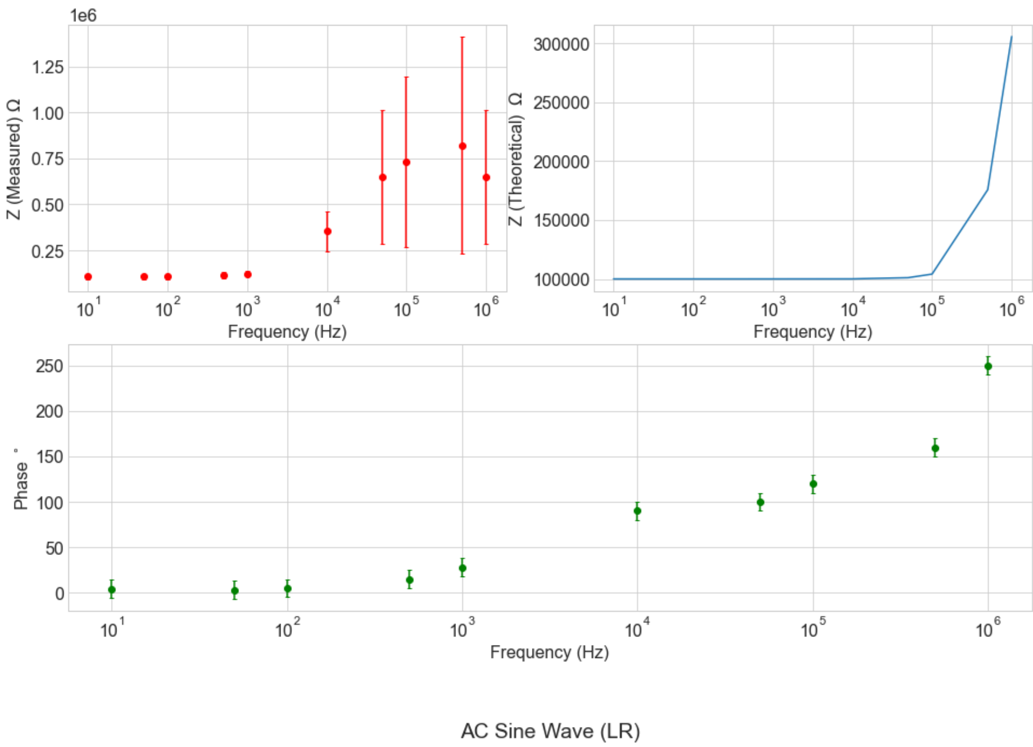


Figure 6: AC Sine Wave (LR): Z vs Frequency and Phase vs Frequency Plots

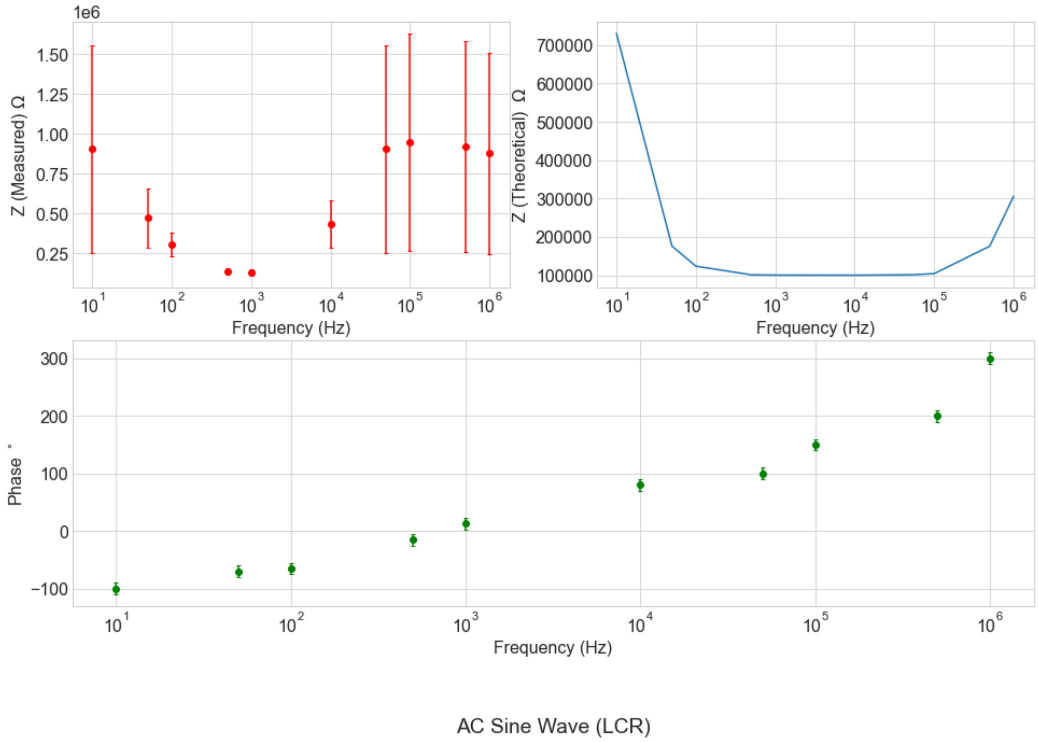


Figure 7: AC Sine Wave (LCR): Z vs Frequency and Phase vs Frequency Plots

3 Uncertainty

For the purpose of this experiment, we estimated the uncertainty in V_R as the thickness of the voltage line plotted on the Oscilloscope for channel 2. This channel measures the voltage across the Resistor. In case of the DC circuit, with the RC configuration, we observe that $\Delta V_R = \frac{1}{4} \times$ Voltage Division $= .25 \times 500\ mV = 125\ mV$. And for the DC Square Wave, it is $2\ V$. Similarly for the LR Square Wave case, $\Delta V_R = 1\ V$. Overall, for the experiment we can assume the uncertainty in the measurement of V_R to be in order of magnitude $1\ V$.

4 Discussion

In the RC circuit, the accuracy of the measurement of V_R in the DC case, is given by $\frac{|1.4/e-1.|}{1.4/e} \times 100 = 35\%$. And the precision of measurement of V_R is given by $\frac{0.125}{0.7} \times 100 = 17\%$. The accuracy value being large can be attributed to instrument error/inaccuracy.

In the LR circuit, the accuracy of V_R in the DC case is given by $\frac{|12.64-10.0|}{12.64} \times 100 = 20\%$. And the precision of measurement of V_R is given by $\frac{1}{10} \times 100 = 10\%$.

In the LCR circuit for lower frequencies, the phase difference is negative. As the frequency is increased, the phase gradually shifts to positive values. The resonance frequency can be estimated to be around 1 kHz. Furthermore, in accordance with theory, $\omega_r = 1/\sqrt{LC} = 1.4$ kHz.

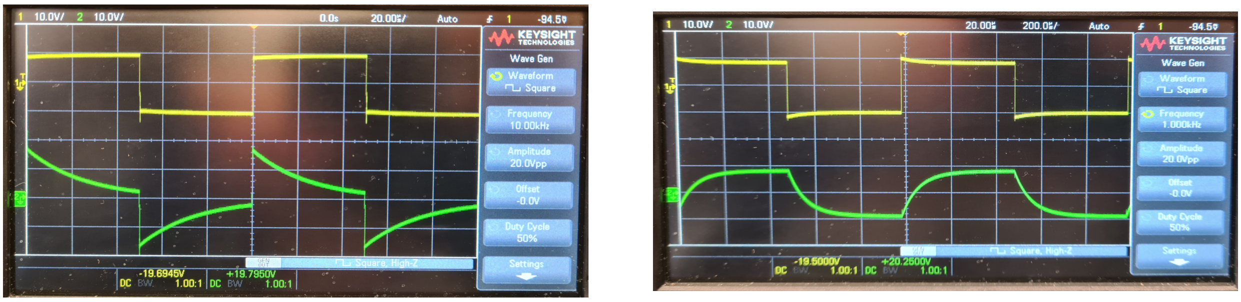
From our LC circuit, we saw that we still had some resistance within the circuit. This is because an LC circuit would be an idealization of the actual LCR circuit for which we were really testing. Though the resistance was small, there was some internal resistance within the capacitor and the inductor in our circuit, along with some internal resistance within the Oscilloscope. Thus, we never managed to get data from a pure LC circuit. If one were to make a circuit diagram of our

"LC" setup, using only ideal resistors, ideal capacitors, ideal inductors and an ideal oscilloscope, one would have to draw the LC circuit diagram for which can be found in the manual, along with a small resistor in series right behind or right before the capacitor, the inductor and the oscilloscope, to account for their internal resistance. In fact, the inductor does actually have some small capacitance as well, from accumulated charge between each wiring. Thus, a small capacitor would need to be drawn in the circuit diagram in series with the inductor as well, to capture the full picture of our imperfect LC circuit.

Given the capacitance, the inductance and the resistance within our LCR circuit, one can calculate the circuits natural resonance frequency. Though driving the circuit with our wave generator at this natural frequency would give the largest resonance, one can also drive the circuit at all natural number multiples of this natural frequency and get some resonance. These are the overtones of the fundamental note, the natural frequency. This is similar to how one with musical instruments can achieve resonance on a string or in the air within a flute by playing the natural frequency or any of the overtones associated to the natural frequency of a string on a violin, a guitar, or associated to the pipes of a flute or an organ.

A Appendix

A.1 Oscilloscope Readings



(a) RC Circuit with Square Wave

(b) LR Circuit with Square Wave

A.2 Experimental Data

A.2.1 RC Sine Wave

```
Frequency, V, V_R, Phase diff
10, 1.3, 0.260, -80
100, 1.3, 0.820, -52
500, 1.19, 1.12, -15
1000, 1.27, 1.15, -10
10000, 1.25, 1.13, -3
50000, 1.22, 1.15, -4
100000, 1.17, 1.11, -3
500000, 1.11, 1.09, -1
1000000, 1.11, 1.11, 0.0
```

A.2.2 LR Sine Wave

```
Frequency, V, V_R, Phase diff
10, 1.13, 1.03, 4.0
50, 1.13, 1.03, 3.0
100, 1.13, 1.03, 5.0
500, 1.13, 0.98, 15.0
1000, 1.17, 0.940, 28
10000, 1.21, 0.340, 90
50000, 1.17, 0.180, 100
100000, 1.17, 0.160, 120
500000, 1.15, 0.140, 160
1000000, 1.17, 0.180, 250
```

A.2.3 LCR Sine Wave

```
Frequency, V, V_R, Phase diff
10, 1.27, 0.140, -100
50, 1.23, 0.260, -70
100, 1.29, 0.420, -65
500, 1.23, 0.900, -15
1000, 1.21, 0.920, 13
10000, 1.31, 0.300, 80
50000, 1.27, 0.140, 100
100000, 1.33, .140, 150
500000, 1.29, .140, 200
1000000, 1.23, .140, 300
```

A.3 Python Code

A.4 Simulated Curves

```
# -*- coding: utf-8 -*-
"""
Created on Thu Nov 25 16:47:07 2021

@author: Fredrik
"""

#Importing modules
import numpy as np
import matplotlib.pyplot as plt

plt.style.use("seaborn-whitegrid")

#Defining constants
R_RC_B = 470*10**3 #Ohm
C_RC_B = 1*10**-6 #Ferad
V0_RC_B = 1

R_RC_W = 500
C_RC_W = 0.022*10**-6
V0_RC_W = 15

R_LR_W = 500
L_LR_W = 46*10**-3 #Henry
V0_LR_W = 20

#Tau:
Tau_RC_B = R_RC_B*C_RC_B
Tau_RC_W = R_RC_W*C_RC_W
Tau_LR_W = L_LR_W/R_LR_W

#Making array of points for plotting graphs:
t_RC_B = np.linspace(0,0.2*9,1000)
V_RC_B = V0_RC_B*np.exp(-t_RC_B/Tau_RC_B)
t_RC_W = np.linspace(0,10*10**-6*5,1000)
V_RC_W = V0_RC_W*np.exp(-t_RC_W/Tau_RC_W)
t_LR_W = np.linspace(0,200*10**-6*3,1000)
V_LR_W = V0_LR_W*np.exp(-t_LR_W/Tau_LR_W)


#we then plot the corresponding curves of exponential decay or increase
#For the RC circuit with battery:
fig = plt.figure(figsize=(16,10))
ax = fig.add_subplot(1,2,1)
ax.axis('off')
ax.set_title("Observed Exponetial Decay \nFor RC Curcuit with Battery")
ax.grid()
```

```
ax = fig.add_subplot(1,2,2)
ax.set_title("Exponential Decay of the Potensial Over the Resistor"+
             "\n For an RC Curcuit with a Battery")
ax.plot(t_RC_B,V_RC_B,c='r', ls='', marker='.',lw=1)
ax.set_xlabel("Time in seconds (s)")
ax.set_ylabel("Voltage (V)")
ax.set_ylim(-1, 4)
ax.figure.savefig("Exponential Decay of the Potential Over the Resistor"+
                  "For an RC Curcuit with a Battery"+".png")

#For the RC cirvuit with wave generator
fig = plt.figure(figsize=(16,10))
ax = fig.add_subplot(1,2,1)
ax.axis('off')
ax.set_title("Observed Exponetial Decay \nFor RC Curcuit with Wave Generator")
ax.grid()

ax = fig.add_subplot(1,2,2)
ax.set_title("Exponential Decay of the Potensial Over the Resistor"+
             "\n For an RC Curcuit with the Wave Generator")
ax.plot(t_RC_W,V_RC_W,c='r', ls='', marker='.',lw=1)
ax.set_xlabel("Time in seconds (s)")
ax.set_ylabel("Voltage (V)")
ax.set_ylim(-20, 20)
ax.figure.savefig("Exponential Decay of the Potential Over the Resistor"+
                  "For an RC Curcuit with the Wave Generator"+".png")


#For the LR circuit with wave generator
fig = plt.figure(figsize=(16,10))
ax = fig.add_subplot(1,2,1)
ax.axis('off')
ax.set_title("Observed Exponetial Increase \nFor LR Curcuit with Wave Generator")
ax.grid()

ax = fig.add_subplot(1,2,2)
ax.set_title("Exponential Increase of the Potential Over the Resistor"+
             "\n For an LR Curcuit with the Wave Generator")
ax.plot(t_LR_W,10-V_LR_W,c='r', ls='', marker='.',lw=1)
ax.set_xlabel("Time in seconds (s)")
ax.set_ylabel("Voltage (V)")
ax.set_ylim(-17, 17)
ax.figure.savefig("Exponential Increase of the Potential Over the Resistor"+
                  "For an LR Curcuit with the Wave Generator"+".png")


plt.show()
```

A.5 Impedance and Phase Plots

The Python code for this exercise is divided into two files. The statslab.py file contains utility methods which we will be frequently using in this course. The lab_7.py file contains the code which analyzes the data.

A.5.1 statslab.py

```
import numpy as np
import scipy.optimize as optim
import matplotlib.pyplot as plt


##################################################################################
# Utility Methods Library
#
# This file contains some utility method which are common to our data analysis.
# This library also contains customized plotting methods.
##################################################################################

# use bigger font size for plots
plt.rcParams.update({'font.size': 16})

def chi2(y_measure,y_predict,errors):
    """Calculate the chi squared value given a measurement with errors and
    prediction"""
    return np.sum( np.power(y_measure - y_predict, 2) / np.power(errors, 2) )

def chi2reduced(y_measure, y_predict, errors, number_of_parameters):
    """Calculate the reduced chi squared value given a measurement with errors
    and prediction, and knowing the number of parameters in the model."""
    return chi2(y_measure, y_predict, errors)/ \
            (y_measure.size - number_of_parameters)

def read_data(filename, skiprows=1, usecols=(0,1), delimiter=","):
    """Load give\n file as csv with given parameters,
    returns the unpacked values"""
    return np.loadtxt(filename,
                      skiprows=skiprows,
                      usecols=usecols,
                      delimiter=delimiter,
                      unpack=True)

def fit_data(model_func, xdata, ydata, yerrors, guess=None):
    """Utility function to call curve_fit given x and y data with errors"""
    popt, pcov = optim.curve_fit(model_func,
                                 xdata,
                                 ydata,
```

```
                                absolute_sigma=True,
                                sigma=yerrors,
                                p0=guess)

    pstd = np.sqrt(np.diag(pcov))
    return popt, pstd

# y = ax+b
def linear_regression(xdata, ydata):
    """Simple linear regression model"""
    x_bar = np.average(xdata)
    y_bar = np.average(ydata)
    a_hat = np.sum( (xdata - x_bar) * (ydata - y_bar) ) / \
            np.sum( np.power((xdata - x_bar), 2) )
    b_hat = y_bar - a_hat  * x_bar
    return a_hat, b_hat
```

A.5.2 lab_7.py

```
#!/usr/bin/env python3
# @author: Pankaj
# -*- coding: utf-8 -*-

import math
import statslab as utils
import matplotlib.pyplot as plt
import numpy as np

# import the measured data from data files
datasets = [
    {
     "name": "AC Sine Wave (RC)",
     "file": "../data/rc_sine.csv",
     "R": 100000,
     "C": 0.022 * 10.0 ** (-6),
     "L": 0,
     "impedance_func": lambda w, R, C, L:
         np.sqrt(1.0 / (w * C) ** 2 + R ** 2)
    },
    {
     "name": "AC Sine Wave (LR)",
     "file": "../data/lr_sine.csv",
     "R": 100000,
     "C": 0,
     "L": 46*10**-3,
     "impedance_func": lambda w, R, C, L:
         np.sqrt( (w * L) ** 2 + R ** 2)
    },
    {
     "name": "AC Sine Wave (LCR)",
     "file": "../data/lcr_sine.csv",
     "R": 100000,
     "C": 0.022 * 10.0 ** (-6),
     "L": 46*10** (-3),
     "impedance_func": lambda w, R, C, L:
         np.sqrt( R ** 2 + np.power(w * L - 1.0 / (w * C), 2 ) )
    }
]

voltage_uncertainty = 0.1 #Volts
phase_uncertainty = 10 #degrees

def analyze_data(data):
    print(data["name"])
    print("\tAnalyzing file %s" % data["file"])
    frequency, v_total, v_r, phase = utils.read_data(data["file"],
```

```
                       usecols=(0, 1, 2, 3),
                       skiprows=1)

# frequency = omega / 2pi
omega = 2*math.pi*frequency

# z = v/v_r * R
z_measured = v_total / v_r * data["R"]
z_errors = z_measured * np.sqrt(
    (voltage_uncertainty/v_total) ** 2 + (voltage_uncertainty/v_r) ** 2)

# z = sqrt((omega L - 1/omega C)^2 + R^2)
z_theory = data["impedance_func"](omega, data["R"], data["C"], data["L"])

fig = plt.figure(figsize=(16,10))
fig.tight_layout()

plt.subplot(2,2,1)
plt.errorbar(frequency,
         z_measured,
         yerr=z_errors,
         marker="o",
         c="r",
         label="Z (Measured) $\Omega$",
         capsize=2,
         ls="")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Z (Measured) $\Omega$")
plt.semilogx()


plt.subplot(2,2,2)
plt.semilogx(frequency, z_theory)
plt.xlabel("Frequency (Hz)")
plt.ylabel("Z (Theoretical)  $\Omega$")

plt.subplot(2,1,2)
plt.errorbar(frequency,
         phase,
         yerr=phase_uncertainty,
         marker="o",
         c="g",
         label="Phase (Measured) $^\circ$",
         capsize=2,
         ls="")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Phase $^\circ$")
plt.title(data["name"], y=-0.5)
plt.semilogx()
```

```
    plt.savefig("%s.png" % data["name"], bbox_inches='tight')


for data in  datasets:
    analyze_data(data)
```

References

[1] Currents in LCR - currents-l-c-r.pdf (https://q.utoronto.ca/courses/235154/files/15436318/download?wrap=1).