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

Assignment 9a - Software Lab, Issued: Tuesday, Oct. 30th

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 9.1) 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 9 that are due on Thursday (Part 9.2).
3. Read through the entire description of Thursday's lab (a separate handout from this one).

...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 *and* the Thursday lab handout.

On Athena machines make sure you do:
`athrun 6.01 update`
so that you can get the `Desktop/6.01/lab9` directory which has the files mentioned in this handout.

- You need the files **resolveConstraints.py, circuitConstraints.py, genKCL.py (your code) or genKCL.pyc (our code)** 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.

Getting a head with circuits

In all the previous labs, we have mostly abstracted away the electrical nature of the signals being processed and generated by the robot. In this laboratory you will design and build an electrical feedback system which controls a robot "head". In subsequent labs, you will put "eyes" on the head to help your robot "see". As with the previous labs, you will start by building a python tool to help you analyze circuits, and then you will use that tool to help you design an improved robot head controller. So this week you will be extending the constraint resolver, analyzing an electromechanical system, and designing a circuit. Whew!

Tuesday's Software Lab: Adding an Op-amp to the Constraint Resolver

You will be extending the constraint resolver program for analyzing circuits by adding an op-amp model to the program. You can then use your augmented constraint resolver to help you design a circuit to control the motor.

Add an op-amp model to the constraint resolver

In class we discussed the operational amplifier circuit referred to as an inverting amplifier shown in figure 1

In order to simplify the analysis of the op-amp circuit, we assumed that $V_+ - V_- \approx 0$ and that no current enters the op-amp inputs. Using those assumptions, we determined that the relationship between V_{in} and $V_{out}$was

$$V_{out} = -\frac{R_f}{R_{in}} V_{in}.$$

The main reason we could assume that $V_+ - V_- \approx 0$ was that the op-amp was connected in a feedback configuration and the gain of the op-amp was very large. You will now investigate the accuracy of that approximation using the constraint resolver. Please use the version of the constraint resolver that does *NOT* automatically generate KCL equations. As part of an exploration, you can investigate how to add the op-amp to your derived class for two-terminal devices.

The abstract model we used for the op-amp was a voltage-controlled voltage source model, and using that model in the inverting amplifier yields a somewhat different looking schematic shown in figure 2.

where the gain of the voltage-controlled voltage source, K, is a very large number. We would like you to please add this voltage-controlled voltage source model to circuitConstraints.py, allowing K to be a parameter, analogous to way resistance is a parameter for resistors. Keep in mind that this model has four terminals, but there is only one nonzero current, i_{out}. Therefore, your constraint should involve four voltages and one current. Please carefully consider what to do with i_{out}, the current generated at the output terminals of the op-amp. Note that this output current is not used in the op-amp constitutive relations (recall that the same situation occurs with a voltage source element).

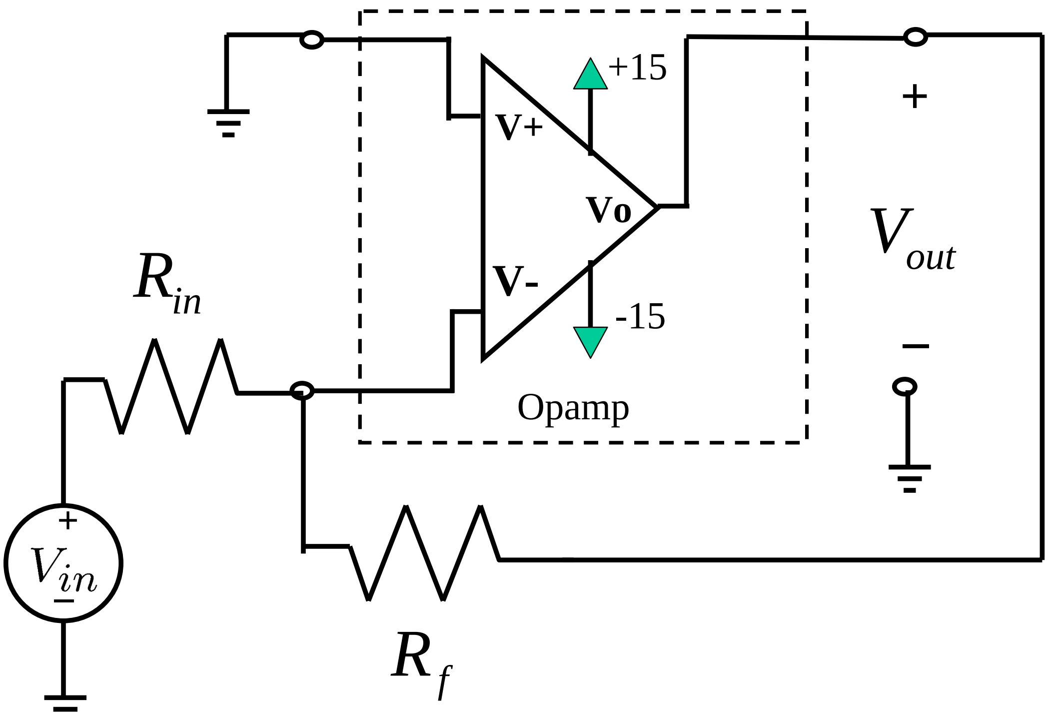

Figure 1: Operational amplifier connected in an inverting amplifier configuration

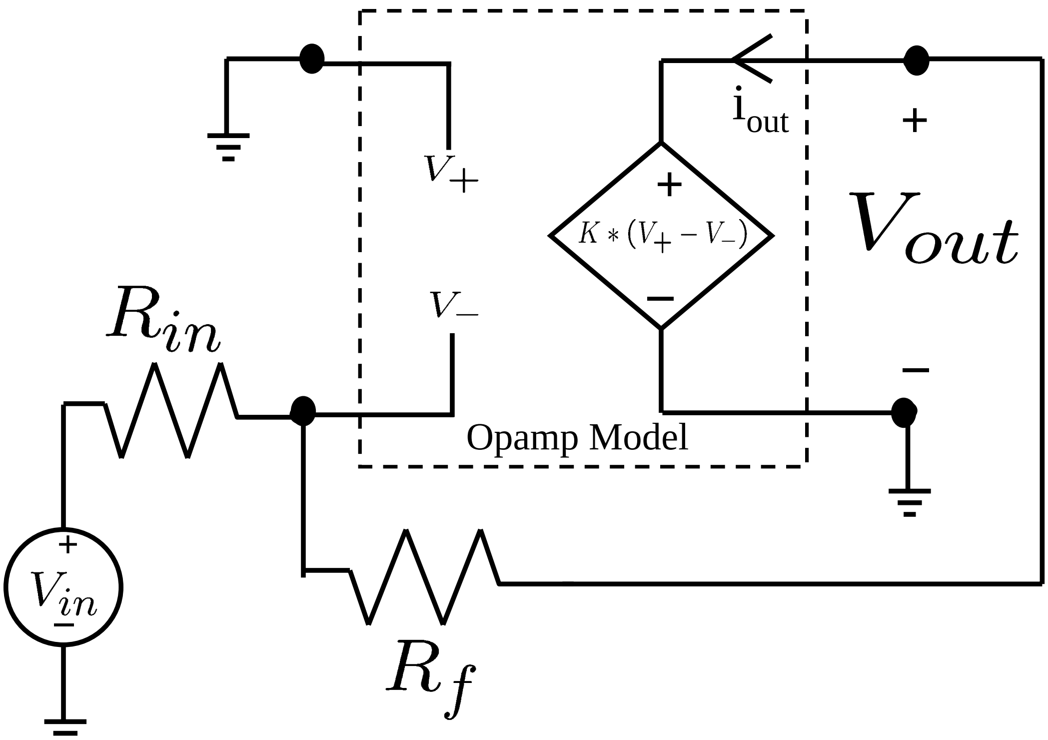

Figure 2: Inverting amplifier using a voltage-controlled voltage source op-amp model

Question 1. Describe your modification to `circuitConstraints.py` to implement the voltage-controlled voltage source model of the op-amp

Question 2. What constraint will make use of the the op-amp current?

Test your implementation of the op-amp constraints by analyzing the inverting amplifier in figure 2 using the constraint resolver. Use $R_{in} = 10,000$, $R_f = 100,000$, and try two cases: one where the op-amp gain, K, is $K = 1000$ and then when $K = 100$. Also, try several different input voltages to test the gain.

Question 3. Describe your python file that generates the constitutive equations and the conservation laws for the negative feedback amplifier circuit.

Question 4. What was the gain of the amplifier when $K = 1000$ and when $K = 100$? Did the gain you measured depend on the input voltage?

Question 5. If the input voltage is 10 volts, you would expect the output voltage to be near -100 volts. In the lab, we use protoboard power supplies that generate voltages in the range -15 volts to $+15$ volts? So what do you think would happen if you built the circuit in figure 2 using the protoboard power supplies and tested it with an input of 10 volts?

Question 6. Determine approximately the smallest value of op-amp gain, K, for which

$$\frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}}$$

is accurate to within ten percent.

Go to the on-line Tutor at `http://sicp.csail.mit.edu/6.01/fall07`, choose PS9, 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 along with your answers to the questions in Thursday's lab before lecture on Tuesday, November 6th.

You will need your op-amp extension for Thursday's lab, please be sure to save a copy where you can access it later.

Exploration - Using the Op-amp with your KCL generator

How would you update `genKCL.py` so that you can automatically generate KCL equations for circuits with a mix of resistors, voltage sources, and voltage-controlled voltage sources (like the op-amp)?