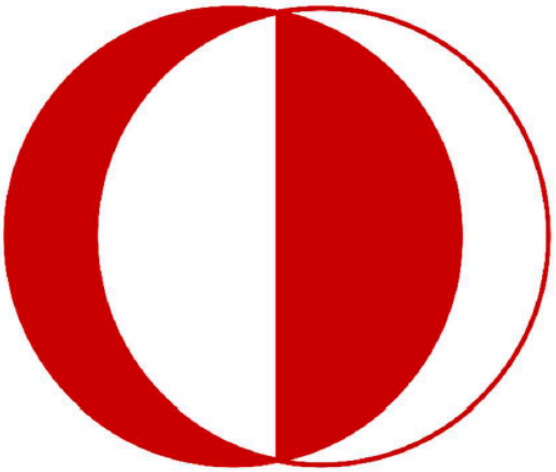

MIDDLE EAST TECHNICAL UNIVERSITY

ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT

EE443 NUMERICAL METHODS AND INTRODUCTION TO OPTIMIZATION

TERM PROJECT FINAL REPORT

Fall, 2023

Rabia Nur Ünsal 2376051

Erdem Canaz 2374676

Introduction

In the domain of electronic circuit analysis, the limitations of conventional linear models become evident when dealing with components exhibiting nonlinear behavior. This project aims to address the challenges posed by nonlinearities in circuit components such as resistors, inductors, capacitors, and voltage sources whose characteristics vary with time. The primary focus is on developing a numerical simulation script that provides a versatile tool for accurate analysis and comprehension of circuits containing such nonlinear elements.

Component models, linearization and updating

1. Voltage Sources

Three types are defined: *dc*, *sine*, and *pulse*. *dc* produces a constant voltage, *sine* generates sinusoidal waves, and *pulse* produces square waves with different duty cycles. Component is considered entirely ideal, its value depending solely on the simulation's specific time.

Type	I(t) in Volts
dc	A
sine	$A + Bsin(2\pi f + \theta)$
pulse	$if\ (t\ mod\ (T)\ <\ DT) \rightarrow A$ $else \rightarrow 0$

2. Current Sources

Three types are defined: *dc*, *sine*, and *pulse*. *dc* produces a constant current, *sine* generates sinusoidal waves, and *pulse* produces square waves with different duty cycles. These are considered entirely ideal, depending on the current value at the simulation's specific time.

Type	I(t) in Amperes
dc	A
sine	$A + Bsin(2\pi f + \theta)$
pulse	$if\ (t\ mod\ (T)\ <\ DT) \rightarrow A$ $else \rightarrow 0$

3. Resistors

Resistors are modeled to have resistance dependent on their temperatures. The resistance is recalculated at the beginning of each iteration considering the temperature and the nonlinear model is considered as a simple linear resistor during that step. Two factors affecting temperature were modeled: the first being the electrical energy lost per second, and the second being the cooling based on the difference with ambient temperature. At the end of each step, considering step-duration and heat capacity, the temperature is updated. The

updating equation is given below. Where $T,\ c,\ R,\ I,\ \lambda\ and\ \Delta t$ denotes temperature, heat capacity, resistance, current, heat transfer coefficient and step-duration respectively.

$$T_{n+1} = T_n + \frac{1}{c}\left(R_n I_n^{\ 2} - (T_n - T_{ambient})\lambda_R\right) \cdot \Delta t$$

4. Inductors

Inductors are modeled to have inductance dependent on their currents. The inductance is recalculated at the beginning of each iteration considering the previously calculated current value and the nonlinear model is considered as a simple current source. At the end of each step, considering step-duration, inductance and voltage drop on the inductor, the current is updated. The updating equation is given below. Where $I,\ L,\ V_P,\ V_N,\ and\ \Delta t$ denotes current, inductance, positive & negative node voltage and step-duration respectively.

$$I_{n+1} = I_n + \frac{1}{L_n}(V_{P,n} - V_{N,n})\Delta t$$

5. Capacitors

Capacitors are modeled to have capacitance dependent on their voltages. The capacitance is recalculated at the beginning of each iteration considering the previously calculated voltage value and the nonlinear model is considered as a simple voltage source. At the end of each step, considering step-duration, capacitance and current of the branch, the voltage is updated. The updating equation is given below. Where $V,\ I,\ C,\ and\ \Delta t$ denotes voltage, current, capacitance and step-duration respectively.

$$V_{n+1} = V_n + \frac{I_c \Delta t}{C_n}$$

Development Overview

This section provides an in-depth explanation of the script's key components.

1. Define Simulation Parameters

The simulation script defines essential simulation parameters for characterizing an electrical circuit. These include ambient temperature, time step constraints, and maximum allowable changes in component states etc..

2. Circuit Elements Definition

The initial phase of the algorithm involves the definition of circuit components, such as voltage sources, current sources, resistors, inductors, and capacitors. These components are essential for constructing the circuit and capturing the dynamic behavior of non-linear elements. The script allows flexibility in specifying various parameters for each components as functions of other parameters. An example code snippet of such definition is given in Fig. 1 for nonlinear inductor model.

```
# DEFINE INDUCTORS

inductor_10mH_v1 = lambda inductor_current: 1e-2

inductor_10mH_saturation_v1 = lambda inductor_current: 0.001 + 0.02*(math.pow(math.e, -(abs(inductor_current/2))))

L1 = component_classes.inductorModel(name = "L1", node_p = 3, node_n = 1, inductance_function = inductor_10mH_saturation_v1 , initial_current = 0)

components.append(L1)
```

Figure 1: Definition of a inductor with saturation characteristic

3. Determining the Unknowns

The numerical simulation is grounded in formulating node equations based on Kirchhoff's current law (KCL). All of the node voltages are considered as unknowns. The voltage source and capacitor currents are also considered as unknowns who yield a voltage constraint between two nodes thus rank is not lost.

4. Updating Time-step

The time step adapts dynamically, ensuring a balance between computational efficiency and accuracy. Time step in this simulation is updated dynamically based on the behavior of the circuit components:

- If temperature change in resistors, voltage change in capacitors are small enough the time step should increase.
- If at least one of the components experienced a significant change, and the time step should be decreased to capture the dynamics more accurately.

The code snippet that this logic is executed is given in Fig. 2..

```
#If the state changes are small enough increase time-step, otherwise decrease it
if should_increase_time_step:
    time_step_now = min(max_time_step, time_step_now*time_step_multiplier)
else:
    time_step_now = max(min_time_step, time_step_now/time_step_multiplier)

time_now = time_now + time_step_now
```

Figure 2 Time-step adjusting logic at the end of each step

5. Component State Updates

As the simulation progresses, the script continuously updates the states of non-linear components based on their respective characteristics. The changes in resistor temperatures, capacitor voltages, and inductor currents are tracked, considering maximum allowable changes to maintain simulation stability.

```python
if component.get_component_category() == "RESISTOR":
    #TODO: check if the temperature change is too high, if so decrease the time step
    positive_node_voltage = approximated_solution[unknowns[f"V_{component_p}"]][0]
    negative_node_voltage = approximated_solution[unknowns[f"V_{component_n}"]][0]
    voltage_difference = positive_node_voltage - negative_node_voltage
    b, del_T = component.update_resistor_temperature(voltage_difference, ambient_temperature, time_step_now, MAX_RESISTOR_TEMPERATURE_CHANGE)
    should_increase_time_step = should_increase_time_step and b
    #print(f"Resistor temperature: {component.get_temperature()}, temperature change: {del_T}")
```

Figure 3 Nonlinear Resistor Temperature Updates

6. Sample Recordings and Result Plotting

The script records samples at specified time intervals, allowing users to analyze the transient response of the circuit. The recorded samples include node voltages, inductor currents, capacitor voltages, and resistor temperatures. The final step involves plotting these results using matplotlib, providing a clear visualization of the circuit's behavior over time.

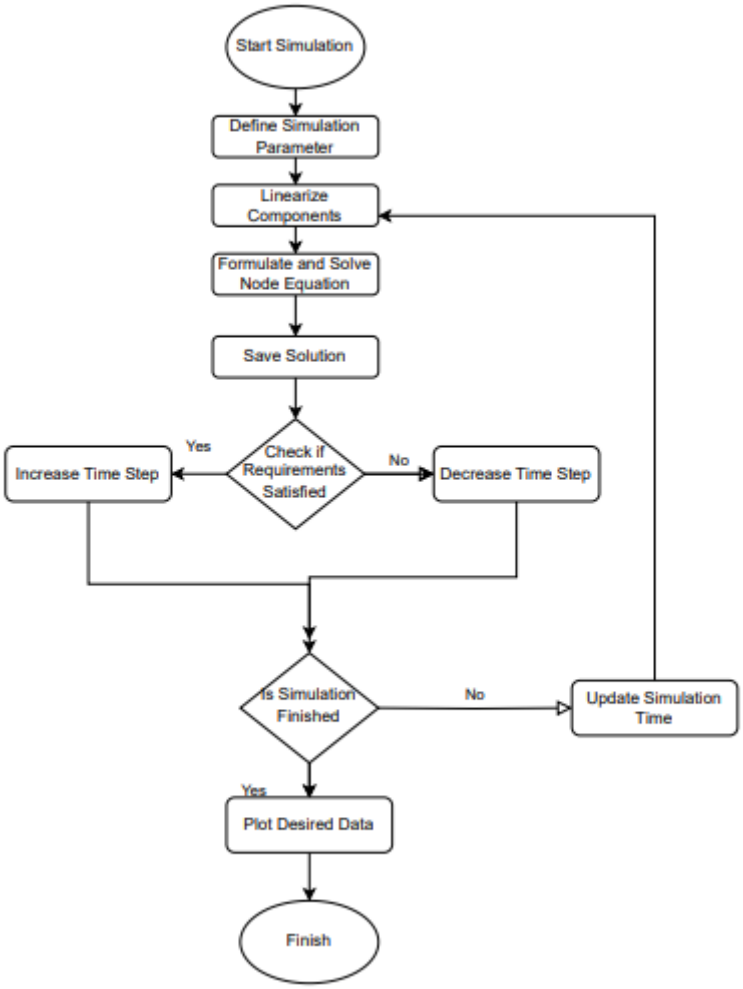

Figure 4 Flowchart of the Algorithm

Comparison

To validate the accuracy and reliability of our numerical simulation script, we conducted a comparative analysis with LTspice, a widely used electronic circuit simulator. LTspice provides a benchmark for circuit analysis, and we aim to evaluate how well our script captures the real-world behavior of nonlinear components. The comparison involves implementing the same circuit configurations in both our simulation script and LTspice, followed by a side-by-side analysis of the results.

1. The Effect of Simulation Accuracy on the Execution Time

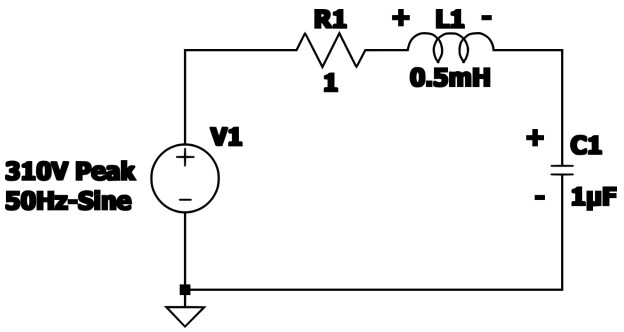

Figure 5 The Modeled RLC Circuits

In this part, the circuit given in Figure 5 is analyzed. When the circuit is simulated in LT-spice, the capacitor voltage and inductor current are found as in Figure 6.

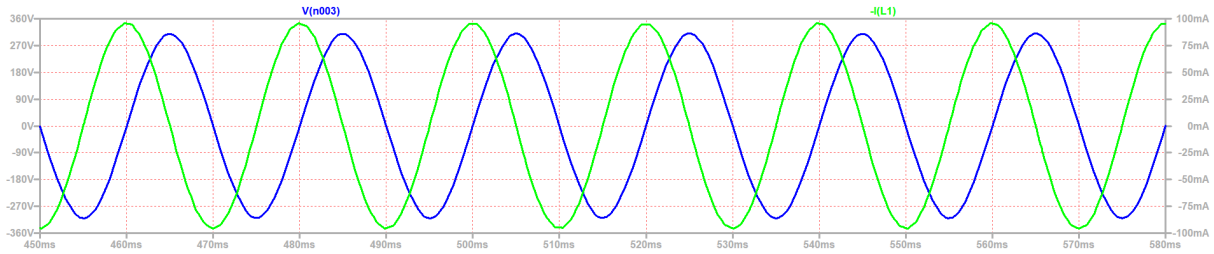

Figure 6 The Capacitor Voltage and Inductor Currents Simulation in LT-spice

It can be seen in Fig. 6 that the peak inductor current and capacitor voltage are 85mA and 310V respectively. The accuracy requirements can be normalized according to these values.

Normalization	Maximum allowed voltage change of the capacitor per step (K)	Maximum allowed current change of the inductor per step (mA)	Maximum allowed voltage change of the capacitor per step (V)	Total simulation time (s)
1.000	1000	85	310	Less than 1 (LT-spice)
0.200	1000	17	62	157.97 (Our approach)
0.010	1000	0.85	3.1	180.54 (Our approach)
0.001	1000	0.085	0.31	271.34 (Our approach)

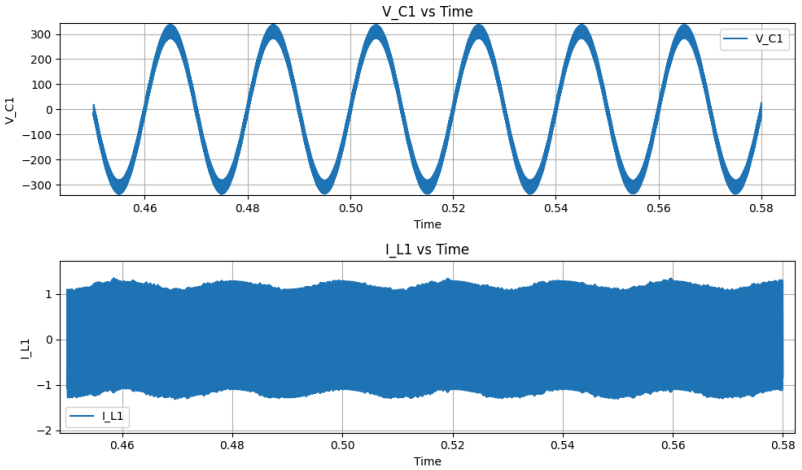

Figure 7 Result of the simulation for normalization = 0.200 (worst accuracy, fastest simulation)

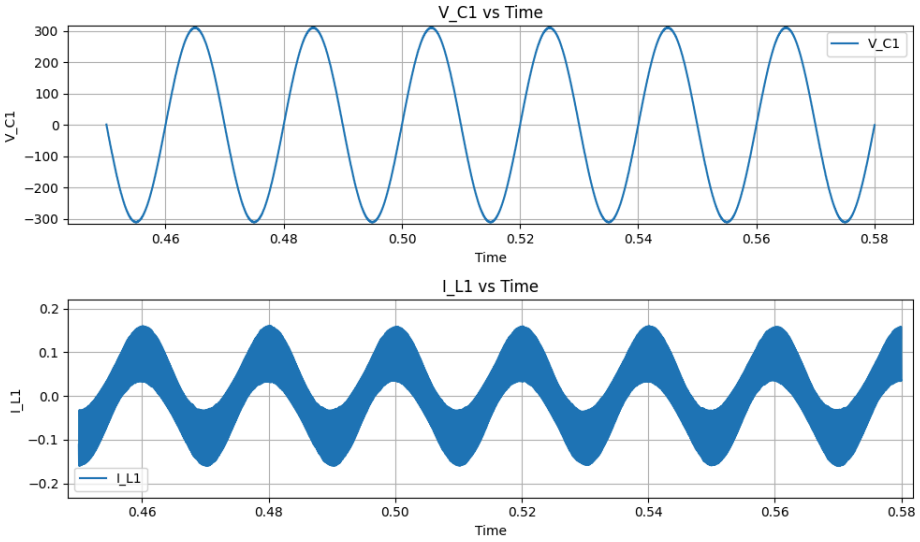

Figure 8 Result of the simulation for normalization = 0.010 (moderate accuracy and simulation time)

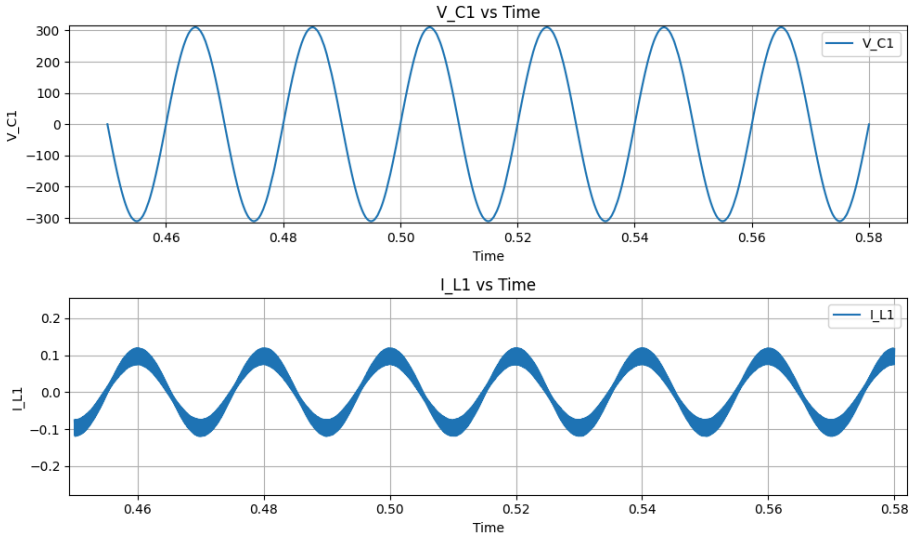

Figure 9 Result of the simulation for normalization = 0.001 (best accuracy, slowest simulation)

The first one, with the lowest accuracy (normalization = 0.200) shown in Figure 6, was the fastest to run but its results were too different from the LT-spice results to be useful. The most accurate simulation (normalization = 0.001), also shown in Figure X, had good results with a bit of noise, but it took a very long time to run. The best overall option was a middle-ground simulation (normalization = 0.010), presented in Figure X, which had results close to LT-spice and didn't take too long. Also note that using a higher inductance value could reduce the high frequency error, that also explains why capacitor waveform seems more accurate. Another solution to reduce error is to increase accuracy which would also increase the simulation time. One should also notice that LT-spice was faster almost 100 times. It is assumed that LT-spice solves this equation analytically by utilizing the fact that the circuit is linear time invariant.

2. Effect of Resistor Nonlinearity

In order to see the effect of the factors like temperature on the components, the same circuits have been simulated in our simulation and in LTSpice. The circuit (b) in Figure 9 was simulated without considering the temperature effect on the components in both LTSpice and our simulation code. As it can be seen in Figure 10 and Figure 11 our simulation gave the same results with the LTSpice. Subsequently, the circuit was simulated by considering the temperature effect. In Figure 12 temperature effect on the circuit can be seen.

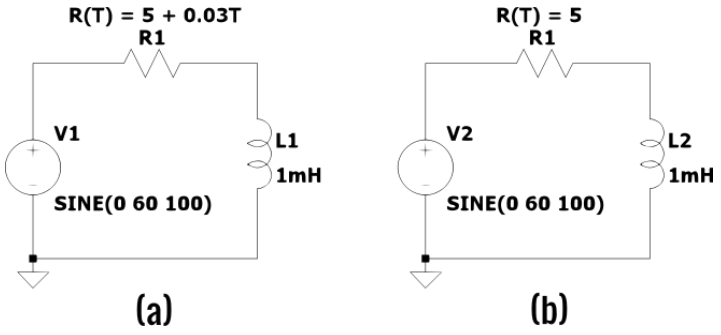

Figure 10 The Modeled RL Circuits Where Linear Inductor and (a) The Temperature Dependent (b) Temperature Independent Resistors are Considered.

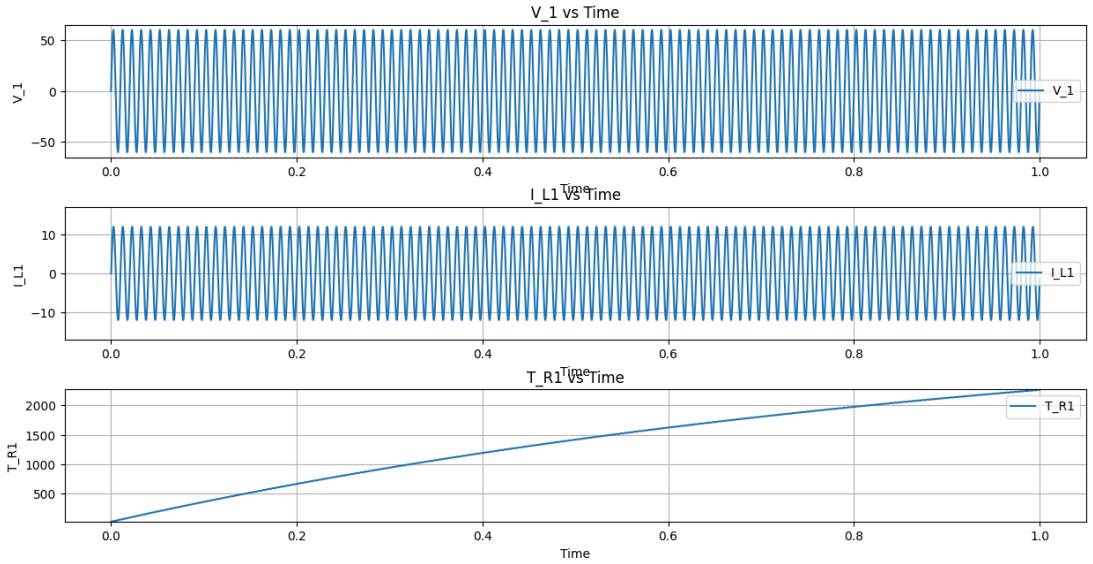

Figure 11 The Simulation Results of RL Circuit with Temperature Independent Resistor.

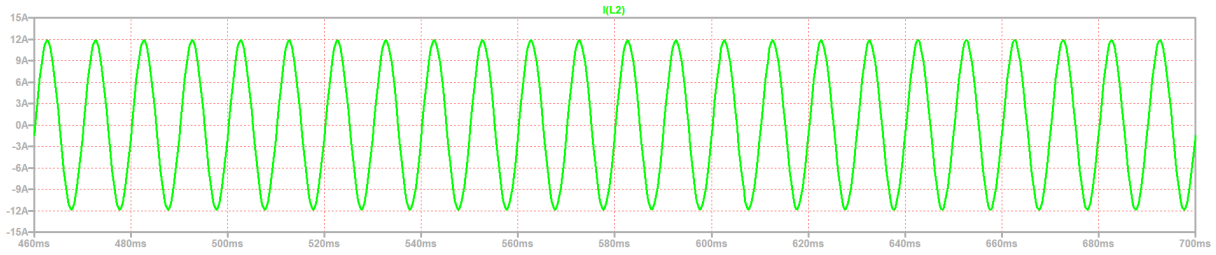

Figure 12 The simulation results of RL circuit performed in LT-spice with temperature independent resistor.

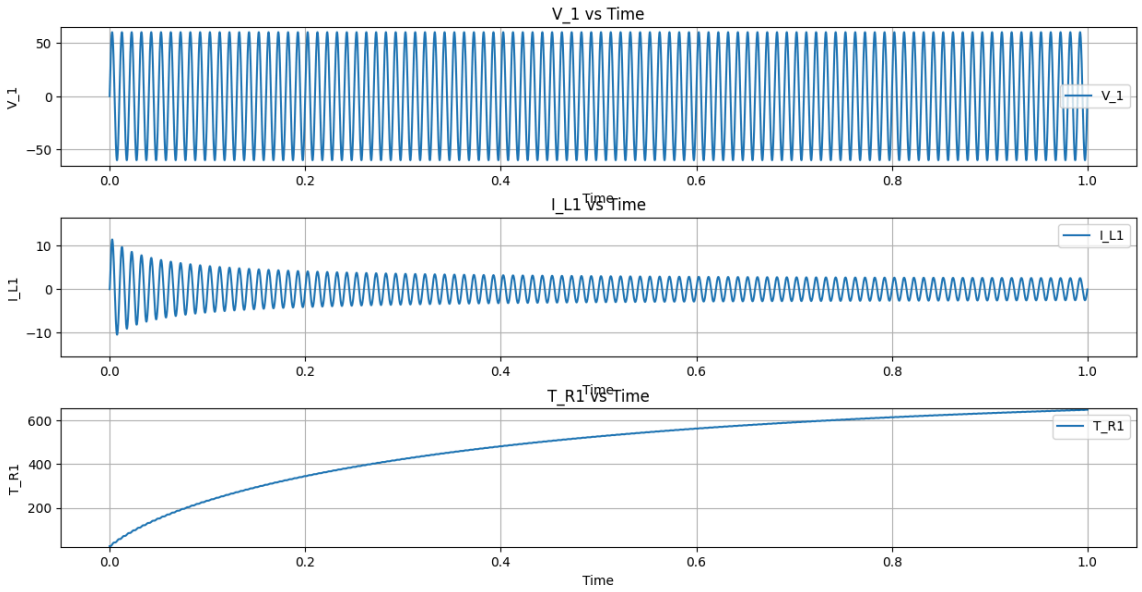

Figure 13 The Simulation Results of RL Circuit with Temperature Dependent Resistor.

The reason for the temperature is very high in Figure 13 is that its resistance is independent of the temperature. Thus, there is no negative feedback is observed to decrease overal current flowing.

3. Inductor Current Overshoot

The nonlinearity in inductance is often attributed to the saturation effect in the magnetic core of the inductor. When the magnetic core reaches saturation, it can no longer absorb additional magnetic flux, disrupting the linear relationship between current and magnetic flux. This nonlinearity causes the inductance to vary with frequency and current intensity. As a result, the AC current waveform undergoes distortion, exhibiting different forms. The circuit in Figure 14 is simulated and as can be seen in Figure 15, the current has a different form as a result of this nonlinearity.

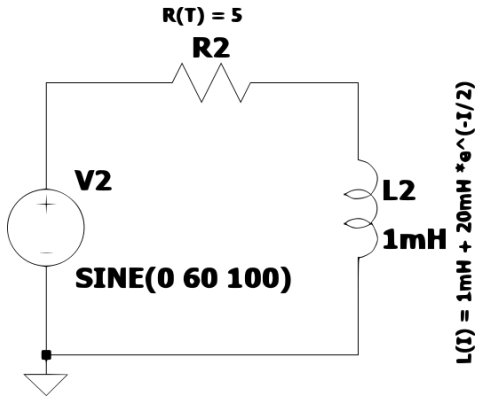

Figure 14 The Modeled RL Circuit where Linear Resistor and The Inductor with Current Dependent Inductance is Modeled.

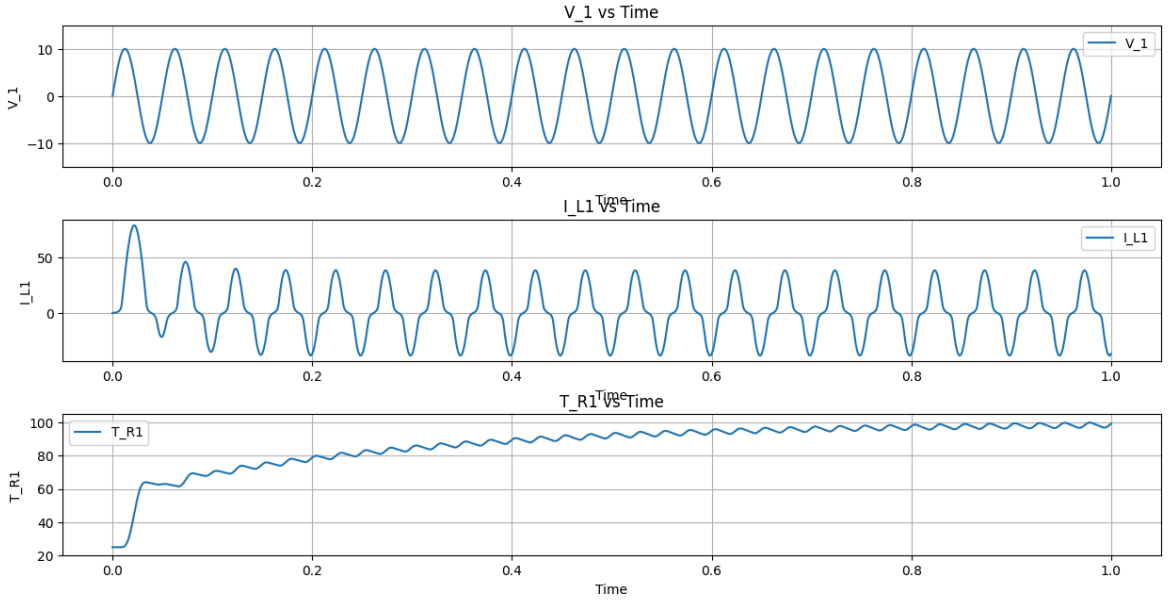

Figure 15 Simulation Result of RL Circuit

Source Code

The source code of the project is shared on GitHub under the GPL-3.0 license.

Conclusion

In conclusion, the developed numerical simulation script provides a robust tool for analyzing electronic circuits featuring nonlinear components. The script's consideration of dynamic parameter changes over time enhances the accuracy of real-world circuit representation. Rigorous checks implemented within the script ensure the integrity of the circuit model. The time-domain plots presented in the results offer a visual grasp of the circuit's transient response. Notably, as we modeled the simulation, we observed an increase in simulation time with improved accuracy. Introducing non-idealities revealed diverse responses in different aspects of the circuit. This highlights the script's capability to capture nuanced behaviors, contributing to an advanced understanding of electronic circuit analysis.