MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.01—Introduction to EECS I
Fall Semester, 2007

Assignment 8, Issued: Tuesday, Oct. 23rd

To do this week

...in Tuesday software lab

1. Start writing code and test cases for the numbered questions in the software lab. Paste all your code, including your test cases, into the box provided in the "Software Lab" (Part 8.1) problem on the on-line Tutor. This will not be graded.

...before the start of lab on Thursday

1. Read the lecture notes.
2. Do the on-line Tutor problems for week 8 that are due on Thursday (Part 8.2).
3. Read through the entire description of Thursday's lab.

...in Thursday robot lab

1. Answer the numbered questions in the robot lab and demonstrate them to your LA.
2. Do the nanoquiz; it will be based on the material in the lecture notes and the on-line Tutor problems due on Thursday.

...before the start of lecture next Tuesday

1. Do the lab writeup, providing written answers (including code and test cases) for **every** numbered question in this handout.

On Athena machines make sure you do:
`athrun 6.01 update`
so that you can get the `Desktop/6.01/lab8` directory which has the files mentioned in this handout.

- You need the files **resolveConstraints.py, circuitConstraints.py, genKCL.py (your code) or genKCL.pyc (ours), circuitt.py, circuitLine.py, genGrid.py, and thntest.py** for the software lab, and may find them helpful for the circuit design lab.

During software lab, if you are using your own laptop, download the files from the course Web site (on the Calendar page). Be sure you have `numpy` installed.

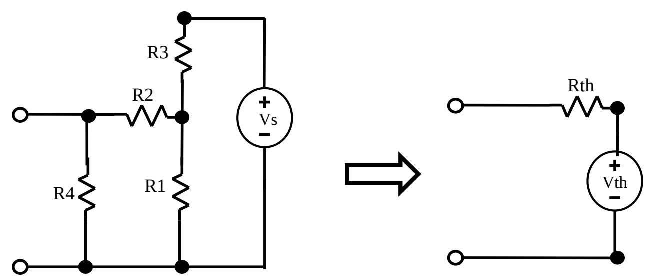


Figure 1: Simple Circuit and its Thevenin Equivalent

Thevenin Equivalents

Last week you learned to use `resolveConstraints` to analyze circuits. You also wrote a class derived from `ConstraintSet` to automatically generate KCL constraints for circuits with two-terminal elements, though your program had to assume that the constitutive equations for the elements were written in a very particular form.

For this software lab, you will be writing a program that computes the Thevenin equivalent resistance and Thevenin equivalent voltage source given an arbitrary circuit of resistors and voltage sources, and a specification of the terminals of your Thevenin equivalent.

Your method should take three arguments. The first argument should be an instance of `constraintSetKCL`, one which includes only the constitutive relation constraints, but for which the KCL equations have not been set, nor has a ground node been defined. The second two arguments are the two node voltage names that will be the terminals of your Thevenin equivalent. Your program should return the Thevenin equivalent resistance, R_{th} and the Thevenin equivalent voltage source, V_{th}. You may find that you will need of copy of the `constraintSetKCL` instance, if so, you should import the module `copy` and use `copy.deepcopy` to make sure you actually generate a completely seperate copy of the instance. Also, `constraintSet` has a method `translate` which you will find useful. The file thntest.py is an example of how to compute Thevenin equivalents.

If you do not have a copy of your genKCL.py module, you can download a compiled version from the course web page.

In order to test your method for computing Thevenin equivalents, demonstrate that you can compute the correct Thevenin equivalents for the circuits in figure 1 and figure 2. In each of the figures, the pair of open circles are used to indicate the terminals for the Thevenin equivalent. Note that for these two examples, the circuits only differ in that different circuit nodes are the terminals for the Thevenin equivalent. Therefore, the two Thevenin equivalents are modeling how the circuit will respond to changes across two different pairs of terminals. For your test examples, you can assume $V_s = 5V$, and $R_1 = R_2 = R_3 = R_4 = 1K\Omega$, where $1K\Omega$ means 1000 ohms.

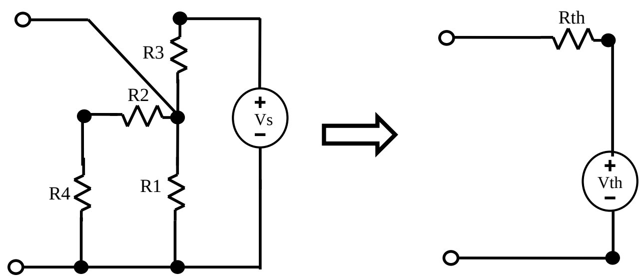


Figure 2: Simple Circuit and its Thevenin Equivalent for different terminals

Question 1. Describe how you implemented your program for computing Thevenin Equivalents.

Question 2. Demonstrate that your program works correctly for the circuits in figure 1 and 2.

Question 3. Explain why the Thevenin equivalent voltages and resistances are different for the two test cases, even though the original circuits are identical.

As you recall from last week, the programs `genGrid.py` and `circuitLine.py` can be used to generate the circuit in figure 3. Modify the `genGrid.py` and `circuitLine.py` files, and use your program for computing Thevenin equivalents to answer questions about the Thevenin equivalent for a part of the resistor line shown in figure 4.

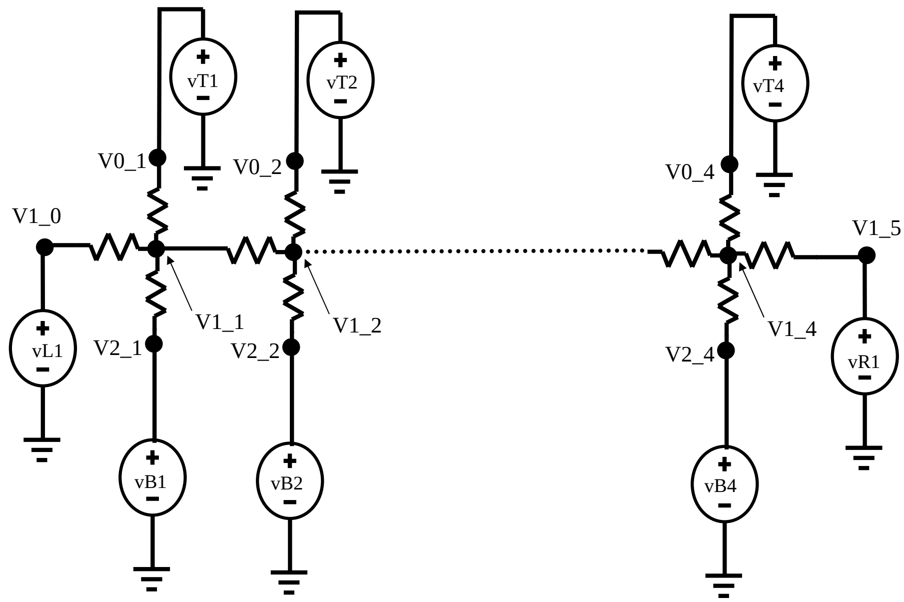


Figure 3: Resistor Line circuit generated by `circuitLine.py`.

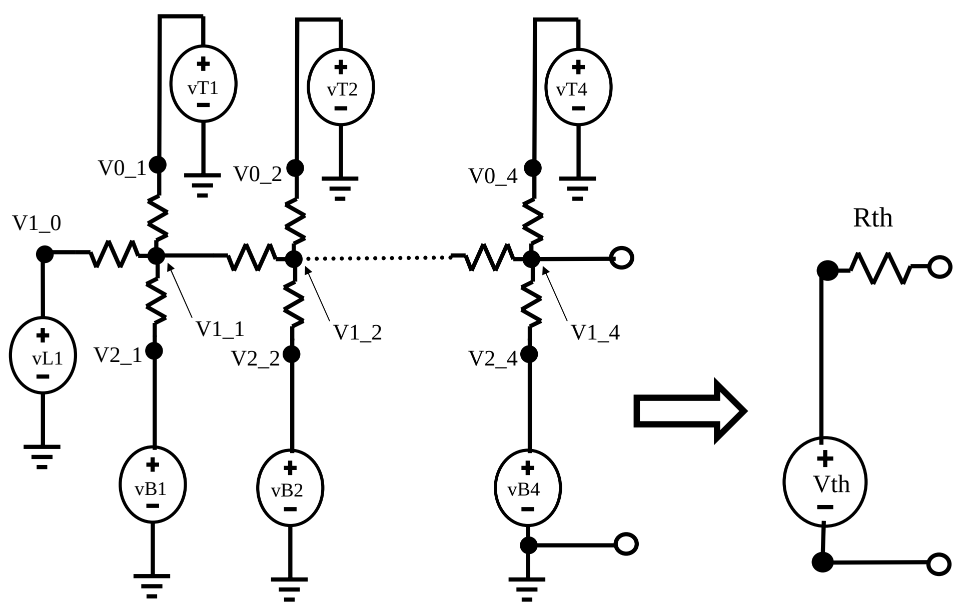


Figure 4: Unterminated resistor line and its Thevenin equivalent.

Question 4. Modify `genGrid.py` and `circuitLine.py` to eliminate the right hand voltage source and associated resistor, so that you can generate the unterminated four-column resistor line in the left-hand circuit of figure 4.

Question 5. Suppose all the resistors in your unterminated line are one ohm, and the top and bottom voltage sources are all zero. How does the Thevenin equivalent resistor and voltage source for the unterminated line vary as you vary the left-hand voltage source? Can you explain why?

Question 6. Modify the grid generator so that you can use different values for the vertical and horizontal resistors, if you did not do so last week.

Question 7. Use your modified version of the grid generator, and modify `circuitLine.py` so that the horizontal resistors in your unterminated line are one ohm, the vertical resistors are all 10,000 ohms, and the top and bottom voltage sources are all zero. Now how does the Thevenin equivalent resistor and voltage source for the unterminated line vary as you vary the left-hand voltage source? Can you explain why?

Question 8. Use your modified version of the grid generator, and modify `circuitLine.py` so that the horizontal resistors in your unterminated line are one ohm, the top and bottom voltage sources are all zero, and the left-hand voltage source is one volt. How does the Thevenin equivalent resistor and voltage source for the unterminated line vary as you vary the horizontal resistors over the range from one ohm to 10,000 ohms? Can you explain why?

Go to the on-line Tutor at `http://sicp.csail.mit.edu/6.01/fall07`, choose PS8, and paste the code that you wrote during lab, including your test cases, into the box provided in the "Software Lab" problem. Do this even if you have not finished everything. Your completed answers to these questions are to be handed in **AT THE BEGINNING OF THURSDAY'S LAB.**

Thursday Lab - Designing and using a virtual oscilloscope

In this lab you will design a resistor network that will allow you to use the virtual oscilloscope you used in lab 4 to measure voltages over a wide range. You will then use your virtual oscilloscope to measure the relationship between voltage and current for a LEGO motor, and use your knowledge of circuits to develop a circuit model for the motor.

For this lab you will need the robot, a laptop, and

- A USB cable, and screwdriver.
- A number of resistors.
- Wire.
- A Lego motor with connector.
- A protoboard with built-in power supply.

Using the NI box to measure voltages

In previous labs, you used the National Instruments digital-to-analog converter (NIDAQ) (the white box on mounted using velcro on top of the robot) to read analog voltages from optical distance sensors, convert the results to digital numbers, and transmit the numbers to your laptop using a USB connection. As you saw in earlier labs, the NIDAQ is managed by running a server program on your laptop. This server program makes the converted analog voltage data available to Python programs, such as the virtual oscilloscope you used briefly in lab 4.

This time you will be generating your own analog signals and using the NIDAQ, NIDAQ server, and virtual oscilloscope software, to examine those signals.

In order to use the NIDAQ, **you must turn on the power to the robot**, then open a terminal window and change to the `lab8` directory. Then make sure the USB cable from the NIDAQ is plugged into the USB port of your laptop (if so, the green light on the NIDAQ will be blinking). In the terminal window, type

```
./NIDAQserver
```

After a few seconds, the terminal screen will start filling with numbers. Minimize this terminal window, but do **NOT** kill the window.

Open a new terminal window and start the virtual oscilloscope program by typing

```
python v_oscillo.py
```

An oscilloscope window should appear on your laptop screen, and the first channel (Ai0) from the NIDAQ (which is connected to the first optical sensor) should be plotted in the oscilloscope window. This is the voltage returned by one of the optical sensors. Check that the NIDAQ is working by blocking the optical sensors and notice what happens in the oscilloscope window.

In this lab, you will be connecting a pair of wires to the Ai5 and ground connections on the NIDAQ, and then monitoring the voltage across these wires by watching the virtual oscilloscope. You can display the voltage between Ai5 and ground on the oscilloscope by right clicking on the oscilloscope display and toggling off Ai0 and toggling on Ai5. If the oscilloscope does not seem to be working, close the oscilloscope window, unplug the NIDAQ USB connection, wait three seconds, replug the USB connection and restart the server and oscilloscope.

You will also be using a protoboard to build some small circuits. A protoboard is used for making easily modifed electrical connections between wires and circuit elements. If you look at your protoboard, you will notice many rows of five holes. These holes are electrically connected, so if you plug two wires in to two holes in the same row, the wires will be connected electrically. Resistor leads can also be plugged directly in to the protoboard holes. Also, each protoboard has several long columns of holes which are used for nodes in a circuit that have a large number of connections. These columns are often used for ground and power. **CAREFUL: the long columns are only electrically connected for HALF THE COLUMN**. If you have never used a protoboard, have one of the staff members demonstrate the board's use.

Try using the NI Box

Connect the NI box to your laptop (it has a USB connector), turn on the robot power, and try running the virtual oscilloscope program to read the voltage at *AI5*. Please note that you will need to connect external wires to the *AI5* and *GND* terminals to connect circuitry to the NI Box. You should also take note of the fact that the NI box has terminals which generate two reference voltages, +5 volts and +2.5 volts. We have special screwdrivers, wire, and wire strippers, just ask if you need them.

Try connecting *AI5* to the 2.5 volt reference on the NI box, note the readings on the virtual oscilloscope. Then try connecting *AI5* to the 5 volt reference on the NI box, note the readings on the virtual oscilloscope. In addition, try connecting *AI5* to the five volt supply on your protoboard, and `GND` to the ground on your protoboard, and note the readings on the virtual oscilloscope.

Question 9. When measuring voltages that are external to the NI box, you needed to make two connections, the NI box input and ground. Why?

Checkpoint: 10:15 PM

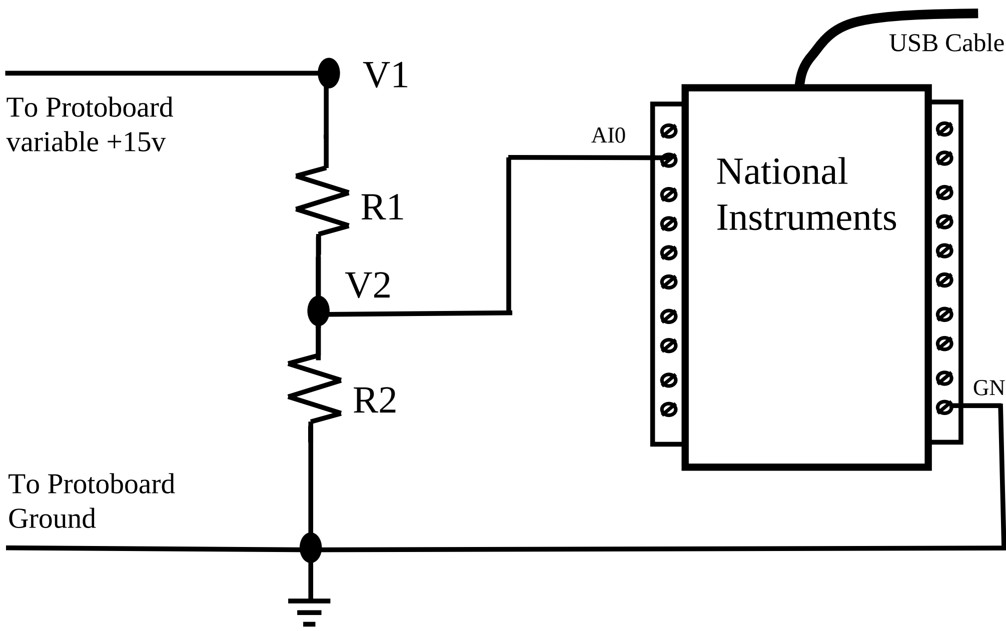
- Demonstrate that you read voltages from the NI box.

Rerange the input using a resistor network

The input voltage range for the NI box *AI5* input is from 0 volts to 10 volts, but the power supply for the protoboard can generate voltages in the range of 0 volts to +15 volts. Your job will be to design a resistor divider network that will reduce voltages in the range from zero volts to +15 volts to voltages in the range from zero volts to ten volts. The reduced voltage range can then be measured using the *AI5* input. You could modify the virtual oscilloscope to post-process the measured voltage and undo the effect of the resistor divider, but that can wait until later. In this way, you can use the NI box to effectively measure a larger range of voltages, making your own volt meter.

To see how to design such a circuit, consider figure below of the NI box, an external resistor network, and the connection points for a test voltage (supplied here by the protoboard +15 variable supply and the protoboard ground). If $R2 = 10,000\Omega = 10k\Omega$, what value should R1 have so that the voltage at *AI5* with respect to *GND* will be ten volts when the protoboard adjustable supply voltage is 15 volts (note: in the figure the NI box input is labeled AI0, but you will be using input AI5).

Once you have decided on a resistor value for R1, explain your solution to your LA. Then, please use the protoboard we have given you to construct your external resistor network. In order to construct your network, you may need to use series and parallel combinations of the resistors we have available to generate the resistance values you need. Ask your LA to show your our labeled box of resistors.



Question 10. Draw a diagram of your circuit, and explain how you designed it.

Checkpoint: 10:45 PM

- Demonstrate that you can measure voltages from 0 to +15 volts using the NI box and your resistor network.

More sophisticated reranging

Your protoboard has two variable supplies, one that can generate voltages as positive as +15 volts, and one that can generate voltages as negative as -15 volts. Suppose the NI box can only measure voltages in the range from zero to ten volts (it can actually measure a wider range, but lets assume the range is only from zero to ten volts). Also, the NI box has a five volt reference. Can you add a resistor between your resistor divider network and the five volt reference, and then adjust the values of your resistors, so that you can measure test voltages from the protoboard in the range from -15 volts to $+15$ volts?

Question 11. Draw a diagram of your circuit, and explain how you designed it.

Checkpoint: 11:30 PM

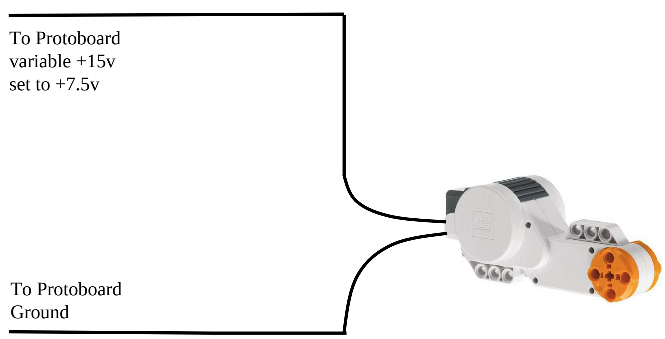
- Demonstrate that your resistor network reranges voltages from -15 to +15 volts to be between zero and 10 volts.

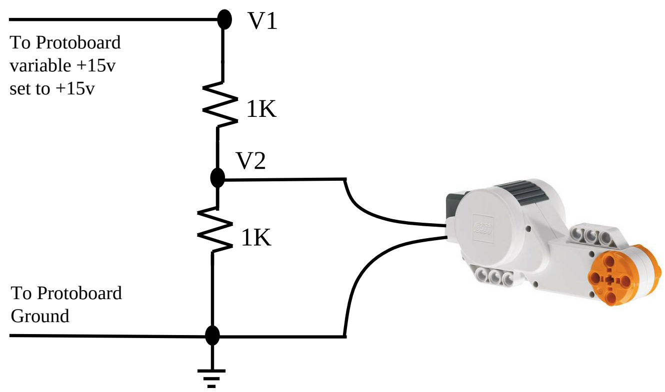
Using the Motor

We have supplied you with a motor, and you will be using the motor in the next lab to make a moving "head" for the robot. For this lab, you will be examining the voltage-current characteristics for the motor.

The speed of the motor depends on the voltage applied to the motor. Since your protoboard has an adjustable power supply, you can experiment with this relationship. As you will notice, the adjustable power supply is not calibrated, so it is quite fortunate that you just designed a volt meter that will let you measure the actual voltage of the adjustable supply. Use your volt meter to set the adjustable supply to 7.5 volts and try the running the motor using the circuit in the figure below. Note that the motor has a orange and green wire. You can interchange these two wires in the circuit, then the motor will spin in the opposite direction.

You can also generate 7.5 volts by using a voltage divider. Then you can connect the divider output to the motor, as shown in the bottom circuit in the figure below. Try this circuit and use your volt meter to understand what happened. Why did the circuit fail?





Question 12. Explain what happened when you tried the motor circuit using a voltage divider.

Checkpoint: 12:00 PM

- Demonstrate to your LA that you understand what happened when you tried the motor circuit using a voltage divider.

Measuring the motor current

Please determine how to use an external resistor (100Ω is a good value), the protoboard power supply, and your virtual oscilloscope to measure the current through the motor as well as the voltage across the motor. Then, try to determine a Thevenin equivalent model for the motor. Explain your approach to your LA.

Question 13. What was your Thevenin equivalent model for the motor?

Question 14. What measurements did you use to determine the model?

Question 15. How did you generate the model from your measurements?

Checkpoint: 12:45 PM

- Demonstrate that you can plot motor currents as a function of time.
- Describe your model to your LA.

Post-Lab Writeup for Tuesday and Thursday's labs: Due before lecture on October 30th

All post-lab hand-ins should be written in clear English sentences and paragraphs. We expect at least a couple of sentences or a paragraph in answer to **all** of the numbered questions in this lab handout (both Tuesday's software and Thursday's design lab).

We also want the code you wrote for Tuesday's or Thursday's lab (though there will not be much code to hand in for Thursday's lab).

Concepts covered in this assignment

Here are the important points covered in this assignment:

- Learn about how to reason about circuits.
- Learn about Thevenin equivalents, voltage dividers, and superposition.
- Learn about loading effects.
- Learn a little about circuit models.