

Lab 6: Importing LTspice Data to Python and Python Plotting

Background:

Python is an open source (free), popular, and very powerful programming languages that can be run on all of the major operating systems and platforms. Its uses are numerous: application development, automation testing, database accessibility, machine learning, data analytics, pattern recognition, web development, scientific research, engineering analysis, digital signal processing, image processing, data visualization, robotics, web scraping, scripting, hacking, artificial intelligence, face detection, and many more.

Python is one of the world's most popular programming languages. It is very well supported, as there are countless resources online for learning and troubleshooting Python code. There are python packages (basically add-ons) for just about anything one can think of. In this lab, we will be using a Python package someone developed in late 2018 called **ltspice**. It makes importing LTspice data into Python very easy.

Whenever you run an LTspice simulation, it outputs all of the simulated voltages and currents to a **.raw** data file. The **ltspice** package reads this file and allows you to put any of those variables into a Python data type called an ndarray, or Numpy array (Numpy is a popular numerical analysis package for Python). Ndarrays can hold multiple numerical values and they make performing calculations with those numbers easy. After LTspice data is imported, you will be doing calculations with it and then plot your results using Python. Python plots can be saved as image files.

Objectives:

- Learn how to make LTspice do multiple simulations over a variety of values of a circuit variable.
- Learn to import data from LTspice to Python
- Perform calculations with Python ndarrays
- Plot data with Python

Required Equipment:

- LTspice XVII
- MS Excel or equivalent spreadsheet program
- A text editor, such as Notepad++

1 Prelab:

1. Go through the tutorial on how to create data plots in Python in the `Lab_6-Plotting_with_Python.ipynb` file.

2 Part A Procedure:

1. Create a new LTspice that simulates the two circuits from our last lab, as shown below.
 - (a) Use the values shown and the values of V_{th} and R_{th} from the previous lab. Use as much precision as possible.
 - (b) For now, do not specify any values for R_L .
 - (c) In the Thevenin equivalent circuit, label the node between R_{th} and its RL as C (not A or B, as those labels are already used).

3 Part A Procedure:

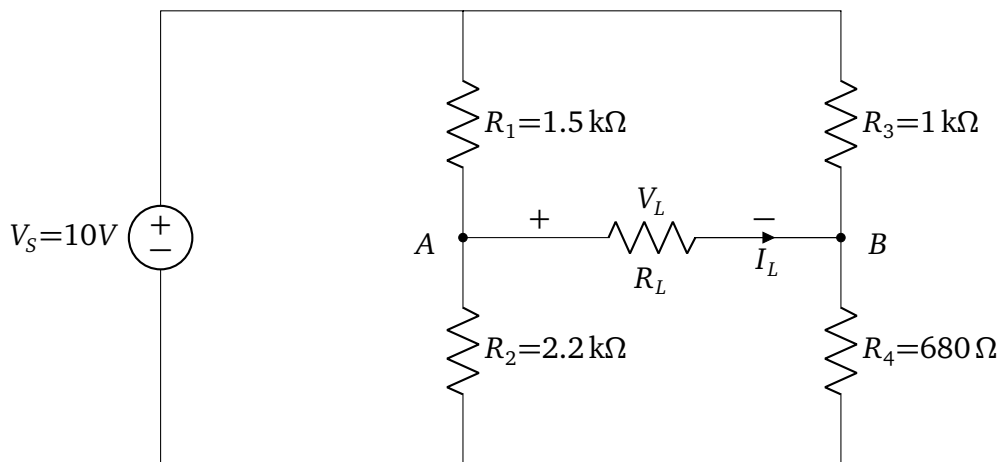


Figure 1: Bridge Circuit

2. Watch this tutorial on how to do use stepping parameters in LTspice: <https://www.youtube.com/watch?v=hH74uZvEm6I>
3. In both LTspice circuits, set R_L as your parameter you will want to vary and have it vary from $1\ \Omega$ to $5000\ \Omega$ in ten-ohm steps. This means LTspice will do 500 simulations as R_L increases from 1 to 5000 ohms.
4. Run your simulation using the DC operating point type of simulation. You should notice a **.raw** file appear in the same folder as your LTspice **.asc** file.

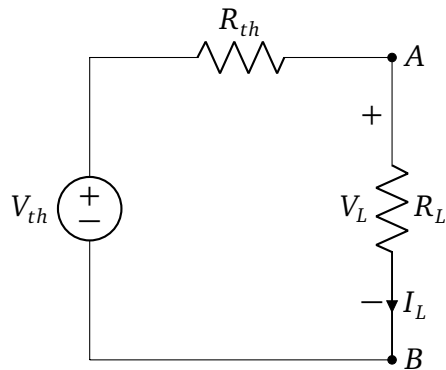


Figure 2: Thevenin Equivalent of Bridge Circuit From the Perspective of Nodes A and B

5. After sim is run, hover mouse over both of your RLs. In the original circuit, make sure current arrow flows from A to B. In the Thevenin equivalent circuit, make sure the it flows from the top of RL down to ground. If this is not the case, pick up the resistor with the Move button, hit ctrl+R twice to rotate it 180 degrees and put it back in the circuit. Re run your simulation.
6. After the simulation is run, LTspice will have written all currents and voltages from all 500 simulations to this file. You should also notice a plot window open up. This does not usually happen for DC simulations , as it usually returns on-screen text as output. Now you get a plot window because a parameter is changing.
7. Open the **.raw** file in a text editor. At the top of the file, you should see a list of the voltages and currents for which the file has data. You should note that I(RL) does not show up. Since there were to objects called RL, LTspice internally renamed each resistor to something like R5 and R5b. If you R5 will be the first of the two RLs you created, and R5b will be the second.
8. Make a copy of the **.raw** file and call it **ltdata.raw** and upload it to your **repl.it** project by dragging it to the Files area.
9. Begin a new Python project by going to <https://repl.it/@dankruger/ENGR221-Lab06-LTspice>. The **ltspice** package has already been in stalled and the file has example code demonstrating how to import a voltage and a current. Notice that the values of RL are obtained using the command `l.getTime()`. This is because most LTspice simulations with large data sets are transient analysis simulations where time is the x-axis variable.
10. Import whatever data variables you need to calculate power dissipated by both the RL in the bridge circuit and the RL in the Thevenin equivalent circuit.
11. Calculate power dissipated by R_L for each vale of R_L for the bridge circuit (store in a variable called **P1**) and for the Thevenin equivalent circuit (store in a variable called **P2**). With the correct LTspice data imported, this should take only 1 line of code each because arithmetic with ndarrays of the same size is very straight forward. Convert **P1** and **P2** into micro watts.
12. Create a plot of the **P1** (y axis) versus **RL**.

- (a) Add a title, x ticks, y ticks, x-axis grid lines, and y-axis grid lines.
- (b) Save the file as **plot1.png** with this line of code:

```
plt.savefig('plot1.png', dpi=300, bbox_inches='tight')
```
- (c) Clear out the **plt** object using the following command after your plot is saved:

```
plt.clf()
```

- 13. Repeat step 10 for **P2** versus **RL** and save this plot as **plot2.png**.
- 14. Repeat step 10 for the difference between **P2** and **P2** versus **RL** and save this plot as **plot2.png**.

4 Deliverables:

Submit the following files to this lab's D2L folder.

- 1. **ltdata.raw**
- 2. **lab6.py** which should contain all of your working Python code.