RLC Transients

Christopher Hunt

Objectives

The aim of this lab is to investigate the influence of resistance in a RLC circuit by simulating different scenarios using *LTspice*. The lab consists of two parts: the underdamped case and the overdamped case, with two simulations each, using different resistor values. The data collected from these simulations will be analyzed using python to find values for α, ω_d, B_1, and B_2 for the underdamped case, and s_1, s_2, A_1, and A_2 for the overdamped case. The results from the simulations will be compared with theoretical results to better understand the circuit's response and the effect of resistance on the circuit.

Equipment

- LTSpice

Part 1: Underdamped

Case 1: $R_2 = 80\ \Omega$

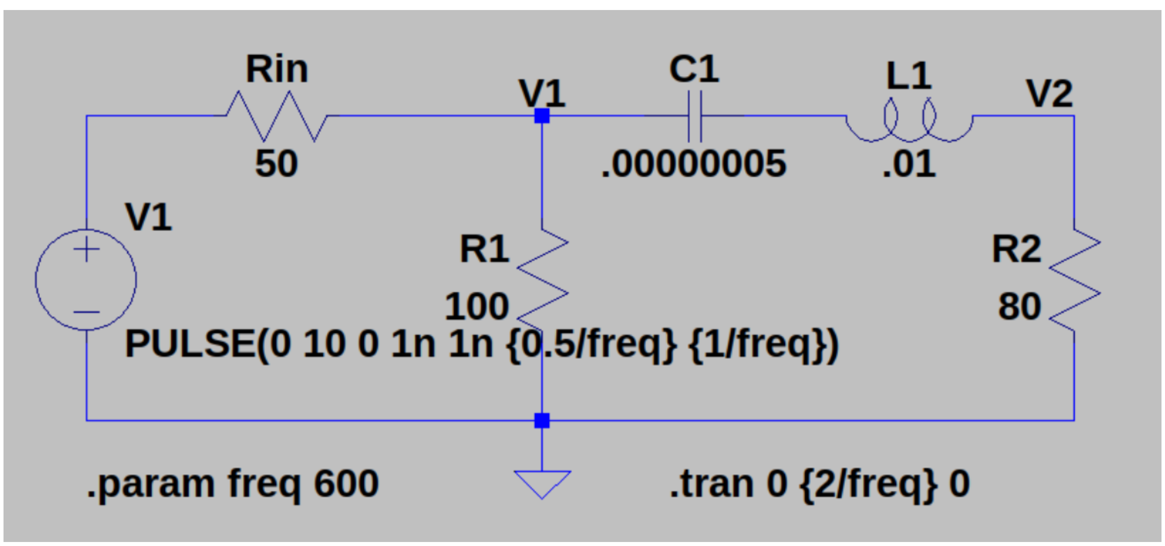

Figure 1: Case 1.1

Case 2: $R_2 = 320\ \Omega$

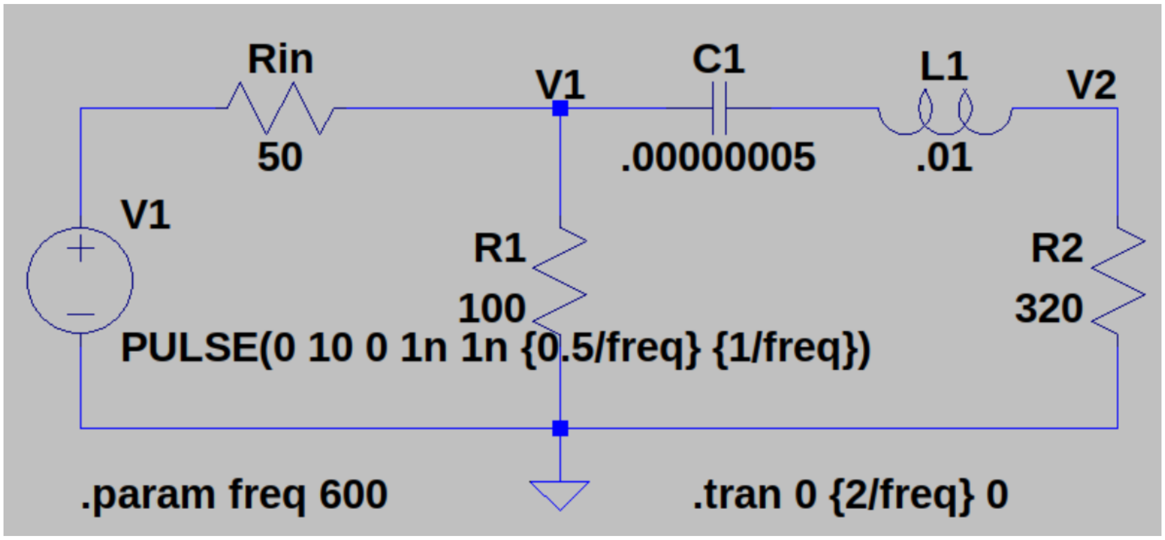

Figure 2: Case 1.2

Part 1 Analysis

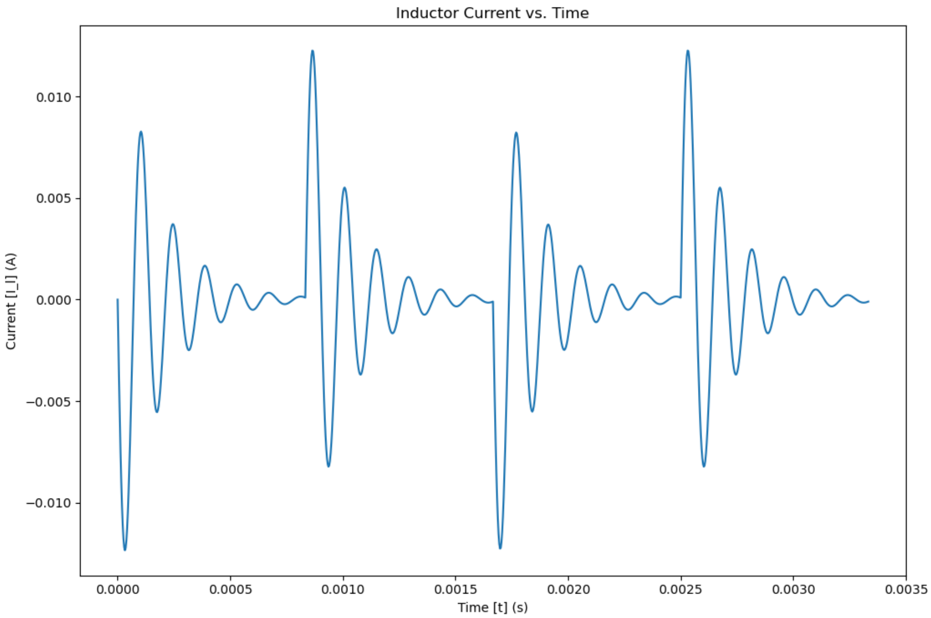

Figure 3: $R_2 = 80\ \Omega$

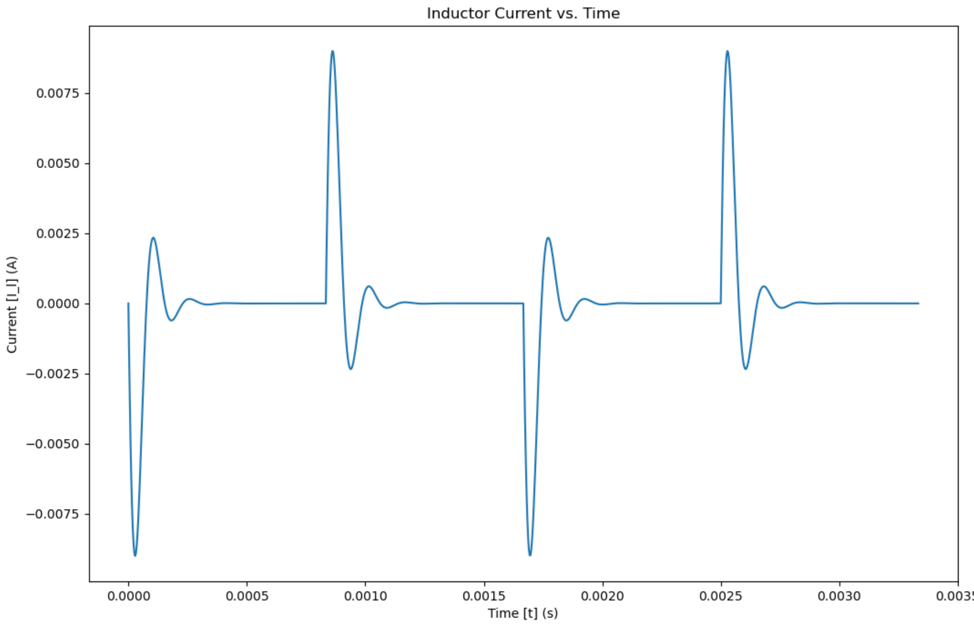

Figure 4: $R_2 = 320\ \Omega$

Upon viewing the current response of the circuit when the step function is applied we see that the greater the value of R_2 the fewer oscillations occur before the current stabilizes towards zero amps. Counting the cycles, case 1 oscillates through 6 periods while case 2 oscillates for 2.
The general form of a 2^{nd} order underdamped response is:

$$i(t) = e^{-\alpha t}(B_1 cos(\omega_d t) + B_2 sin(\omega_d t))$$

From this data it is possible to extract the coefficients α, ω_d, B_1, and B_2. The α term can be found by finding the slope of the normalized decay envelope. The ω_d term can be found by finding the inverse of the period of the oscillations. B_1 can be found if you set a t = 0, at which point the function becomes $i(0) = B_1$. And finally, to find B_2 find the points where the absolutely value of i(t) is at their maximum during each half period. This will form the envelope of the decay component. Normalize this function, the slope is α and the y-intercept is B_2.

Theoretic Analysis

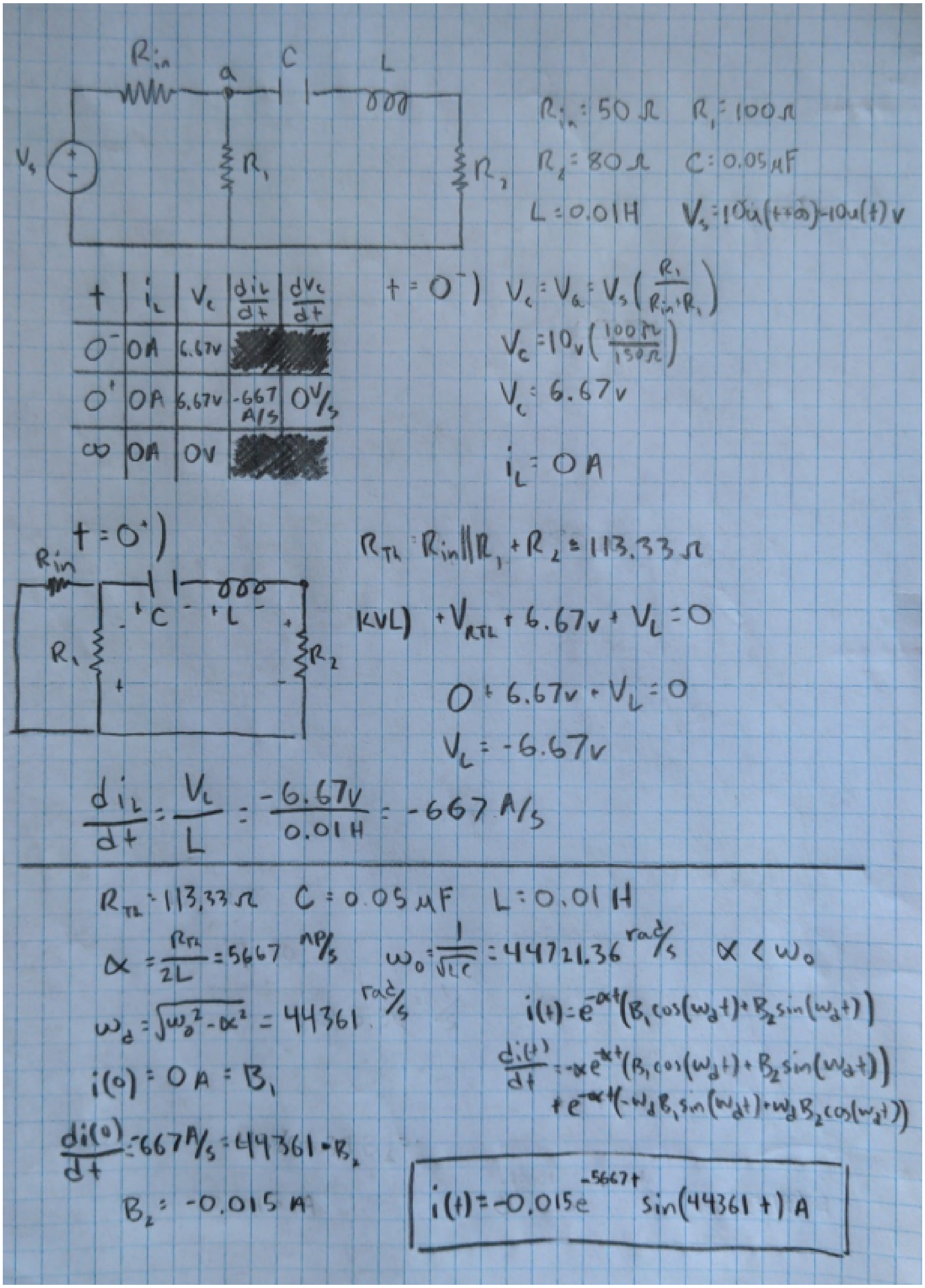

Figure 5: Theoretical Analysis: Underdamped

From the theoretical analysis (Fig. 5) it was found that the coefficients for the circuit are $\alpha = 5667\,\frac{np}{s}$, $\omega_d = 44361\frac{rad}{s}$, $B_1 = 0\ A$, and $B_2 = -0.015\ A$. The current response function when $R_2 = 80\ \Omega$ can be expressed as:

$$\alpha = 5667\,\frac{np}{s} \qquad \omega_d = 444361\,\frac{rad}{s} \qquad B_1 = 0\ A \qquad B_2 = -0.015\ A$$

$$i(t) = -0.015e^{-5667t}sin(44361t)\ A$$

Data Analysis

Simulation data has been taken and processed using the following python code. From this we were able to find values for the current response function coefficients.

$$\alpha = 5633.069\,\frac{np}{s} \qquad \omega_d = 44483.537\,\frac{rad}{s} \qquad B_1 = 0\ A \qquad B_2 = -0.01486\ A$$

$$i(t) = -0.01486e^{-5633.069t}sin(44483.537t)\ A$$

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks

# Define a function to read data from a file
def read_data(filename):
    x = []
    y = []

    with open(filename, 'r') as file:
        for line in file:
            try:
                parts = line.strip().split('\t')
                x_val = float(parts[0])
                y_val = float(parts[1])
                x.append(x_val)
                y.append(y_val)
            except:
                continue

    return x, y

# Read the data from the file
filename = 'ENGR203_lab2_p1_1.txt'
x, y = read_data(filename)

# Isolate the first Source Free Response
x, y= x[:282], y[:282]

# Find the period of oscillation
zero_cross = [x[index] for index, val in enumerate(y[:-1]) \
              if val >= 0 and y[index + 1] < 0]
```

```
periods = [zero_cross[i] - zero_cross[i - 1] \
           for i in range(1, len(zero_cross))]
average_period = sum(periods) / len(periods)
omega_d = 2 * np.pi / average_period  # rad/s
f_d = 1 / average_period  # Hz

# To find the decay envelope, take the absolute value of the function
# and find the peak of each oscillation
y_abs = np.abs(y)
peak_indices, _ = find_peaks(y_abs)
max_y = [y_abs[i] for i in peak_indices]
max_x = [x[i] for i in peak_indices]


# Linearize the exponential function f(t) = Ae^(-alpha*t)
# ln(f(t)) = ln(A)+(-alpha)t
ln_y = np.log(max_y)

# Find the slope of the linearized function
dy = np.array([ln_y[i] - ln_y[i-1] \
               for i in range(1, len(ln_y))])
dx = np.array([max_x[i] - max_x[i-1] \
               for i in range(1, len(max_x))])
slopes = np.array(dy/dx)
slope = sum(slopes/len(slopes))
alpha = slope * -1

# Find the y-intercept: Ln(A)
# ln(f(t)) = -alpha*t + Ln(A) --> Ln(A) = ln(f(t)) + alpha*t
# A = exp(ln(ft)+alpha*t)
y_int = np.exp(ln_y[1] + alpha * max_x[1])

# Print the required values
print(f"Alpha = {alpha}")
print(f"Angular frequency: {omega_d}")
print(f"B_2 = {y_int}")
```

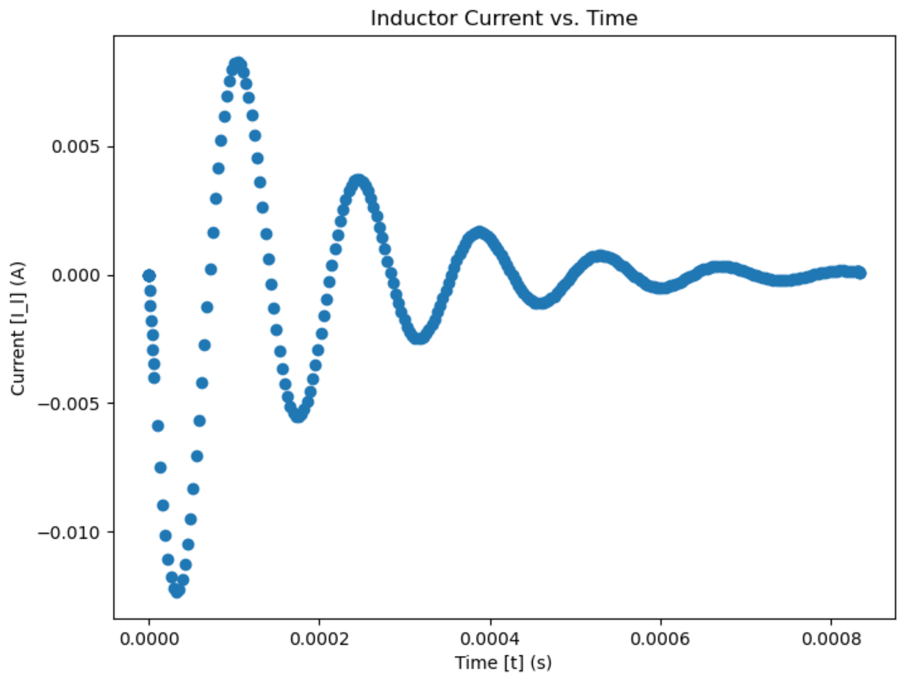

Figure 6: RLC Circuit Response

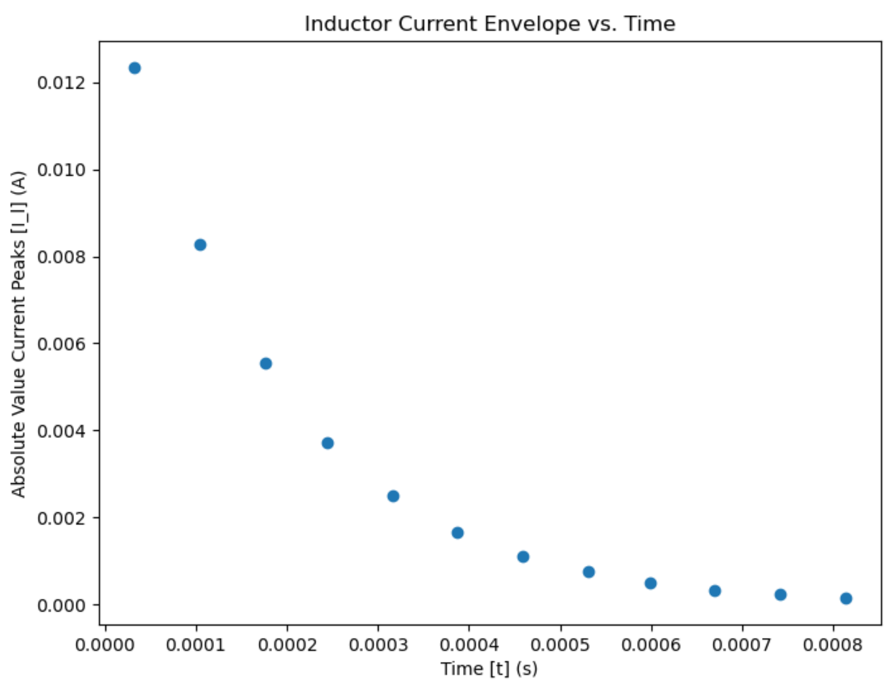

Figure 7: Current Response Decay Envelope

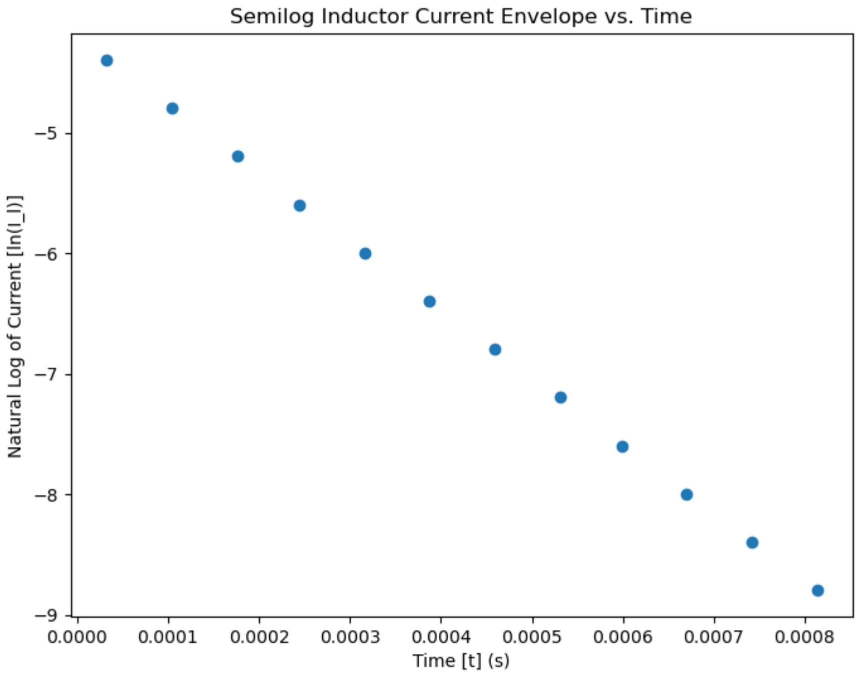

Figure 8: Linearized Current Response Decay Envelope

Part 2: Overdamped

Case 1: $R_2 = 1500\ \Omega$

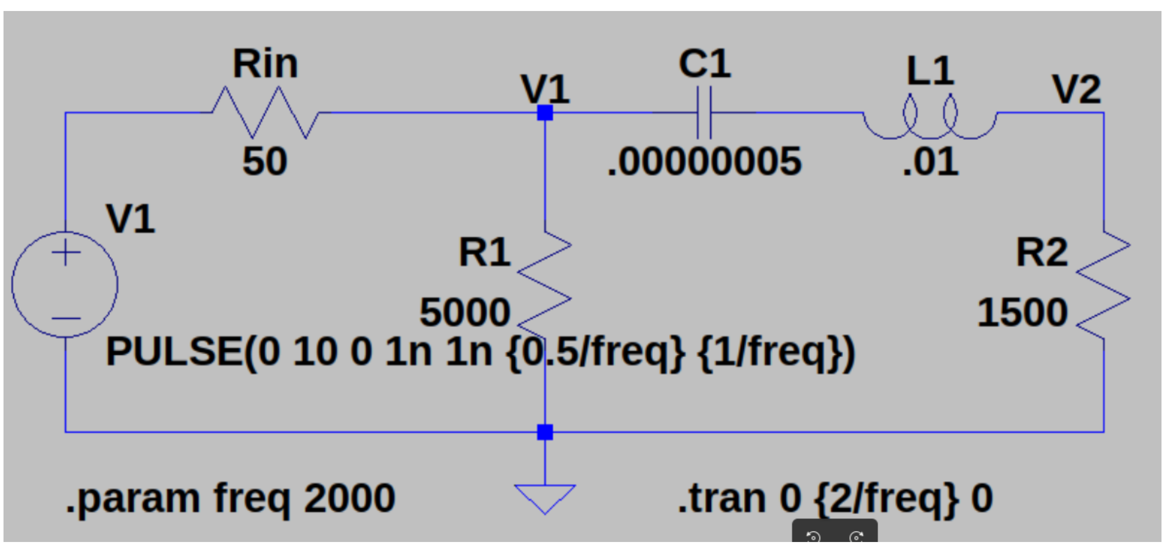

Figure 9: Case 2.1

Case 2: $R_2 = 6000\ \Omega$

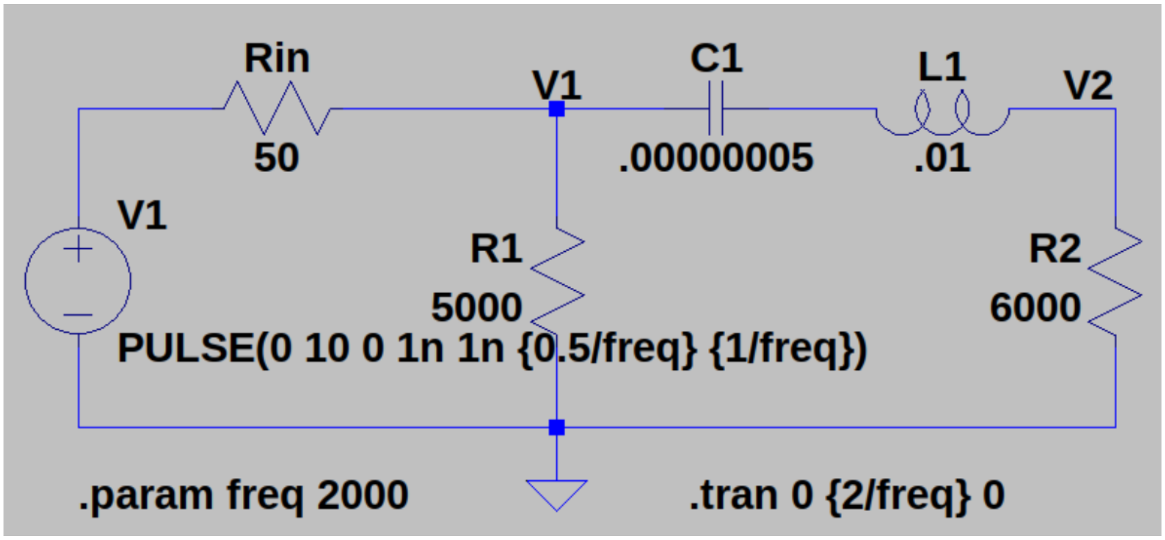

Figure 10: Case 2.2

Part 2 Analysis

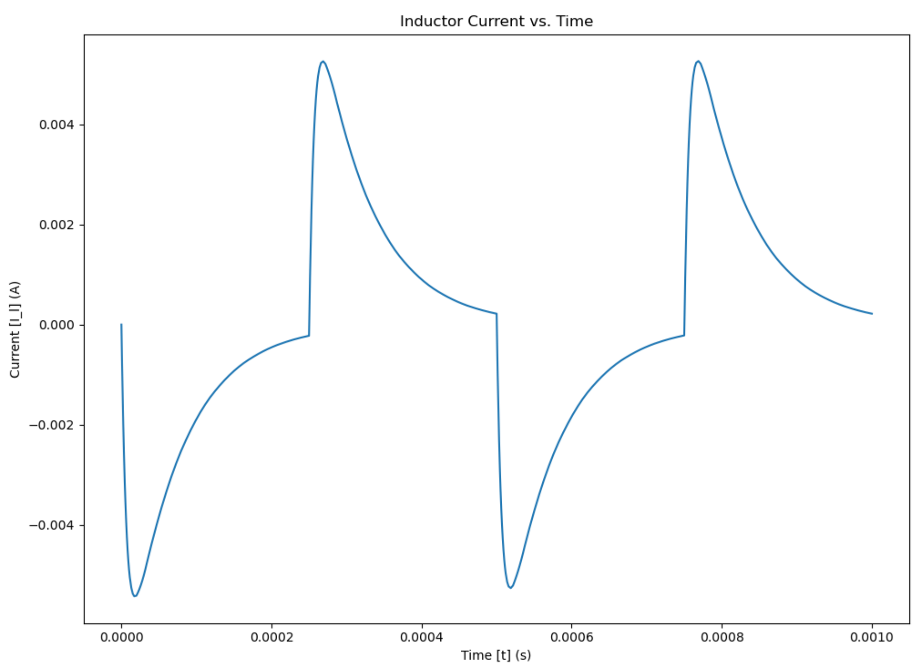

Figure 11: $R_2 = 1500\ \Omega$

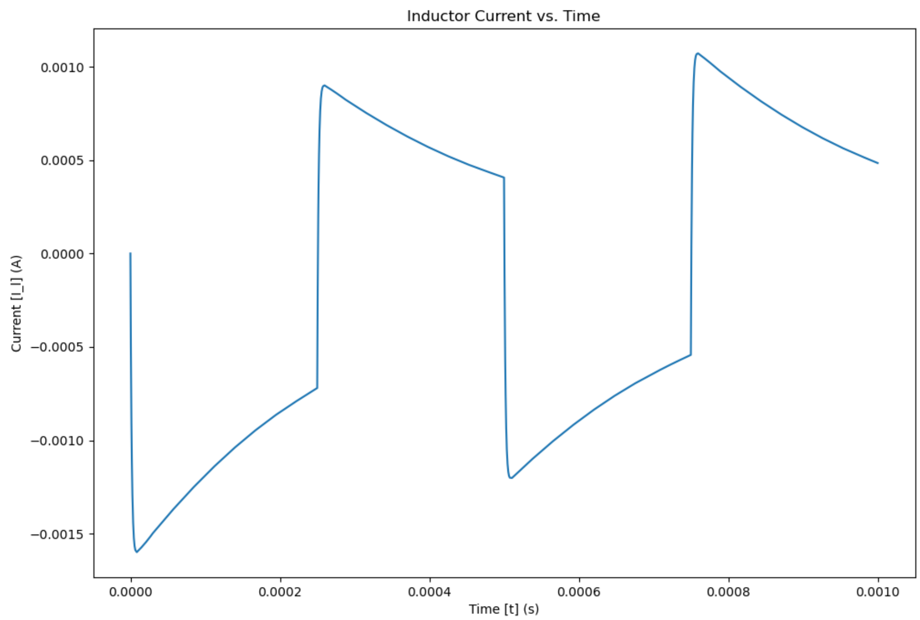

Figure 12: $R_2 = 6000\ \Omega$

Upon examining the inductor current response in the overdamped cases, it is evident that as the value of R2 increases, the time it takes for the current to stabilize towards zero amps also increases. In case 1, where $R_2 = 1500\Omega$, the current settles more quickly than in case 2, where $R_2 = 6000\Omega$.
The general form of a 2nd order overdamped response is:

$$i(t) = A_1 e^{s_1 t} + A_2 e^{s_2 t}$$

From the data, it is possible to extract the coefficients s_1, s_2, A_1, and A_2. This will be accomplished using non-linear least squares optimization to fit the data to the overdamped response function above.

Theoretic Analysis

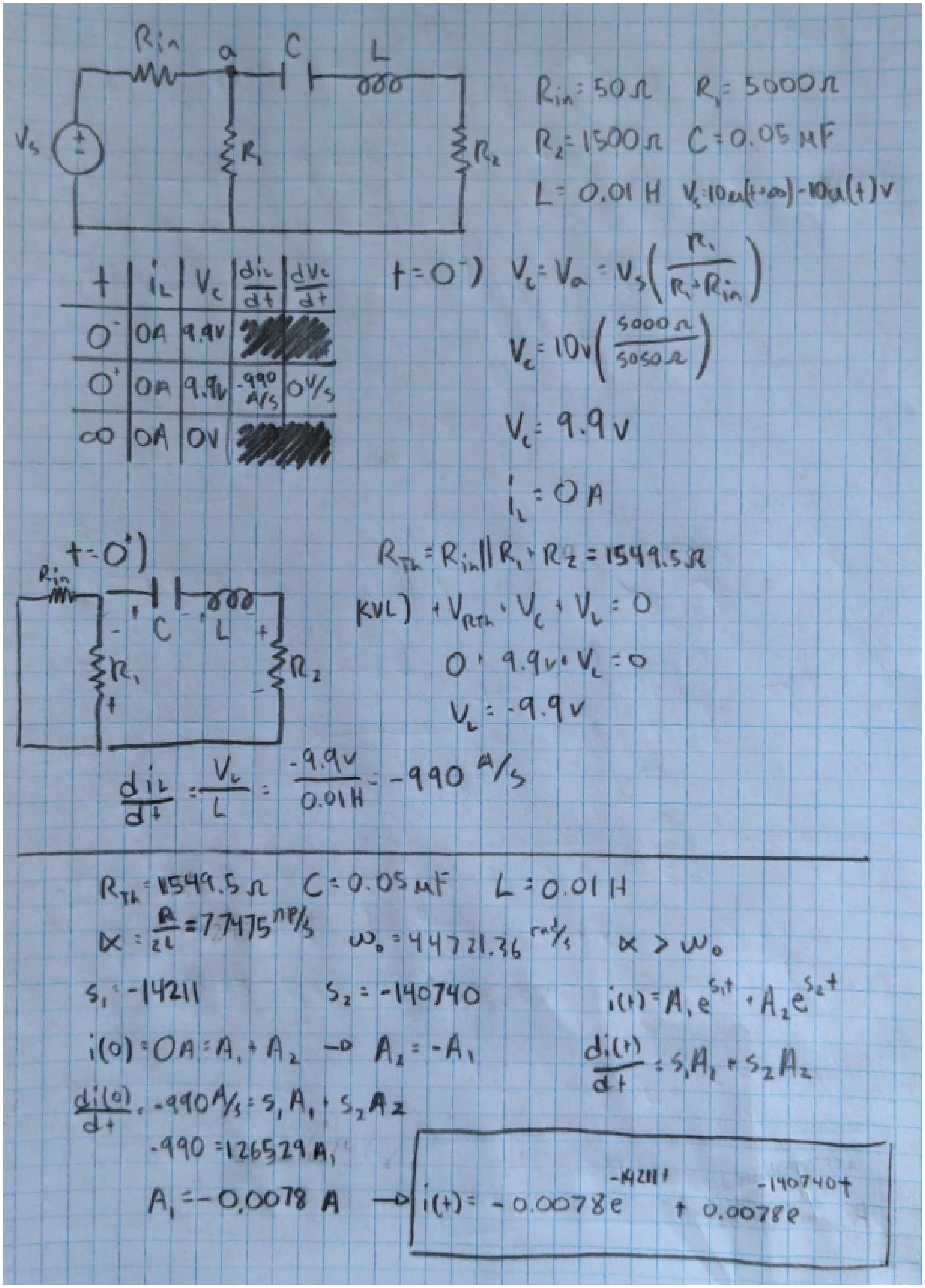

Figure 13: Theoretical Analysis: Overdamped

From the theoretical analysis (Fig. 13) it was found that the coefficients for the circuit are $s_1 = -14211$, $s_2 = -140740$, $A_1 = -0.0078\ A$, and $A_2 = 0.0078\ A$. The current response function when $R_2 = 1500\ \Omega$ can be expressed as:

$$s_1 = -14211 \qquad s_2 = -140740 \qquad A_1 = -0.0078\ A \qquad A_2 = 0.0078\ A$$

$$i(t) = -0.0078e^{-14211t} + 0.0078e^{-140740t}\ A$$

Data Analysis

Simulation data has been taken and processed using the following python code. From this we were able to find values for the current response function coefficients.

$$s_1 = -14225.6414 \qquad s_2 = -139878.5373 \qquad A_1 = -0.007837\ A \qquad A_2 = 0.007837\ A$$

$$i(t) = -0.007837e^{-142225.6414t} + 0.007837e^{-139878.5373t}\ A$$

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

# Define the model function
def rlc_func(x, A, B, s_1, s_2):
    return A * np.exp(s_1*x) + B * np.exp(s_2*x)

def read_data(filename):
    x = []
    y = []

    with open(filename, 'r') as file:
        for line in file:
            try:
                parts = line.strip().split('\t')
                x_val = float(parts[0])
                y_val = float(parts[1])
                x.append(x_val)
                y.append(y_val)
            except:
                continue

    return x, y

# Read the data from the file
filename = 'ENGR203_lab2_p2_1.txt'
x, y = read_data(filename)

# Pull out first response curve
x, y = np.array(x[:81]), np.array(y[:81])
```

```
# Fit the data to the model
popt, pcov = curve_fit(rlc_func, x, y)

fit_out = np.array([rlc_func(x,popt[0], popt[1], popt[2], popt[3])])
fit_out = fit_out.reshape(81,)

# Print the estimated parameters
print("A_1:", popt[0])
print("A_2:", popt[1])
print("s_1:", popt[2])
print("s_2:", popt[3])
```

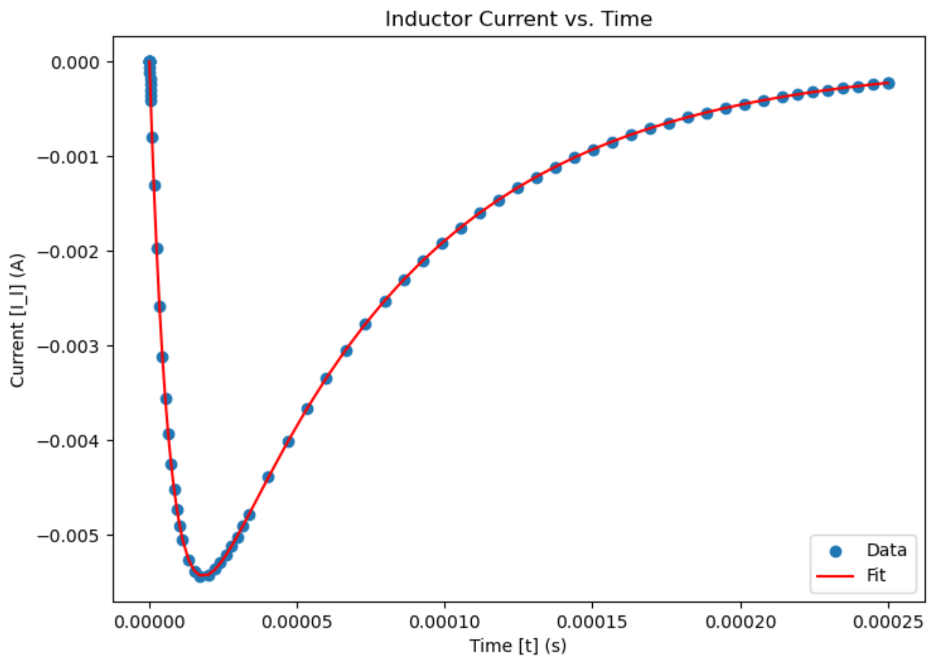

Figure 14: Current Response — Data and Fit Curve

Conclusion

In this lab, we have explored the effect of resistance on RLC circuits by simulating underdamped and overdamped scenarios in LTspice. By analyzing the data from these simulations using Python, we were able to find the α, ω_d, B_1, and B_2 coefficients for the underdamped cases, and the s_1, s_2, A_1, and A_2 coefficients for the overdamped cases. The percent error from data analysis is as follows:

$$\text{Underdamped: } \alpha = 0.6\% \text{ , } \omega_d = 0.03\% \text{ , } B_2 = 0.9\%$$

$$\text{Overdamped: } s_1 = 0.1\% \text{ , } s_2 = 0.6\% \text{ , } A_1 = 0.5\% \text{ , } A_2 = 0.5\%$$

The comparison of the simulation results with the theoretical analysis provided valuable insights into the circuit's response and the impact of resistance on the circuit's performance.