nano 7

10/21

$V = IR$

$I = \frac{V}{R}$

$I = \frac{V}{\frac{R_1 R_2}{R_1 + R_2}}$ ✓ just in terms of R

2. $7 + \frac{30}{13}$

$5 = \frac{37}{13}$ ~~R~~

~~I = 9.8~~ $V = 1.7$

1. $I_1 = \frac{R_2}{R_1 + R_2} I$ ✓

---

2. V = same

$V = 5.17$

85

✓ ~~18~~ $= \frac{37}{13}$ ~~R~~ 5

14.23 x

~~$\frac{\frac{37}{13}}{85 + \frac{37}{13}}$~~

B (2)

Just too pressured

$V = IR$

$10 = R \cdot 1$

$R = 10$

$V = 5$ ⓧ

Just not focused

And whats a current source got to do w/ it

Need ~~sm~~ no tire pressure

give up

2. 50 V
3. 15 V

So for 2 is it just

$V = IR$

$V = 5 \cdot 10$

~~Same~~

but what about the 3 there

different <u>current</u> in each

(3)

3. $V = IR$

$V = 1.5 + 10$?

Not what it is asking

Voltage difference over 1A current source

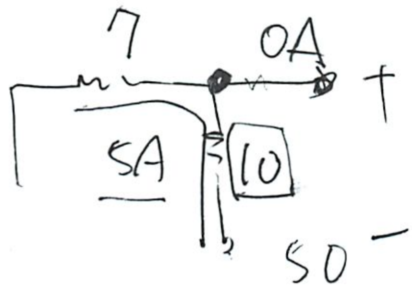
p4 lecture slide 3

current source is current through resistor

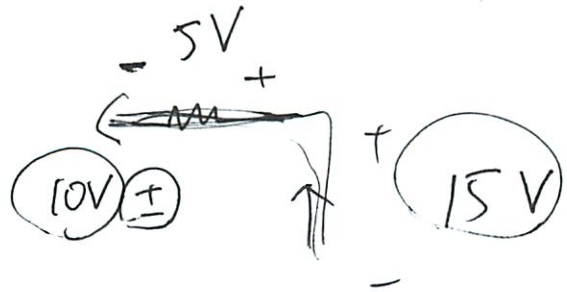
$V = 1.5$

then was adding 10 correct?

~~$V = 17.5$~~



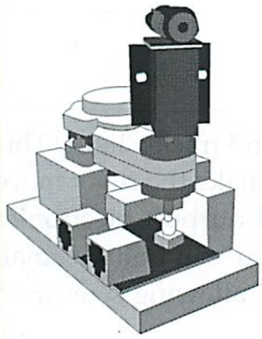
3.



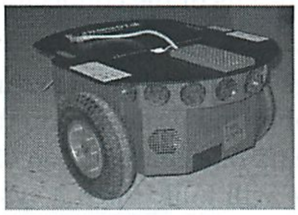
what I had thought

Design Lab 7: For Your Eyes Only

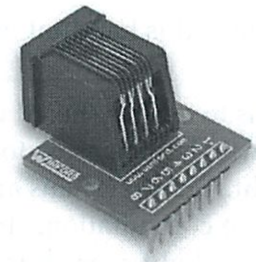
Each partnership should have a lab laptop and a robot. Do `athrun 6.01 update`. Files will be in `Desktop/6.01/lab7/designLab/`. In addition, you will need:



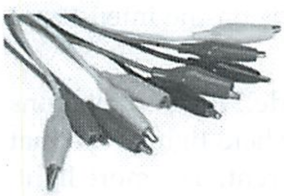
Robot head



Robot



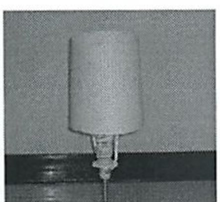
Two eight-pin connectors



Two clip leads

Multimeter

Resistors, tbd

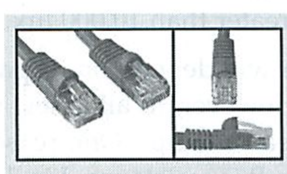


Floor Lamp

Lamp+extension cord

Wirekit

Proto board



Red cable

The relevant files in the distribution are:

- `eyeDataBrain.py`: A brain that rotates the robot a fixed amount; used for data collection.
- `turnToLightBrainSkeleton.py`: A brain with a place for you to write a state machine that will turn the robot to face the light.

See the back page of this handout for the pin-outs of the connectors.

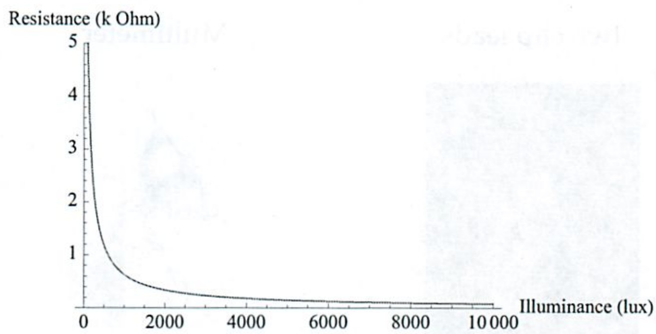
Our ultimate goal is to build a 'head' with eyes and a neck, which can turn and track a light. This week you will connect light sensors to your robot and write a brain that can make the robot move to face a bright light. Many systems are made up of a combination of special-purpose electronics and general-purpose computation. Today, we will build such a system. You will build a small circuit to connect two light sensors to the robot and, thence, to a lab laptop, allowing soar brain software to read voltages from your board and command voltages back to it.

1 Seeing the Light

Our first task is to use a pair of photoresistors to construct light sensors (eyes) for the robot head. A photoresistor is a two-terminal device whose electrical resistance depends on the intensity of light incident on its surface.

A photoresistor is made from a high resistance material (e.g., cadmium sulfide). Incident photons excite electrons – liberating them from the atoms to which they are normally held tightly – so that the electrons can move freely through the material and thereby conduct current. The more light, the higher the current and thus the lower the resistance.

Photoresistor



The net effect can be characterized by plotting electrical resistance as a function of incident light intensity, as shown above. Normal room lighting is between 10 and 100 lux. Illuminance near a 60 watt light bulb (as we will use in lab) can be greater than 10,000 lux.

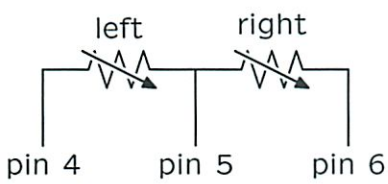
The details of the behavior of the photoresistor will depend on its particular design and will vary substantially even among "identical" parts. However, in all cases, we expect to see a very high resistance in the dark and a low resistance near a lamp. One reasonable approximation to the behavior of a photoresistor is: $R = R_0/I$, where R is resistance, I is illuminance and R_0 is a (large) constant, representing resistance when the illuminance is 1 lux (dark).

When thinking about the behavior of a photoresistor, keep in mind that **illuminance will drop as the square of the distance from a light source.** So, resistance will increase as you move sensor away from the light source.

Step 1. We will start by measuring the resistance of the photosensors in different lighting conditions. Plug an 8-pin connector into an otherwise empty protoboard. Use a red cable to connect that connector to the 8-pin connector on the robot head. We will call this connector on the protoboard the **head connector.** A description of which connections are to be found on which pins of this connector (known as a "pin-out") is at the end of this handout.

Step 2. Notice that there are two photosensors on the head. The photosensors are wired to the head connector like this:

not need to wire



Connect your meter to pins 4 and 5 of the head connector, using short wires as probes into the protoboard. Switch your meter to measure resistance (the scale is marked Ω), and be sure you understand the scales (you can always measure a known resistor to help figure out the scale). Get a silver hand-held lamp and plug it in to a power strip.

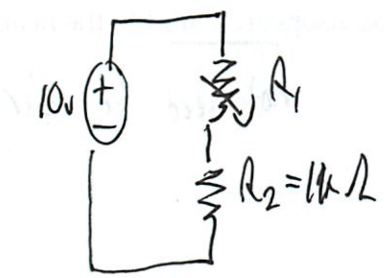
Measure the resistance (in Ω) of **each photosensor individually** under the following lighting conditions. Don't worry if the sensors respond very differently.

	4-5 Left	5-6 Right
Ambient light	7	10.2
One foot in front of lamp	.45	.6
Three feet in front of lamp	2	2.4

20kΩ

So all figures in kΩ

Step 3. Design a circuit that uses one photoresistor (plus one or more additional resistors) to generate a voltage that is large under bright conditions and small under dark conditions. The robot provides a 10 V voltage source. There will need to be a location in your circuit where we measure the voltage with respect to ground. Make sure that, at this location, there is at least a 3 V difference between the voltage with respect to ground in ambient conditions and when there is a lamp one foot away. Sketch your circuit below. *Think about how to make a variation in resistance into a variation in voltage.*



What voltage do you expect for the following lighting conditions?

	Left	Right
Ambient light		
One foot in front of lamp		
Three feet in front of lamp		

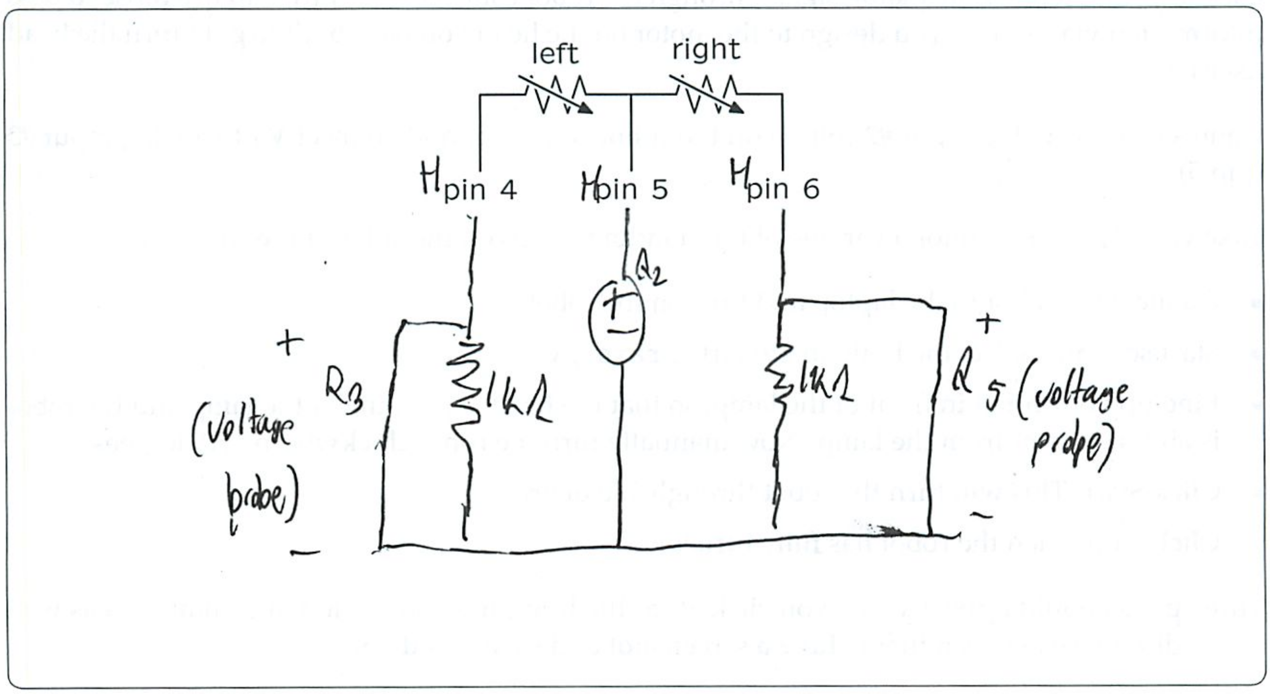
Check Yourself 1. Verify your design and calculations. If you have any questions, ask a staff member.

Step 4. Plug a second 8-pin connector into your proto board; we will call this the **robot connector**. You can connect your circuit to a robot via the yellow 8-pin cable that is coming out of the robot **(don't do it yet though; that would make it awkward to work on your board).** This connector is exactly the same as the head connector; to help keep this distinct from the red cable connecting your board to the head, remember: 'red' — 'head'. The pin-out of the robot connector is described at the end of this handout.

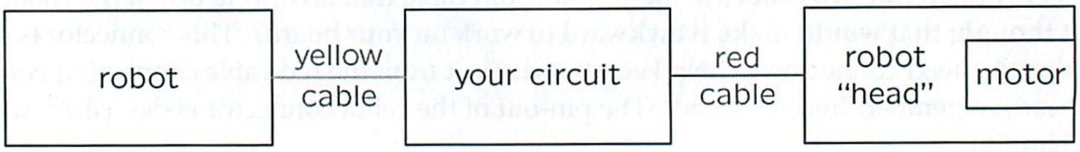
Connect power and ground on your board to the corresponding pins (pins 2 and 4, respectively) on the robot connector.

The robot has within it a set of converters. A-to-D (analog to digital) converters take analog voltages from pins 1, 3, 5, and 7 of the robot connector, sample them on each soar cycle, and encode them in binary so that they may be read by the computer, ultimately showing up as a list of values in the `analogInputs` attribute of a `io.SensorInput` object. A single D-to-A (digital to analog) converter takes a value specified by the `voltage` argument to the `io.Action` constructor, and converts it into a voltage, which is made available on pin 6 of the robot connector; we will be using this output voltage in later weeks.

Step 5. Draw a schematic for two photoresistor circuits, one to generate the voltage V_L from the left photoresistor and one to generate the voltage V_R from the right photoresistor, using pins 4, 5, and 6 on the head connector.



Step 6. Build the circuits you designed. Here is how the whole system should be configured:



Attach the head to the lego plate on the front of the robot (sometimes putting a couple of lego bricks in between will make this easier), connect the yellow robot cable to your board, and **turn on the robot**. Use your multimeter to make sure that you are getting reasonable values for V_R and V_L. You can use your finger to obscure each of the sensors in turn and see that the voltages behave as expected.

Think of this combination as having created a new component to your robot. The robot head has sensors that can convert information about the world (light) into electrical signals, which the head connector delivers to your circuit (through the convention specified by the pin-out). Your circuit then converts that information into voltages, and supplies those analog signals to the robot via the robot connector and its pin-out. The robot can then use that information together with its attached (or internal) computer to decide actions. In some cases these actions will be used directly by the robot (e.g. turning its wheels), but we will see later that those actions can also be passed back to your circuit as an analog signal through the robot connector, and we will use those to pass information via a circuit you design to the motor on the head you have built (e.g. to turn the head itself).

Step 7. Connect V_L to analog input #2 (pin 3) on the **robot** connector and connect V_R to analog input #3 (pin 5).

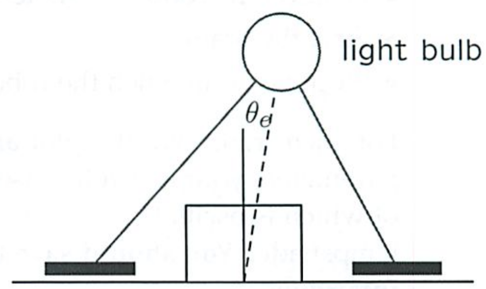
Step 8. Take your robot and laptop near one of the standing lamps on the sides of the room.

- Connect the robot to the laptop and turn on the robot.
- Start soar and select the brain in `eyeDataBrain.py`.
- Line up your robot in front of the lamp, so that the head is pointing at the lamp and the robot is about a meter from the lamp. Now manually turn the robot **clockwise** by 90 degrees.
- Click Start. This will turn the robot through 180 degrees.
- Click Stop when the robot has fully turned.

Three plots should appear when you click Stop: the brightness on the left and right eyes as well as the difference between them. **Take a screenshot and save this data.**

Step 9.

Now we are interested in designing a controller that will make the robot turn so that the head is directly pointed toward a bright light anywhere in front of the robot. It should do this as accurately as possible. Examine your plots and decide how you want to approach this. The photoresistors are separated by a "nose" that casts a shadow, and thereby controls the amount of light on the two photoresistors (see top-down view at right).



Checkoff 1.	**Part a.** Explain your plots to a staff member. **Part b.** Describe your approach to pointing at the light. Be prepared to justify your approach.

2 Bull's Eye

Go back to your original table and debug using a silver lamp. When you think your program is roughly right, then go test it using a standing lamp.

Step 10. Modify `turnToLightBrainSkeleton.py` to implement your approach to pointing the robot towards the light. It will generate one plot, which you can ignore for now. The voltages from the photosensors can be read as follows:

```
# io.SensorInput().analogInputs is a list of all 4 analog inputs
left = inp.analogInputs[1]
right = inp.analogInputs[2]
```

Step 11. To enable us to see how well your pointing approach works, we have attached low-power laser pointers on the robot heads. To power the laser, plug it into the small round connector coming out of the robot near the yellow cable.

Step 12. We want the laser to move quickly and reliably to make the red dot fall on the lampshade. First, tune your controller so that the laser lands reliably on the lampshade. You will probably have to compensate for discrepancies in the responses of the two eyes to the light. Be prepared to discuss your approach for dealing with this.

Step 13. Once your controller's accuracy is adequate, you can focus on its speed. When you stop the brain, you will see a plot of the robot's angle as a function of the number of time steps. An estimate of the *settle time* of the signal will be printed to the output window. A signal S *settles* at time t if it is within some fixed value ϵ of its final value starting at time t and for all time steps after that. If your signal is not converging, the reported number will not be meaningful, and will be some time near the end of the signal.

Use the following procedure to collect data for your controller:

- Position the robot one meter away and pointing at a 45 degree angle away from the lamp.
- Start the brain.
- Stop the brain when the robot stops moving, or when it is clear that its angle will not converge.

For each trial, save the plot and record the associated settle time and gain value (or whatever parameters your controller uses). Generate three plots with substantially different behaviors, one of which is oscillatory. Except in the oscillatory case, each trial should end with the laser on the lampshade. **You should save these plots (and associated parameters and settle times) for your interview.**

Step 14. Finally, experimentally optimize your controller with respect to settle time. You don't need to spend too long on this, but you should try several more parameter settings, and **save the plot, settle time, and parameters for the controller you find to have the best behavior. Save the code for your best controller for your interview as well.**

Checkoff 2.

a. Discuss your approach to the problem.
b. Demonstrate your best controller.
c. Discuss the plots, parameters, and settle times for several controllers with different behaviors.

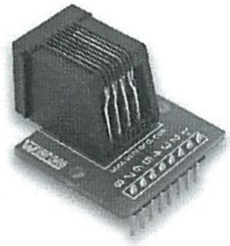
For fun...

After you're done with everything else, change your brain so that it follows the light. It should move forward or backward to keep a desired distance from the light, as well as rotating towards the light.

No matter what

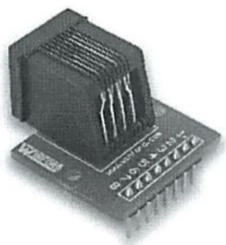
- Mail your brain, plots, parameters, and settle times to your partner.
- Disassemble the circuit board; return both 8-pin connectors and cables to the supply bins.
- Turn off your robot and your multimeter.

Head Connector Pin-out



pin 1:		neck pot (top)
pin 2:		neck pot (center)
pin 3:		neck pot (bottom)
pin 4:		photoresistor (left)
pin 5:		photoresistor (common)
pin 6:		photoresistor (right)
pin 7:	V_{M+}	Motor drive +
pin 8:	V_{M-}	Motor drive −

Robot Connector Pin-out



pin 1:	V_{i1}	analog input #1
pin 2:	$+10$ V	power (limited to 0.5 A)
pin 3:	V_{i2}	analog input #2
pin 4:	ground	
pin 5:	V_{i3}	analog input #3
pin 6:	V_o	analog output
pin 7:	V_{i4}	analog input #4
pin 8:	$+5$ V	power (limited to 0.5 A)

des Lab 7

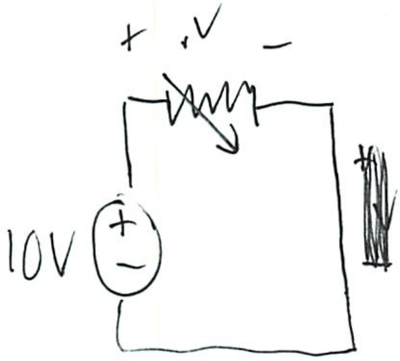
10/21

How to make a variation in resistance w/ variation in V

Voltage divider = series circuit

$V = I R$

$\uparrow$ larger $\uparrow$ small

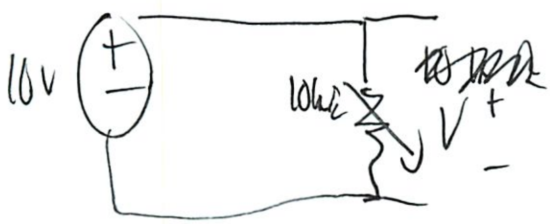


~~ambient~~

$10V = I \cdot$

if this is only resistor

Voltage drop always = 10



$10V = I \underline{R}$

$\uparrow$ Ro

Intensity

~~current~~

Output voltage constant

" i changes

how convert the i to V?

Want 3-5 V drop on photo resistor

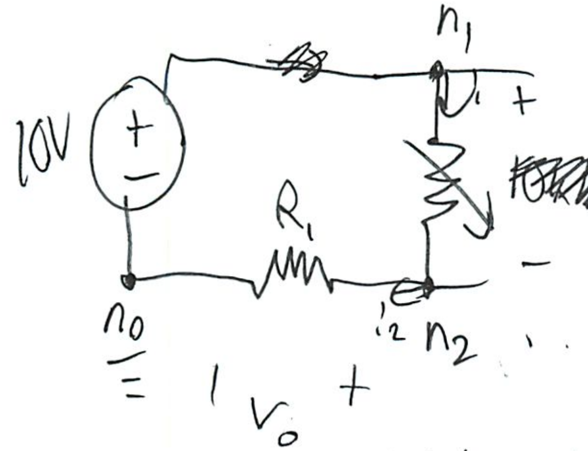
②

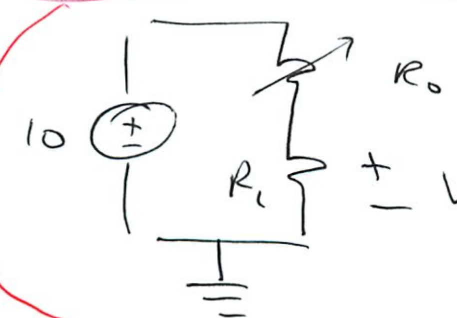
Convert $i \to V$

by running through R

In series = voltage divider

Signs were messed up





$I_0 = I_1$

1 path

Prof.

$V_2 - V_1 = I_0 R_0$ ← want 3-5 V

do MVCC

slow + steady wins the race

$V_2 - V_0 = I_1 R_1$

~~$A_2 \neq A_1$~~

$V_0 = \frac{R_1}{R_1 + R_0} 10$

$\frac{1}{1+?}$

$\frac{1}{1+.45}$

$10V - I_0 R_0 - I_1 R_1 = 0$

$10V = I(R_0 - R_1)$

I changes w/ R

or plug into I - remember MVCC

$\frac{V_2 - V_1}{I R_0} = \frac{V_2 - 0}{R_1}$

← here he guessed 1

See p4 formal

(3)

$\underline{V_2 - V_1}$ $V_1 = 10$

$$\frac{V_2 - 10}{R_0} = \frac{V_2}{R_1}$$

or just 3-5V diff

Want $V_2 = 5V$ $V_2 = \cancel{0V}\ 1V$

if $R_0 = .45$ $R_0 = 7$

So what R_1 satisfies both ?

$$\frac{1-10}{7} = \frac{1}{R_1}$$

$-9R_1 = 7$

$R_1 = .777$

$$\frac{5-10}{.45} = \frac{1}{R_1}$$

$-5R_1 = .45$

$R_1 = .09$

(4)

$$V_H = \frac{R_1}{R_1 + .45} 10$$

$$V_L = \frac{R_1}{R_1 + 7} 10$$

R_1

$$V_H - V_L = 5$$

$$\frac{R_1}{R_1 + .45} \cdot 10 - \frac{R_1}{R_1 + 7} \cdot 10 = 5$$

$$\frac{10 R_1}{R_1 + .45} - \frac{10 R_1}{R_1 + 7} = 5$$

+ solve

$$\frac{10 R_1}{R_1 + .45} = 5 + \frac{10 R_1}{R_1 + 7}$$

$$10 R_1 = 5(R_1 + .45) + \frac{10 R_1 (R_1 + .45)}{R_1 + 7}$$

$$10 R_1 = 5 R_1 + 2.25 + \frac{10 R_1^2 + 4.5 R_1}{R_1 + 7}$$

etc - not going to do
calc solve by hand

(5)

~~Late~~ 8. wolfram alpha

$X = .62 \leftarrow 627\,\Omega$

$X = 5.02 \leftarrow 5k\,\Omega$

(✓) see was able to solve when went slow and did MVCL Got it

What voltages to expect

$\cancel{V_{drop} = I_0 R_0}$

$$\frac{V-10}{R_0} = \frac{V}{R_1} \qquad R_1 = .627$$

So $R_0 = 7$

$V = ?$

$$(V-10)(R_1) = VR_0$$

$$.627V - 6.27 = V \cdot 7$$

$$.627V - 7V = 6.27$$

$$V(.627 - 7) = 6.27$$

$$V = \frac{6.27}{.627-7} = .98$$

(6)

Should do generic

~~V~~ ~~10~~ .

$R_1 V - 10 R_1 = V R_0$

$R_1 V - R_0 V = 10 R_1$

$V(R_1 - R_0) = 10 R_1$

$$V = \frac{10 R_1}{R_1 - R_0}$$

$R_1 = .627$

$$V = \frac{6.27}{.627 - R_0}$$

$R = .45$ / $V = 35$

$R = .6$ / $200V$

$R = 2$ / $V = 4.5$

$R = 2.4$ / $3.5V$

$R = 10.2$ / $V = .65$

drew used 1kΩ resistor
Got vastly diff #

(7)

$R_1 = 1$

$V = \dfrac{10}{1 - R_0}$

$R_0 = 7$ / $V = +1.66$

$R_0 = 10.2$ / 1.08

$R_0 = .45$ / $V = 18$

$R_0 = .6$ / $V = 25$

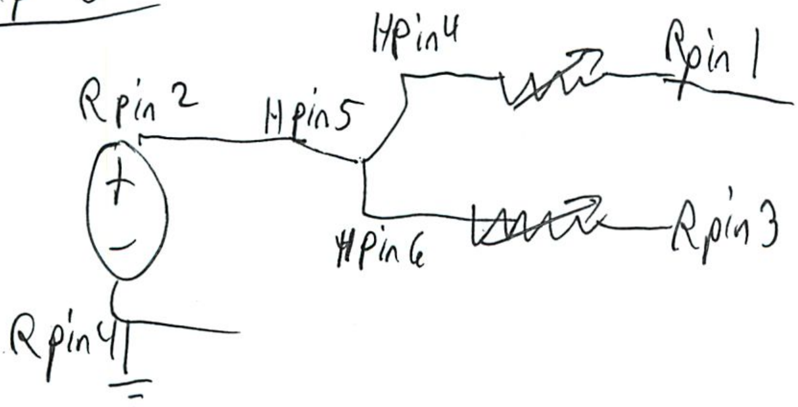
$R_0 = 2$ / $V = +10$

$R_0 = 2.4$ / $V = 7.1$

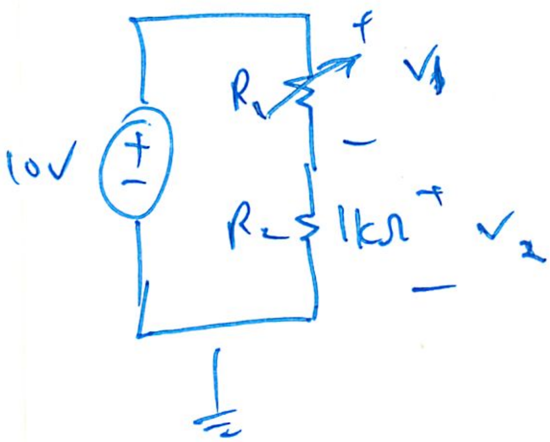
still very different

- as long as there is a difference

Step 5



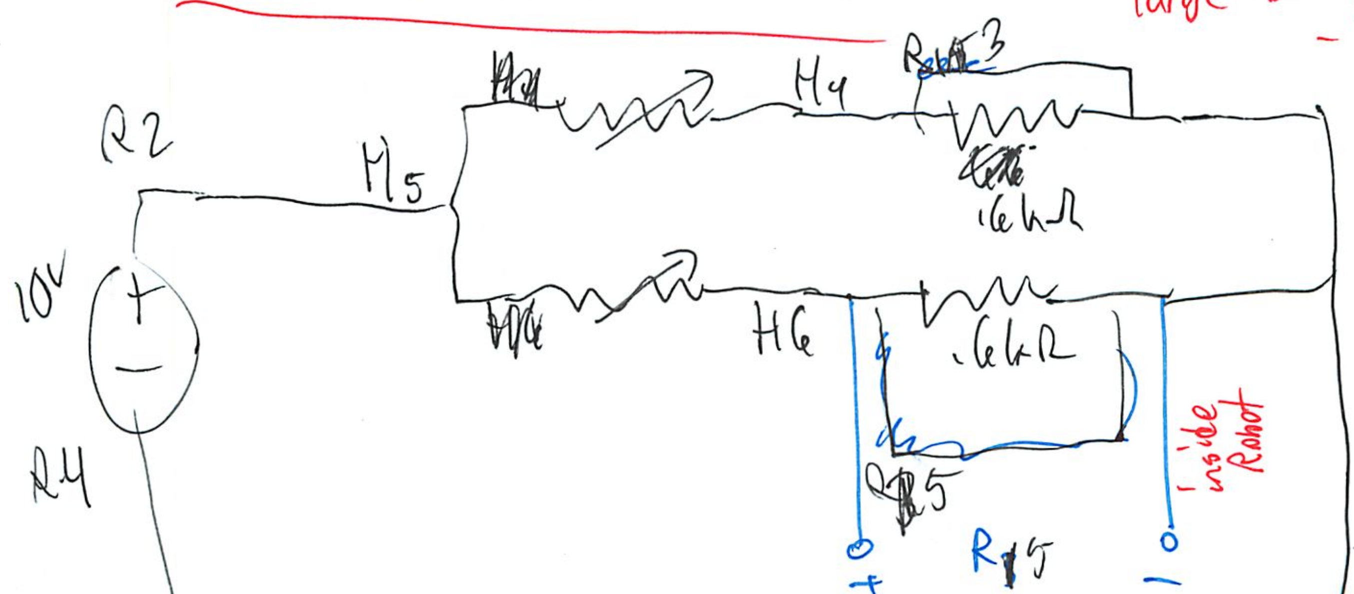
(8)



$$V_1 = 10V \cdot \frac{R_1}{R_1 + R_2}$$

$$V_2 = 10V \cdot \frac{R_2}{R_1 + R_2}$$

Want to measure V_2 ! − depends which is large or small − if wanted inverse could flip



9

Brighter light = higer voltage

Checkoff 1

Our laser did not work?

- no need to push buttom

Why not symetrical?

- different resistors
- correct in HW or SW

add resistor   many things

easier

Want left - right = 0

left > right = turn left

" < right = turn right } proportional

Calibrate mode??

at startup read relative values b/w them

save that as offset

Want left - right = offset ≈ 0

left + offset > + right turn left

right + offset > left turn right

am I flipping + and -? try some #

(10) Actually worked fairly well

- Want proportional now

Theta $\frac{\text{left} - \text{offset}}{\text{right}}$ ~~ML~~

; same thing

$\frac{\text{right} + \text{offset}}{\text{left}}$

$\text{right} + \text{offset} = \text{left}$

$\text{right} = \text{left} - \text{offset}$

$\frac{\text{left} - \text{offset}}{\text{right}}$

; can I do this

* Need to -1

$\frac{\text{left} - \text{offset}}{\text{right}} - 1$

Since before the changes are ~~A~~ $> < 1$

we want $> < 0$

Fast way of thinking - not 6.01

(11)

But for settle time to be valid
Need to hard code offset
Offset = 130 experimentally for this head

Settle time = 28

checkoff 3 We did ratio b/w left and right
this is proportional to difference from 10

ratio - portional to how far off
Subtraction - linear

light better
- ratio is better

input is non linear
- better take ~~log~~ ln
- and work w/ that (subtraction since linear)

Our eyes do ratio

$\ln\left(\frac{I}{ref}\right)$

ratio correction
than linearize

12

We had implicit gain of 1

Worked fairly well

Could we have done better?

Could we draw a system diagram???

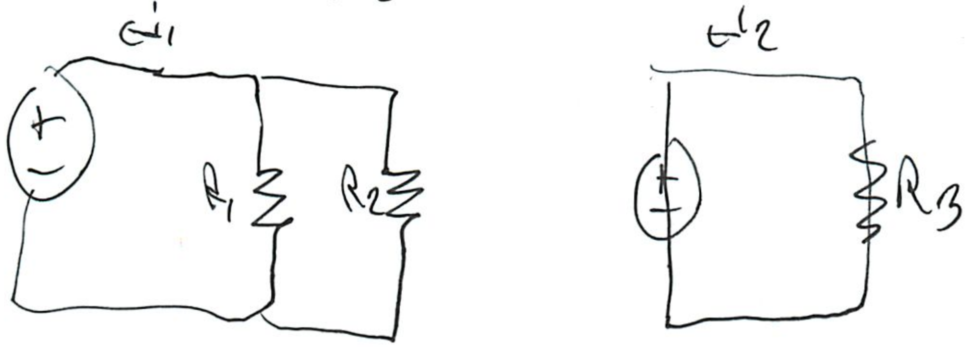
- would take me a while

HW Week 7

10/22

Parallel Resistors

to build intuition
no calculators



$R_1 = R_2 = 100\ \Omega$

$$R_3 = \frac{100 \cdot 100}{100+100} = \frac{10000}{200} = \frac{100}{2} = 50\Omega$$

2. if $R_1 = 100\ \Omega$ $R_2 = 10,000\ \Omega$
what value of R_3 will $i_{s1} = i_{s2}$
- aka where will current be eq
- so where is Req

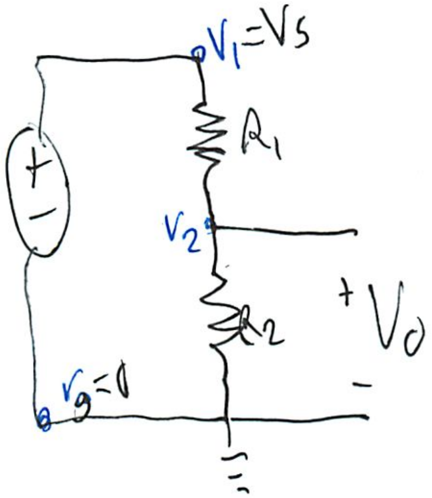
$$\frac{100 \cdot 10,000}{100+10,000} = \frac{1,000,000}{10,100} = 99\ \Omega$$

3. Same qu $R_1 = 100$ $R_2 = 1\ \Omega$

$$\frac{100 \cdot 1}{100+1} = \frac{100}{\cancel{100}\ 101}\ \Omega = .99\Omega$$

(2)

Resistance Dividers



a. For given R_1 and R_2 if $R_2 \uparrow$ V_o will

- current same

$V = IR$

$\downarrow \quad \uparrow$

~~decrease~~ (X) increase

b) $R_1 = 100\ \Omega$ $R_2 = 10{,}000$

What is $\frac{V_o}{V_s}$?

do formal ~~way~~ method

$\cancel{V_2 - V_1} = I R_1$

$\cancel{V_2} \uparrow$

go other way

$V_1 - V_2 = I R_1$

$V_2 - V_g = I R_2$

(3)

$$V_s - V_2 = I R_1$$
$$V_2 - 0 = I R_2 \qquad I = I$$

$$\frac{V_s - V_2}{R_1} = \frac{V_2}{R_2}$$

$$V_2 = V_o$$

$$\frac{V_s - V_2}{100} = \frac{V_2}{10000}$$

$$100 V_2 = 10000(V_s - V_2)$$
$$100 V_2 + 10000 V_2 = 10000 V_s$$
$$V_2 (10100) = 10000 V_s$$

$$\frac{V_2}{V_s} = \frac{10000}{10100} = \cancel{X} \quad \cancel{.904} = \cancel{1099} = .99$$

duh wrong

c)

$$\frac{V_s - V_2}{10000} = \frac{V_2}{100}$$

$$100(V_s - V_2) = 10{,}000 V_2$$
$$100 V_s - 100 V_2 = 10000 V_2$$
$$100 V_s = 10{,}100 V_2$$
$$\frac{V_2}{V_s} = \frac{100}{10100} = .0099$$

(4)

d) if $V_o = \frac{1}{5} V_s$ What is ratio R_1/R_2

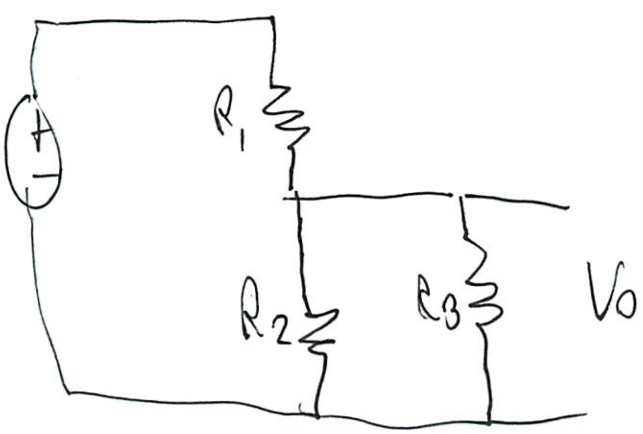
$$\frac{V_s - \frac{1}{5}V_s}{R_1} = \frac{1/5 V_s}{R_2}$$

$$\cancel{V_s - \frac{1}{5}V_s}$$

$$\frac{4}{5} V_s R_2 = \frac{1}{5} V_s R_1$$

$$\frac{R_1}{R_2} = \frac{4/5 V_s}{1/5 V_s} = 4$$

Part 2 loaded



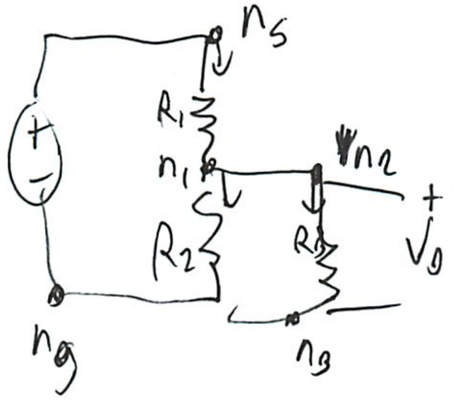
$R_1 = R_2 = 1000\,\Omega$

Interested in addition V_o

Voltage across R_2 (w/o R_3) $= V_d$

5

1. If R_3 has high value (100,000 Ω)
How does new V_o compare to V_d $\quad \frac{V_o}{V_d}$?



$$V_5 - V_1 = I_5 R_1$$
$$V_1 - V_g = I_1 R_2$$
$$V_2 - V_3 = I_2 R_3$$

But $V_3 = V_g$
$V_1 = V_2$

$$V_5 - V_1 = I_s R_1$$
$$V_1 - 0 = I_1 R_2$$
$$V_2 - 0 = I_2 R_3$$

$$I_s = I_1 + I_2$$

NVCC

$$\frac{V_5 - V_1}{R_1} = \frac{V_1}{R_2} + \frac{V_1}{R_3}$$

(4)

$R_1 = R_2 = 1000\,\Omega$

$V_0 = V_1 - 0$

$V_0 = V_1$

$$\frac{V_S - V_1}{1000} = \frac{V_1}{1000} + \frac{V_1}{100{,}000}$$

$$\frac{V_S - V_1}{1000} = \frac{101 V_1}{100{,}000}$$

$$101000\, V_1 = 100{,}000(V_S - V_1)$$

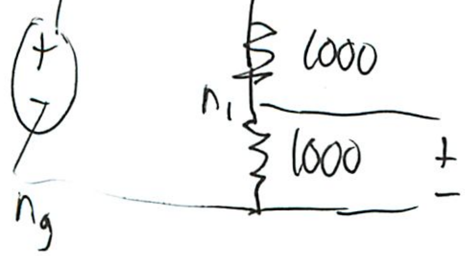
$$101\,000\, V_1 = 100{,}000\, V_S - 100{,}000\, V_1$$

$$(101{,}000 + 100{,}000) V_1 = 100{,}000\, V_S$$

$$V_1 = \frac{100{,}000\, V_S}{201{,}000}$$

$$V_1 = \frac{100}{201} V_S$$

Now w/out



$V_S - V_1 = I\, 1000$

$V_1 - V_g = I \cdot 1000$

$I = I$

$$\frac{V_S - V_1}{1000} = \frac{V_1 - 0}{1000}$$

(a)

$$1000(V_s - V_1) = 1000V_1$$

$$1000V_s - 1000V_1 = 1000V_1$$

$$1000V_s = 2000V_1$$

$$V_1 = \frac{1}{2}V_s$$

$$\frac{V_{1\,now}}{V_{1\,old}} = \frac{\frac{100}{201}V_s}{\frac{1}{2}V_s}$$

Taking really long way but getting them right

$$= \frac{100V_s}{201} \cdot \frac{2}{1V_s}$$

$$= \frac{200}{201} = .995 \text{ (✓)}$$

b) R_3 has low value $10\,\Omega$

V_{now}/

$$\frac{V_s - V_1}{1000} = \frac{V_1}{1000} + \frac{V_1}{100}$$

$$\frac{V_s - V_1}{1000} = \frac{11V_1}{1000}$$

$$1000(V_s - V_1) = 11{,}000V_1$$

(8)

$$1000\,V_s - 1000\,V_1 = 11{,}000\,V_1$$

$$1000\,V_s = 12000\,V_1$$

$$V_1 = \frac{1}{12} V_s$$

$$\frac{V_{new}}{V_{old}} = \frac{\frac{1}{12} V_s}{\frac{1}{2} V_s}$$

$$= \frac{V_s}{12} \cdot \frac{2}{V_s} = \frac{1}{6} = .1667 \quad \otimes$$

Opps I did 100 Ω

$$\frac{V_s - V_1}{1000} = \frac{V_1}{1000} + \frac{V_1}{10}$$

$$\frac{V_s - V_1}{1000} = \frac{101 V_1}{1000}$$

$$1000\,(V_s - V_1) = 101{,}000\,V_1$$

$$1000\,V_s - 1000\,V_1 = 101{,}000\,V_1$$

$$1000\,V_s = 102{,}000\,V_1$$

$$V_1 = \frac{1 V_s}{102}$$

(9)

$$\frac{V_{now}}{V_{old}} = \frac{\frac{1V_S}{102}}{\frac{1}{2}V_S}$$

$$= \frac{V_S}{102} \cdot \frac{2}{V_S}$$

$$= \frac{2}{102} = .0196$$ ✓

c) $R_1 = R_2 = R_3 = 1000\ \Omega$

no intuition, just plug & chug
↑ just confuses you

$$\frac{V_S - V_1}{1000} = \frac{V_1}{2000} + \frac{V_1}{1000}$$

$$\frac{V_S - V_1}{1000} = \frac{V_1}{500}$$

$$500(V_S - V_1) = 1000V_1$$

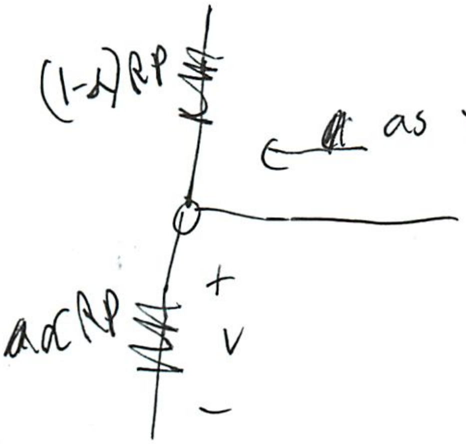
$$500V_S = 1500V_1$$

$$V_1 = \frac{500}{1500}V_S = \frac{1}{3}V_S$$

$$\frac{V_{now}}{V_{old}} = \frac{\frac{V_S}{3}}{\frac{1}{2}V_S} = \frac{V_S}{3} \cdot \frac{2}{V_S} = \frac{2}{3} = .6667$$ ✓

(10)

3. Pot → Potentiometer



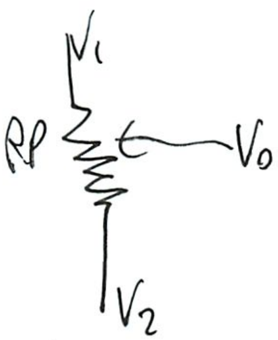
← as you turn, this top ~~part~~ ~~get~~ resistance decreases

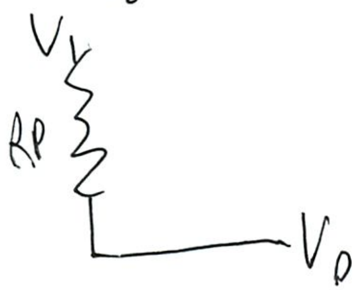
Sum of top + bottom constant

α is between 0-1

a) $\alpha = 0$

Total Resistance = RP =





$V_0 = ?$

$V_1 - V_0 = I\,RP$

assume 1

$V_1 = RP + V_0$

$V_0 = V_1 - RP$ (x)

(1c)

Should I enter w/ I?

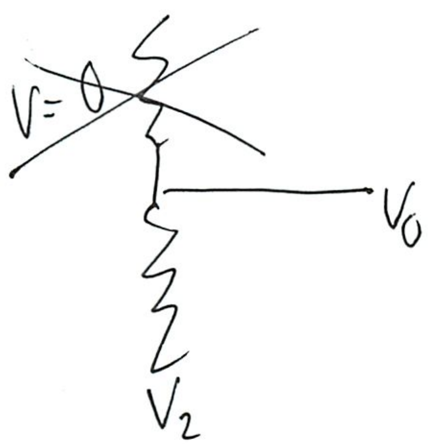
Oh they put α on bottom

No I compensated for that – removed bottom voltage

Or could it flipped

- what is

b) $\alpha = 1$



$V_2 - V_0 = RP$

$V_2 - RP + V_0$

$V_0 = V_2 - RP$

Same as I got above

Course Notes say

$$V_{out} = V_{in} \frac{R_B}{R_A + R_B}$$

$\uparrow$
V_1

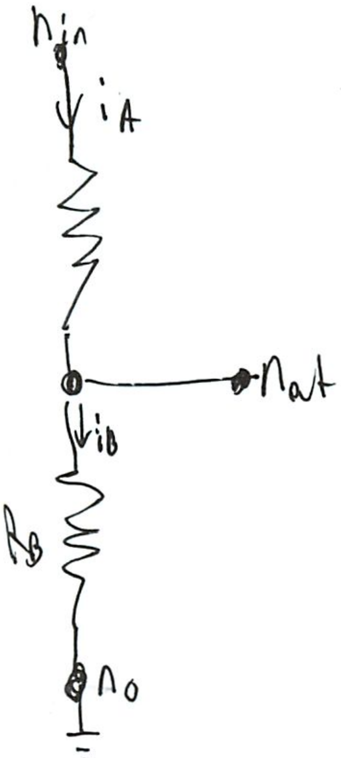
$$= V_1 \frac{RB}{RB + 0}$$

$$= V_1$$

x

12

Why are they using weird variables
- confusing



$R_B = \alpha RP$

$R_A = (1-\alpha) RP$? why not switch?

$V_0 = 0$

$i_A - i_B = 0$

$V_{in} - V_{out} = i_A R_A$

$V_{out} - V_0 = i_B R_B$

$$\frac{V_{in} - V_{out}}{R_A} = \frac{V_{out} - V_0}{R_B}$$

$$\frac{V_{in} - V_{out}}{1 \cdot RP} = \frac{V_{out}}{0}$$

divide by 0 ??

$V_{out} = RP$ ⊗

How about $R_B = .0001$ ($\lim \to 0$)

(13)

$.0001\,(V_{in}-V_{out}) = RP\,V_{out}$

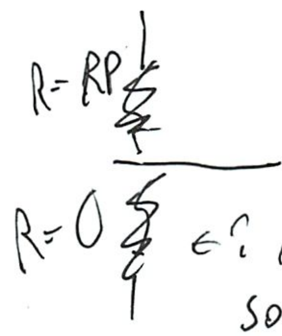
$.0001\,V_{in} = (RP + .0001)\,V_{out}$

$$V_{out} = \frac{.00001\,V_{in}}{RP + .0001}$$

$$= \frac{0}{RP} = 0$$

? so what is it in the weird format they want ?

So again

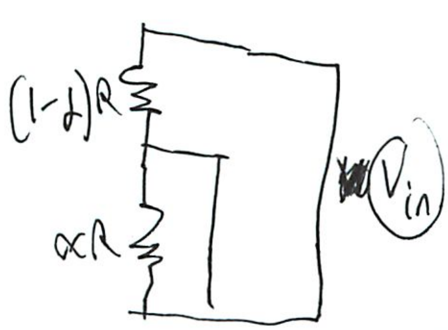


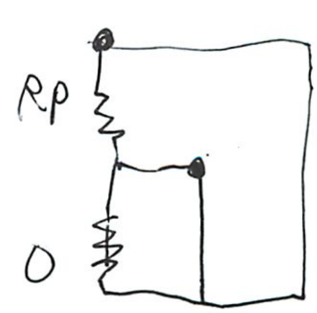
← ? or 0 resistance
so wire

maybe this is why ~~RP~~ is $(1-\alpha)$ on bottom

Where is voltage being applied ???

Look at Sh Lab 7 again



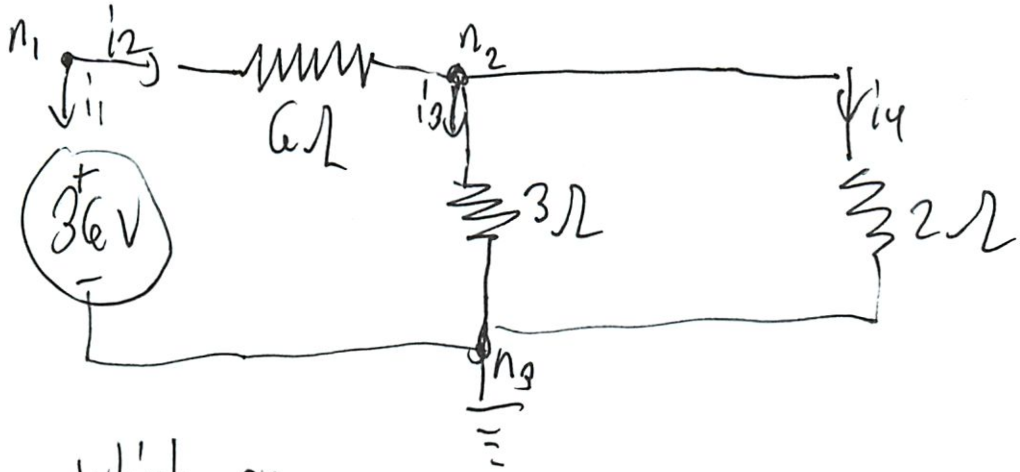


$V - V_0 = IRP$

(14) ~~Sk~~ Skip for now

4. ~~..~~ NVCL

- 10 checks



which eq are true

~~just regenerate here~~

~~$i_2 = -i_2$~~

go thrugh

1. $i_1 - i_2 = 0$

   $i_1 = i_2$ ⊗

2. $-i_1 - i_2 = 0$

   $-i_1 = i_2$ ✓ ✓

3. $-i_1 + i_2 = 0$

   $-i_1 = -i_2$ ⊗

15

4. $-i_3 + i_4 = 0$

$i_3 = i_4$ ✗

5. $-i_3 - i_4 = 0$

$-i_3 = i_4$ ✓ ✗ ← see below I don't say =, need i_1

6. $i_2 - i_3 - i_4 = 0$ ✓ $i_2 = i_3 + i_4$

7. $i_1 + i_3 + i_4 = 0$

$i_1 = -i_3 - i_4$

$-i_1 = i_3 + i_4$ ✓ ✗ idk why - all go out but algebra looks right

8. $-i_2 + i_3 - i_4 = 0$ ✗

9. $n_2 - n_1 = 6i_2$

↑ do ~~them~~ they mean V?

~~✓~~ ✗

10. $n_1 - n_2 = 6i_2$

actually this is right ✓ w/ signs

11. $n_2 - n_1 = i_2$ ✗

12. $n_1 - n_3 = 36$ ✗ ← what is voltage source anyway? just VC ✓ $n_1 - 0 = 36$ true

other way

13. $n_1 - n_3 = i_1$ ✗ ←

14. $-n_1 - n_3 = i_1$ ✗

(16)

$n_3 = 0$

15. $-n_2 = 3i_3$

(X) wrong sign

16. $n_2 - n_3 = i_3$

(X) resistance

17. $n_2 - n_3 = 3i_3$ (✓)

18. $n_2 - n_3 = i_4$

(X) resistance

19. $-n_2 = 2i_4$

(X) wrong singn

20. $n_2 - n_3 = 2i_4$

(✓)

21. $n_3 = 0$

(✓) but out of room!

22. $n_3 = -36$

(X)

first one right
- since order matters

either 3,4,5 is wrong

removed 5, added 21

---

new

17

1. 2
2. 6
3. 7
4. 10
5. 17
6. 20
7. 21

error b/w 7 and ~~10~~ 16
~~the~~ know 7 is wrong
likely move 10 up
and then add one ~~10~~-16

- try 10
12

Now Solve

$-i_1 = i_2$

$i_2 = i_3 + i_4$

$V_1 - V_2 = 6 i_2$

$V_1 - V_3 = 36$

$V_2 - V_3 = 3 i_3$

$V_2 - V_3 = 2 i_4$

$V_3 = 0$

(18)

$$\frac{V_1 - V_2}{6} = \frac{V_2 - V_3}{3} + \frac{V_2 - V_3}{2}$$

$V_1 = 36$

$$\frac{36 - V_2}{6} = \frac{V_2}{3} + \frac{V_2}{2}$$

$$\frac{36 - V_2}{6} = \frac{2V_2 + 3V_2}{6}$$

$$\frac{36 - V_2}{6} = \frac{5V_2}{6}$$

$6(36 - V_2) = 30V_2$

$216 - 6V_2 = 30V_2$

$216 = 36V_2$

$V_2 = \frac{216}{36}$

$V_2 = 6$

$i_3 = \frac{6-0}{3} = 2$

$i_4 = \frac{6-0}{2} = 3$

$i_2 = \frac{36-6}{6} = 5$

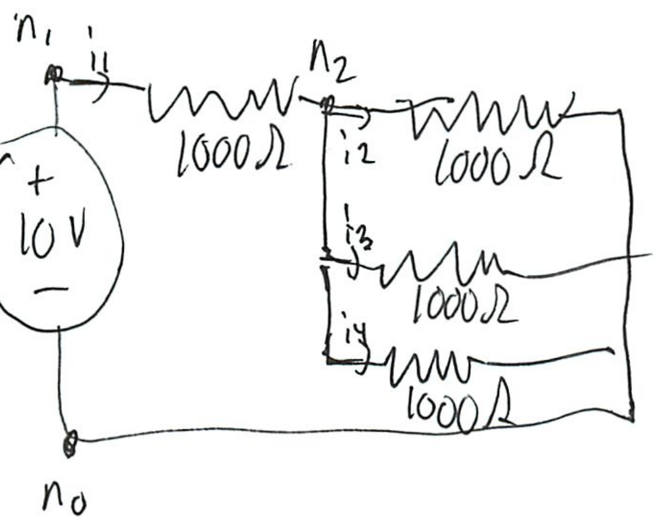
$i_1 = -5$

solved

4 pages

18

5. Resistor Network
- CView
- $1000\,\Omega$ resistors
- give Voltages
  - guess I could build and try ~~it~~ if needed



triple parallel

$n_1 = 10V$

$n_0 = 0V$

$V_{\cancel{1}} - V_2 = I_1\, 1000$

$V_2 - V_0 = I_2\, 1000$

$V_2 - V_0 = I_3\, 1000$

$V_2 - \cancel{V_0} = I_4\, 1000$

n_2/ ~~ma~~ $i_1 = i_2 + i_3 + i_4$

$$\frac{V_1 - V_2}{1000} = \frac{V_2}{1000} + \frac{V_2}{1000} + \frac{V_2}{1000}$$

20

$$\frac{10-V_2}{1000} = \frac{3V_2}{1000}$$

$$10{,}000 - 1000\,V_2 = 3000\,V_2$$

$$10{,}000 = 4000\,V_2$$

$$V_2 = \frac{10}{4} = 2.5\,V$$

H 48 ~~[illegible]~~ V_0 0V

C 48 ~~[illegible]~~ V_0 0V

H 45 V_2 2.5 V

C 45 V_c 0V

H 42 V_2 2.5V

C42 V_1 ·10V

H39 V_2 2.5V Opps 0 V

C 39 V_0 0V

6. Argopt

- now for some code
- argmin takes 2 arguments (proc, list)
- returns tuple (best Value, best Arg)

min value of input function — list value that gives that

So build argmin
- isn't there a better way than what they do w/ best?
- oh well
- back to writing basic python code
- what if they evaluate to same amt
  - update best arg or not?
  - make a list?
- need to ignore ~~ent~~ none
  - aka always run 1st time

- worked 1st try ✓

(22)

Part 2 arg opt

- 3 arguments (proc, list, comparison)
- same retrn
- can't pass standard comparison operators
- but operator lib
  - ↳ use that

Wow that was easy ✓

---

7. Float Range
   - generalize python range function
     - ↳ since only works for int
   - arguments
     - low ← low range
     - hi ← high
     - size step → size of sucessive steps in range

     (low and hi: float or int)

So start at low, increment stepSize till high

assume hi > low

stepsize > 0

test ✗

(2.3)

- <u>don't</u> include high value!

easy fix ✓

---

8. opt Over Line

Given a function $f(x)$ how can find a value x^* so that $f(x^*) \leq f(x)$ for all x?

If f differentiable
- take derivitive
- set $=0$
- solve for x

- Complex when multiple minima or multiple arguments
- or if max, min, abs ~~hard to do~~ can't differentiate
- So sample different values of x, evaluate f at each and return min
- arguments
  - objective: function w/ 1 argumen
  - x min, x max
  - num x steps
  - Comparison operator from ~~opt~~ operator lib
- Output (bestObjValue, bestX)

24

- used this before
- use FloatRange + argopt from before

---

make float range

feed into arg opt

try 1/3 ✓

2/3 ✗

they do # steps

and float range = step size

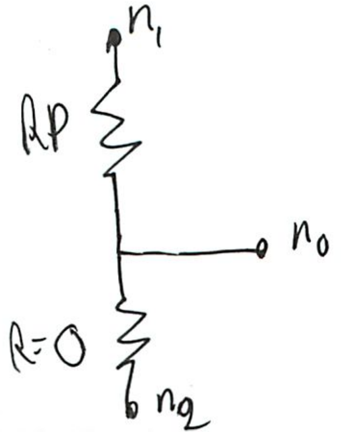
$$\frac{hi - lo}{\# steps} = step size$$

oh python float problem

- float both #

ok works now ✓

Now just pot qv left



$n_1 - n_0 = I RP$

$n_2 - n_0 = I \cdot 0 = 0$

$V_1 - V_0 = I RP$

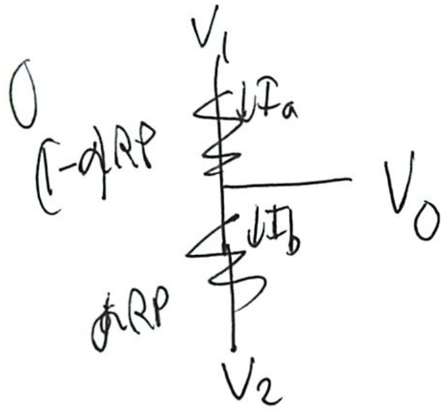
$V_1 - I RP = V_0$

Go for OH

25

Email response in 10 min

- all voltage related to ground



Calculate current from $V_2 \to V_1$ get v drop $\to$ that is V_0

~~the~~ for first to V_0 is ~~xx~~ connected to V1, V2 directly

$$V_1 - V_0 = I(1-\alpha)RP$$

$$V_0 - V_2 = I\alpha RP$$

$I_A = I_B$ ← i any current going at n out

- no load here
- just want voltage drop

always work w/ current MVCC!

$$\frac{V_1 - V_0}{(1-\alpha)RP} = \frac{V_0 - V_2}{\alpha RP} \quad \text{Find } V_0$$

$$(V_1 - V_0)\alpha RP = (V_0 - V_2)[(1-\alpha)RP]$$

(26)

$$\alpha RP V_1 - \alpha RP V_0 = (1-\alpha)RP\, V_0 - (1-\alpha) RP V_2$$

$$V_0\left((1-\alpha)RP + \alpha RP\right) = \alpha RP V_1 + (1-\alpha) RP V_2$$

$$V_0 RP = \alpha RP V_1 + 1-\alpha RP V_2$$

$$V_0 = \alpha V_1 + 1-\alpha V_2$$

when $\alpha = 0$

$V_0 = V_2$ ✓

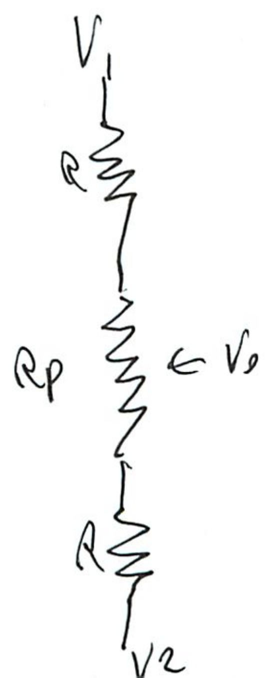
when $\alpha = 1$

$V_0 = V_1$ ✓

when $\alpha = .5$

$V_0 = .5 V_1 + .5 V_2$ ✓ wont!

4. Now this circuit



(27)

Now $R + \alpha RP$

$R + (1-\alpha)RP$

$$\frac{V_1 - V_0}{R + (1-\alpha)RP} = \frac{V_0 - V_2}{R + \alpha RP}$$

$$(V_1 - V_0)(R + \alpha RP) = (V_0 - V_2)(R + (1-\alpha)RP)$$

$$(R + \alpha RP)V_1 - (R + \alpha RP)V_0 = (R + (1-\alpha)RP)V_0 - (R + (1-\alpha)RP)V_2$$

$$V_0(R + (1-\alpha)RP + R + \alpha RP) = (R + \alpha RP)V_1 + (R + (1-\alpha)RP)V_2$$

$$V_0(2R + RP) = (R + \alpha RP)V_1 + (R + (1-\alpha)RP)V_2$$

$\alpha = 0$

$$V_0(2R + RP) = RV_1 + (R + RP)V_2$$

$$V_0 = \frac{RV_1 + (R + RP)V_2}{2R + RP}$$

X lots of illegal python results in answer

- Fixed 1 mistake in syntax V now

28

$\alpha = 1$

$$V_o(2R+RP) = (R+RP)V_1 + (R)V_2$$

$$V_o = \frac{(R+RP)V_1 + RV_2}{2R+RP}$$

$\alpha = .5$

↳ should be a very simple ans

$$\frac{(R+.5RP)V_1 + (R+.5RP)V_2}{2R+RP}$$

? should something simplify?

that worked

their answer

$$\frac{V_1+V_2}{2}$$

oh

$$\frac{\cancel{(R+.5RP)}(V_1+V_2)}{2\cancel{(R+.5RP)}}$$ I see

6.01: Introduction to EECS I

Op-Amps

Week 8 *October 26, 2010*

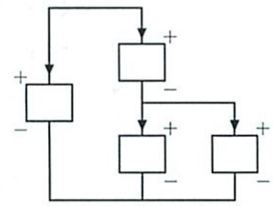
Reading: 7.8

Last Time: The Circuit Abstraction

Circuits represent systems as connections of elements

- through which currents (through variables) flow and
- across which voltages (across variables) develop.

"fluid"



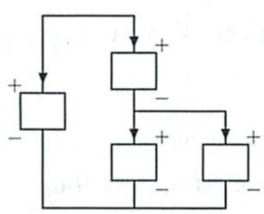
Last Time: Analyzing Circuits

differences

Circuits are analyzed by combining three types of equations.

- KVL: sum of voltages around any closed path is zero.
- KCL: sum of currents out of any node is zero.
- Element (constitutive) equations
  - — resistor: $V = IR$
  - — voltage source: $V = V_0$ (I = anything)
  - — current source: $I = I_0$ (V = anything)

in each elements



Last Time: Common Patterns

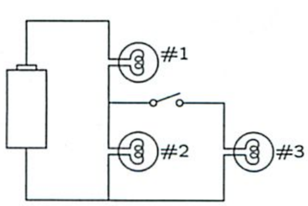
basic patterns

There are a number of **common patterns** that facilitate design and analysis:

- series resistances
- parallel resistances
- voltage dividers
- current dividers

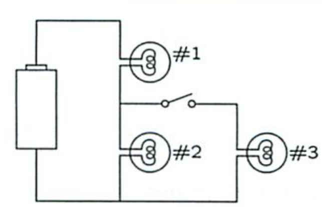
Caveat: Loading (the bad news)

Closing the switch changes the brightnesses of the left bulbs.



Remember last time's "Check Yourself"

What happens if we add third light bulb?



Closing the switch will make

1. bulb 1 brighter
2. bulb 2 dimmer
3. 1. and 2.
4. bulbs 1, 2, & 3 equally bright
5. none of the above

└ See seperate sheet

Want to get rid of loading effects to try + maintain PCAP

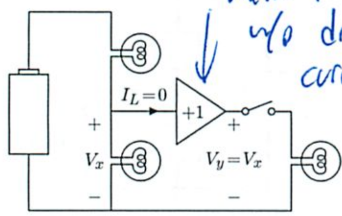
Buffering (the good news)

Effects of loading can be diminished or eliminated with a buffer.

An "ideal" buffer is an amplifier that

- senses the voltage at its input **without** drawing any current, and
- sets its output voltage equal to the measured input voltage.

Wants to find voltage w/o drawing any current



Today: Designing with Op-Amps

To analyze op-amps, we must introduce a new kind of element: a dependent source.

A dependent source generates a voltage or current whose value depends on another voltage or current.

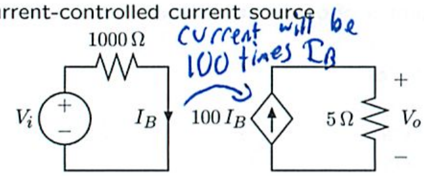
design circuits that are independent

this one a voltage controlled voltage source

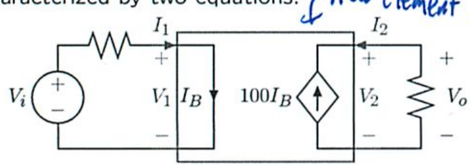
Dependent Sources

Example: current-controlled current source

current will be 100 times I_B



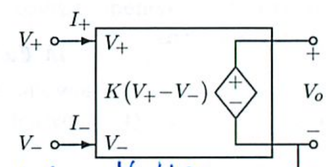
Formally, dependent sources involve two currents and two voltages, and are characterized by two equations. new element - 4 connections



Here $V_1 = 0$ and $I_2 = -100\,I_1$.

Op-Amp

An op-amp can be represented by a voltage-controlled voltage source.



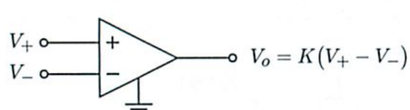
takes V difference + amplifys it

$I_+ = I_- = 0$ and $V_o = K(V_+ - V_-)$ where K is large ($\approx 10^5$). or bigger

No current flows through input terminals or between input and output. just generating voltage difference

Op-Amp

An op-amp can be represented by a voltage-controlled voltage source.

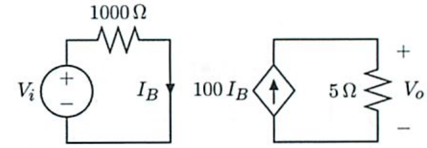


$I_+ = I_- = 0$ and $V_o = K(V_+ - V_-)$ where K is large ($\approx 10^5$).

What you connect to the output terminal does not affect the voltage at the input terminals. buffer no current flows

Check Yourself

Find $\frac{V_o}{V_i}$.



1. 500
2. $\frac{1}{20}$
3. 1
4. $\frac{1}{2}$
5. none of above

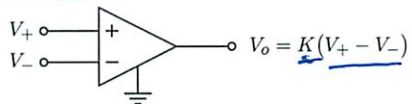
got separation so can do analysis

$I_B = \frac{V_i}{1000\Omega}$

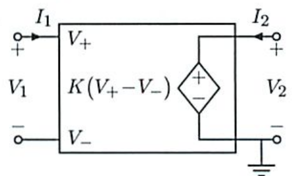
$V_o = 100 \cdot I_B \cdot 5\Omega = \frac{5V_i}{10} = \frac{1}{2} V_i$

Op-Amp

An op-amp (operational amplifier) can be represented by a voltage-controlled voltage source.



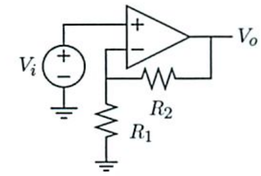
A voltage-controlled voltage source is a two-port.



$I_1 = I_+ = I_- = 0$ and $V_2 = V_o = K(V_+ - V_-)$ where K is large ($\approx 10^5$).

Op-Amp: Analysis

Example: find $\frac{V_o}{V_i}$ for the following circuit.



$V_+ = V_i$

$V_- = \frac{R_1}{R_1 + R_2} V_o$ Voltage divider $V_o - 0$ through

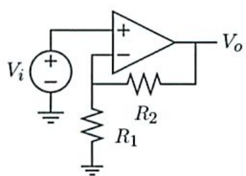
$V_o = K(V_+ - V_-) = K(V_i - \frac{R_1}{R_1 + R_2} V_o)$

algebra $\frac{V_o}{V_i} = \frac{K}{1 + \frac{KR_1}{R_1+R_2}} = \frac{K(R_1 + R_2)}{R_1 + R_2 + KR_1} \approx \frac{R_1 + R_2}{R_1}$ (if K is large)

k is really big - k terms dwarf other terms

Non-inverting Amplifier

For large K, this circuit implements a non-inverting amplifier.



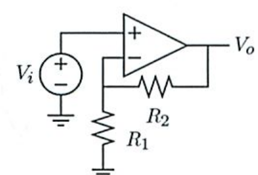
$$\frac{V_o}{V_i} = \frac{R_1 + R_2}{R_1}$$

Remember, however, that op amp does this without drawing any current into its input ports, and hence does not affect parts of the circuit on the input side of the op amp.

Creates range of amplification

Check Yourself

For which value(s) of R_1 and/or R_2 is $V_o = V_i$.



1. $R_1 \to \infty$
2. $R_2 = 0$
3. $R_1 \to \infty$ and $R_2 = 0$
4. all of the above
5. none of the above

$\frac{V_o}{V_i} = \frac{R_1 + R_2}{R_1} = 1$

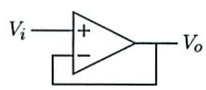
and here

make k really big

The "Ideal" Op-Amp

As $K \to \infty$, the difference between V_+ and V_- goes to zero.

Example:



$V_o = K(V_+ - V_-) = K(V_i - V_o)$

$\boxed{V_o = \frac{K}{1+K} V_i}$ is a Voltage drop

$V_+ - V_- = V_i - V_o = V_i - \frac{K}{1+K} V_i = \frac{1}{1+K} V_i = \frac{1}{K} V_o$

$\lim_{K\to\infty} (V_+ - V_-) = 0$ but want it to be 0 so

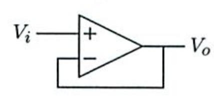
If the difference between $V+$ and V_- did not go to zero as $K \to \infty$ then $V_o = K(V_+ - V_-)$ could not be finite.

Under this model the voltage difference is 0

The "Ideal" Op-Amp

The approximation that $V_+ = V_-$ is referred to as the "ideal" op-amp approximation. It greatly simplifies analysis.

Example.

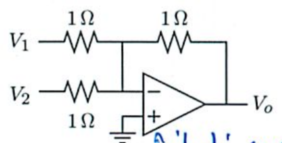


If $V_+ = V_-$ then $V_o = V_i$!

here

Check Yourself 2

Determine the output of the following circuit.



ideal; same Voltage

1. $V_o = V_1 + V_2$
2. $V_o = V_1 - V_2$
3. $V_o = -V_1 - V_2$
4. $V_o = -V_1 + V_2$
5. none of the above

$V_- = V_+ = 0$

KCL $\frac{V_1-0}{1} + \frac{V_2-0}{1} + \frac{V_o-0}{1} = 0$

$V_o = -V_1 - V_2$

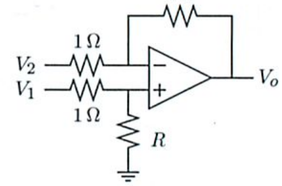
had this part

did not think to do this next step

? "inverting adder"

Check Yourself 3

Determine R so that $V_o = 2\,(V_1 - V_2)$.

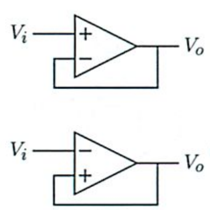


1. $R = 0$
2. $R = 1$
3. $R = 2$
4. $R \to \infty$
5. none of the above

Seperate page

Paradox

The ideal op-amp approximation implies that both of these circuits function identically.



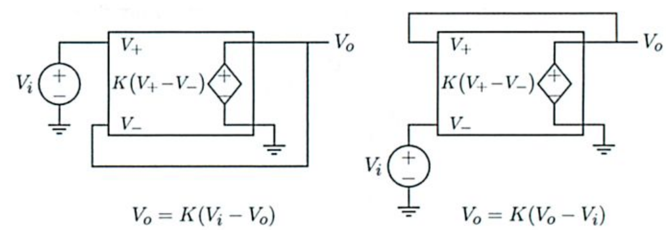
are these the same

No!

- one works
- one = smoke

$V_+ = V_- \quad \rightarrow \quad V_o = V_i$!

Paradox



$$V_o = K(V_i - V_o)$$
$$(1+K)V_o = KV_i$$
$$\frac{V_o}{V_i} = \frac{K}{1+K} \approx 1$$

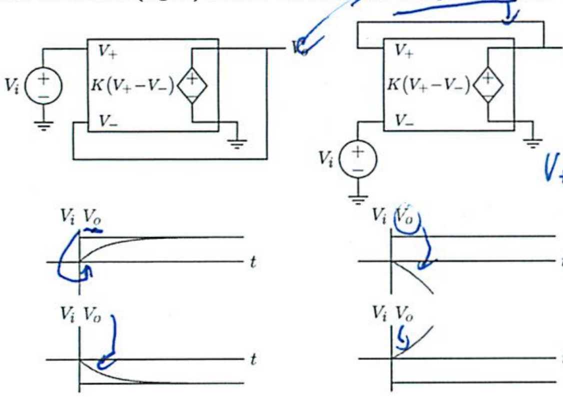
$$V_o = K(V_o - V_i)$$
$$(1-K)V_o = -KV_i$$
$$\frac{V_o}{V_i} = \frac{-K}{1-K} \approx 1$$

These circuits seem to give similar responses if K is large.
Something is WRONG!

in limit same

Positive and Negative Feedback

Negative feedback (left) drives the output **toward** the input.
Positive feedback (right) drives the output **away from** the input.



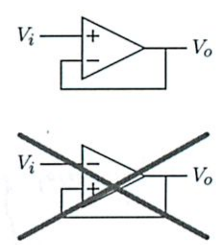
$V_+ - V_- = \ominus$
negative
drive it
down

but initally

= smoke

Paradox Resolved

Although both circuits have solutions with $V_o = V_i$ (large K), only the first is stable to changes in V_i.



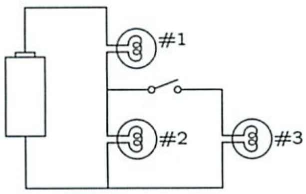
both work in steady state but not in transition to them

Moral: feedback to negative input of op-amp

Buffers

Op-amps can be used to partition a circuit into conceptually separate pieces.

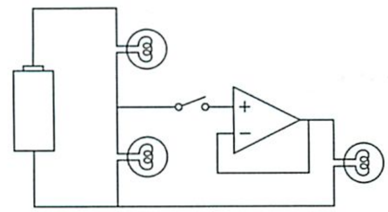
Recall that closing the switch adds a third light bulb, and also alters the brightness of the original two bulbs.



We can use an op-amp to eliminate the loading effects of the third bulb.

Check Yourself

What will happen when the switch is closed?

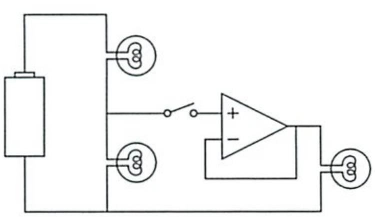


1. top bulb is brightest 2. right bulb is brightest
1. right bulb is dimmest 4. all 3 bulbs equally bright
5. none of the above

Power Rails

The output of an op-amp can provide power to a circuit.

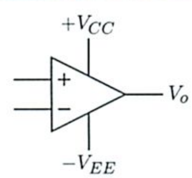
Example: The battery provides the power to illuminate the left bulbs. Power for the third bulb comes from the op-amp.



But where does the op-amp get the power?

Power Rails

Op-amps derive power from connections to a power supply.



The power supply limits the output of an op-amp.

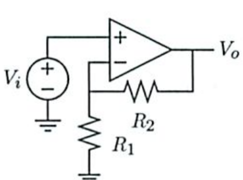
Typically, the output voltage of an op-amp is constrained so that

$-V_{EE} < V_o < V_{CC}$. usually are 0, 10V
notice other

-limit due to 2 rails

-has to - powering down the chain

Non-inverting amplifier



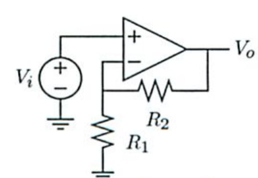
get set of common patterns

$$V_+ = V_i$$
$$V_- = V_i$$
$$\frac{V_o - V_i}{R_2} = \frac{V_i - 0}{R_1}$$
$$R_1 V_o - R_1 V_i = R_2 V_i$$
$$V_o = V_i \frac{R_1 + R_2}{R_1}$$

non-inverting amplifer

Check Yourself

For which value(s) of R_1 and/or R_2 is $V_o = 4V_i$?

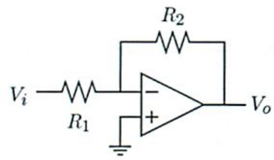


anything so $4 = \frac{R_1 + R_2}{R_1}$

$R_1 = 300\,\Omega$
$R_2 = 100\,\Omega$ (not 1, 3, hard to deal w/)

$V_o = .5V_i$? -one of them has to be -
-not possible!

Inverting Amplifier



the ground

$$V_- = V_+ = 0$$

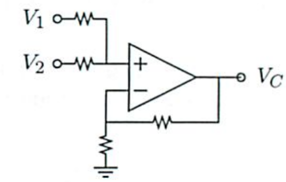
$$\frac{V_i}{R_1} + \frac{V_o}{R_2} = 0$$

$$V_o = -\frac{R_2}{R_1} V_i$$

- Changes sign (relative to voltage V_+)
- Can decrease magnitude
- Cannot exceed power-supply rails

Op-amp as a summer

Assume all resistances are R.



$$\frac{V_C - V_-}{R} = \frac{V_- - 0}{R}$$

$$\frac{V_1 - V_+}{R} = \frac{V_+ - V_2}{R}$$

So:

$$V_- = \frac{V_C}{2}$$

$$V_+ = \frac{V_1 + V_2}{2}$$

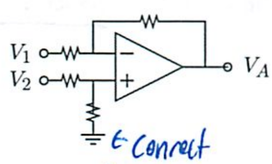
Using the ideal op-amp model:

$$V_+ = V_-$$

$$V_C = V_1 + V_2$$

Op-amp as a subtractor

Assume all resistances are R.



← connect to whatever you want to center it around

$$\frac{V_1 - V_-}{R} = \frac{V_- - V_A}{R}$$

$$\frac{V_2 - V_+}{R} = \frac{V_+ - 0}{R}$$

So:

$$V_- = \frac{V_1 + V_A}{2}$$

$$V_+ = \frac{V_2}{2}$$

Using the ideal op-amp model:

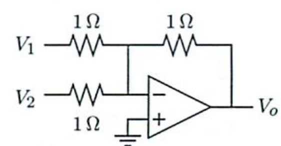
$$V_+ = V_-$$

$$V_2 = V_1 + V_A$$

$$V_A = V_2 - V_1$$

Check Yourself

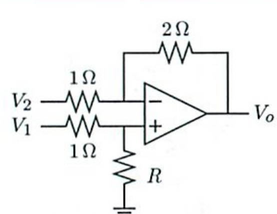
Determine the output of the following circuit.



1. $V_o = V_1 + V_2$
2. $V_o = V_1 - V_2$
3. $V_o = -V_1 - V_2$
4. $V_o = -V_1 + V_2$
5. none of the above

Check Yourself

Determine R so that $V_o = 2\,(V_1 - V_2)$.



1. $R = 0$
2. $R = 1$
3. $R = 2$
4. $R \to \infty$
5. none of the above

This Week

Software lab: Software to solve circuits

Design lab: Controlling a motor using resistors and op-amps

HW 2: Due before design lab

HW 3: Handed out in design lab; due the following week.

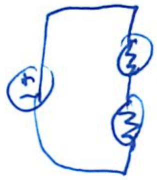
①

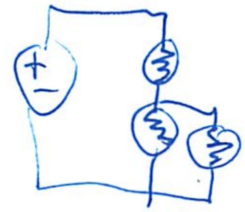
Lecture 8 Additional

10/26

Last time's check yourself

$$V_o = \frac{R}{R+R} V_i = \frac{1}{2} V_i \qquad V_o = \frac{.5R}{R+.5R} V_i = \frac{1}{3} V_i$$





parallel combo

$$R_{eq} = \frac{RR}{R+R} = \frac{R}{2}$$

Now just using tools
- not doing it out in great detail

Extra light

- top bulb gets $\frac{2}{3} V_i$ → <u>brighter</u>
- bottom bulb gets $\frac{1}{3} V_i$ → <u>dimmer</u>
- new bulb gets $\frac{1}{3} V_i$

② Check Yourself 2

Ideal Op amp

$$V_- = V_+ = 0$$

KCL at Vo

$$\frac{V_1 - 0}{1} + \frac{V_2 - 0}{1} + \frac{V_0 - 0}{1} = 0$$

$$V_0 = -V_1 - V_2$$

---

negitive voltage?
- ground is arbitrary value
- reduces voltage instead of adding it

Check Yourself 3

Want R so $V_0 = 2(V_i - V_2)$

V_+ = voltage divider
2 resistances in series

$$= \frac{R}{R+1} V_1$$

take advantage of isolation - tools

$$\frac{V_2 - V_-}{1} + \frac{V_0 - V_+}{2} = 0$$

net flow in must be 0

$$V_2 = \frac{3}{2} V_- + \frac{1}{2} V_0 = 0$$

(3)

$$V_- = \frac{2}{3} V_2 + \frac{1}{3} V_0$$

Ideal Opt Amp

$$V_+ = V_-$$

$$\frac{R}{1+R} V_1 = \frac{2}{3} V_2 + \frac{1}{3} V_0$$

$$V_0 = \frac{3R}{1+R} V_1 - 2V_2$$

$$\frac{3R}{1+R} = 2$$

$$R = 2\,\Omega$$

Opt - Amps

- isolate parts of circuits
  - modular design
- amplify voltages
- perform math operations
  - add. + sub.

- draws no current
- $V_+ - V_0 = 0$ for ideal opt-amp

$(PCAP)^4$ abstraction at 1 level become primitives at next

(4)

Opt amp very complex

- Resistors + transitors
- won't talk about
- take more classes after this

SW Lab 8: Describing Circuits

- **Using a lab laptop or desktop machine**: Log in using your Athena user name and password; in the terminal window, type `athrun 6.01 update`.
- **Using your own laptop**: Download the zip file for software lab 8 from the *Calendar* tab of the course web page.

You can work in the file `circSkeleton.py` in `lab8/swLab`.

The software lab for this week is to develop a method for describing circuits at a high level of abstraction, and convert that description into linear equations.

1 Specifying and solving linear equations

Consider the problem of finding values for x and y that satisfy the two equations:

$$5x - 2y = 3$$

and

$$3x + 4y = 33.$$

You would probably approach this with the *substitution method*, in which you solve the first equation for x, getting $x = \frac{2}{5}y + \frac{3}{5}$, and then substituting that into the second equation, getting

$$\frac{6}{5}y + \frac{9}{5} + 4y = 33 \quad .$$

Then, solving for y, we find that $y = 6$. And substituting $y = 6$ into our expression for x tells us that $x = 3$.

That was a relatively simple system of two equations in two unknowns. More generally, we'll be interested in solving systems of n equations in n unknowns, sometimes for very large values of n. This is the sort of problem at which humans are typically bad (how many lines of algebra can you do before making a sign error, or multiplying 2 by 3 and getting 5?), but that we can get a computer to do for us, cheerfully and reliably.

We could try to write a computer program to perform the substitution method, but it involves a lot of manipulation of intermediate algebraic expressions, and can be computationally inefficient. There is another method, called *Gaussian elimination*[1], which is efficient and relatively easy to implement on a computer; we'll use a standard implementation of it from the Python `numpy` library.

[1] The only thing this method has to do with Gaussian probability distributions is that Carl Friedrich Gauss worked on both.

The numpy implementation of Gaussian elimination requires, as input, a matrix of coefficients for every variable in every equation. But when we're describing the linear constraints for a circuit, each equation only mentions a few of the variables, which means we'd need a big matrix filled with lots of zeros. We'll use a simpler notation, which can be converted into a coefficient matrix for numpy, thus enabling us to leverage numpy's Gaussian elimination method.

We will use classes in the 6.01 software module `le` to represent sets of equations[2].

An **equation** is represented with an instance of class `le.Equation`, which takes, at initialization time, three arguments:

- `coeffs`: a list of numerical coefficients for the variables mentioned in the linear equation
- `variableNames`: a list of strings, naming the variables in the equation; the variable names have to be listed in the same order as the coefficients
- `constant`: the numerical constant in the equation, with the sign chosen appropriately for the situation in which the constant is the only term on one side of the equality.

So, for example, we could represent the equation $-3x + 9.2z - 4 = 0$ as

```
le.Equation([-3, 9.2], ['x', 'z'], 4)
```

Note how the value of -4 changed since we need to move the constant to the right hand side of the equality.

There are several other ways to represent the same equation, including these two:

```
le.Equation([9.2, -3], ['z', 'x'], 4)
le.Equation([3, -9.2], ['x', 'z'], -4)
```

We can represent a **set of equations** using an instance of the class `le.EquationSet`. It takes no parameters at initialization time, but supports the following methods:

- `addEquation(self, eqn)`: adds eqn, which must be an instance of `le.Equation`, to the set of equations
- `solve(self)`: computes and returns the solution to the set of equations or generates an error; the solution is an instance of the class `le.Solution`

A **solution** to a set of equations is represented using an instance of the class `le.Solution`. You won't need to construct a solution (you will get a solution by calling the `solve` method of an `EquationSet`); but you can look up the value of a variable in a solution with the method

`translate(self, name)`: `name` is a string naming a variable that occured in the equation set that was solved; this method returns the value of that variable in the solution.

Using these classes, you can describe, then solve, our simple example like this:

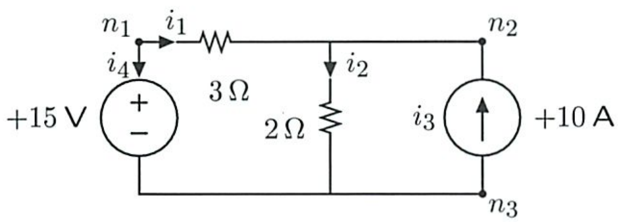
---

[2] Read the online documentation for the module `le` (under **Software Documentation** in the **Reference Material** tab of the 6.01 web page) for more details.

```
>>> small = le.EquationSet()
>>> small.addEquation(le.Equation([5, -2], ['x', 'y'], 3))
>>> small.addEquation(le.Equation([3, 4], ['x', 'y'], 33))
>>> sol = small.solve()
>>> sol.translate('x')
3.0
>>> sol.translate('y')
6.0
```

2 Circuit equations

Using the NVCC method, we can write a set of equations that characterizes the following circuit (see Section 7.6 in the Course Notes).



We will use 'n1', 'n2', etc., as the names of the node voltages and 'i1', 'i2', etc., as the names of the component currents. We start by creating an equation set:

```
ckt = le.EquationSet()
```

Now, we have an equation for each of the four constituents:

$$n_1 - n_3 = 15$$
$$n_1 - n_2 = 3i_1$$
$$n_2 - n_3 = 2i_2$$
$$i_3 = 10$$

which we add to the equation set as follows:

```
ckt.addEquation(le.Equation([1.0, -1.0], ['n1', 'n3'], 15.0))
ckt.addEquation(le.Equation([1.0, -1.0, -3], ['n1', 'n2', 'i1'], 0.0))
ckt.addEquation(le.Equation([1.0, -1.0, -2], ['n2', 'n3', 'i2'], 0.0))
ckt.addEquation(le.Equation([1.0], ['i3'], 10.0))
```

Next, we need to specify an equation that sets the voltage of the ground node to be zero. We have chosen, somewhat arbitrarily, n_3 to be ground; any other choice would have been fine.

$$n_3 = 0$$

We add it to the equation set with

```
ckt.addEquation(le.Equation([1.0],['n3'], 0.0))
```

Finally, we specify KCL equations for the remaining nodes

$$-i_4 - i_1 = 0$$
$$i_1 - i_2 + i_3 = 0$$

which are added to the equation set with

```
ckt.addEquation(le.Equation([-1.0, -1.0], ['i4', 'i1'], 0.0))
ckt.addEquation(le.Equation([1.0, -1.0, 1.0], ['i1', 'i2', 'i3'], 0.0))
```

Now, we can solve the circuit, with the following result:

```
>>> ckt.solve()
i1 = -1.0
i2 = 9.0
i3 = 10.0
i4 = 1.0
n1 = 15.0
n2 = 18.0
n3 = 0.0
```

This is convenient, because it saves us from our own algebra errors. Unfortunately, it can be hard to remember to construct exactly the right set of equations (do I have one for each constituent component? do I have all of the current equations for each node? did I use the right names and coefficients for the currents?).

In this lab, we will develop software that allows the following, much more compact, specification.

```
c = circ.Circuit([circ.VSrc(15, 'n1', 'n3'),
                  circ.Resistor(3, 'n1', 'n2'),
                  circ.Resistor(2, 'n2', 'n3'),
                  circ.ISrc(10, 'n3', 'n2') ])
```

Here, the idea is to create classes whose instances represent components, together with identifiers for their connectors. So long as we consistently use the same identifier to indicate how the components are connected (e.g., `n1` connects a voltage source to a resistor), we can construct a `Circuit` class with a method that will do the work for use. Specifically, the major simplification is that we don't have to mention the currents when specifying the components and we don't have to specify the KCL equations at all. When we call `c.solve('n3')`, a set of equations is automatically constructed, with node `'n3'` as ground, and then solved (by automatically using the `le` method just as we did above):

```
>>> c.solve('n3')
Solving equations
****************
+n1-n3 = 15
+n1-n2-3*i_n1->n2_14 = 0
```

```
+n2-n3-2*i_n2->n3_15 = 0
+i_n3->n2_16 = 10
+i_n1->n3_13+i_n1->n2_14 = 0.0
-i_n1->n2_14+i_n2->n3_15-i_n3->n2_16 = 0.0
+n3 = 0
****************
i_n1->n2_14 = -1.0
i_n1->n3_13 = 1.0
i_n2->n3_15 = 9.0
i_n3->n2_16 = 10.0
n1 = 15.0
n2 = 18.0
n3 = 0.0
```

The currents are automatically given names. So, `i_n1->n2_14` is a current that flows between nodes `n1` and `n2`.[3] So, we can see that the first four equations (listed above between the ******* borders) describe the components, the next two are KCL equations, and the last specifies the ground. The solution (listed after the ******* borders) tells us the currents in the connectors between components and the voltages at the nodes.

Wk.8.1.1	Write the `EquationSet` and write the more abstract representation for the circuit shown in the tutor problem.

3 Overview of the `Circuit` class

A `Circuit` class instance is created with a list of component instances, as shown above. The key method is the `solve` method, which constructs an equation set from the components and solves it. We will define each component type as a class that can construct the relevant equation for that type of instance (see below).

However, the `solve` method will also need to construct the KCL equations at every node (except the ground). So, we will need to know which components are connected to which nodes. In our implementation, we use the `NodeToCurrents` class to keep track of which component current enters (or leaves) each node. Each component has a method that provides this information (see below).

Every type of component, for example voltage source, resistor, and op amp, will be a subclass of the `Component` class. Every subclass of the `Component` class must supply two methods:

- `getEquation`, which returns an instance of `le.Equation` that constrains the voltage across the terminals of the component, and
- `getCurrents`, which returns the list of currents that this component adds to the nodes to which it is connected. Each current is represented as a tuple `(i, node, sign)`, where `i` is the name

[3] We append an additional unique number (in this case 14) to the name, because, if there are multiple components in parallel between `n1` and `n2`, we need to be able to speak of several different currents flowing between those nodes.

of a current variable, node is the name of a node, and sign is the sign of that current at that node, either +1 or -1.

All two-input components have the same pattern of currents: they make a new current variable when created, and then assert that it flows into their node n1 and out of their node n2. So, we have implemented this pattern as the default getCurrents method in the Component class.

```
class Component:
    def getCurrents(self):
        return [[self.current, self.n1, +1],
                [self.current, self.n2, -1]]
```

Here is how the Resistor component is implemented.

```
class Resistor(Component):
    def __init__(self, r, n1, n2):
        self.current = util.gensym('i_'+n1+'->'+n2)
        self.n1 = n1
        self.n2 = n2
        self.r = r
    def getEquation(self):
        # your code here
```

The util.gensym procedure takes a string as an argument and returns a string which is the argument with a unique integer appended to it.

Wk.8.1.2	This problem will guide you through implementing the getEquation method for the Resistor class.

Wk.8.1.3	This problem will guide you through implementing the OpAmp class as a voltage-controlled voltage source; see Section 7.8.1 of the Course Notes.

4 Implementing the Circuit class

The Circuit class has two methods;

```
class Circuit:
    def __init__(self, components):
        self.components = components

    def solve(self, gnd):
        es = le.EquationSet()
        n2c = NodeToCurrents()
```

```
for c in self.components:
    es.addEquation(c.getEquation())
    n2c.addCurrents(c.getCurrents())
es.addEquations(n2c.getKCLEquations(gnd))
return es.solve()
```

A circuit is just a list of instances of the `Component` class. When we ask the circuit to solve itself, we provide the name of a node, passed in as parameter gnd, which will be the ground node and have voltage 0; then the `solve` method:

1. Makes a new empty equation set `es`.
2. Makes a new instance, `n2c`, of the `NodeToCurrents` class. This class keeps track of which currents are flowing into and out of each node.
3. For each component, adds the equation that describes the relationship between voltage and current that the component induces, and it adds the currents to the appropriate nodes in `NodeToCurrents`.
4. Adds the KCL equations that result from the node-current relationships stored in `n2c`, and one that sets the node named by the gnd variable to have voltage 0.
5. Solves the equations.

You can read about the `NodeToCurrents` class and its methods in the software documentation.

Wk.8.1.4	Implement the `NodeToCurrents` class. Please debug your code in the `circSkeleton.py` file and then paste it into the Tutor.

swLab 8

10/26

circSkeleton.py
building a circuit solver

solving linear eq
Computers are good at
humans do substitution + make a lot of mistakes
Computers like Gaussian eliminations
- numpy does
- w/ matrix of equations
  - (oh I remember this - inverse matricies?)

We will write in simpler notation
let computer convert to numpy's Guassian method

Use module le. to represent linear equations
- coeffs
- variable names
- constant

$-3x + 9.2z - 4 = 0$

= le.Equation([-3, 9.2], ['x', 'z'], 4)

= le.Equation([3, -9.2], ['x', 'z'], -4)

= " " ([9.2, -3] ['z', 'x', 4)

Set of equations
le.EquationSet
- no init params

(2)

Methods

add Equation(self, eqn)

adds eq to the set

Solve (self)

Solves the eq in the set

le.Solution outputs

translate(self, name)

name of variable in eq set

returns value of variable in solution

Se

```
Set = le. EquationSet()
Set. add Equation(le.Equation([5,-2],['x' 'y'],3))
Set. add Equation(  "    "   ([3,4],  "   "  , 33))
Sol = set. Solve()
Sol. translate('x')
   3.0
Sol. translate('y')
   6.0
```

③

So can do for circuits w/ NVCC method

- I get how to write eq
- just add them to set
- ~~At~~ tricky to make sure have all equations, variables
- in this lab will use a much more compact notation

```
c = circ.Circuit([circ.VSrc(15, 'n1', 'n3'),
                  circ.Resistor(3, 'n1', 'n2'),
                  circ.Resistor(2, 'n2', 'n3'),
                  circ.ISrc(10, 'n3', 'n2')])
```

- So want classes which represent components
- make sure to pay attention to n1, n2, etc
- when call c.solve('n3') it treates n3 as ground
- currents give names automatically

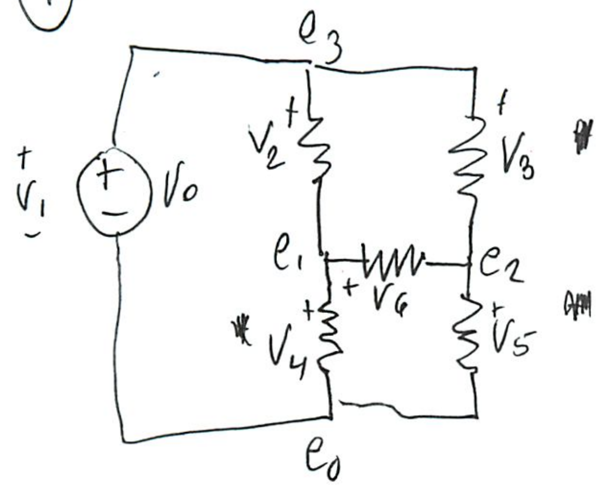
i_n1→n2_14

unique # auto. appended

8.1.1 Write EquationSet and Circuit ~~class~~ just translate circuit to the values

↓

(4)



Current each component

$+ \to -$

i_k corresponds to V_k

all R except 4 $= 100\,\Omega$ $\quad V_0 = 10\,V$

$R_4 = 10\,\Omega$ $\quad e_0 =$ ground

arrow $(+)\to(-)$

~~$i_1 = i_2 + i_3$~~

$-i_1 - i_2 - i_3 = 0 \to -i_1 = i_2 + i_3$

$i_2 - i_6 - i_4 = 0 \to i_2 = i_4 + i_6$

$i_3 + i_6 - i_5 = 0 \to i_5 = i_3 + i_6$

$i_4 + i_5 - i_1 = 0 \quad -i_1 = i_4 + i_5$

$V = IR$

alway direction $(+)\to(-)$ *

$e_3 = 10V + e_0 \quad e_3 - e_0 = 10 \quad e_3 = 10$

$e_0 - e_3 = -10\,V$

$e_2 - e_0 = 100\,\Omega\, i_5$

$e_3 - e_1 = 100\,\Omega\, i_2$

$e_0 = 0$

$e_1 - e_0 = 10\,\Omega\, i_4$

$e_3 - e_2 = 100\,\Omega\, i_3$

$e_1 - e_2 = 100\,\Omega\, i_6$

(5)

So

$$\cancel{e_0 - e_3 - 10 = 0}$$

1 2

$$e_0 - e_3 = 10$$

1 -1 constant

$$e_3 - e_1 - 100 i_2 = 0$$

1 -1 -100 constent

do I need the current eq?

- yes w/ just voltage have 9 variables 7 equ

- need to ~~pick~~ pick the unique 2 current eq
- well adding one of them added another variable
- so put them all in
- singular matrix
- oh I had [0], [e0], 0

is $0 e0 = 0$

want $1 e0 = 0$

[1], [e0], 0

(6)

Also needed to comment out last circuit eq

$i_1 = i_4 + i_5$ ← redundant info

Then solved out, check in tutor

$e_0 =$ (X) slightly off in values

$e_1 =$ What was wrong?

$e_2 =$ Entering eq correctly (had i_1 instead of i_2) ← big issue

$e_3 =$ Sign on i_1 - but worked out the same

Using .0 for floats?

Now w/ Circuit class

Vsource (10, n_3, n_0)
(+) → (−)

Resistor (100, n_3, n_1)
Resistor (10, n_1, n_0)
Resistor (100, n_3, n_2)
Resistor (100, n_2, n_0)
Resistor (100, n_1, n_2)

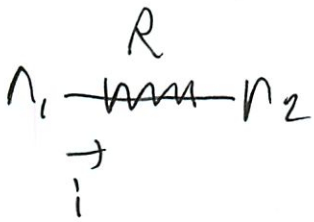
order does not matter

but enter w/ e_s for some reason

(✓) woot - So much easier!

7

8.1.2 Modeling Resistors



what are coefficents

first non-zero coefficent should be positive

$V = IR$

$n_1 - n_2 = ir$

$n_1 - n_2 - ir = 0$

1   -1   r   0

✓   ✓   ✗   ✓

-r opps, had written that did not type in

✓

Part ~~2 Now express as~~ an le.equation

R=1000

use n1, n2, i

le.equation([1, -1, -1000], [n1, n2, i], 0)

↑ in quotes of course

✓

8

Part 3 Get Equation

Class Resistor

- oh uses those values

- ? so what should get Eq return?

  - should return le.Equation w/ values

le.Equation([n_1, $-n_2$, $-r$], [n_1, n_2, i], 0)

self. self. self.

bad operand type for unary

- is it the names? - yeah says name

- ? so return

le.Equation [1, -1, -self.r] ...

- oh supposed to give equ for resistor!

  - well include return

where is there x?

Use their names

return le.Equation([1, -1, -self.r], [self.n1, self.n2, self.current], 0)

(9)

Since they call ~~it~~ e1
names e2

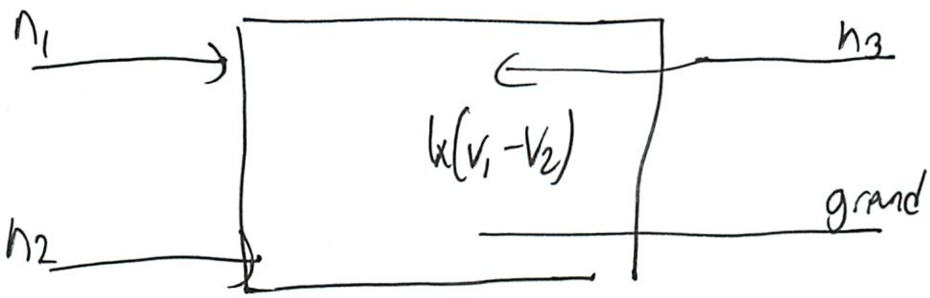
i_e1 → e2_51

have the names pass through

Done for day

8.1.3 Modeling Op-Amps

10/27
at home



gain = k_1 $\qquad$ n_3 = output

n_1 = positive

n_2 = negitive

what are coefficents

$\_n_1 + \_n_2 + \_n_3 = 0$

hmm don't fully understand opt amps - look at lecture 8 notes

$V_o = k(V_+ - V_-)$

(10)

$V_3 = k V_1 - k V_2$

$k V_1 - k V_2 - V_3 = 0$

part 2 opt amp currents

- oh good I was wondering about currents

$i_+ = i_- = 0$

$i_1 = i_2 = 0$

so only include in KCL not opt-amp eq

Not include in any

i_3 = must be related to V_3

$V = IR$

$I = \frac{V}{R}$

greater than 0

Only include in KCL

- Since opt amp just works w/ voltages

Unconstrained by opt amp

Part 3 implement opt amp class

- just get eq function, right?

11

return le.Equation([self.k, -self.k, -1],
[self.nPlus, self.nMinus, self.nOut], 0)

- current not used
  - like said earlier

- so all correct except forgot to type in -1
  on the n in nOut

works now

Now back to law

Circuit class instance
- created w/ list of component instances from above
- each component type is a class
- solve method generates KCL eq at every node
- so need to keep track of which components are at which nodes
- use NodeToCurrents class to track which component current enters or leaves node
- each component must implement the 2 values
  - getEquation - w/ le Equation that constrains voltage across terminals (what I wrote)

(12)

get Currents () - returns list of current this component adds to nodes
represented as tuple (i, node, sign)

value name +1, -1

all 2 input components make a new current variable and then assert

- so implement in base Component class

```
class Resistor(Component):
    def __init__(self, r, n1, n2):
        self.current = util gen sym(i_+n1 + -> n2)
        self.n1 = n1
        self.n2 = n2
        self.r = r

    def get Eq
```

Oh opps I ~~am~~ skipped ahead + did that already

(13)

Implement circuits class

```
class Circuit:
    def __init__(self, components):
        self.components = components

    def solve(self, gnd)
        es = le.EquationSet()
        n2c = NodetoCurrent()
        for c in self.components
            es.addEquation(c.getEquation())
            n2c.addCurrents(c.getCurrents())
        es.addEquations(n2c.getKCLEquations(gnd))
        return es.solve
```

What it does

1. Makes an empty es
2. Makes a Node to Current class to track which currents flowing in + out of each node
3. From each Component take voltage eq add current w/ Node to Currents
4. Add these node - current relationships
5. Solve

(14)

Implement ~~Node~~ Node To Currents

- good, wanted extra current practice
- oh whole class
    - init
    - add Current
    - add Currents
    - get KCL eq
- more docs in 6.01 ~~doc~~ software doc.
- so need to get picture of what it does
- uses each components get currents

- so like current matrix; switch board
- where know in → out
- so for each node $\sum in - \sum out = 0$
- and it outputs those full les

[what do we want coming out?
what's going in?

↑ oh add currents sends

[['i_10v → gnd_5', '10V', 1] ['i_10v → gnd_5', gnd, -1

15

~~So i - 10v → gnd - 5 · 10V = 1~~

- no different here?

yeah (i, node, sign)

- really long i names and just names

So know

$V = IR$

$I = \frac{V}{R}$ ← don't care here

need to combine all the nL

- in this case 10V

↑ just a more descriptive name

whats there

· 10V

+ i - 10V → gnd - 5

+ i - 10V → Vo - 6

↑ but both positive ???

- could have negative

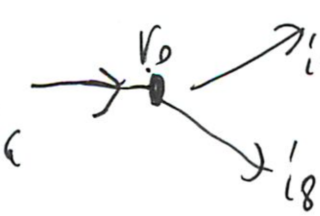
Vo

- i - 10V → Vo - 6

+ i - Vo → gnd - 7

+ i - Vo → gnd - 8

So



← giving shorter names

(16)

Then need to aggregate into eq

$$i'_6 = i'_7 + i'_8$$

$$i'_6 - i'_7 - i'_8$$

$$([1, -1, -1], [i_6, i_7, i_8], 0)$$

and i print v. eq too

- so how to aggregate?
- need to save first
- what about gnd
  - worry about later

(I am thinking about n times from my ~~python~~ interview)

- sort ✓
- gather all of that type
- oh don't need to sort
  - python will auto do it I think!
- hmm gives key error - why?
  - i starts w/ #?
- do a try, catch block
- cool
  - this is where ground could fit in

(17) Now what ~~to~~ do at each node
return

$([sign\ 1, sign\ 2, sign\ 3], [i_1, i_2, i_3], 0)$

- iterating though a dictionary - not happy
- cool got a quick hack
- seems to work
- but code reports iteration over non seq when try to use
- prob outputted wrong type

- ~~there~~

- oh in meanwhile
  just return ground = 0 and skip that one
  - that seems to have worked
  - signs and names seems to work
- or should KCL be returning voltages too?
  - and at ground is V_0 or i_0?
  - look back at real life ones
  - did <u>not</u> do that
- also never did add current
  - easy
- oh should be returning in format for es.addEquation
- so just return eq, <u>not</u> in a set

18

Ran sucessfully!

Test in tutor

Wow worked ✓

This was a really challenging, fun problem

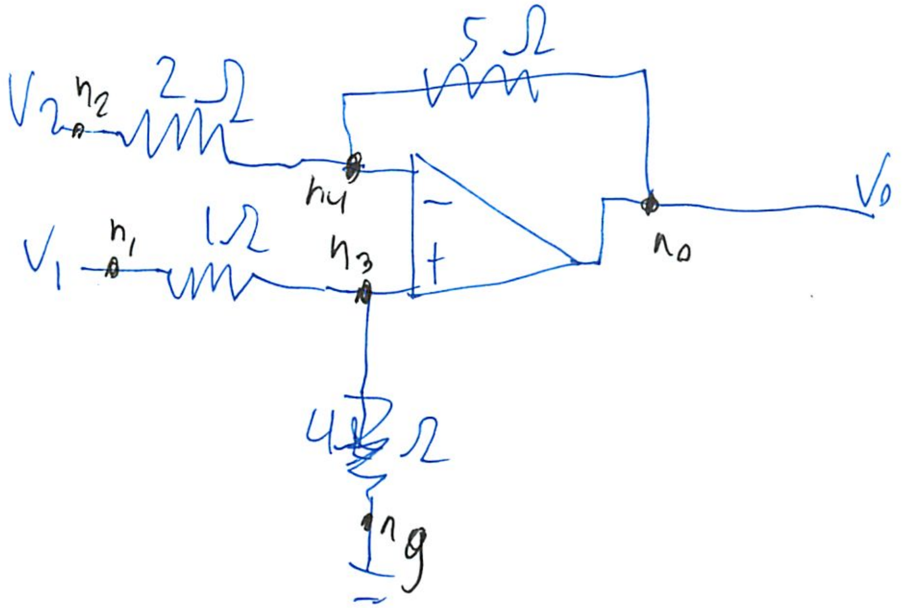
Did in ~1.5 hrs

So what did we solve for?

- Voltages b/w each pt + ground

Nano 8

10/28



do w/ current $V=IR$

$$\frac{n_4 - n_2}{2} = \frac{n_3 - V_1}{1}$$

$$\frac{n_4 - V_2}{2} = \frac{n_3 - V_1}{1}$$

$$n_4 - V_2 = 2n_3 - 2V_1$$

$$V_1 = -\frac{n_4 + V_2 + 2n_3}{2}$$

HW $\quad kV_1 - kV_2 - V_3 = 0 \qquad i_1 = i_2 = 0$

? do in relation to this?

(X) they want IT

②

$V_0 = 2(V_1 - V_2)$

$V_+ = \frac{4}{4+1} V_1$ ✓

$\frac{V_2 - V_-}{2} - \frac{V_0 - V_+}{5} = 0$? net flow = 0

solve for

$5V_2 - 5V_- = V_0 - V_+$

$\cancel{5V_- = 5V_2 - V_0 + V_+}$

$\cancel{V_- = V_2 - \frac{1}{5}V_0 + \frac{1}{5}V_5}$

$\cancel{V_- = V_2 - \frac{1}{5}V_0 + \frac{1}{5}\left(\frac{4}{5}\right)V_1}$

$V_- = V_+$

$\cancel{5V_- - V_+ = 5V_2 - V_0}$

$\cancel{4V_- = 5V_2 - V_0}$

$\cancel{V_- = \frac{5}{4}V_2 - \frac{1}{4}V_0}$

$2V_2 - 2V_- = V_0 - V_+$

$V_2 = \frac{V_0 - V_+ - 2V_-}{2}$

$\frac{-3V_0}{2}$

(2) $V_- - V_2$ ← current is going other direction for consistant direction

$$\frac{\cancel{V_2 - V_-}}{2} = \frac{V_o - V_{\cancel{+}}}{5} = 0$$

$$5V_2 - 5V_- = 2V_o - 2V_+$$

$$5V_- - 2V_+ = 5V_2 - 2V_o$$

$$3V_- = 5V_2 - 2V_o$$

$$V_- = \frac{5}{3}V_2 - \frac{2}{3}V_o$$ ⊗

$$\frac{V_o - V_-}{5} = \frac{V_2 - V_-}{2}$$

$$2V_o - 2V_- = 5V_2 - 5V_-$$

$$2V_o = 5V_2 - 3V_-$$

$$2V_o + 3V_- = 5V_2$$

$$3V_- = 5V_2 - 2V_o$$

$$\underline{V_- = \frac{5}{3}V_2 - \frac{2}{3}V_o}$$ ⊗ same

~~3~~ 4

3.

$$\frac{4V_1}{5} = \frac{5}{3}V_2 - \frac{2V_0}{3}$$

solve for V_0

$$\frac{2V_0}{3} = \frac{5}{3}V_2 - \frac{4}{5}V_1$$

$$2V_0 = 5V_2 - \frac{12}{3}V_1$$

$$V_0 = \frac{5}{2}V_2 - \frac{12}{10}V_1$$

Times up

$$V_- = \frac{5}{7}V_2 + \frac{2}{7}V_0$$ ← so where 7 in denom

$$V_0 = \frac{28}{10}V_1 - \frac{25}{10}V_2$$

$7V_-$

5 + 2 ?

not 5 − 2 ?

(5) $\frac{V_- - V_2}{2} = \frac{V_0 - V_-}{5} = 0$ * direction

$5V_- - 5V_2 = 2V_0 - 2V_-$

$\cancel{5V_- - 3V_2 = 2V_0}$

$7V_- - 5V_2 = 2V_0$

$7V_- = 2V_0 + 5V_2$

$V_- = \frac{2}{7}V_0 + \frac{5}{7}V_2$

$V_+ = V_-$

3. $\frac{4}{5}V_1 = \frac{5}{7}V_2 + \frac{2}{7}V_0$

$\frac{2V_0}{7} = \frac{4}{5}V_1 - \frac{5}{7}V_2$

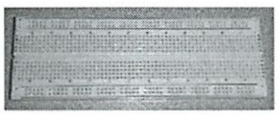
$2V_0 = \frac{4 \cdot 7}{5}V_1 - \frac{5 \cdot 7}{7}V_2$

$V_0 = \frac{4 \cdot 7}{5 \cdot 2}V_1 - \frac{5}{2}V_2$ there we go

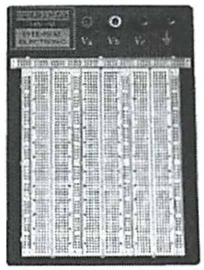
Why was I wrong - based off course notes
- thought about that reason but thought prof was right

DL8, SL9, HW3: Turning Heads

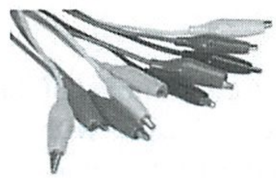
You can use any computer that runs CMax. Do `athrun 6.01 update`. Files will be in `Desktop/6.01/lab8/designLab/`. In addition, you will need:



Proto board

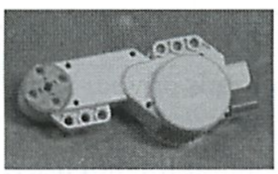


Power supply



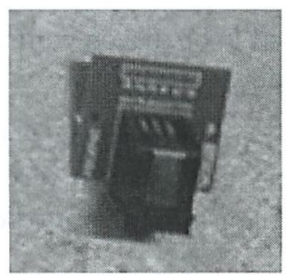
Four clip leads

Multimeter

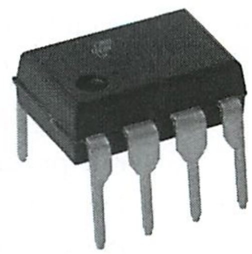


Lego motor

Black motor cable



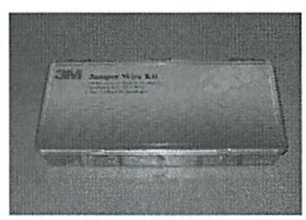
Six-pin connector



Op-amp package

Potentiometer

Resistors, as needed



Wirekit

The relevant files in the distribution are:

- `circuitSimulateTest.py`: A file with the right imports for defining circuit simulations.
- `lib601/oneStep.py`, `lib601/threeSteps.py`, `lib601/eyeServo.py`: Input signals for dynamic simulations of circuits.

See the back page of this handout for the pin-outs of the connectors.

Our ultimate goal is to build a 'head' that we can put on the robot that will be able to sense and track light. In lab 7, you designed a brain to control the robot to point at a light. The goal for this week and the next is to build a circuit that controls the neck motor on the head to get faster tracking of the light.

1 Lego Motor

We use a Lego motor for the "neck" of the robot head. The motor attaches to a 6-pin proto board connector via a short black cable with connectors that are similar to those used for telephones. Notice that the two ends of the cable are different: the locking clip is centered on one end and offset on the other. The end with the centered clip goes into the connector, the end with the offset clip goes into the motor. The motor is driven by the voltage difference between pins 5 and 6 of the connector.

Step 1. The motor is designed to be driven with a voltage difference between 0 and 10 V across its terminals. Try it out as follows.

- Connect the power supply terminals labeled +15 V and **GND** to the power rails of your separate proto board using clip leads. Adjust the power supply voltage to 0 V. (Yes, really 0).
- Plug a 6-pin connector into the proto board and connect it to a standalone motor (do **not** use a pre-built head).
- Turn off the power supply; wire pins 5 and 6 of the connector to the power and ground rails, respectively, of the proto board.
- Turn on the power supply.
- Connect a multimeter to measure the voltage from the power supply.
- Adjust the power supply voltage between 0 and 10 V and note the relation between motor speed and applied voltage.
- Swap the connections to power and ground. What happens?
- What is the minimum voltage required to make the motor turn?

$V_{min} =$ .26

Step 2.

- Remove the connection between the motor and power rails of the proto board.
- Re-adjust the power supply back to +10 V (in preparation for next part), then turn it off.
- Measure the resistance R_m of the disconnected motor using the multimeter.

$R_m =$ 4,8 Ω

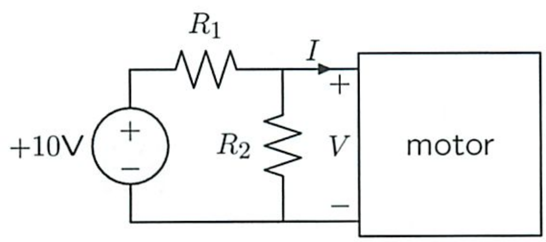
motor

1.1 Controlling the motor with resistors

Our goal is to control the motor electronically. We will ultimately mount the motor on the robot and use the robot's power supply, which is constant at 10 V. The point of this section is to find a way to use a constant-voltage power supply to get a range of motor speeds.

First, think about how we might control the velocity with resistors. One way might be to use a voltage divider to generate a control voltage between 0 and 10 V, and then use this control voltage to drive the motor.

Consider the following resistor circuit for generating the control voltage, where $R_1 = R_2 = 1000\Omega$.



Step 3. Build the circuit on your proto board. Turn the power supply back on and measure the voltage across the motor and observe the motor's behavior.

$V_{motor} =$ .044

no buffering

Check Yourself 1. Does the motor turn? Explain.

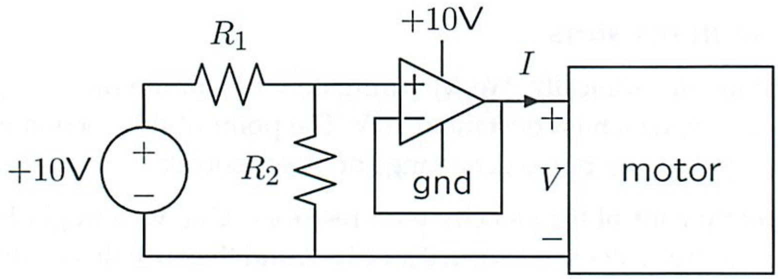
Step 4. Use circuit theory, treating the motor as a resistor, to determine the voltage across the motor. Use the resistance value you measured in step 2.

$V_{motor} =$ .05

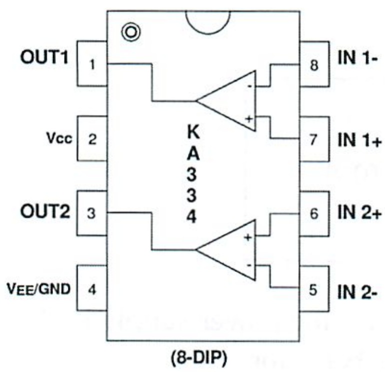
Check Yourself 2. Does the theory match the measurement in the previous part? Explain.

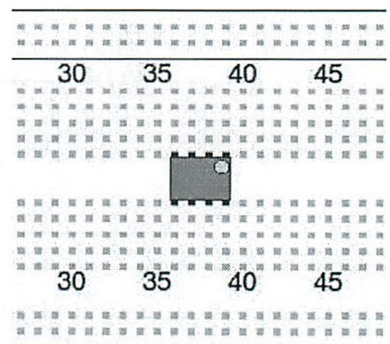
1.2 Buffering the motor voltage

We can add an op amp to *buffer* the output of the resistor network so that the resistors function as a voltage divider while the resulting voltage drives the motor. A simple buffer circuit is shown below.



We use op amps (KA334) that are packaged so that two op amps fit in an 8-pin (dual inline) package, as shown below.





The spacing of the pins is such that the package can be conveniently inserted into a breadboard as shown above (notice that a dot marks pin 1). The top center wire in the picture above shows the connection of the op amp pin labeled V_{CC} to the positive power supply (+10 V). The shorter wire to the left of center shows the connection of the op-amp pin labeled V_{EE} to ground. The diagonal wires indicate connections to the outputs of the two amplifiers, and the short horizontal wires indicate connections to the two inverting (−) inputs.

Step 5. Build the buffered divider circuit on your proto board. Measure the voltage across the motor and observe the motor's behavior.

> *Check Yourself 3.* Compare the behaviors of the circuit with and without the buffer. 5.0V

Step 6. Replace the two resistors in the voltage divider with a potentiometer.

Step the potentiometer through various settings (1/4 turn, 1/2 turn, 3/4 turn) and observe the behavior of the motor. We'll ask you to compare this behavior to a simulation below.

> *Checkoff 1.* Explain to a staff member the results of your experiments, with and without buffering.

Turn off your meter, disassemble your board, and put the wires, op-amp, pot, and connectors back in the appropriate places. Throw away the resistors.

2 Circuit Simulations

We saw in Lab 7 that we can simulate circuit layouts using CMax. As we explore more complex circuit designs, we will want to simulate them to see if they have the desired behavior, without going through the trouble of doing a complete wire layout in CMax. In fact, CMax derives a `Circuit` component level (as in this week's Software lab) representation of the circuit from the layout and then simulates it.

Here's a procedure for simulating the buffered divider circuit (don't try to run this procedure yet — we'll do that in the next step).

```
def motorTest(test):
    (nsteps, signal) = test
    circ = [OpAmp('v+', 'v-', 'vo'),
            Wire('vo', 'v-'),
            Pot('gnd', 'v+', '10v'),
            Connector('Motor', ['1', '2', '3', '4', 'vo', 'gnd']),
            Power('10v'),
            Ground('gnd'),
            Probe('Pos', 'vo'),
            Probe('Neg', 'gnd')]
    runRealCircuit(circ, signal, nsteps = nsteps)
>>> motorTest(oneStep.testSignal())
```

Note that we cannot use the command `OpAmp('v+', 'vo', 'vo')` to build a buffer; instead, we must use a `Wire` to connect the output (`'vo'`) to the negative input (`'v-'`).

The procedure `runRealCircuit` takes a list of circuit component instances and a test signal, runs the simulations and produces graphs. The Python definition of an appropriate signal can be imported from the test files that you use with CMax, for example `oneStep.testSignal()` returns a tuple (number of simulation steps, signal), as shown above.

We have the following classes of components, some of which you saw in software lab, and some of which are new:

- `Resistor(value, node1, node2)`: `value` is a number indicating resistance in Ohms; `node1` and `node2` are strings representing node names.
- `Wire(node1, node2)`: `node1` and `node2` are strings representing node names.
- `OpAmp(posNode, negNode, outNode)`: `posNode`, `negNode`, and `outNode` are all strings representing node names.
- `Connector(type, pinNodes)`: the `type` argument can be `'Motor'` or `'Head'`; `pinNodes` is a list of node names.

- `Power(node)` and `Ground(node)`: you can have at most one of each of these 'components', which specify which nodes are connected to power and ground.
- `Probe(type, node)`: the `type` argument is either `'Pos'` or `'Neg'`. You can have at most one positive and one negative probe.
- `Pot(nodeLeft, nodeCenter, nodeRight)`: represents a 5KΩ potentiometer, with `nodeCenter` being the center terminal, whose voltage will vary. The resistance between `nodeLeft` and `nodeCenter` is $\alpha(5K\Omega)$, while the resistance between `nodeRight` and `nodeCenter` is $(1-\alpha)(5K\Omega)$, where $0 \leqslant \alpha \leqslant 1$.

Step 7. **Run the simulation.** Start idle with the `-n` flag. Load the file `circuitSimulateTest.py` from the lab distribution (this file contains the code from above). Evaluate the procedure by pressing Run Module in Idle. Then type `motorTest(oneStep.testSignal())` in the Python shell, where `oneStep.testSignal()` specifies the conditions that we wish to simulate (here we simulate a potentiometer that starts at $\alpha = 0$ for $0 < t < 0.5$ seconds and changes abruptly to $\alpha = 0.1$ for $0.5 < t < 1$ seconds). This will run the simulation and produce several graphs. They all have time steps on the x axis, and some other quantity on the y axis. The signal is sampled at intervals of 0.02 seconds.

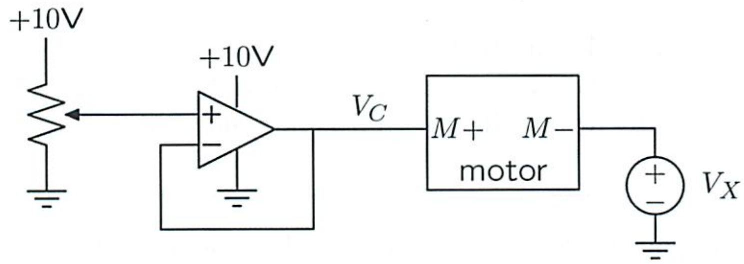
- **Probe** (in green): When there is a probe in the circuit, this graph shows the voltage measured across the probes. Voltmeter
- **Motor** (in red): When there is a motor in the circuit, then simulations **assume that the motor is attached to a potentiometer**, which turns as the motor turns. One of the motor graphs is the α value of the motor potentiometer, which measures the motor potentiometer angle. The other motor graph shows rotational velocity (radians/sec) for the motor. Remember that the potentiometer has a finite range of rotation (0 to $3\pi/2$ radians), so with a constant voltage, the motor will quickly reach the end of the range and stop. **When you're using a real robot head, driving it up against the end of the range in this way risks tearing the head apart.**
- **Input** (in blue): When there is an external input to the simulation, such as a potentiometer, this graph shows the input value, for example, the value α for a potentiometer, which goes between 0 and 1.

> *Check Yourself 4.* Make sure you understand the meaning of the motor rotational velocity and motor pot alpha graphs. Compare the simulated behavior to the actual behavior of the circuit you built.

3 Bidirectional Speed Controller

The circuit you built for Checkoff 1 controls the speed of a motor. That circuit allows the motor to turn fast or slow (depending on the choice of resistors or the pot setting), but only in one direction. To make our robot head turn both left and right, we need to design a bidirectional speed controller.

The circuit in the previous parts of this lab only turns in one direction because the op-amp operates from a single +10 V power supply. We are limited to a single +10 V power supply, because it is the only power supply available from the robots for which we are building the "head." A simple approach to this problem (using a (5KΩ) potentiometer) is to connect the motor as follows:



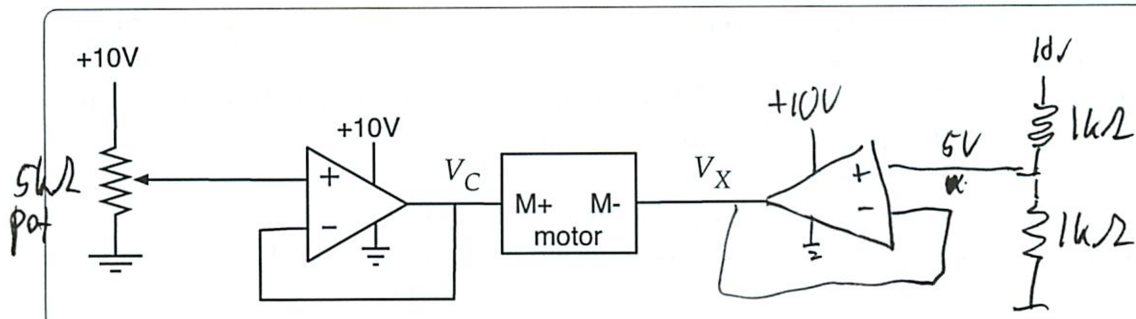
Step 8. The key new component in the bidirectional speed controller is the voltage source V_X. What value of V_X gives the most symmetric (around 0) range of speeds for the motor?

$V_X =$ 5V

Design a circuit to implement this voltage source (using only a fixed 10 V supply, which is all that's available from the robot).

Check Yourself 5. Can you implement V_X with just a voltage divider? Explain.

Modify the circuit diagram below to include your circuit for supplying V_X.



Step 9. Build a `Circuit` simulation of your bidirectional control circuit by completing the definition of the `biDirectional` procedure in `circuitSimulateTest.py`. It should contain a circuit definition that is a modification of the one in the `motorTest` procedure. When the α value of the pot is near zero, the motor should spin quickly in one direction; when α is 0.5, the motor should be stopped, and when α is near 1, the motor should spin quickly in the other direction.

The `biDirectional` circuit can be tested with the `threeSteps.testSignal(1.0)` test, which simulates turning the potentiometer first to $\alpha = 0.25$ and holding it there, then turning it to $\alpha = 0.5$ and holding it there, and finally turning it to $\alpha = 0.75$ and holding it there.

Checkoff 2. Demonstrate your working simulation. Explain the relation between motor speed and potentiometer angle. Demonstrate that you can generate both positive and negative speeds. Explain how your circuit accomplishes bidirectional speed control.

Save a plot of the input signal and the associated output signal, as well as the procedure that defines your circuit. Mail these results to your partner. We will discuss these at your interview.

4 Head Controller Design

- This should be done individually. You can discuss with your partner but your written submission should be your own.
- See the back page of this handout for a detailed listing of what to turn in. It is due, in lab, on paper, at the beginning of your Design Lab 9 section.
- This section of the lab contains:
  - A tutor problem that is due in software lab (but can be done before).
  - Two checkoffs that can be obtained during design lab 8, software lab 9, or **the beginning** of design lab 9.

Your goal for this assignment is to design and compare two circuits for controlling the neck motor on the head so as to point the head quickly and accurately towards a bright light (as you did in Lab 7 (last week) by turning the robot, but hopefully more quickly).

4.1 Design Criteria

We will start by considering what properties we want our circuit to have:

- **Fast**: The head should line up with the light as quickly as possible.
- **Stable**: The head should not oscillate.
- **Uniform**: The behavior of the head should be nearly independent of the brightness of the light and of the distance of the light to the head.
- **Accurate**: The head should point accurately at the light, as demonstrated by the laser pointer.

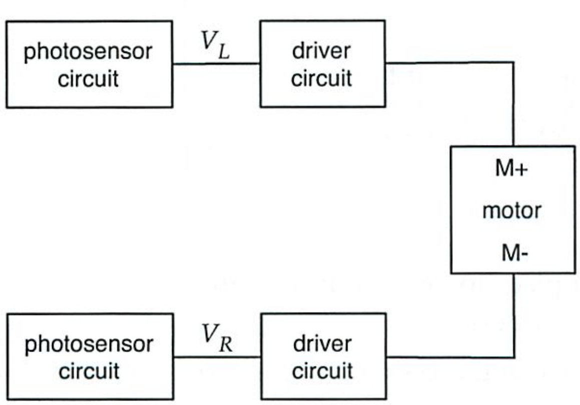
There are a few key points to keep in mind when thinking about a design.

- Op-amps can't produce voltage values outside the range of supplied voltages (0V to 10V in our case).
- The speed of the motor is proportional to the voltage difference across it; for fast response you want this difference to be large.
- You can think of the circuit as a controller in a feedback system, with a gain. Higher gains will give you higher speeds, but may cause oscillation. Think about what controls the overall gain and how you can change the gain. Think about the limits to the gains that you can choose in your circuit.
- The sensors you have in your circuit do not have identical response to light. How will this affect the behavior of the circuit? Can you compensate?

4.2 A reference design

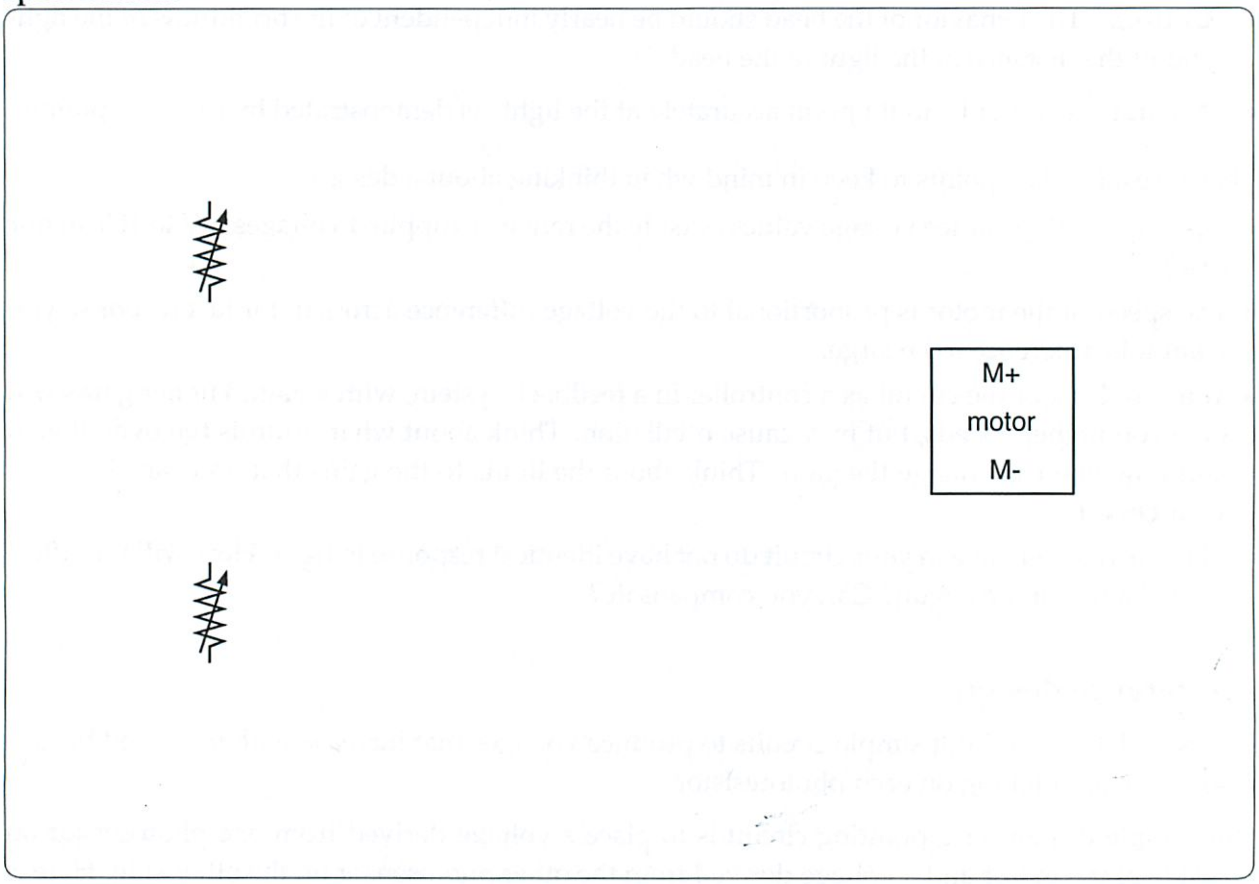
In design lab 7, you built simple circuits to produce voltages that increase with increased brightness of the light falling on each photoresistor.

One simple design for a pointing circuit is to place a voltage derived from one photosensor on one side of the motor and a voltage derived from the other photosensor on the other side. Here is the basic structure of the circuit:

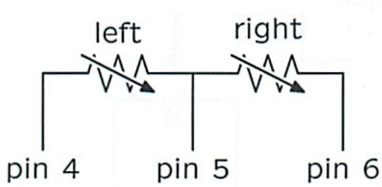


The difference in voltages across the motor can be described as $k(V_L - V_R)$, where k can be thought of as a gain. Be sure that you understand how to adjust k in your circuit.

Step 10. Fill in the design, using op-amps and resistors to make sure you can get a good range of voltages across the motor. **Label each node in your diagram with a name you will use in the `Circuit` specification.**



Remember that the photoresistors are connected on the head as shown below, and indicate the pin numbers on your diagram above.



Step 11. The file `circuitSimulateTest.py` has a place to define a procedure called `eyeNeckCircuit`. Create a list of components representing your design.

Use a **head connector** in your circuit, to give you access to the photoresistors and the motor on the head assembly.

```
Connector('Head', ['neck pwr', 'neck signal', 'neck gnd', 'left eye',
                   'common eye', 'right eye', 'motor pos', 'motor neg']),
```

You can simulate your circuit's behavior by running

```
eyeNeckCircuit(eyeServo.testSignal(dist=3.0))
```

The test signal simulates moving a light instantaneously back and forth between two positions. The positions have angles $\pi/2$ and π; when the head is pointing directly at these angles, the motor potentiometer values should be $\alpha = 0.333$ and $\alpha = 0.666$. The head starts at $\alpha = 0.5$. The ideal circuit would cause the robot's neck angle to track this input as closely as possible.

Debugging: You might want to simplify your problem while debugging so that you can better understand what's going on.

- One way to simplify is to disconnect the motor (for example, by temporarily connecting pins 7 and 8 of the head connector to ground) and place probes on the nodes that you would have connected to pins 7 and 8 of the head connector. That way you can observe what your circuit is "commanding" the motor without making the head turn and change the light values.
- Another way to simplify is to use a simpler input signal; `eyeServo.simpleSignal(dist=3.0)` models a light source at a constant angle. You can use that with the motor connected to make sure that your circuit converges to the target angle.

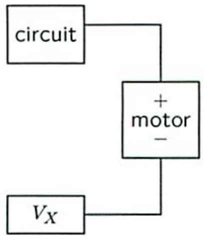
You can change the simulated distance (in meters) between the robot and the light by changing the argument `dist` in `eyeServo.testSignal(dist=3.0)`.

Step 12. Simulate your circuit at distances of 1 and 3 for gains 1, 5 and 10 (or a high enough gain so that it stops working). Why does it stop working?

Checkoff 3.	Discuss the design and behavior of your circuit. Be sure you can address all the issues in the Design Criteria section (speed, accuracy, stability, uniformity). Keep your circuit diagrams, the `Circuit` descriptions, simulation results for your next interview.

4.3 Alternative design

We can use the bidirectional motor controller from earlier in this lab as the basis for an alternative design in which both photoresistors are used to compute a voltage on the positive terminal of the motor and there is a fixed voltage on the negative terminal of the motor.



Step 13. How should the voltage on the positive terminal of the motor relate to the amount of light on the photosensors (L and R)? Two good choices are something proportional to $L/(R+L)$ or to $L-R$. Using $L/(R+L)$ leads to a simpler design, but either one is fine.

Design a circuit that establishes such a voltage. You can use as many op-amps as you need, but try to keep it simple.

Step 14. With this voltage on one terminal of the motor and a fixed voltage on the other, write an algebraic expression for the voltage difference across the motor.

Wk.9.1.1	Your circuit design must have bidirectional behavior: the motor must be able to turn both ways. This tutor problem will help you think about the ranges of voltages and ways of introducing gains higher than 1.

Step 15. Draw your complete alternative design and **label the nodes** with names you'll use in the circuit definition in the next step.

Step 16. Make a new procedure in `circuitSimulateTest.py`, similar to `eyeNeckCircuit`, that creates a list of circuit components corresponding to your alternative design and simulates it. Run simulations using `eyeServo.testSignal` as input at distances of 1 and 3 for gains 1, 5 and 10. Save the plots of the motor potentiometer angle that results from these simulations.

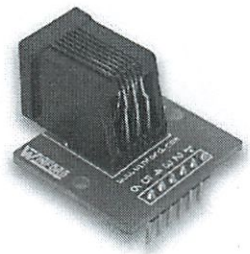
Checkoff 4.	Discuss the design and behavior of your alternative circuit. Be sure you can address all the issues in the Design Criteria section (speed, accuracy, stability, uniformity). Keep your circuit diagrams, the `Circuit` descriptions, simulation results for your next interview.

HomeWork3: What to hand in at the beginning of design lab 9:

A printed, stapled, legible collection of pages containing:

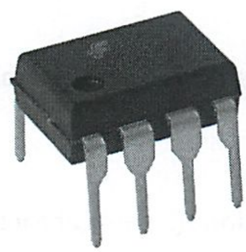
- Reference Design
  - A detailed circuit diagram with the nodes labeled.
  - The code describing your circuit.
  - The motor potentiometer graph for each of your simulations (3 gains at each of 2 distances).
  - A discussion of what the simulations tell you about the circuit, in reference to the design goals.
- Alternative Design
  - An algebraic expression for the voltage difference across the motor, in terms of L and R, the brightness of the light on each photosensor.
  - A detailed circuit diagram with the nodes labeled.
  - The motor potentiometer graph for each of your simulations (3 gains at each of 2 distances).
  - A discussion of what the simulations tell you about the circuit, in reference to the design goals.

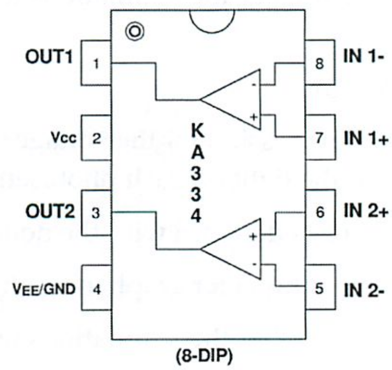
Motor (6-pin) Connector Pin-out



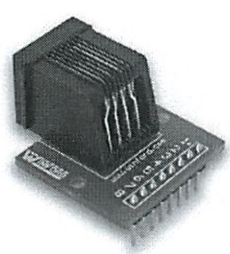
pin 5:	V_{M+}	Motor drive +
pin 6:	V_{M-}	Motor drive −

Op-Amp Pin-out





Head (8-pin) Connector Pin-out



pin 1:		neck pot (top)
pin 2:		neck pot (center)
pin 3:		neck pot (bottom)
pin 4:		photoresistor (left)
pin 5:		photoresistor (common)
pin 6:		photoresistor (right)
pin 7:	V_{M+}	Motor drive +
pin 8:	V_{M-}	Motor drive −

DesLab 8

10/28

Circuit Simulate test.py - defines the circuit for simulations
Various input signals

goal: build head to sense + track head w/ circuit

Wired up

Swapping = swap direction

.26 V

$V = IR$

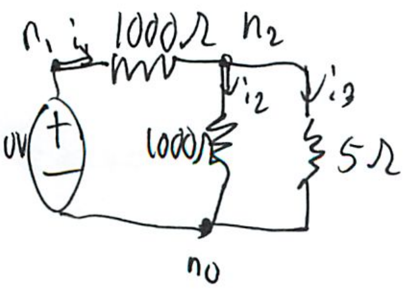
$R = \frac{V}{I}$ ← need I?

$R = 4.8\,\Omega$ smaller is better

1.1 $V_{supply} = 10$

Real $V_{motor} = .044$ motor does not turn

Theory



$n_2 - n_1 = i_1\,1000$

~~1000~~

$n_2 - n_0 = i_2\,1000$

$n_2 - n_0 = i_3\,5$

$n_0 = 0$

(2) $i_1 = i_2 + i_3$

$$\frac{n_2 - n_1}{1000} = \frac{n_2}{1000} + \frac{n_2}{5}$$ want n_2

$$\frac{n_2 - n_1}{1000} = \frac{201 n_2}{1000}$$

$$\frac{n_2 - 10}{1000} = \frac{201 n_2}{1000}$$

$$n_2 - 10 = 201 n_2$$

$$-10 = 200 n_2$$

$$n_2 = \frac{-10}{200} = \frac{1}{20} V$$

.05V ✓ theory matches reality

1.2 Buffering motor voltage

w/ op amp

5.0 V real life

note pins of op-amp

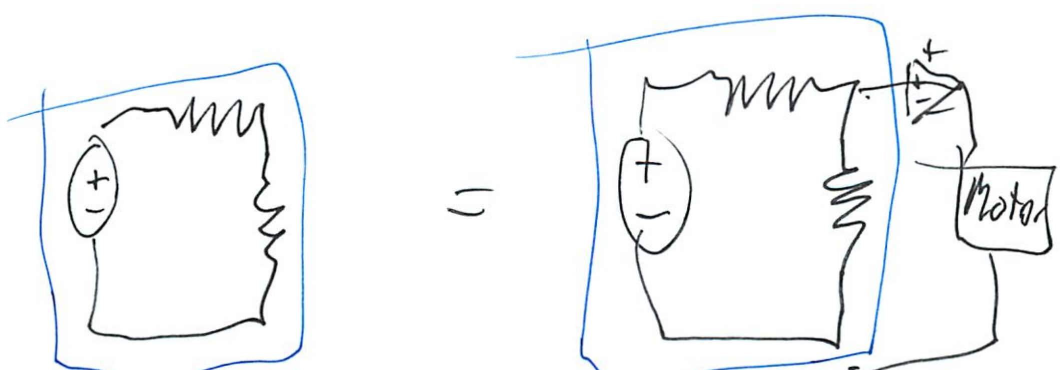
pull R_2 → M → 10 V speed doubles

③

w/ 5 - such easier path
shorts out $1 k\Omega$

* buffering left
    - add motor w/o changing left
    - left protected
right depends on left's specification

(I am all sequential logic - left side of brain
not patterns (ie music) - right side)



Same for left

but changing left → affects the ~~[illegible]~~ voltage
specified for right!

(makes a lot more sense now)

(4) w/ pot

0 turn = 0 V

~ $\frac{1}{4}$ turn = ~~###~~ 7 V

~ $\frac{1}{2}$ turn = 9 V

$\frac{3}{4}$ turn = 9 V

1 turn = 9 V

maxes out 9V at $\frac{1}{3}$ turn

believe 5k Ω pot

- so 1000 Ω we had was like $\frac{1}{5}$ pot → 5V

So $\frac{1}{3}$ pot → 9V makes sense

Part 2 Circuit Simulations

- note w/ Op amp need wire to $V_o \to V_-$
- run Real Circuit
- use test signal
- various types
- -pot right center left

$$\underbrace{\text{right center}}_{1-d}\ \underbrace{\text{left}}_{d}$$

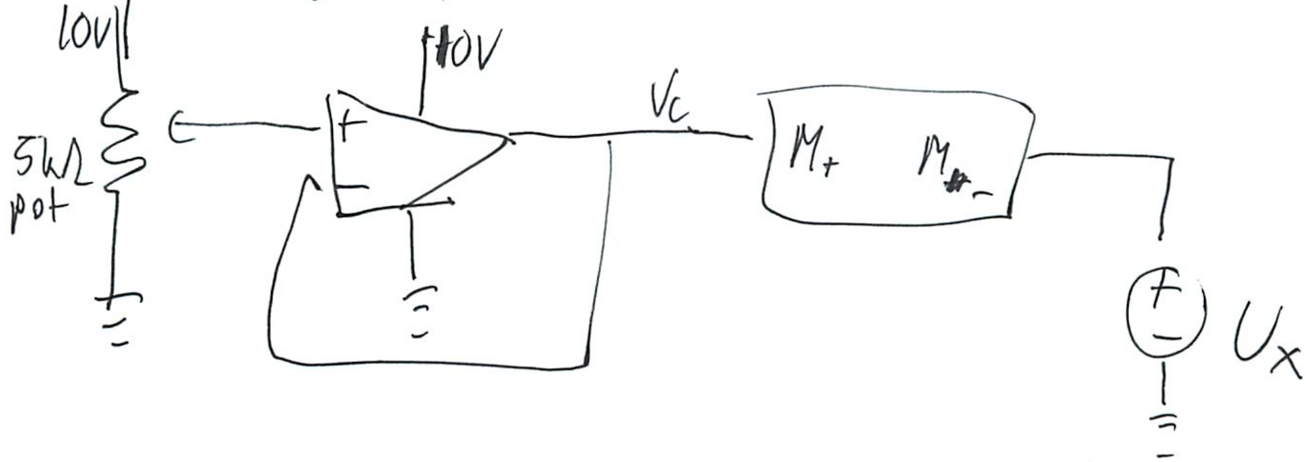
- how is motor ~~###~~ pot and input pot different

(5)

- potentiometer is connected to motor's output
  - so see how far it turned (output)

3 Bi Directional Speed controller

need to go ↻ ↺
before could only go 1 direction
their solution



Whats new is V_x
What value of V_x gives symetric range of speed?

Voltage drop on motor $V_c - V_x$

if $V_c > V_x$ goes main dir
$V_c < V_x$ goes opposet dir

so somewhere in between
$V_c = V_x$

(6)

Solve for V_x

V_c depends on what pot is

$\frac{1}{3} = 9V$

$\frac{1}{5} = 5V$

$\frac{1}{2} = 9V$ ← do we want it to charge at 9V?

So want V_x so V_c can go either way

5V ✓

when $V_c > 5$ one dir

$V_c < 5$ other dir

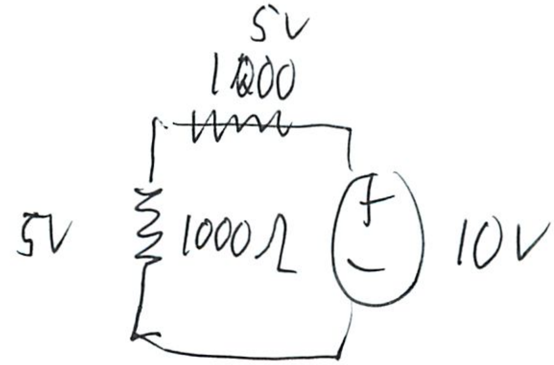
what I was thinking

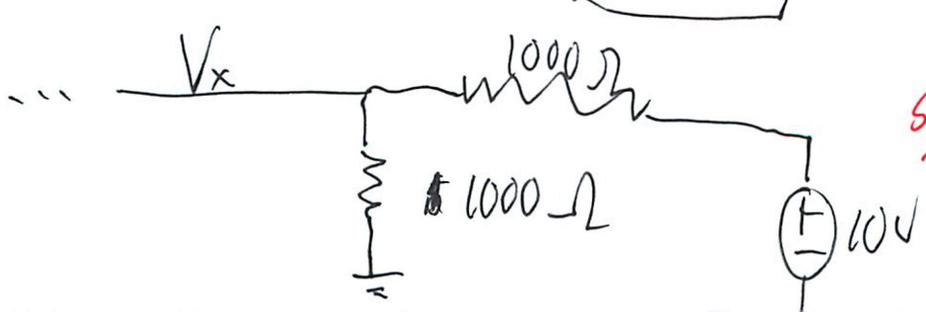
$\alpha > \frac{1}{5}$

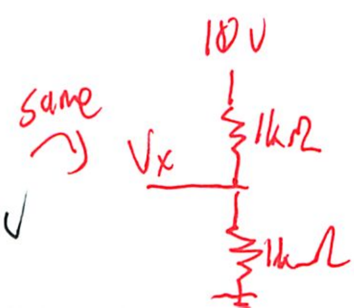
$\alpha < \frac{1}{5}$

Now need to build the 5V output

Want something to be always 5V

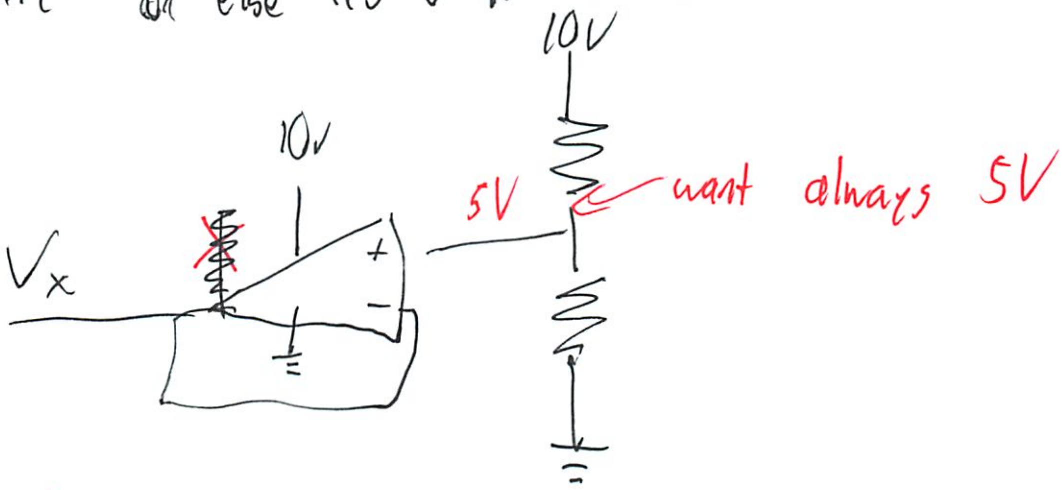






7

TA: Motor is ~~xx~~ loading voltage divider
need buffere ~~or~~ or else .05 v like before

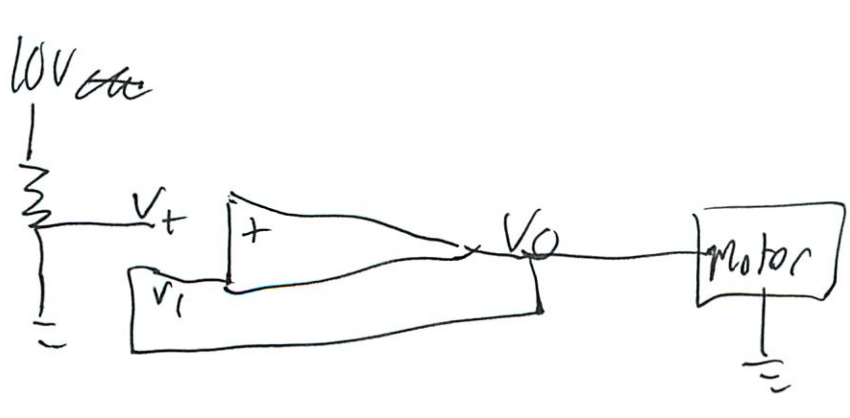


when plug motor in - want still to be 5V
thats why need buffer
not fall to $\frac{1}{20}$ V which would happen

---

build circuit simulation w/ biDirectional proc
what we had

OpAmp (V+ , V− , Vo)
wire (Vo , V−)
Pot (gnd , V + , 10V)
~~xx~~ Motor (1, 2, 3, 4, Vo , gnd)



need to add

8

Pot (gnd, $V1+$, 10V)

OpAmp ($V1_+$ $V1-$, $V1_o$)

wire ($V1_o$, $V1-$)

Motor change (1, 2, 3, 4, V_o, $V1_o$)

---

try running - singular matrix !
- gets 1 graph
- then fails

($V1_o = V_x$)

---

don't make it a pot - static resistor

Resistor (1000, ~~...~~ 10v, V1+)

Resistor (1000, V1+, gnd)

Works this time!

but test w/ 3 step

---

note $\frac{d}{dt}$ motor pot output = motor velocity

9 4. Head Controller Design

10/29
Home

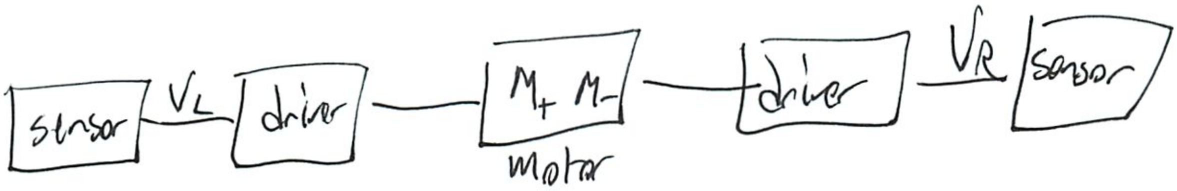
- fast
- stable
- uniform
- accurate

} Criteria

Speed of motor ~~*~~ a proportional to voltage

* Controller in feedback system w/ gain
  - pick right light

balance out sensors & light



↑
voltage difference
$k(V_L - V_R)$

it asks how to adjust k - do they mean ~~set~~ set it or actively adjust

- need to draw out circuit
- go back and review last design lab
- sensors are variable resistors
  - resistance drops in front of light

(10)

Other was

$$V_o = \frac{R_1}{R_1 + R_o} 10$$

?: is this was resistor to put as R_1

- my notes are not clear what Correct was

- Oh right had to find resistor that would put out right Votage drop (over what?)

- 627 Ω
  - found by solving eq on Wolfram Alpha
  - just want a differential voltage drop
    ~~I guess in resistor~~
    No I think its over that 627/1kΩ resistor
    yeah confirmed w/notes

- So before fed this drop into robot's analog input (aka voltage probe)
- brighter light = higher voltage
  ↳ ? opposed what we want - if dim want it to turn quickly

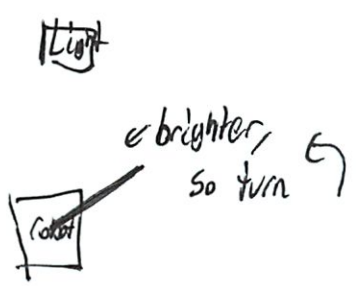
11

But what about cross

~~If L = dim, turn~~

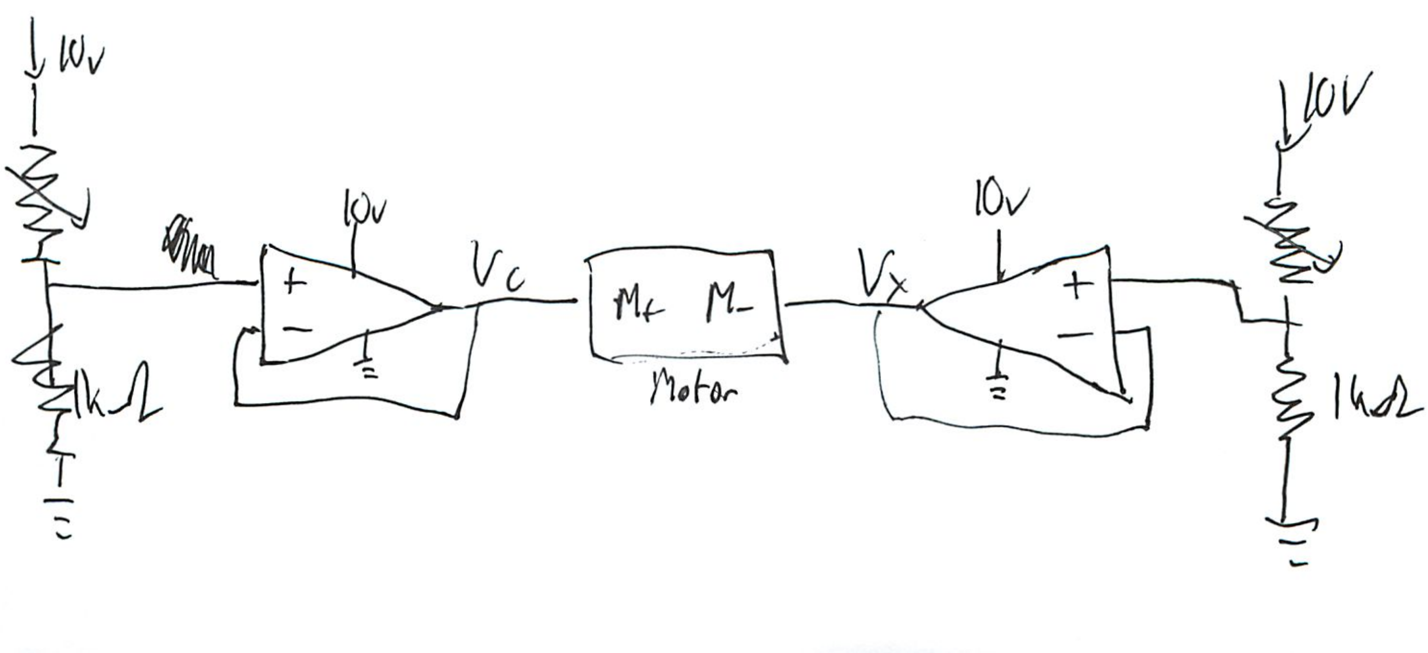
What did we do before?

if L voltage > R voltage → turn left



So high voltage left, turn left

- what we had right?
- well if wrong, flip motor connection
- it would turn <u>towards</u> greater voltage source
- So no need to flip

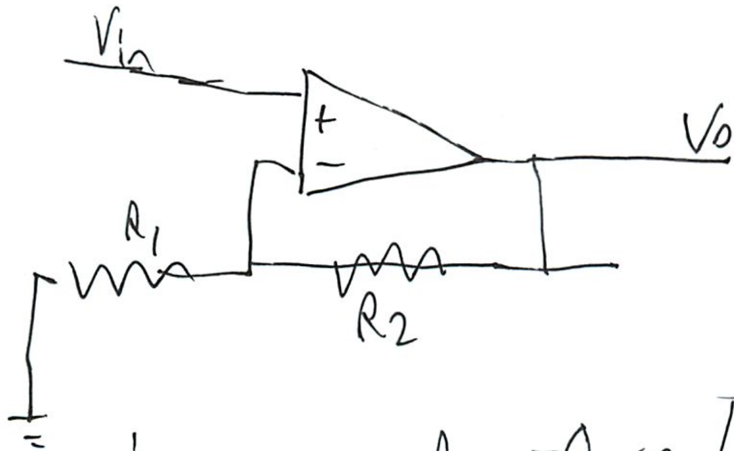


(12)

Why does wire go from V_- to V_c again?

"non inverting amplifer" ← use case

just amplifies a voltage



WP

$$V_{out} = V_{in}\left(1 + \frac{R_2}{R_1}\right)$$

~~Also~~ Lecture

$$\frac{V_o}{V_i} = \frac{R_1 + R_2}{R_1}$$

- oh ends up being same thing

in our case $R_2 = 0$ so $\boxed{V_{out} = V_{in}}$

don't care about impediance - not learning

Currents may not be matched

Opt amps always ~~Vout = Vin~~ ~~$V_+ = V_-$~~ this also $= V_{out}$ (if $k \to \infty$) ideal

↳ are you sure?

but why that line?

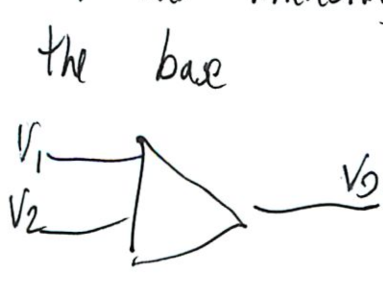
- So $V_o = V_{in}$?

otherwise

never mind - ask prof - but for now treat as black box

Ok WP has interesting pattern

the base



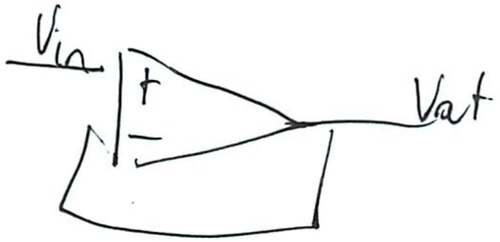
is comparer - outputs larger voltage

13

for inverting talks about feedback
Some output returned to input
output 180° out of phase - so subtracted from input
reducing input into op amp - reducing gain of ampifier

---

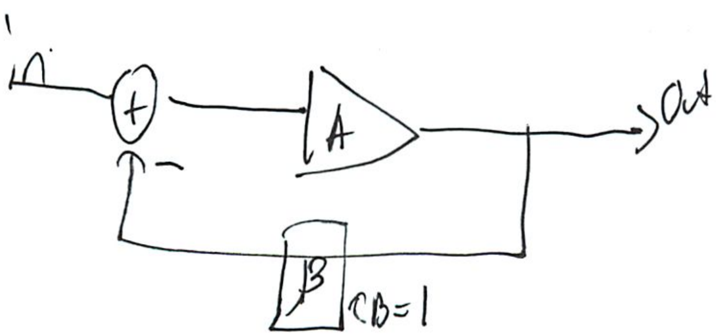
what we have w/o resistors = <u>Voltage follower</u>
- eliminate loading effect



$V_{out} = V_{in}$
$Z_{in} = \infty$ ← impedience

~~has poor stability measurements~~ don't care

WP: Votage buffer example
Oh! like a negitive feedback system

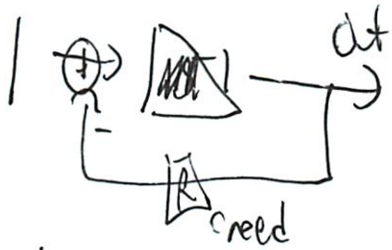


"full series negitive feedback"
entire output voltage placed "in contrary and in series"
w/ input voltage
Voltages subtracted (KVL)
≈ sum of voltages in circle = 0

(14)

Forces op amp to adjust output voltage to input voltage
(might be clearer if saw inside - but inside is very complex!)

Think more about that feedback subtract



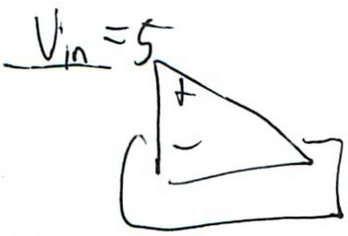
in	out	R
1	1	1
1	0	0
1	1	1

but ocilates - how does help?

WP: Negitive feedback : - improves stability + reduces sensitivity
Oh that "history" thing I had trouble w/ in interview

knows the past, so smoother

But does not explain why op amp does this



outputs bigger 5
and now $V_- = 5$
so just outputs 5?

but what about getting there? - class does not cover

(15)

and switching wire to V_+ V_+?

- that has an paradox

$V_+ - V_- = \ominus$ would be – driving it down

called positive feedback – drives away
adds to in?

- but then that is not really "picking" V_+/V_-
- I guess think $k(V_+ - V_-)$

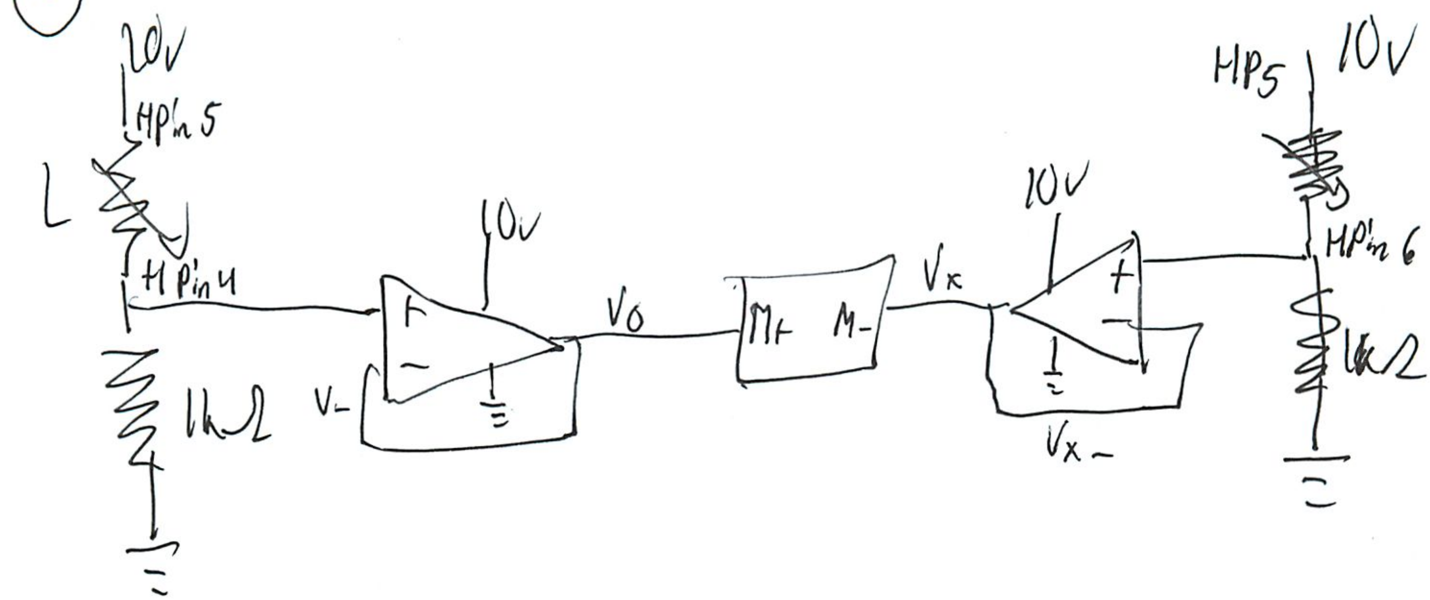
negitive feedback drives to input
reduces slowly to V_{in} / $V_{desired}$

Starting to get control systems more
Should ask prof could also do kirchoff's loop rule

Ok back on task

- now need code it up in eyeNeckCircuit, procedure

16



~~With not [illegible]~~

motor on head

Vo, Vx

Connector ('Head', [~~[illegible]~~ 0, 0, 0, 'h4', '10V', h5', ~~rest~~)

Resistor (H4, Ground) 1000Ω

↑ they give these dummy # - all different - will do

OpAmp (h4, V-, Vo)

Wire (V-, Vo)

Connector → ~~Motor (0, 0, 0, 0, Vo, Vx)~~

OpAmp (h6, Vx-, Vx)

Wire (Vx-, Vx)

~~Connector~~ done already

Resistor (1000, h6, ground)

↑ very rough format

don't forget the standard

Power (10V)

Ground (gnd)

Probe (Pos, Vo)

Probe (Neg, gnd)

(17)

test in python eyeNeckCircuit(eyeServo.testSignal(dist=3.0))

(X) error only 1 motor at a time supported

- what I have!

- comment the motor out

(X) Motor not connected!

- WTF

- emailed in

So tes is moving angle back and forth

references motor potentorer

they recommend disconecting motor

↳ oh it says pins 7+8 of head

- oh I see that is the motor we want

- integrated head

Ok works now

as in it runs, but does not produce proper results

motor never moves - probe voltage same

- ~~does~~ what is neck power, signal, grand?

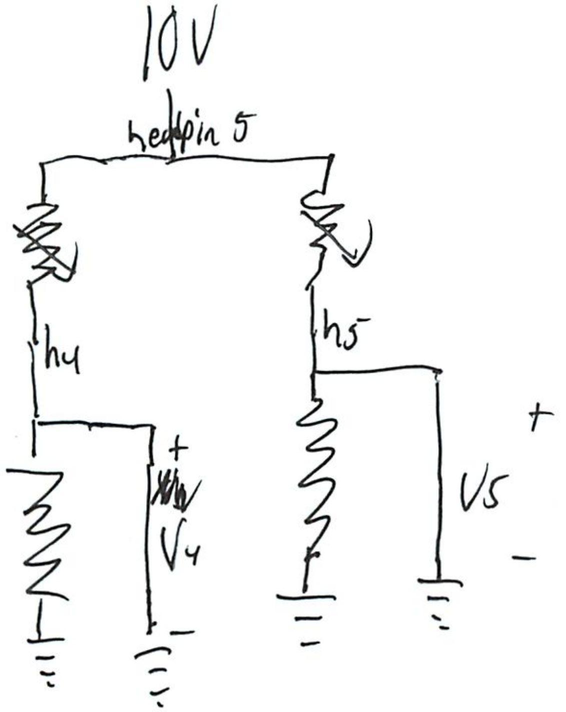
- ~~pot~~ for the pot which we do not care about right?

(18)

test w/ varring probes

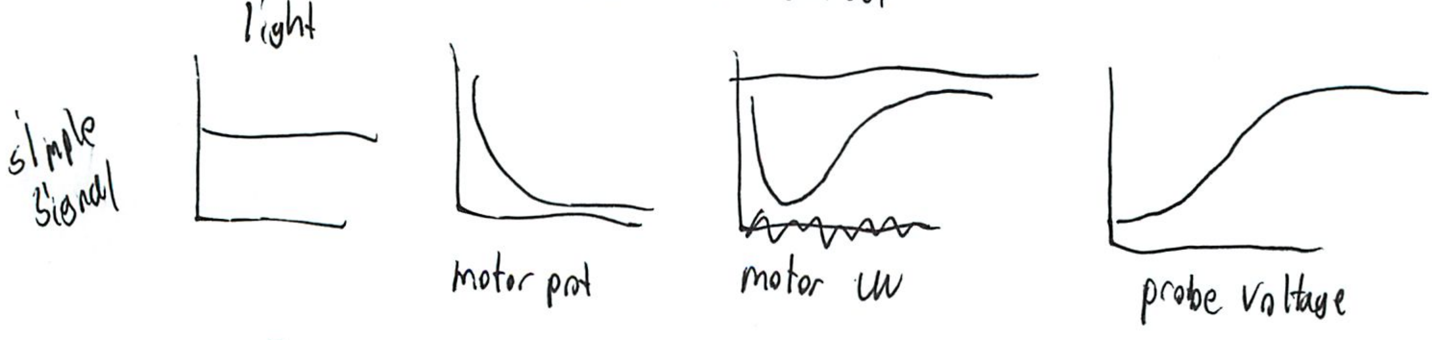
h4 always constant voltage

- why no voltage drop b/w h4 and gnd



basic

- try building just this
- oh realized was connecting resistor to h5 not h4
- oh now original circuit is doing somethin
  - but why did that value V5 - V4 affect h4
    - should still have 1 effect



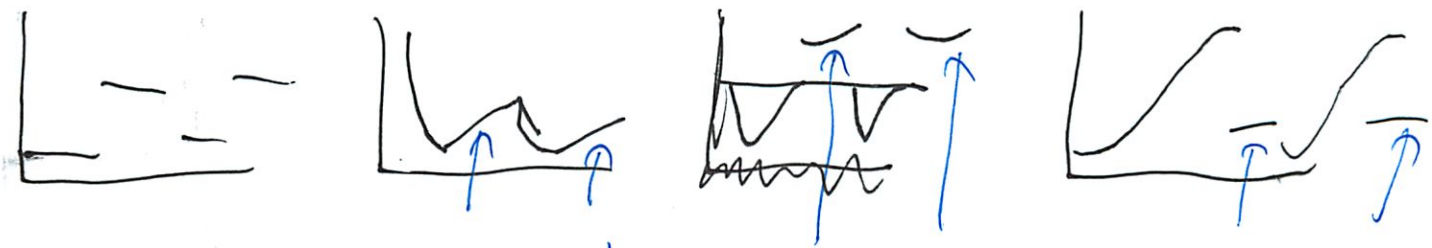
exactly what we want

(19)

But what did we set gain to?
- if anything

Oh $k \to \infty$ w/ our ideal op amp

Now try w/ test signal



These parts worry me

Why is it not moving ~~later~~ back fast?

Is moving at linear speed / in wrong direction?
? no Voltage is increasing but very slightly!

? are they simulating differ voltages?
- Oh probe is at h4
- when at h6?
  - same thing
- try other resistance for right side?
- nope ~~does not~~ seems to mess up stuff more
- its like each op amp has different k
- They actually mention gain - but where would you put it in?

(20)

I thought k was in opt amp

Lab says difference in Voltages across motor

$= \cancel{\ } k(V_L - V_R)$

^ I just have k = 1 now, I guess

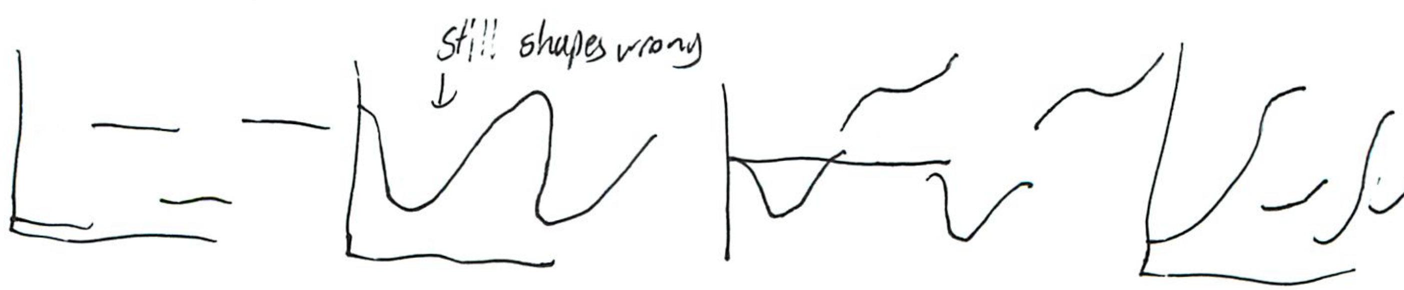
but how to change it ??

- not in opt amp
- ? not in the 1000 resistors
- ? or add something to circuit across motor ??
  - I don't think so
- Somehow boost Voltage gain on right
- but that is w/ other resistors
- ~~with~~ guess try smaller adjustments (guess + ✓)
- 950 ~~looks~~ on right looks a lot more balanced

how to tell scientifically

- last lab !
- need to measure Voltages drop on photo resistor at various lights and then do math

- 940 Ω



(21)

Still confused on ~~bu~~ if this is right

Test at distance = 1

Think thats good - will ask in lab, or via email

~~B~~ut I don't have numeric gains

4.3 Alternate Design

- where both controllers on 1 side
- and other side is fixed
- so like earlier in lab

- So voltage on + relate

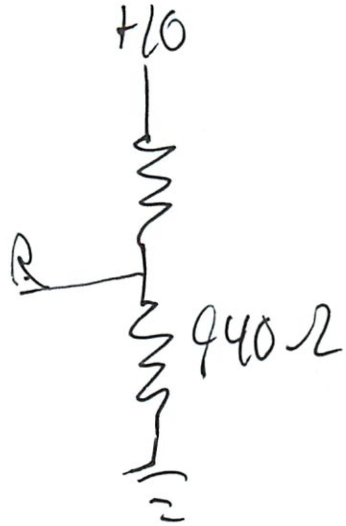
$$\frac{L}{R+L}$$

- keep right to 5V
- So we want $\frac{L}{R+L} < 5V$ turn right ? - yeah since 5 dominates

$\frac{L}{R+L} > 5V$ turn left

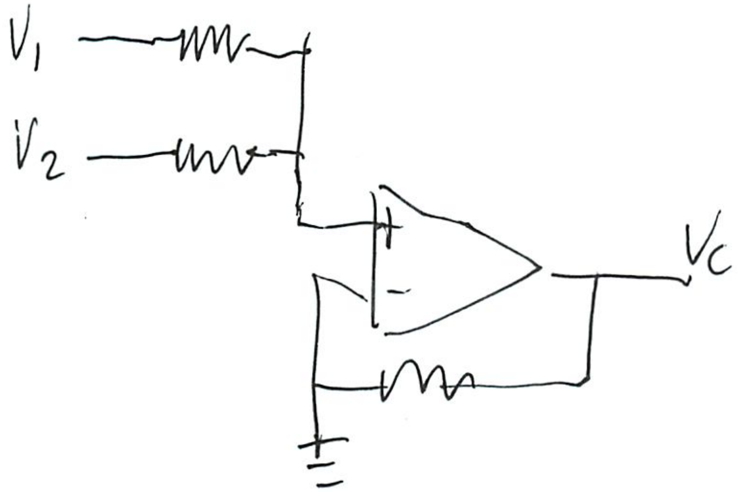
but they gave us $\frac{L}{R+L}$ so now just need to big

22



how to add voltages?

- amplifer w/ feedback loop

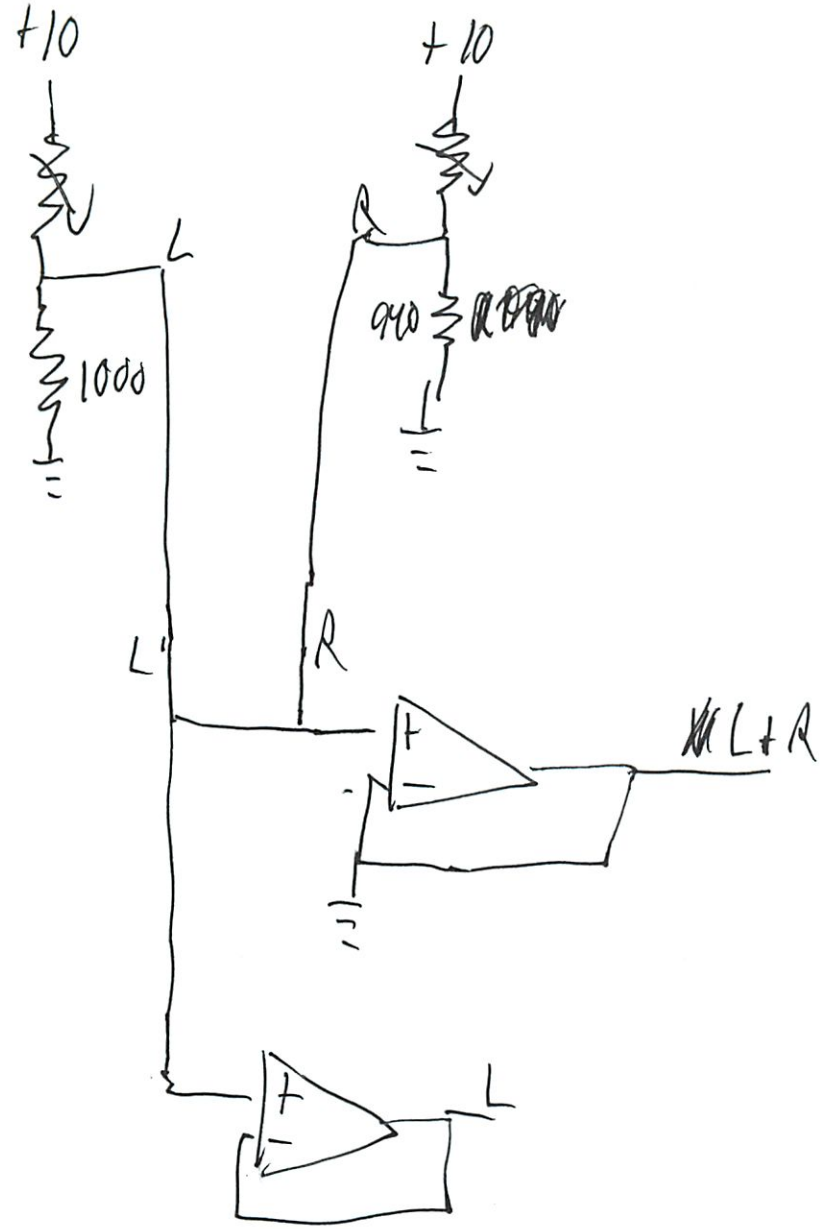


$V_c = V_1 + V_2$

then need to do V_L on its own as well

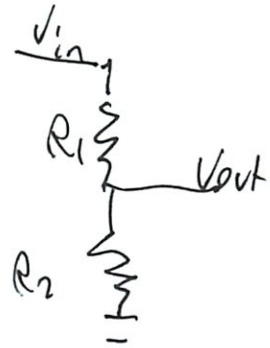
and divider - resistors in series

(23)



; but are resistors the "division" I am looking for?

Yeah WP!

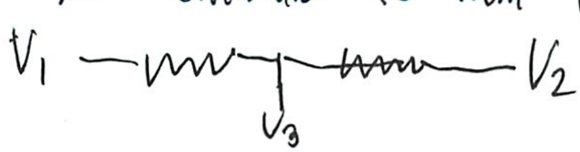


$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

but I have voltages, not resistances

interesting! or can you calculate R from V_1 V_2

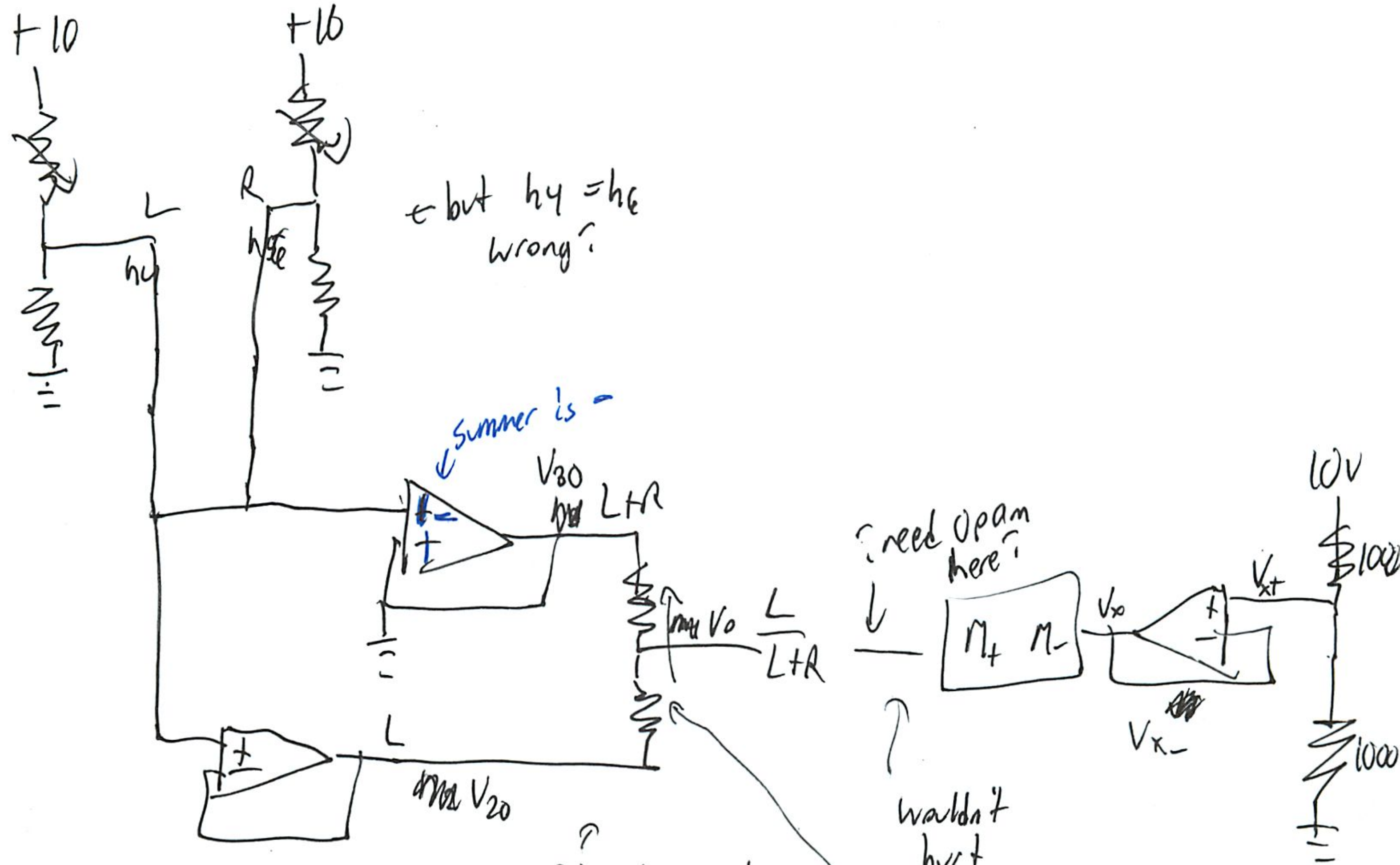
other site!



where $V_3 = \frac{V_1}{V_2}$

(24)

That makes sense somewhat



← but $h_4 = h_6$ wrong?

wouldn't hurt

? how do you know numerator
– that ans sounds suspicious from internet

and what are resistors? guess 1000

how try to build
I feel this is wrong
- singular matrix
- wait for OH
- is tutor hw will do another day
- read course notes
- blue fix
  - still singular matrix

(25)

How to do sum h_1 + h_2 w/o it overlapping?

Or is that ok

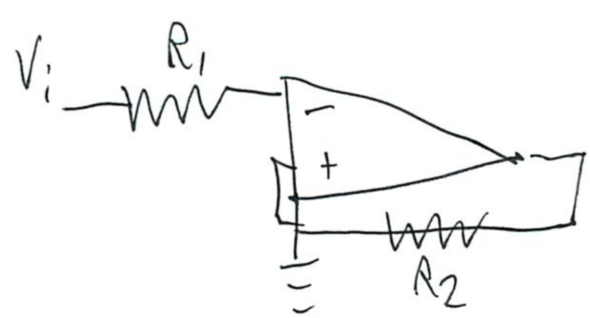
- but not for L
- have some R_i ~~each~~ and R_F that are =
- idk will just ask

Prof reply

10/31 OH

- several diff ks
- op amp = voltage controlled voltage source
- op amp circuits implement aritmetic function
  - like "inverting amplifer" has gain $k = -\frac{R2}{R_1}$ ← this is k
- I used op-amp as a "unity buffer" ← gain = 1
- Use a different op-amp design
- I'm guessing inverting amplifer to change
- so add R_2 and R_1s

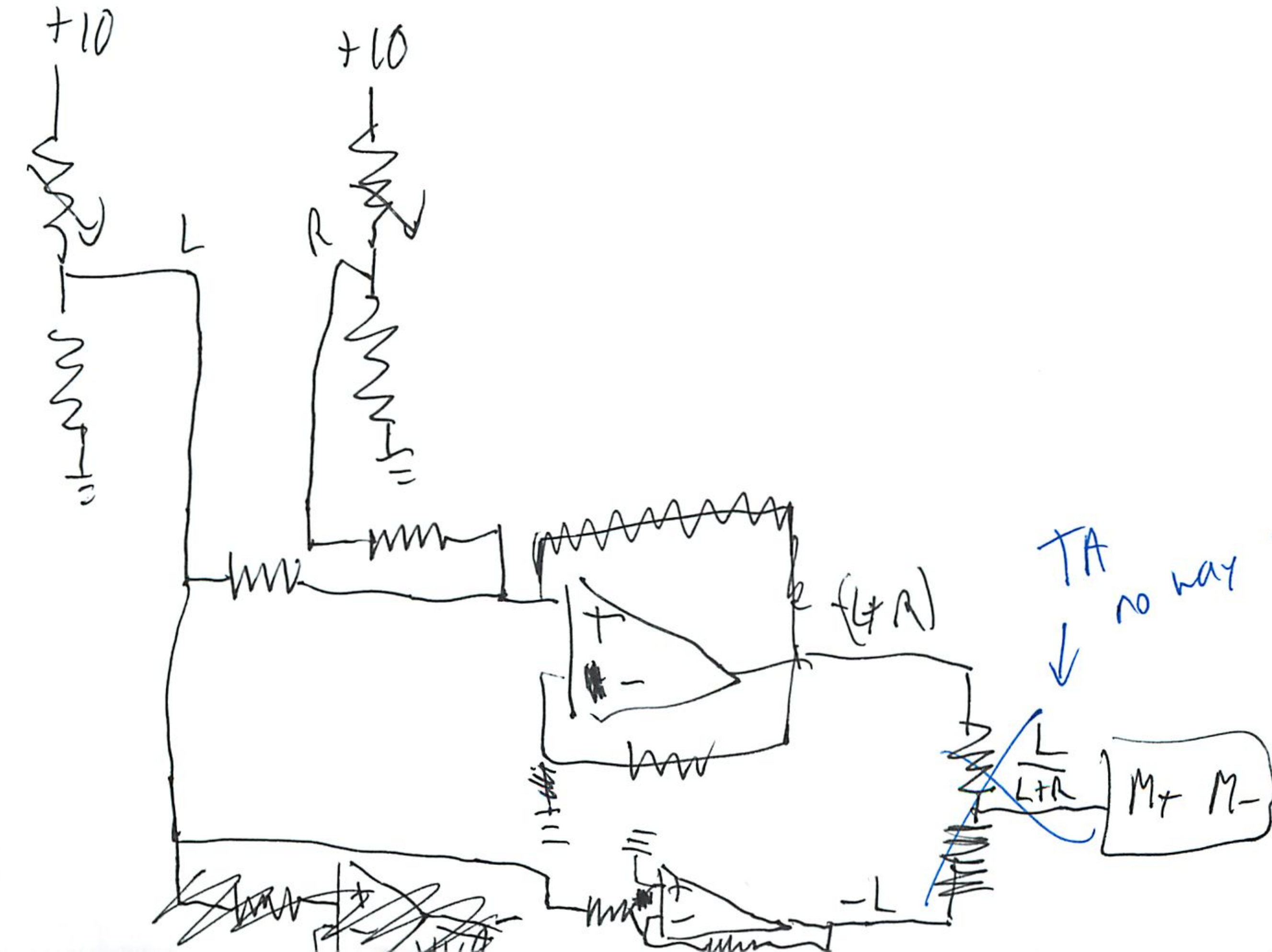
(26)



but inverts – don't want it to do ?
but prof said pick something different
although $\dfrac{-L}{-(L+R)}$ would be positive

---

need resistor b/w L and R

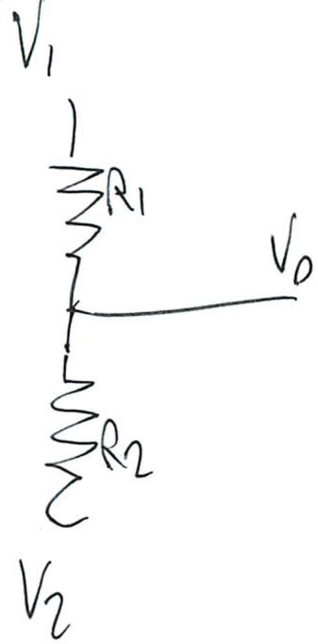


TA no way to do this

... etc same

27

What is ~~Votage~~ voltage divider



$$V_0 = \frac{R_2}{R_1+R_2} V_1$$

but $V_2 \neq 0$

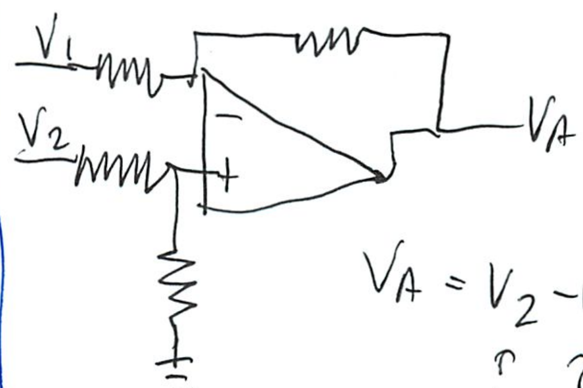
$$V_0 = \frac{R_2}{R_1+R_2}(V_1 - V_2)$$

not what I thought

TA: Says tend towards subtractor solution

L-R

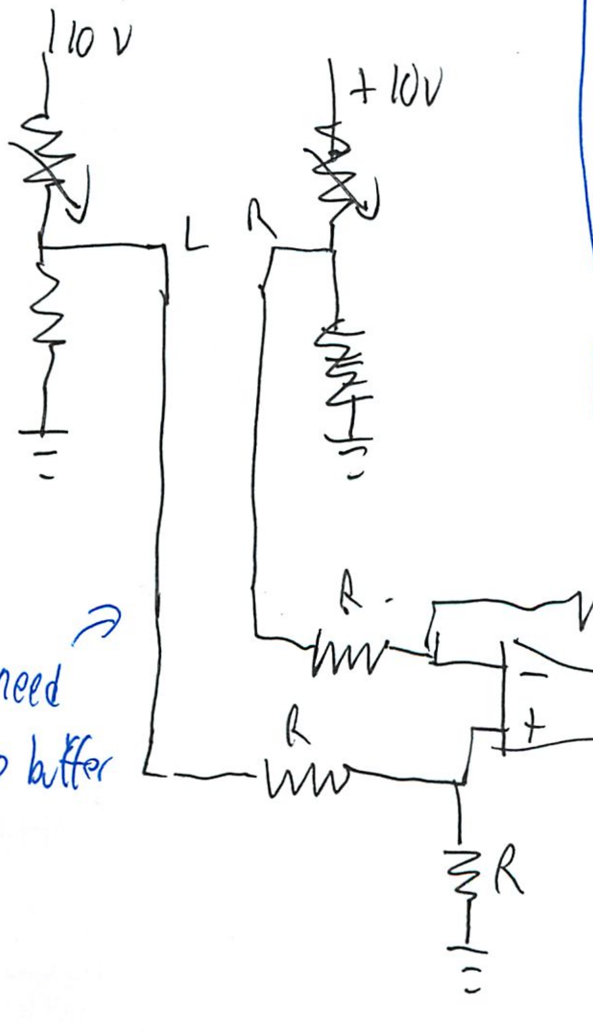
- So op amp as subtractor

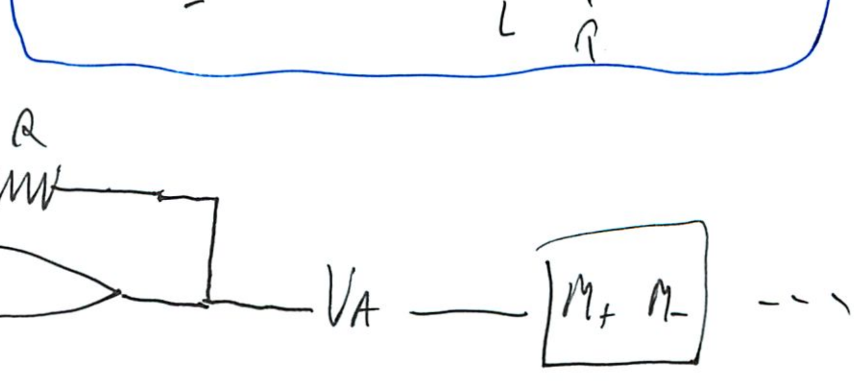


all R are same

$$V_A = V_2 - V_1$$

L R

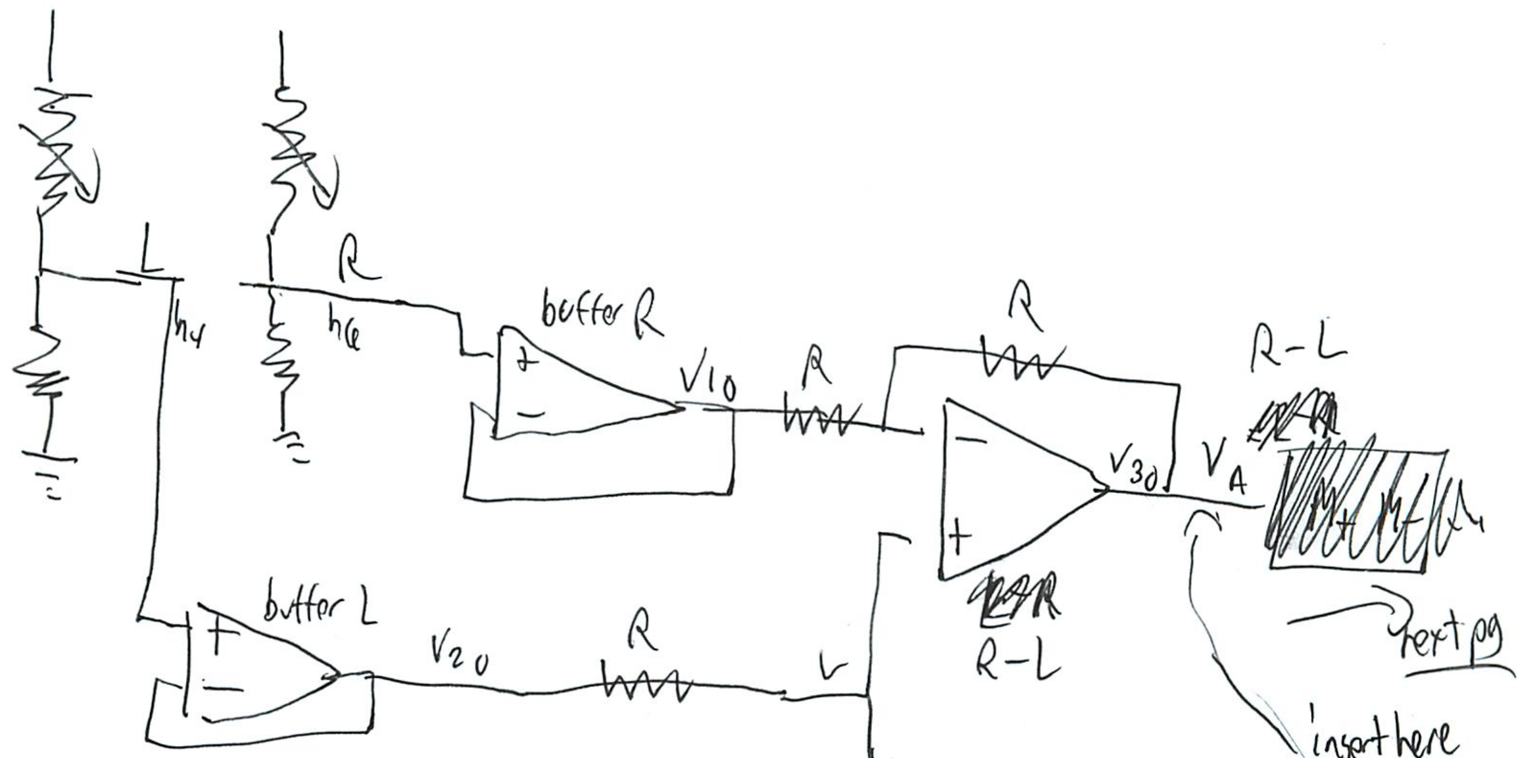




need to buffer

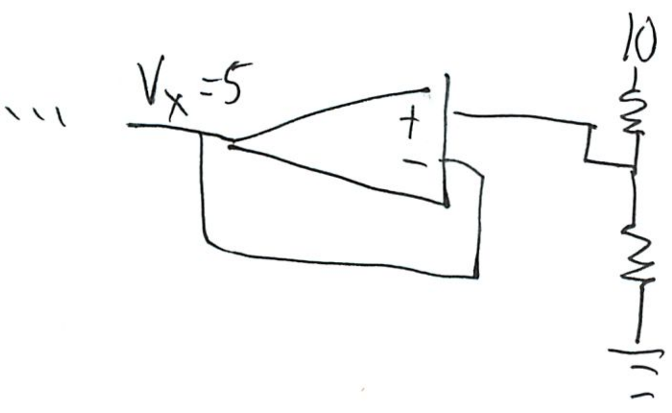
seems easier

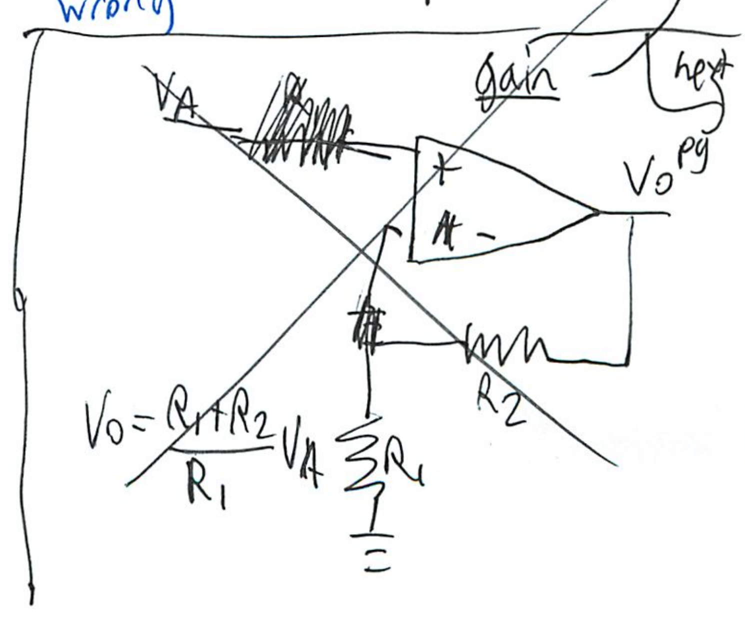
(28)



L & R are same
except want
gain - but where
changes
- run through noninverting
amp

Centered around
5 - slides
wrong

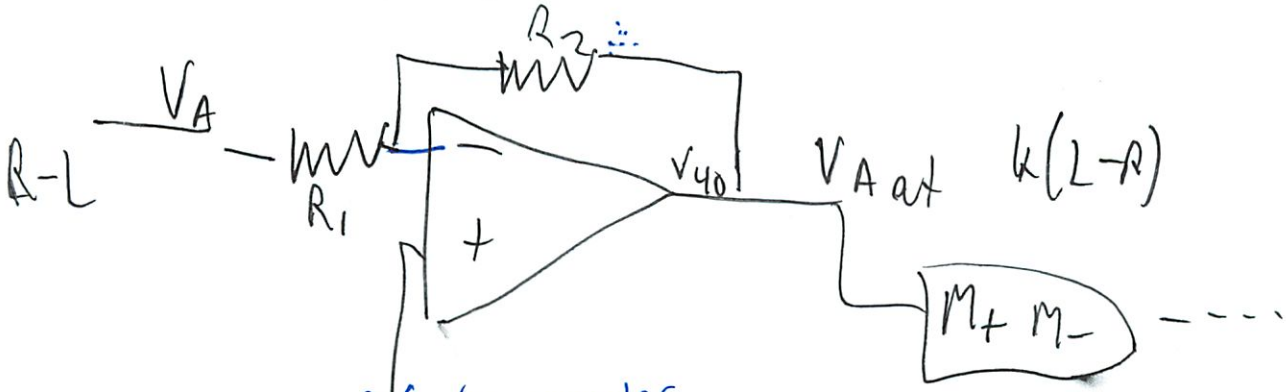




$V_0 = \frac{R_1+R_2}{R_1} V_A$

(29)

Add inverting gain for gain and to invert



V_A at $k(L-R)$

~~ground~~ V_x ← center
around here too!

$$V_0 = -\frac{R_2}{R_1} V_i$$

$$k = \frac{R_2}{R_1}$$

↑ now need something to make gains actually work

---

just V divider

V_{20} varry b/w ~~[illegible]~~ 3−7

pull to near 5

↖ always half way in middle if R=R

So
$V_{20} = 7$ $V_+ = 6$
$V_x = 5$

$V_{20} = 3$ $V_+ = 4$
$V_x = 5$

} what we want

(30)

✓ works – now need gains 1, 5, 10

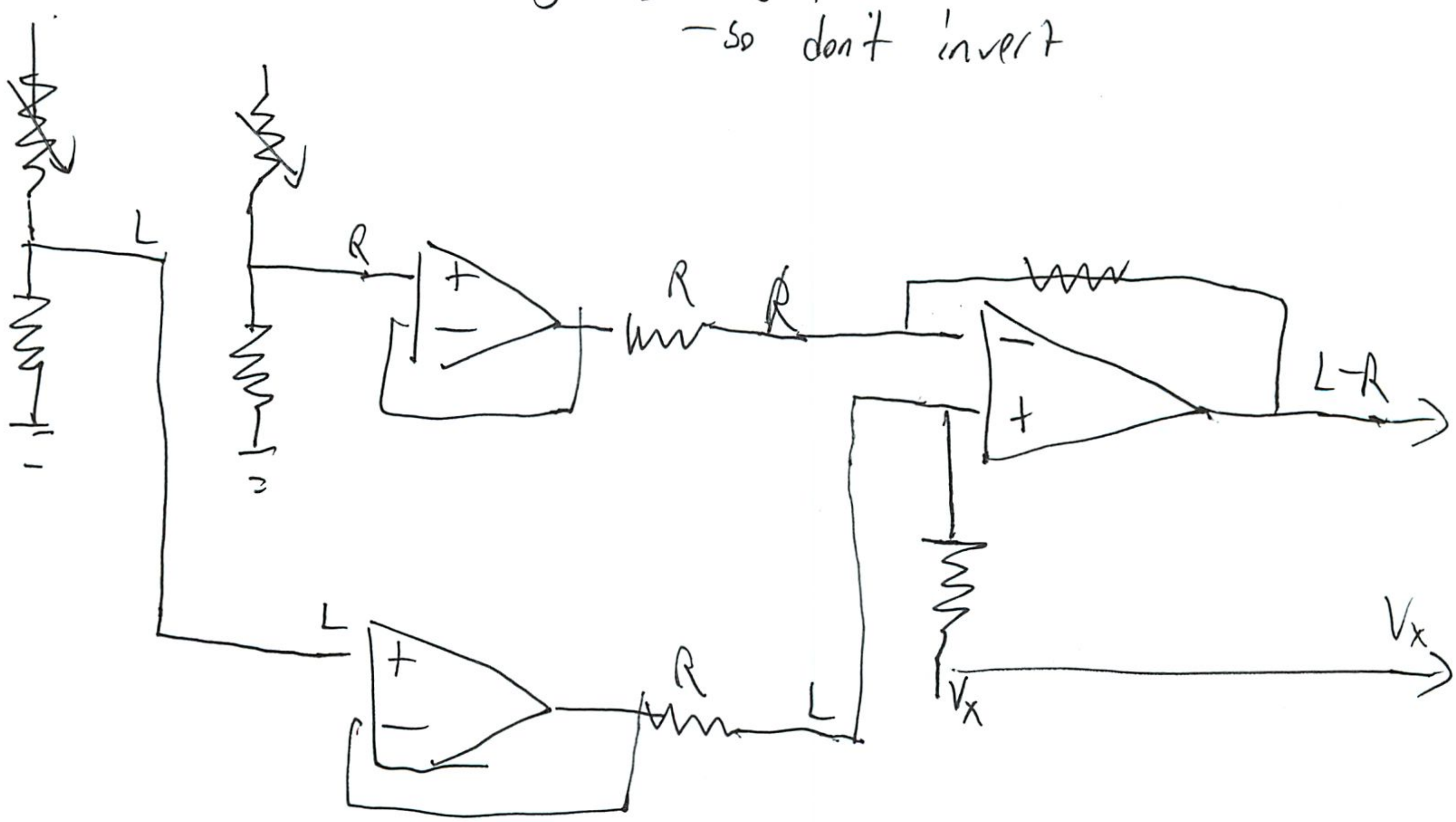
distance 1, 3

but $5 = \frac{R_2}{R_1} = \frac{5000}{1000}$ seems to hit wall

$R_1 = 5000$ $R_2 = 1000$ seem to works

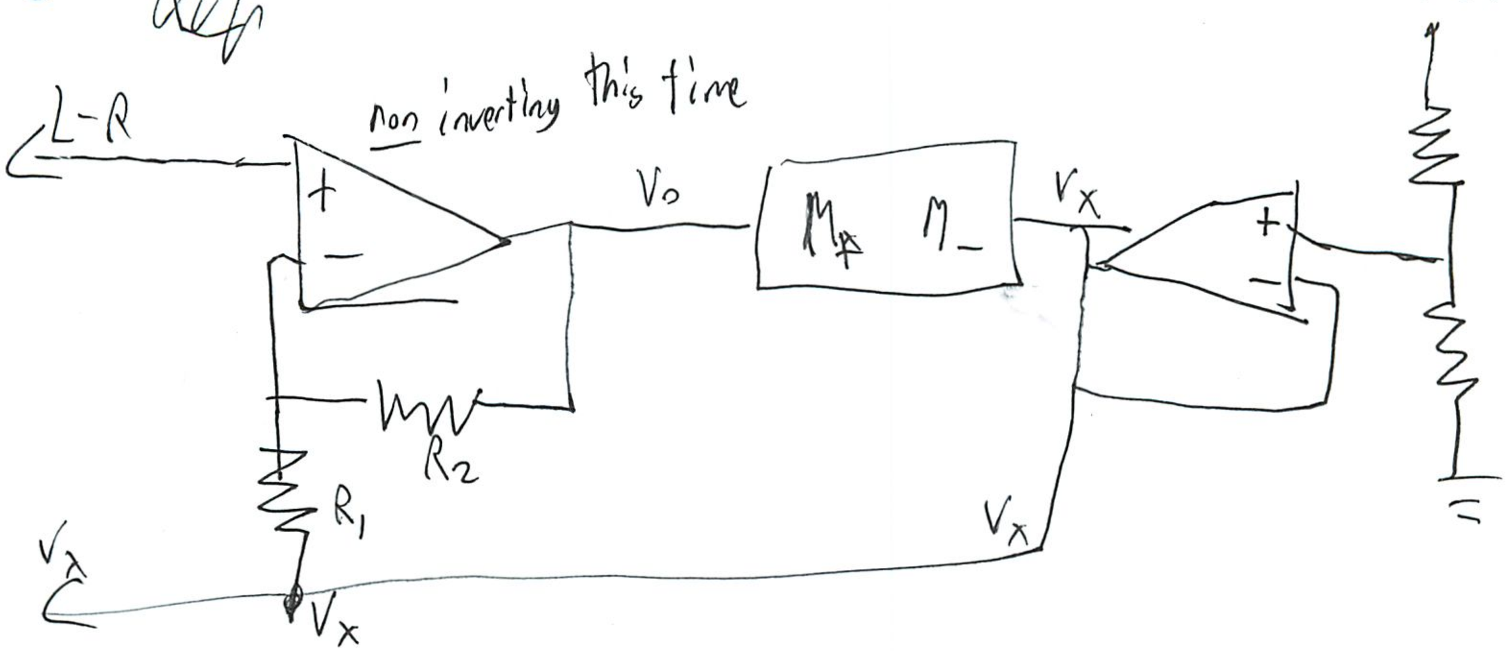
Oh I was doing L–R !

– so don't invert



(31)

non inverting this time



$$V_0 = \frac{R_1 + R_2}{R_1} V_i$$

? k

- oh need $R_2 = 0$ for $k = 1$

$k = 5$

$R_1 = 1000$

$R_2 = 4000$

$k = 10$

$R_1 = 1000$

$R_2 = 9000$

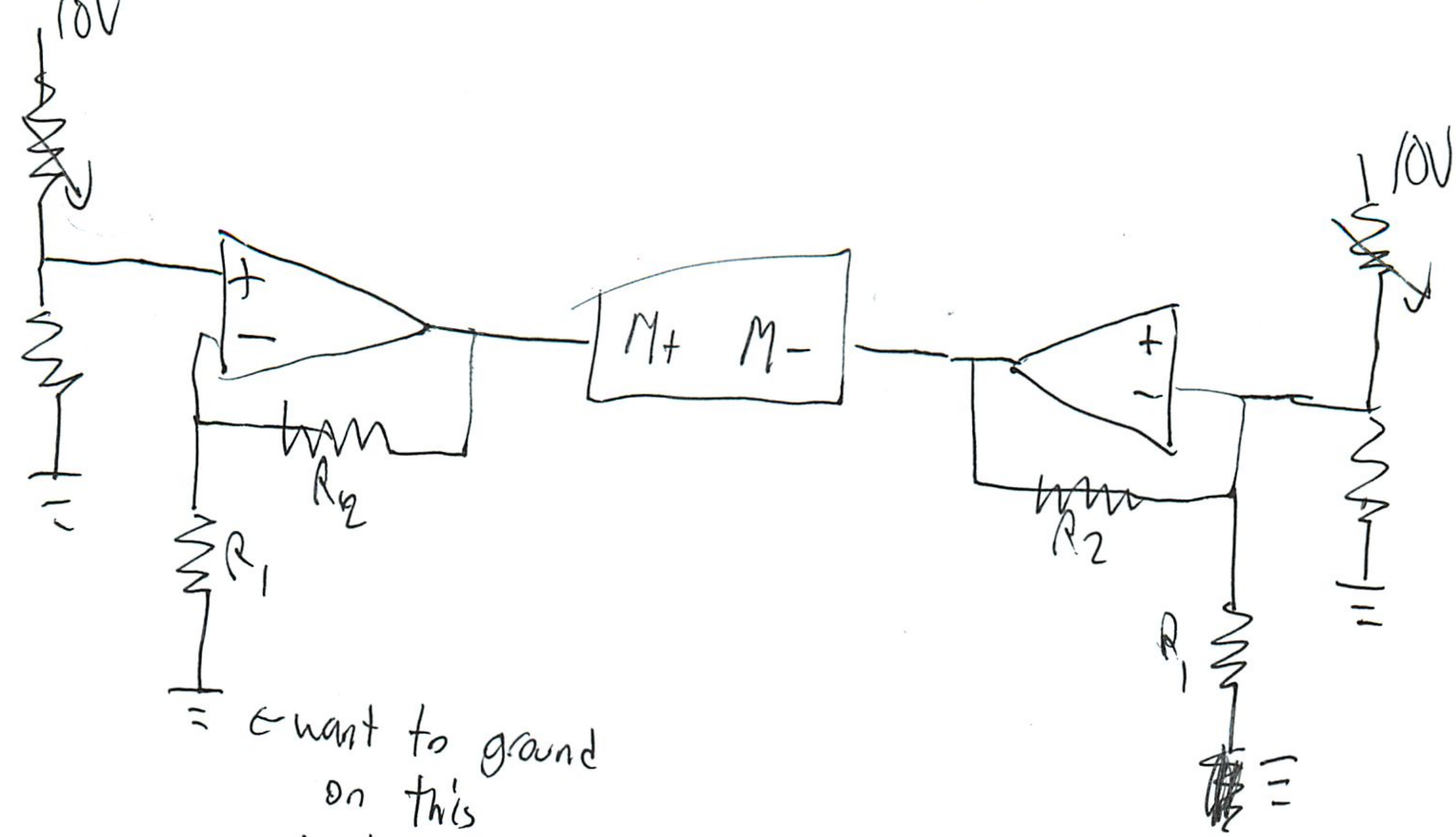
Checkoff 4 ✓

I really understand it much better!

32

Now back to Checkoff 3

Need to add gain - use non inverting



← want to ground on this

0 is our ref

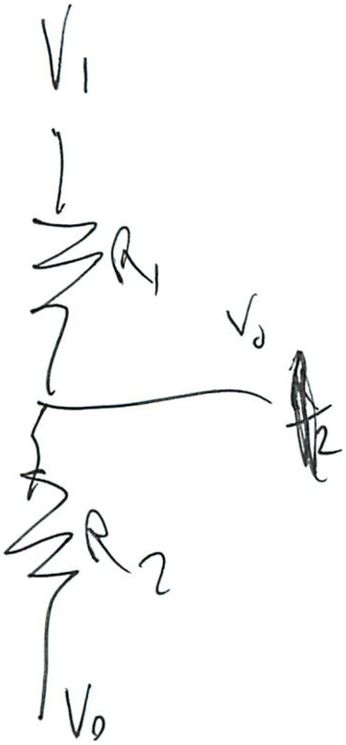
$$k = \frac{R_1 + R_2}{R_1}$$

$k = 1$ $R_1 = 1000$
$R_2 = 0$

$k = 10$ not work?

- yeah supposed to
- test where stop working $k = 5$ max

$k = 5.5$



$$V_o = \frac{R_2}{R_1+R_2}(V_1 - V_2)$$

bright light = ~~high~~ low resistance

Bri

$V_1 = 10$

$V_2 = 0$

$R_1 = 10000$ 1

$R_2 = 1000$ = 10

$$V_o = \frac{10}{11} \cdot 10 \approx 9.5$$

Is it accurate? does not point at light

- gain of 2 is most accurate

motor pot = .33, .66

34

Not very uniform

- kinda uniform on gain 5
- not on 1

6.01 HW week 8 10/30

Name that node

look at photo and match scematic to photo of real Circuit bord

So for each point on photo

a) A

b) pin 1 of opamp → resistor D

c) pin 1 of op amp C

d) none

e) pin 7 of opamp B

f) pin 8 op amp C

g) pin 7 of opamp B

h) ~~pin 7 + Resistory~~ +10 A

i) Ground E

all right 1st try

(2)

Summer Vacation

- ans should be ∦
- -> its the relationship b/w the ∦

$$V_{out} = -\frac{R_F}{R_I}(V_1 + V_2)$$

$$V_{out} = -\frac{R_F}{1}V_1 + \frac{R_F}{1}V_2 \qquad R_I = 1$$

no - R_F must be same

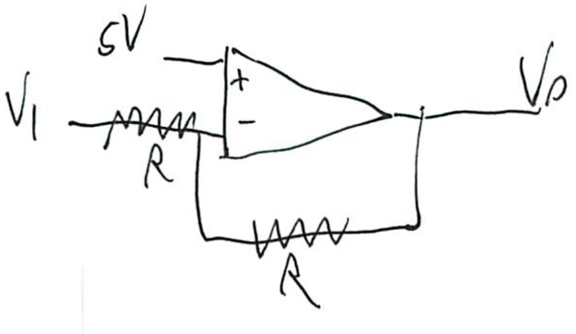
$$\frac{6}{3 \leftarrow R_A} \leftarrow R_F/R_C \rightarrow \frac{6}{2 \leftarrow R_B}$$

nice

then $R_A = RC/2$
$RB = RC/3$
$RC = RC$

#3 Opamp practice

Part 1



want $V_+ = V_-$

$$5V = \frac{V_1 - V_-}{R}$$

$$\frac{V_o - V_-}{R} = ?$$

$$V_o = k(V_+ - V_-)$$

③ So what to do again?
-confused

remember $V_+ = V_- = 5V$

$$5V = \frac{V_1 - V_+}{R} = \frac{V_0 - V_+}{R}$$? can you say this?

-no =5V

$$\frac{V_1 - 5V}{R} = \frac{V_0 - 5V}{R}$$? am I putting curent in right dir?

$$R(V_1 - 5V) = R(V_0 - 5)$$

$$V_1 = V_0$$?? ⓧ wrong

Think back to now

$V_+ = 5V$

$$\frac{V_- - V_i}{R} = \frac{V_0 - V_-}{R} = ?$$

?0? said net flow =0

find V_i

know $V_+ = V_- = 5$, right?

$$\frac{5V - V_1}{R} = \frac{V_0 - 5V}{R}$$ ①

? so same, but other direction?

(1)

$5V - V_1 = V_0 - 5V$

$V_0 = 10V - V_1$

So $V_1 = 4,\ V_0 = 6$

$V_1 = 8,\ V_0 = 2$

$V_1 = 10,\ V_0 = 0$

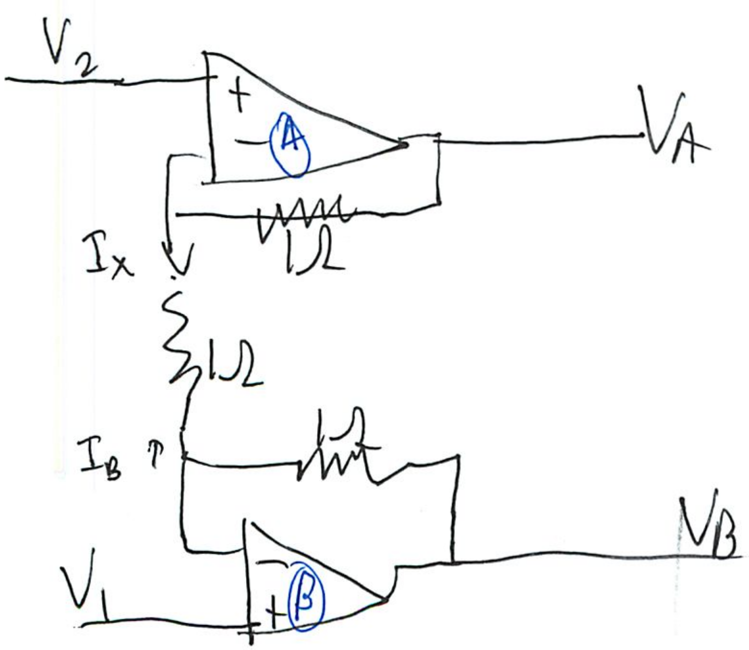
So what did I fix?

- consistant current direction $\leftarrow$

Could have done $\rightarrow$

$$\frac{V_1 - V_-}{R} = \frac{V_- - V_0}{R}$$

Part 2



(5)

Find I_x if $V_1 = 1$ $V_2 = 2$

Let me try getting NVCL right

Lable Op Amps

direction $\longrightarrow$

$V_2 = V_{A+} = V_{A-} = 2$

$\frac{V_{A-} - V_A}{1} = ? = ?\ \cancel{V_2}$? what does it connect to ? I_x

V_2

$V_1 = V_{B+} = V_{B-} = 1$

$\frac{V_{B-} - V_B}{1} = ?$? what $= V_1 ?$

$\frac{V_{A-} - V_{B-}}{1} = I_x$

$\frac{2-1}{1} = I_x = 1$ ✓ oh wow actually right

Determine V_A

$\frac{V_{A-} - V_A}{1} = \frac{2 - V_A}{1} = 2 ?$

$V_A = 0 ?$ ✗

(a)

ve know $I_x = 1$ $\frac{V}{R} = I$

$$\frac{V_{A-} - V_A}{1} = I_x$$

$$\frac{2 - V_a}{1} = 1$$

$V_A = 2$ (X)

---

or loop like before!

$$\frac{V_2 - V_i}{R} = \frac{V_0 - V_-}{R}$$

$$\frac{2 - \cancel{V_x}}{0} = \frac{V_A - V_x}{1}$$

$V_{Ax} = 2 = V_2$

0 ? no, resistance

- but can't do /0
- or do 1 for the other one

$$\frac{2-2}{0} = \frac{V_a - 2}{1} = 0$$

$V_a - 2 = 0$ $V_a = 0$ (X) tried already

(7)

How much does bottom play a roll?

try other direction

$$\frac{V_A - V_{A-}}{1} = -I_x \qquad V_{A+} = V_{A-} = 2V$$

$$\frac{V_A - 2}{1} = -1$$

$$V_A - 2 = -1$$

$$V_A = 1$$

How much does bottom play a roll

$V_{B-} = 1$

~~A~~ V

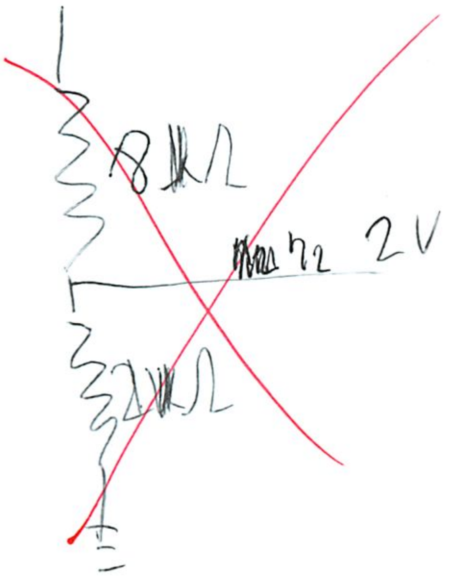
?. Confused, not focusing

next need as coeff

Try entering it into circuit test

- but only can do 10 V
- Could do Resistor trick

(8)



don't need
- use SW lab 8 code
- not ~~des~~ desLab 8

It gives $V_A = 3$ ✓

Now find pattern

$V_2 = 3$	$V_1 = 1$	$V_A = 5$
4	1	7
4	2	6
4	3	5

$V_A = -V_1 + 2V_2$ ✓

I totally cheated through that, but why does it work like that?

Is $V_B = 2V_1 - V_2$? yeah

Ask in OH - 3rd thing

Also the simular reports I_x in wrong dir

9

When $V_1 = V_2$ $I_x = 0$

~~When $V_1 > V_2$ I_x~~

$I_x = V_2 - V_1$ w/ tutor sign convention

but why all this?

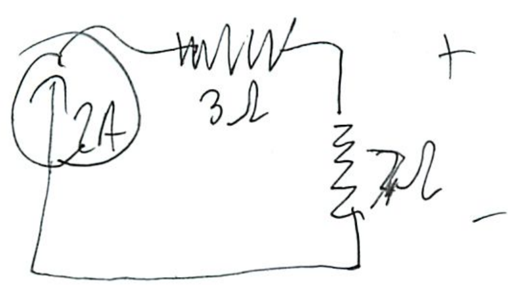
- ask

See after these notes

Voltages

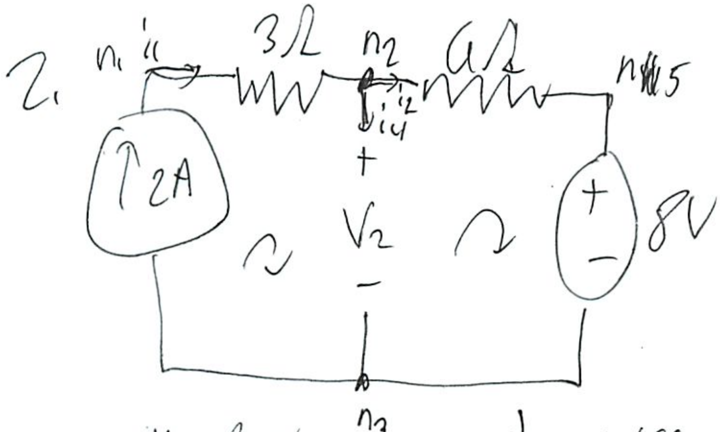
$V = IR$

1.



$V_1 = 2 \cdot 7 = 14V$

2.



heres where it gets tricky 2 lectures ago

~~$V = 2/3$ same + drop~~

do NVCC

$i_1 = i_2 + i_4$ $V_3 = 0 = n_3$

$\frac{n_1 - n_2}{3} = i_1$

(1c)

~~high low~~ ← no must do same dir

$$\frac{n_2 - n_5}{6} = i_2 \qquad n_5 - n_3 = 8$$

$$i_4 = 0 \qquad V_2 = n_2 - n_3 = \text{want}$$

$$\frac{n_1 - n_2}{3} = \frac{n_2 - n_5}{6}$$

$$\cancel{6n_1 - 6n_2 = 3n_2 - 3n_3}$$
$$\cancel{6n_1 = 9n_2}$$
$$\cancel{2n_1 = 3n_2}$$

$n_5 = 8$

$$6n_1 - 6n_2 = 3n_2 - 3n_5$$
$$6n_1 = 9n_2 - 24$$

now how does 2 am play in

$$\frac{n_1 - n_2}{3} = 2$$
$$n_1 - n_2 = 6 \checkmark$$
$$n_1 = 6 + n_2$$

← oh opps sign error when copying

$$6(6 + n_2) = 9n_2 - 24$$

18

$36 + 6n_2 = 9n_2 - 24$

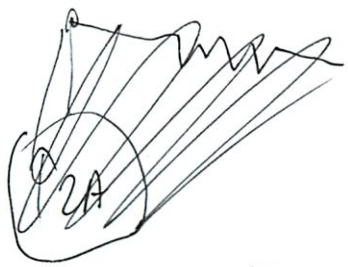
$+6n_2 \qquad +24$

$60 = \cancel{15}\,3n_2$

$n_2 = \cancel{4}\,20$ ✗ ✓

I am always confused when both current + voltage source

Could build simulation again



remember its 1 circle - just want to measure v_1

$\boxed{\text{Ok computer print out}}$

$i(n_3 \to n_1) = 2$

$n_1 - n_2 - 3\,i(n_1 \to n_2) = 0$

oh $\dfrac{n_1 - n_2}{3} = i(n_1 \to n_2)$

$\dfrac{n_2 - n_5}{6} = i(n_2 \to n_5)$

(12)

$\cancel{n_3 - n_5}$ $n_5 - n_3 = 8$

$i(n1 \to n2) = i(n3 \to n1)$

$i(n2 \to n5) = i(n1 \to n2)$

$i(\cancel{n3 \to n5}$ $n_5 \to n_3) = i(n2 \to n5)$

$n3 = 0$

So answers

$i(n_1 \to n2) = 2$

$i(n2 \to n5) = 2$

$i(n3 \to n_1) = 2$

$i(n3 \to n5) = +2$

$n_1 = \cancel{10}$ 26

$n_2 = \cancel{4}$ ← they got 4 too! 20

$n_3 = 0$

$n_5 = +8$

Oh my voltage source was backwards =20

So what did I do wrong

$n_5 - n_3 = 8$

$+ \quad -$

← right makes sense

$n_3 = $ gnd $= 0$

$n_5 = 8$

* always $+ \to -$

had that when worked manually

13

I called $n_1 \to n_2 = i_1 = n_3 \to n_1 = 2$

$n_2 \to n_5 = i_2$

$n_2 \to n_3 = i_4$

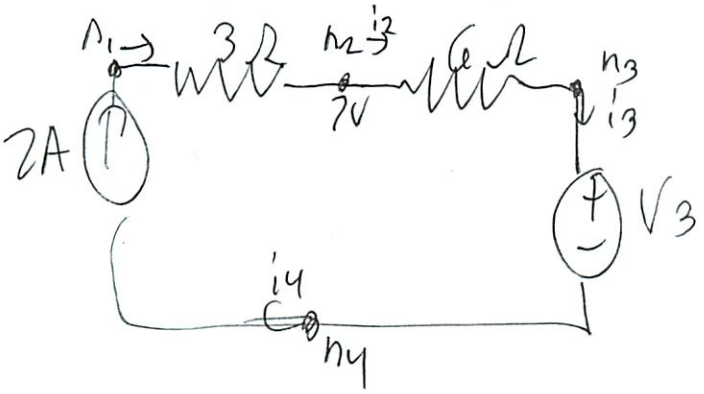
oh I never wote this
— Yes I did

Oh I see sign error notes p 10

So I did everything right, I just flipped sign.

---

3. Very similar → see if I can do



Note this + last one were just one circuit

That made last problem harder
Should still be able to solve

I should be same $i_1 = i_2 = i_3 = i_4$ since 1 circuit

~~$n_4 \to n_1$~~ $i_1 = 2A$

$$\frac{n_1 - n_2}{\cancel{2}\,3} = i_1$$ ← where did I get 2 from?

$$\frac{n_2 - n_3}{6} = i_2$$

(14)

$$\frac{n_1 - n_2}{\cancel{7}\,3} = \frac{n_2 - n_3}{6}$$

$$6n_1 - 6n_2 = \cancel{7}^{3} n_2 - \cancel{7}^{3} n_3$$

$$6n_1 = \cancel{13}^{9} n_2 - \cancel{7}\,3 n_3$$

and I dropped a coefficent there

$$n_2 = 7V$$

$$\cancel{n_1} = n_4 = 0$$

right

$$0 = \cancel{13}^{9}(7) - 3n_3$$

$$n_3 = \cancel{91}^{21} V$$

- seems way to high

Computer simulator guess + check says 5V

note - yeah want $n_3 - n_4 = V_3$ ($n_4 = 0$)

but voltage sorce is $n_4 - n_3 = V_3$

So want -3

- still

Check my math

Oh and I had Vsorce wrong again

~~but can't get~~

but can't get V_2 down to 7

unless go negitive $-5V$

(15)

So its −5 − but why can I not get that

$$\frac{n_1 - n_2}{3} = \frac{n_2 - n_3}{6}$$ ✓ testing w/ calculator ans

$$6n_1 - 6n_2 = 3n_2 - 3n_3$$ ✓

36, 21, −15

$$6n_1 = 9n_2 - 3n_3$$ ✓

78

want $n_2 = 7$

Oh my mistake $n_1 \neq 4V$

(why is this so error prone)

current source has 13 V gain over it!

!!!

$$6n_1 = 9(7) - 3n_3$$

↑ want n_3

$$n_3 = \frac{63 - 6n_1}{3}$$ ✓

$$n_3 = 21 - 2n_1$$ ✓ tests at

↑ find w/ eq resistance

$V = 2 \cdot 9$

$V = 18$

⊗ this should be 13

Oh $n_1 = 2 \cdot 9 + n_3$ ⊗ ~~wrong~~ was right

(16)

~~$n_3 = 21 - 18 - n_3$~~

~~$2n_3 = 3$~~

~~$n_3 = \frac{3}{2}$ (x) not right either~~

~~looking for -5~~

So my n_1 is wrong

want $13 = 2 \cdot 9 \pm -5$

$-18 \quad -18$

$-5 = -5$

so (+) was right

Oh I see

$n_3 = 21 - \underline{2}\, n_1$

$n_3 = 21 - 2(18 + n_3)$

$n_3 = 21 - 36 - 2n_3$

$3n_3 = -15$

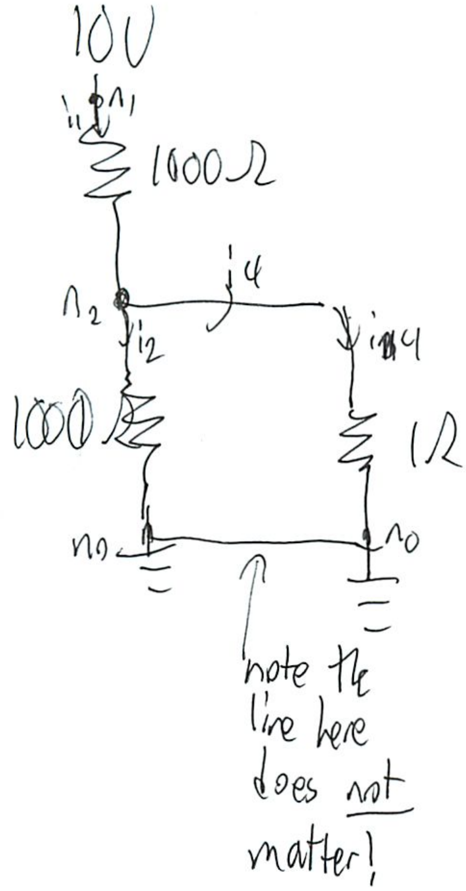
$n_3 = -5$

So made 1 mistake $n_1 \neq 4$

and then copy errored the $\underline{2}(n_1)$

(17)

4.



Want I_4

$i_1 = i_2 + i_4$

$$\frac{n_2 - 0}{1000} = i_2$$

$$\frac{10V - n_2}{1000} = i_1 \qquad \frac{n_2 - 0}{1} = i_4$$

$$\frac{10V - n_2}{1000} = \frac{n_2}{1000} + \frac{n_2}{1}$$

$$\frac{10V - n_2}{1000} = \frac{1001 n_2}{1000}$$

(18)

$10V - n_2 = 1000 n_2$

$10V = 1002 n_2$

$n_2 = 100.2 V$

$i_4 = \frac{100.2}{1} = 100.2$ (X)

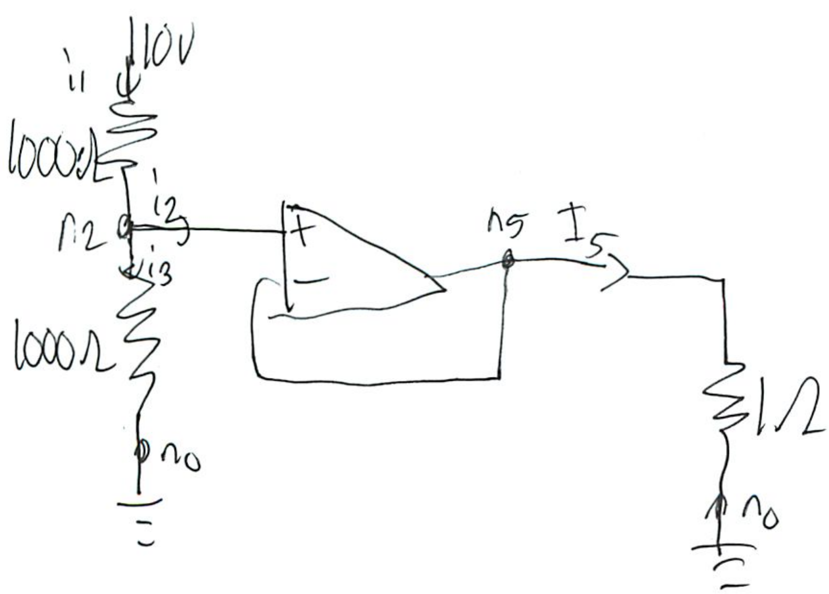
Wrong way division math error

$\frac{10}{1002} = \frac{5}{501} = .0099$ (✓)

[illegible]

finally - but even here stupid math error

5. First intuition : same
- no 5V I think since no loading effects
- do it at



(19)

$$\frac{10 - n_2}{1000} = i_1$$ $i_1 = i_2 + i_3$

$$\frac{n_2 - 0}{1000} = i_3$$

Op amp
- think back

$n_2 = V_+ = V_-$

$V_{out} = k(V_+ - V_-)$ ← not relevant here

well this is the simple $V_o = V_+ = V_- = V_5$

i_3 ? i_5 ← don't think so

$n_3 = n_5$ fresh current at that voltage

$i_2 = 0$ ← no current pull

$$\frac{10 - n_2}{1000} = \frac{n_2}{1000}$$ so will be different

$10{,}000 - 1000\,n_2 = 1000\,n_2$

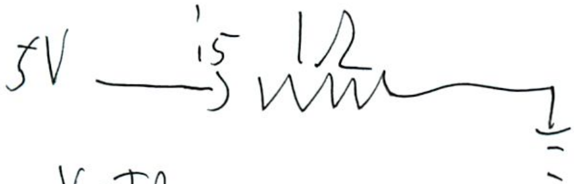
$10 - n_2 = n_2$

$10 = 2n_2$

$n_2 = 5$ ← like I predicted

(20)

Op amp 5V at what ever curent



$V = IR$

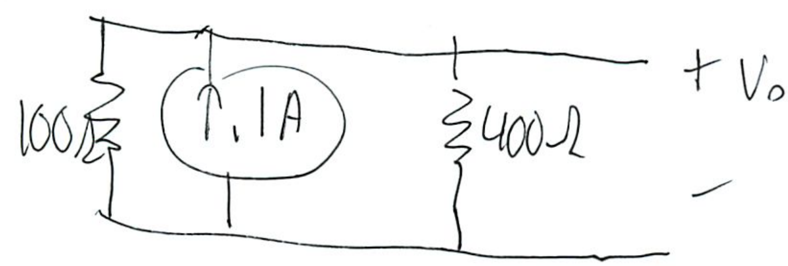
$5 = I \cdot 1$

$I = 5$ ✓ finally got one 1st try!

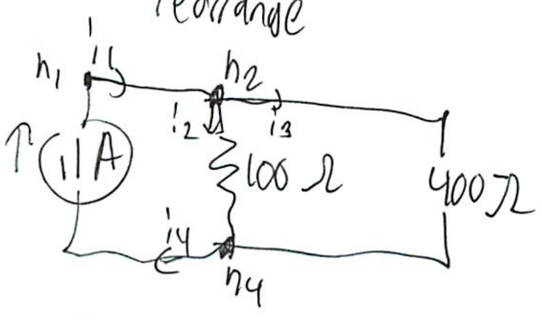
#5. Circuits

1. Find V_o

- oh no same type of problems)



reorrange



no tricks, nvcc

$i_1 = i_2 + i_3$

$i_1 = .1$

$$\frac{\overset{+}{n_2} - \overset{-}{n_4}}{100} = i_2$$

21

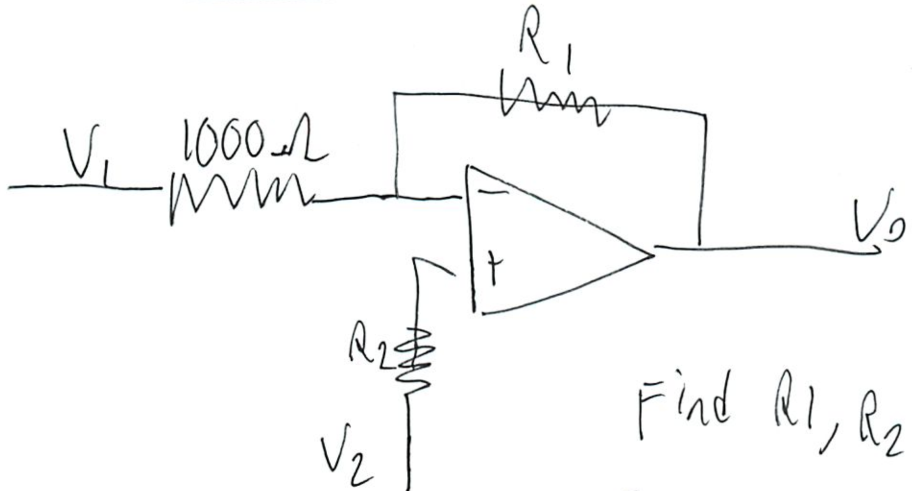
$$\frac{n_2 - n_4}{400} = i_3 \qquad n_4 = 0$$

$$i_1 = \cancel{} \frac{n_2}{100} + \frac{n_2}{400} \qquad \text{want } n_2 - n_4 = n_2$$

$$i_1 = \frac{n_2}{80}$$

$n_2 = 8$ ✓ woot another one

2.



Find R_1, R_2, V_2

So $V_0 = 1 - \frac{V_1}{4}$

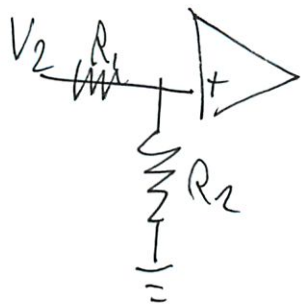
~~look up patterns~~ try nvcc

$V_- = V_+$

$$\frac{V_1 - V_-}{1000} = \frac{V_- - V_0}{R_1}$$

$\frac{V_2 - V_+}{R_2} = i$ I does not matter

(22)

if it was  it would be

$$\frac{V_2 - V_+}{R_1} = \frac{V_+ - 0}{R_2}$$

note different R_1, R_2

that line not there

R_2 like ∞

so this is like 0

just say = 1 then ??

$V_2 - V_+ = R_2$

$V_+ = V_2 - R_2$

$V_+ = V_{\cancel{a}} -$

$$\frac{V_1 - (V_2 - R_2)}{1000} = \frac{(V_2 - R_2) - V_0}{R_1}$$

$$R_1 V_1 - R_1 V_2 - R_1 R_2 = 1000 V_2 - 1000 R_2 - 1000 V_0$$
$$+R_1 V_2 \quad +R_1 R_2 \qquad +R_1 V_2 \quad +R_1 R_2$$

$$R_1 V_1 = (1000 + R_1) V_2 + (R_1 - 1000) R_2 - 1000 V_0$$

$$V_0 = \frac{(1000 + R_1) V_2 + (R_1 - 1000) R_2 - R_1 V_1}{1000}$$

$$V_0 = \left(1 + \frac{R_1}{1000}\right) V_2 + \left(\frac{R_1}{1000} - 1\right) R_2 - \frac{R_1 V_1}{1000}$$

(23)

Want $V_0 = 1 - \frac{V_1}{4}$

So find R_1, R_2, V_2 for that

~~V_2 = any~~

$V_2 = 0$

$V_0 = $ ~~$\left(\frac{2000}{1000}\right)$~~

$$1 - \frac{V_1}{4} = \left(\frac{R_1}{1000} - 1\right) R_2 - R_1 V_1$$ solving simlataneous eq

↑ R_1 needs to be $\frac{1}{4}$

$$1 = \left(\frac{1/4}{1000} - 1\right) R_2$$

$R_2 = -1$

$R_1 = 1/4$ ⊗
$R_2 = -1$ ⊗
$V_2 = 0$ ⊗

all of these are disasters!

diff than "real" R2

Or is that line to ground (what I called R_2) $= \infty$

So $\frac{1}{\infty} = 0$

(24)

$$\frac{V_2 - V_+}{R_1} = 0$$

$$V_2 - V_+ = 0$$

$$V_2 = V_+$$

now go back

$$\frac{V_1 - V_2}{1000} = \frac{V_2 - V_0}{R_1}$$

$$R_1V_1 - R_1V_2 = 1000V_2 - 1000V_0$$

$$R_1V_1 = (R_1 + 1000)V_2 - 1000V_0$$

want V_0

$$V_0 = \frac{(R_1 + 1000)V_2 - R_1V_1}{1000}$$

$$V_0 = \left(\frac{R_1}{1000} + 1\right)V_2 - \frac{R_1V_1}{1000}$$ ← oh never divided this out

$$1 - \frac{V_1}{4} = \left(\frac{R_1}{1000} + 1\right)V_2 - \frac{R_1V_1}{1000}$$

R_2 = any ✓

(25)

$$1 - \frac{V_1}{4} = \frac{R_1 V_2}{1000} + V_2 - \frac{R_1 V_1}{1000}$$

$R_1 = 250$ ✓

$$1 - \frac{V_1}{4} = \frac{250 V_2}{1000} + V_2 - \frac{V_1}{4}$$

$$1 = \frac{250 V2}{1000} + V_2$$

~~or did I mess this up and $V_2 = 1$~~

$V_2 = .8$ ✓

Failure analsis - only thing was not evaluating that missing line as ∞ Resistance and $\frac{1}{\infty} = 0$

#6. Period of Pole

- last problem
- Notes 6.4 - this is so un related!
- if altenates period is 2
  - period $= \frac{2\pi}{\Omega}$ ← angle of poll
- complex #, returns 2nd half - so just $\frac{2\pi}{\text{complex pole}(1)}$?

(26)

Oh wow worked on all pt 1

when did 15

- so when denom = 0

- so what does that mean period is?

- oh need to test if real, negitive → 2

↑ have

- but if real + positive → monotonic

↳ ~~like~~ I see return none

(✓) Done HW

Want more circuit pratice

Should make a list of ~~what~~ what to watch for

$V_+ - V_-$

↑ always!

line not there $R = \infty$

divide right way / change signs right / algebra / copying

Current sources have voltage change!

Still confused what op amp = when does not appear to = anything

well $i_+ \neq i_-$ but all of i_+ are same *

↳ i_-

did not get the connected op amps ← only thing still confused on

↑ why they are =

Part 2 again

10/31

$V_+ = V_- = V_2$

$$\frac{V_{A-} - V_A}{1} = I_x$$

$$\frac{V_{B-} - V_B}{1}$$

$$\frac{V_{A-} - V_{B-}}{1} = I_x$$

$$\frac{V_{A-} - V_A}{1} = \frac{V_{B-} - V_B}{1} = \frac{V_{A-} - V_{B-}}{1}$$

$$V_{A-} - V_A = V_{B-} - V_B = V_{A-} - V_{B-}$$

$$+V_{B-} \qquad +V_{B-} \qquad +V_{B-}$$

$$V_{A-} + V_{B-} - V_A = 2V_{B-} - V_B = V_{A-}$$

$$-V_{A-}$$

$$V_{B-} - V_A - 2V_{B-} - V_B = 0$$

never did 3 inequalities before

(2) Solve for V_A

$$\begin{array}{cccc} +V_A & +V_A & +V_A & \\ V_{B-} = 2V_{B-} - V_B + V_A = V_A \end{array}$$

need to get rid of !

$V_{A-} = V_{A+} = V_2$

$V_{B-} = V_{B+} = V_1$

$V_1 = 2V_1 - V_B + V_A = V_A$

? discegard extra info ? should disappear

should be V_2

---

try from start in case messed up

$V_2 - V_A = V_1 - V_B = V_2 - V_1$

Solve for V_A in terms V_1, V_2

$\cancel{V_2 = V_A}$? but how get rid of V_B ?

$$\begin{array}{ccc} -V_2 & -V_2 & -V_2 \\ -V_A = V_1 - V_2 - V_B = -V_1 \end{array}$$

$\cdot -1$

$V_A = V_2 - V_1 + V_B = V_1$ $\quad V_A \neq V_1$

(3)

Or just ~~V_A~~ first 2

$V_2 - V_A = V_1 - V_B$

~~$V_2 = V_1$~~

~~$V_2 = V_1 - V_B + V_A$~~

$-V_A = V_1 - V_B - V_2$

$V_A = -V_1 + V_B + V_2$ (X) won't get us there

V_B must $= V_2$; but can't see that being true

Or sign is wrong

$$\frac{V_A - -V_A}{1} = -\left(\frac{V_B - -V_B}{1}\right)$$

~~V_A~~ $V_2 - V_A = -V_1 + V_B$ solve V_A

$-V_A = -V_1 + V_B - V_2$

$V_A = V_1 - V_B + V_2$

; now $V_B = -V_2$

wrong

(4) With TA

$$V_2 = V_{A+} = V_{A-}$$

$$\frac{V_{A-} - V_A}{1} = I_x = \frac{V_2 - V_A}{1}$$

$$V_1 = V_{B+} = V_{B-}$$

$$\frac{V_{B-} - V_B}{1} = I_B \quad \text{Not } I_x, -I_x \quad = \frac{V_1 - V_B}{1}$$

$$\frac{V_{A-} - V_{B-}}{1} = I_x = \frac{V_2 - V_1}{1}$$

Set = $\frac{V_2 - V_A}{1} = \frac{V_2 - V_1}{1}$

$$V_2 - V_A = V_2 - V_1$$

$$-V_A = V_2$$

TA suggests $\frac{V_A - V_B}{3} = I_x$ ← but has V_B

(5) Work backwards

$$V_A = 2V_2 - V_1$$

$$V_A + V_1 = V_2 + V_2$$

$$V_2 - V_1 = V_A - V_2$$

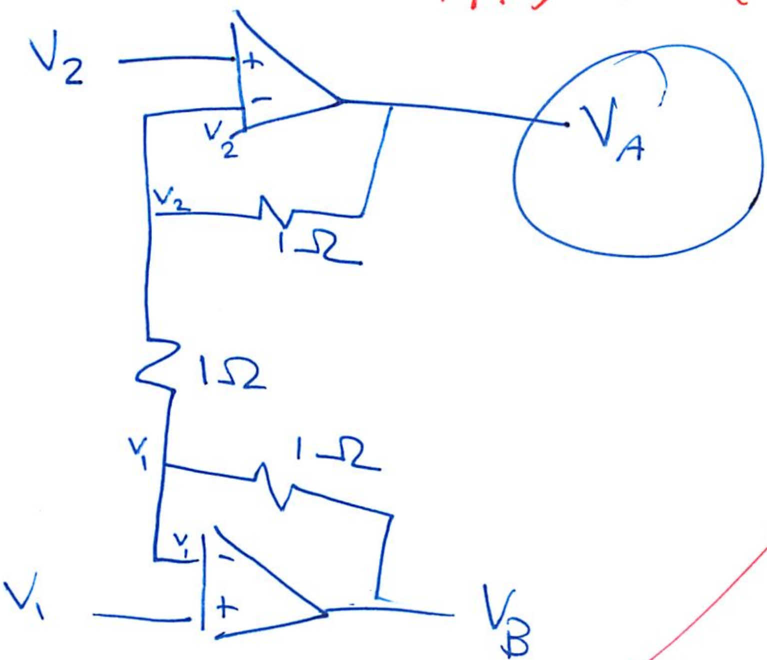
? so this was wrong way

look at direction

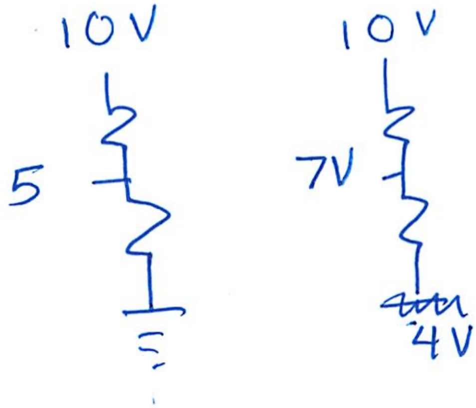
need to decide which current will flow

6

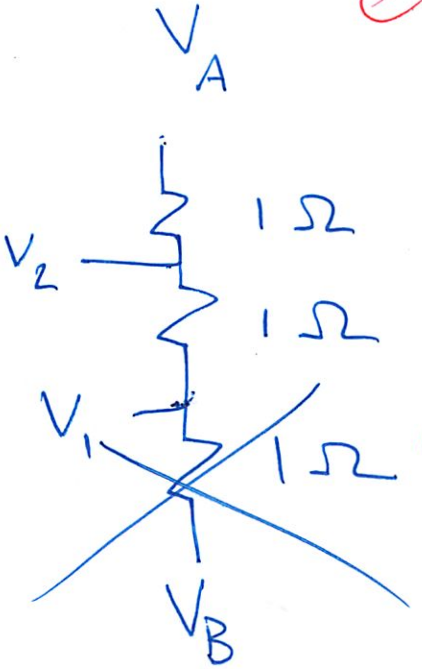
TA's work



think of it as the 3 resistors



← straighten out



← don't care here

or → $V_2 - V_1 = (V_A - V_1) \cdot \left(\frac{1}{2}\right)$

$V_2 = (V_A + V_1) \times \left(\frac{1\,\Omega}{2\,\Omega}\right)$

$V_2 = \frac{1}{2} V_A + \frac{1}{2} V_1$

$2V_2 = V_A + V_1$

$V_A = 2V_2 - V_1$

$V_2 + (V_2 - V_1) = V_A$

have	have	want
3	5	7
4	6	8

think more about this
still don't get why my method did not work - something w/ direction

6.01: Introduction to EECS I

Circuit Abstractions

Week 9 *November 2, 2010*

Reading: 7.7

PCAP Framework for Managing Complexity

Managing complexity in Python and in Signals and Systems.

Python:

- **procedures** are abstractions that combine primitive operations to capture common patterns
- **classes** are abstractions that associate data (attributes) and procedures (methods) that are related

Signals and Systems:

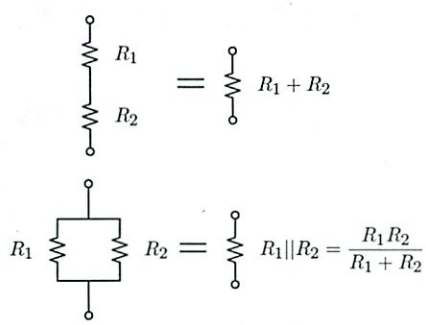
- **signals** are abstractions that collect all of the samples at different times into a single object
- **system functions** are abstractions that combine operations (delays, gains, and adders) to capture common patterns

Today: managing complexity in design and analysis of circuits

I came in 5min late

Equivalent Resistors

Series or parallel resistors can be replaced by a single equivalent resistor.



Replacing two elements by one reduces conceptual complexity.

Circuit Abstractions

Today: Similar simplifications for other types of elements

The key concept for generalizing from equivalent resistors to equivalent circuits is the **current-voltage relation**.

cool graph

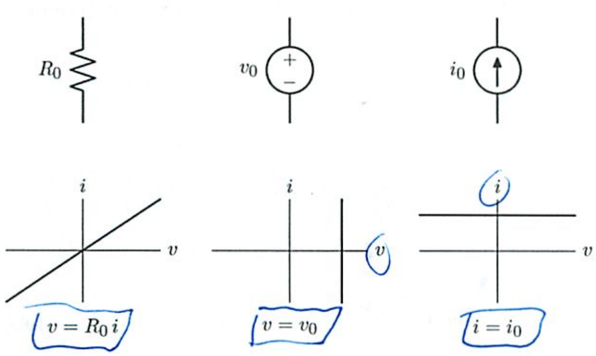
$i = \frac{1}{R} v$

linear

Two circuits are "equivalent" if they have the same current-voltage relation.

Current-Voltage Relations

The current-voltage relations for resistors and sources are **linear**.



Key result: arbitrary combinations of linear elements produce linear current-voltage relations.

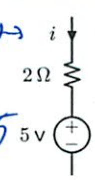
abstraction

any combo will have linear relationship

Check Yourself

current coming in = 0

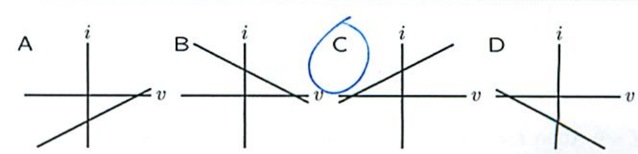
then voltage = 5



voltage drop

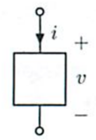
current ↑

Which plot shows corresponding current-voltage relation?



One-Ports

A "one-port" is a circuit that can be represented as a single, generalized element.



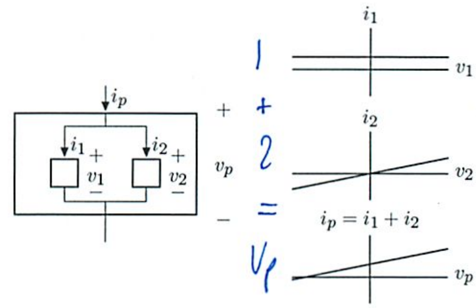
A one-port may contain any number of primitive elements (e.g., resistors, current sources, and voltage sources) as well as one-ports!

The defining feature of a one-port is that it has two terminals, such that current enters one terminal (+) and exits the other (−), producing a voltage v across the terminals.

Parallel One-Ports

The parallel combination of two linear one-ports is a linear one-port.

The proof follows from the current-voltage relations.



1 + 2 = V_p

Voltage drop is the same

The sum of two linear relations is linear.

Slope is $\frac{1}{R}$

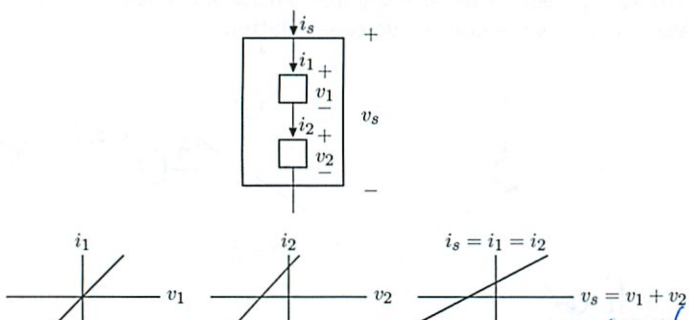
$V_p = \underbrace{\frac{R_1 R_2}{R_1+R_2}}_{\text{slope}} i_p + \underbrace{\frac{R_2 V_1^0 + R_1 V_2^0}{R_1+R_2}}_{\text{offset}}$ did several examples

math not on handout

Series One-Ports

The ~~parallel~~ series combination of two linear one-ports is a linear one-port.

The proof follows from the current-voltage relations.



sum of voltage drops

The "horizontal sum" of two straight lines is a straight line.

resistor

resistor + voltage source

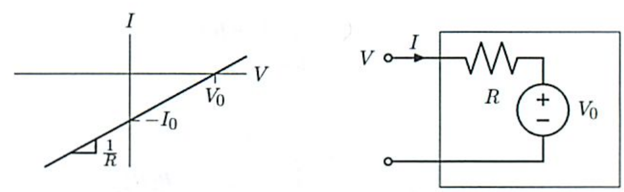
$V_s = (R_1 + R_2) i_s + V_1^0 + V_2^0$

or

Thevenin Equivalents

If the relation between terminal voltage and current can be represented by a straight line, then the terminal behavior of the one-port can be represented by a voltage source in series with a resistor.

abstraction



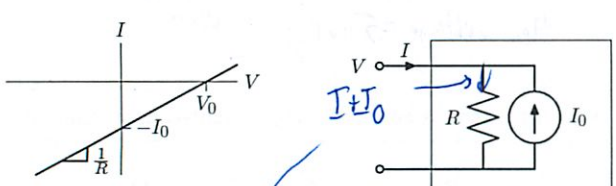
From the circuit, $I = \frac{V - V_0}{R}$.

If $I = 0$, then $V = V_0$ (the x-intercept of the plot).

The rate of growth of $I = \frac{V}{R} - \frac{V_0}{R}$ with V is the slope $1/R$.

Norton Equivalents

If the relation between terminal voltage and current can be represented by a straight line, then the terminal behavior of the one-port can be represented by a current source in parallel with a resistor.



$I + I_0$

From the circuit, $V = (I + I_0)R$.

If $V = 0$, then $I = -I_0$ (the negative of the y-intercept of the plot).

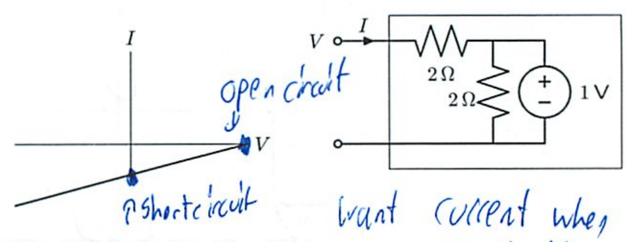
The rate of growth of $I = -I_0 + V/R$ with V is the slope $1/R$.

powerful tools

Open-Circuit Voltage and Short-Circuit Current

If a one-port contains just resistors and current and voltage sources, then its terminal behavior can be characterized by determining just two points on its v-i curve.

Example: open-circuit voltage and short-circuit current.



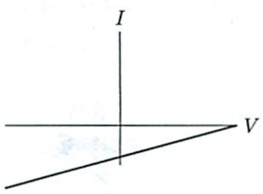
$V = V_0$ when $I = 0$, i.e., $V_0 = 1\,\text{V}$.

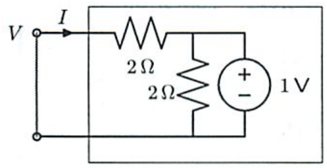
want current when
- open circuit
- short circuit

Open-Circuit Voltage and Short-Circuit Current

If a one-port contains just resistors and current and voltage sources, then its terminal behavior can be characterized by determining just two points on its v-i curve.

Example: open-circuit voltage and short-circuit current.



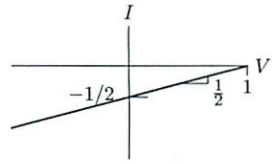


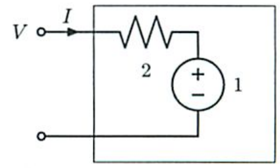
$V = V_0$ when $I = 0$, i.e., $V_0 = 1\,\mathrm{V}$.

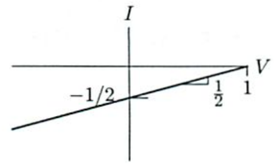
$I = -I_0$ when $V = 0$ (i.e., add the red wire), i.e., $I_0 = 1/2\,\mathrm{A}$.

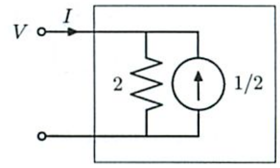
Thevenin and Norton Equivalents

Given the open-circuit voltage $V_0 = 1\,\mathrm{V}$ and short-circuit current $I_0 = 1/2\,\mathrm{A}$, find the Thevenin and Norton equivalent circuits.





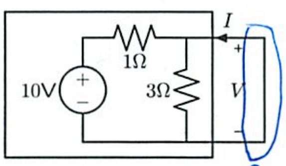




Given any circuit can do this

Thevenin Example

Find the Thevenin equivalent of this circuit.



Open-circuit voltage (i.e., $I = 0$)

$$V_0 = V = \frac{3}{3+1} \times 10 = 7.5\,\mathrm{V}$$

Short-circuit current (i.e., $V = 0 \rightarrow$ add a wire!)

$$I_0 = -I = \frac{10\,\mathrm{V}}{1\,\Omega} = 10\,\mathrm{A}$$

Bypasses the 3Ω resistor

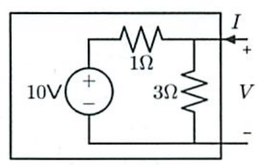
Equivalent resistance

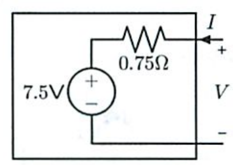
$$R = \frac{V_0}{I_0} = \frac{7.5\,\mathrm{V}}{10\,\mathrm{A}} = 0.75\Omega$$

Kirkoff's Law

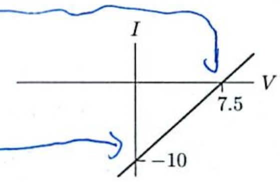
Thevenin Example

This is the Thevenin equivalent.





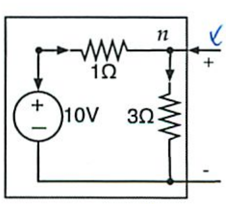
Do these circuits have the same current-voltage relations?



Thevenin Example – Another Solution

Another way to do it

Find **open-circuit voltage** (v, when $i = 0$):



no current

$$\frac{10-n}{1} = \frac{n-0}{3}$$

Watch direction!

current law

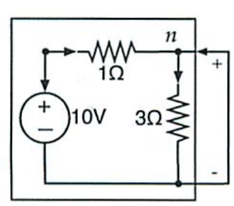
$$30 - 3n = n$$

$$n = 7.5$$

So, one point on our graph is $(7.5, 0)$.

Thevenin Example – Another solution

Find **closed circuit current** (i when $v = 0$). Add wire, so that $n = 0$.



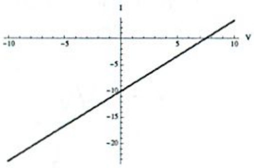
$$\frac{10-n}{1} = -i$$

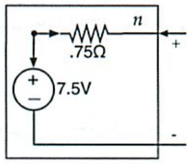
bypass the 3Ω

$$i = -10$$

So, another point on our graph is $(0, -10)$.

Thevenin Example – Another solution





$i = mv + b$

$0 = m(7.5) + b$ ← one piece — voltage drop open case

$-10 = m(0) + b$ ← other piece — current short circuit

↓ solve!

$i = 1.333v - 10$

So, equivalent source is $7.5V$ and equivalent resistance is $\frac{1}{1.333} = 0.75\Omega$.

Check Yourself

Determine the Thevenin voltage and resistance as seen from the V_o port.

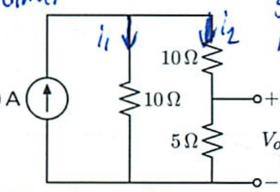
Open

i_2 ← need current divider

$i_2 = \frac{10}{10+15} I_{10}$

$= 4A$

$V_o = I_2 R = 20V$ → $R_{TH} = \frac{V}{I} = \frac{20}{5} = 4$



Short

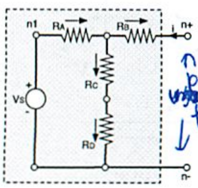
5Ω bypassed

need i_2 current divider

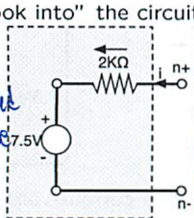
$i_2 = \frac{10}{10+10} I = 5A$

Thevenin depends on how we view the circuit

Assume $R_A = \cancel{2\Omega}$ 2kΩ and $R_B = R_C = R_D = \Omega$. What is the equivalent combination if we "look into" the circuit through the shown ports.



picked these



Example circuit — Thévenin

Short

$15-n = 2i_1$ $i = n - 7.5$

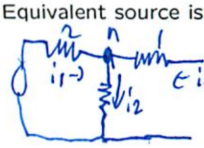
$n - 0 = 2i_2$ $i = -i - 7.5$

$n - 0 = -i$ $i = -3.75$

$i = i_2 - i_1$

$= \frac{n}{2} - \frac{15-n}{2}$

Equivalent source is $7.5V$ and equivalent resistance is $\cancel{2k\Omega}$. $\boxed{2k\Omega}$



open $i_1 = i_2$

$15 - n = 2i_1$

$n - 0 = 2i_2$

$15 - n = n$

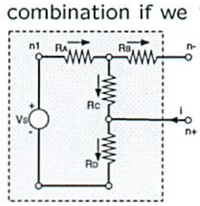
$n = 7.5$

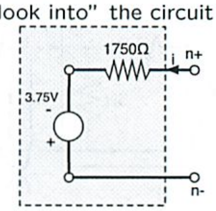
Or could just say voltage divider

$i = .5V - 3.75$

Thevenin depends on how we view the circuit

Here is the same circuit. We want to determine the equivalent combination if we "look into" the circuit through the shown ports.





Same circuit, different ports — Its Thévenin equivalent

Here, equivalent source is $-3.75V$ and equivalent resistance is 1750Ω.

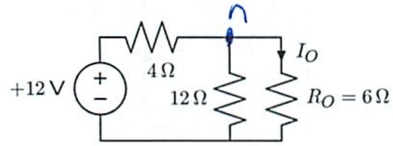
different values out

Check Yourself

Find V_B and R_B so that $I_B = I_O$.
Choose values so that $I_B = I_O$ even if $R_O \neq 6\,\Omega$.

$V_{th} = \frac{12}{12+4} \cdot 12$

$= 9$



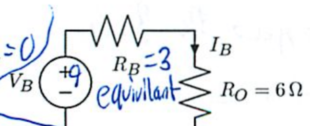
$12 - n = 4 \cdot i_1$

when connected $n = 0$

$i = 3$

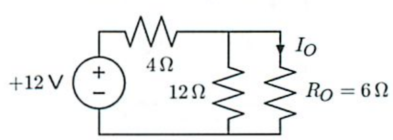
but current dir we want

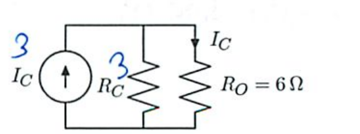
$I_{th} = -3$



Check Yourself Norton

Find I_C and R_C so that $I_C = I_O$.
Choose values so that $I_C = I_O$ even if $R_O \neq 6\,\Omega$.

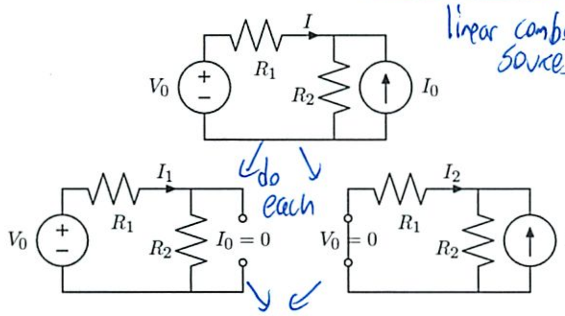




Superposition

If a circuit contains only linear parts (resistors, current and voltage sources), then any voltage (or current) can be computed as the sum of those that result when each source is turned on one-at-a-time.

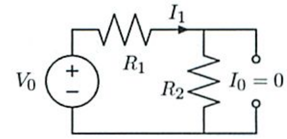
linear combo of sources



every linear combo can be represented as a straight line

Superposition

First component, with $I_0 = 0$

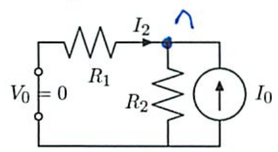


Resistances just add, since no current flow through open element

$$I_1 = \frac{V_0}{R_1 + R_2}$$

Superposition

Second component, with $V_0 = 0$



Use KCL at nodes: just use KCL

$$I_2 + I_0 + i_2 = 0$$ for n

Voltage drop across both resistances must be same, since parallel resistances

$$i_2 R_2 = I_2 R_1$$

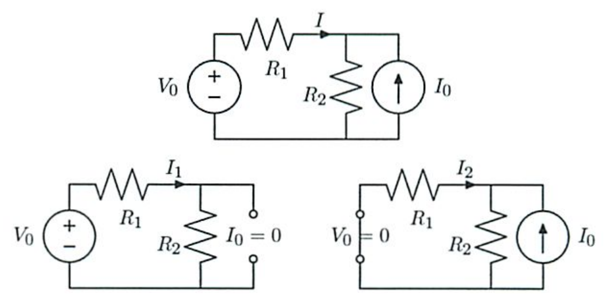
now have parallel resistors

Combining gives:

$$I_2 = \frac{-R_2}{R_1 + R_2} I_0$$

Superposition

If a circuit contains only linear parts (resistors, current and voltage sources), then any voltage (or current) can be computed as the sum of those that result when each source is turned on one-at-a-time.

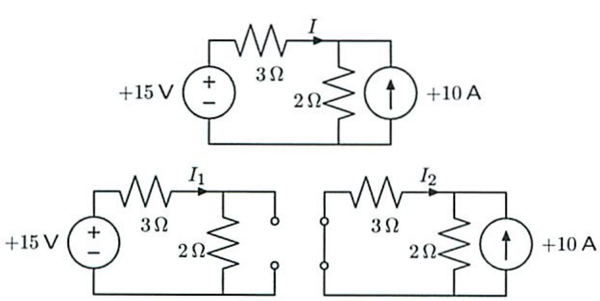


$$I = I_1 + I_2 = \frac{V_0}{R_1 + R_2} - \frac{R_2}{R_1 + R_2} I_0$$

pick which tool to use, or use 1 to check another

Superposition

For many circuits, superposition is even easier to apply than the node or the loop methods.



$$I = I_1 + I_2 = \frac{15}{2+3} - \frac{2}{2+3} 10 = 3 - 4 = -1\,\text{A}$$

Summary: PCAP for Circuits

Python:

- **procedures** are abstractions that combine primitive operations to capture common patterns
- **classes** are abstractions that associate data (attributes) and procedures (methods) that are related

Signals and Systems:

- **signals** are abstractions that collect all of the samples at different times into a single object
- **system functions** are abstractions that combine operations (delays, gains, and adders) to capture common patterns

Circuits

- **one-ports** are abstractions that combine any number of primitive elements (e.g., resistors, current sources, and voltage sources) into generalized circuit elements
- **Thevenin and Norton equivalents** provide a common framework for analyzing linear one-ports

This Week

Software lab: Work on HW3. Good time to get checkoffs.

HW 3: Due at the beginning of design lab.

Design lab: Building one of your circuit designs from HW3, to make the robot head turn to the light.

Midterm 2:

- Tuesday, November 9, 7:30–9:00PM, 32-141 or 32-155 14.01 conflict
- Any printed material okay

Conflict exam:

- Wednesday, November 10, 8:00–9:30AM, 34-501
- Email welg@mit.edu before Monday, November 8 if you need to take this exam.

6.01: Introduction to EECS I

Circuit Abstractions

Week 9 *November 2, 2010*

Reading: 7.7

PCAP Framework for Managing Complexity

Managing complexity in Python and in Signals and Systems.

Python:

- **procedures** are abstractions that combine primitive operations to capture common patterns
- **classes** are abstractions that associate data (attributes) and procedures (methods) that are related

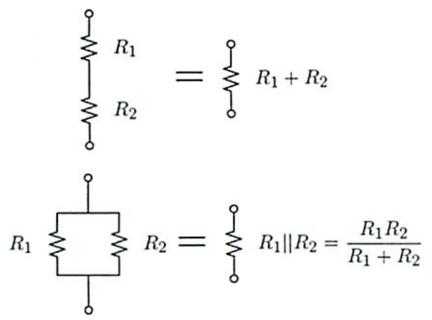
Signals and Systems:

- **signals** are abstractions that collect all of the samples at different times into a single object
- **system functions** are abstractions that combine operations (delays, gains, and adders) to capture common patterns

Today: managing complexity in design and analysis of circuits

Equivalent Resistors

Series or parallel resistors can be replaced by a single equivalent resistor.



Replacing two elements by one reduces conceptual complexity.

Circuit Abstractions

Today: Similar simplifications for other types of elements

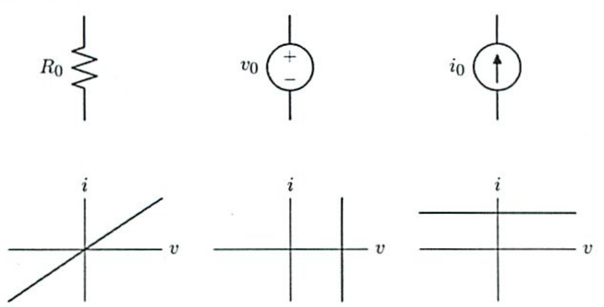
The key concept for generalizing from equivalent resistors to equivalent circuits is the **current-voltage relation**.

$i = \frac{1}{R} v$

Two circuits are "equivalent" if they have the same current-voltage relation.

Current-Voltage Relations

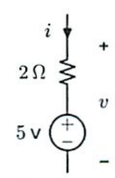
The current-voltage relations for resistors and sources are **linear**.



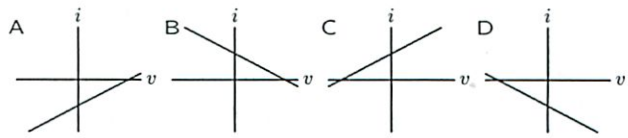
$v = R_0\, i$ $\quad v = v_0$ $\quad i = i_0$

Key result: arbitrary combinations of linear elements produce linear current-voltage relations.

Check Yourself

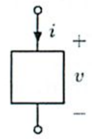


Which plot shows corresponding current-voltage relation?



One-Ports

A "one-port" is a circuit that can be represented as a single, generalized element.



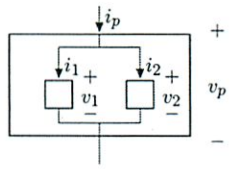
A one-port may contain any number of primitive elements (e.g., resistors, current sources, and voltage sources) as well as one-ports!

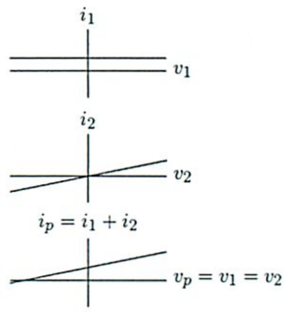
The defining feature of a one-port is that it has two terminals, such that current enters one terminal (+) and exits the other (−), producing a voltage v across the terminals.

Parallel One-Ports

The parallel combination of two linear one-ports is a linear one-port.

The proof follows from the current-voltage relations.



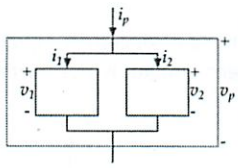


The sum of two linear relations is linear.

Parallel combination

Components have a linear description:

$$v_1 = i_1R_1 + V_1^0$$
$$v_2 = i_2R_2 + V_2^0$$



Voltages $v_1 = v_2 = v_p$, so

$$i_p = i_1 + i_2$$
$$= \frac{v_p - V_1^0}{R_1} + \frac{v_p - V_2^0}{R_2}$$
$$R_1R_2i_p = R_2(v_p - V_1^0) + R_1(v_p - V_2^0)$$
$$= v_p(R_1 + R_2) - R_2V_1^0 - R_1V_2^0$$
$$v_p = \frac{R_1R_2}{R_1+R_2}i_p + \frac{R_2V_1^0 + R_1V_2^0}{R_1+R_2}$$

Combination is a line with parameters $\frac{R_1R_2}{R_1+R_2}$ and $\frac{R_2V_1^0+R_1V_2^0}{R_1+R_2}$.

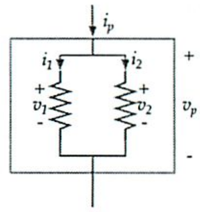
Parallel: Resistor and Resistor

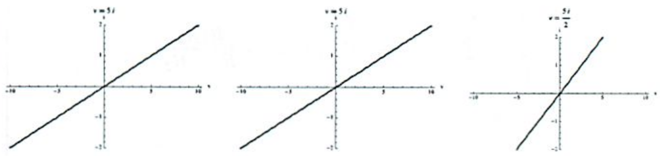
$$v_1 = i_1R_1$$
$$v_2 = i_2R_2$$

Voltages $v_1 = v_2 = v_p$, so

$$v_p = i_p\frac{R_1R_2}{R_1+R_2}$$

A line with parameters $\frac{R_1R_2}{R_1+R_2}$ and 0.



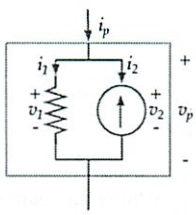


$R_1 = 5$ $R_2 = 5$ $R_p = \frac{5}{2}$

Parallel: Resistor and Current Source

$$v_1 = i_1R_1$$
$$i_2 = -I_2$$
$$i_1 + i_2 = i_p$$
$$\frac{v_p}{R_1} - I_2 = i_p$$
$$v_p = R_1i_p + R_1I_2$$

A line with parameters R_1 and R_1I_2.

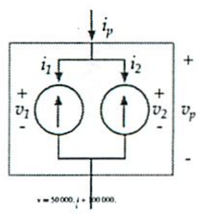


$R_1 = 5$ $i_2 = -1$ $R_p = 5, V_p^0 = 5$

Parallel: Current Source and Current Source

$$i_1 = -I_1$$
$$i_2 = -I_2$$
$$i_1 + i_2 = i_p$$
$$-I_1 - I_2 = i_p$$

A line horizontal line at $-I_1 - I_2$.



$i_1 = -1$ $i_2 = -1$ $i_p = -2$

Parallel: Resistor and Voltage Source

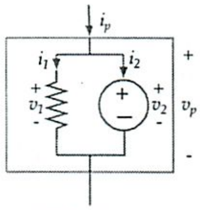
$v_1 = i_1 R_1$

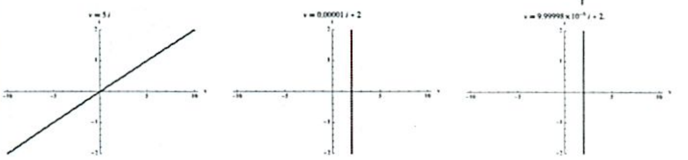
$v_2 = V_2^0$

Voltages $v_1 = v_2 = v_p$, so

$v_p = V_2^0$

A vertical line at V_2^0.





$R_1 = 5$ $V_2^0 = 2$ $V_p^0 = 2$

Parallel: Current Source and Voltage Source

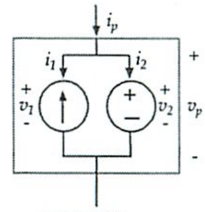
$i_1 = -I_1$

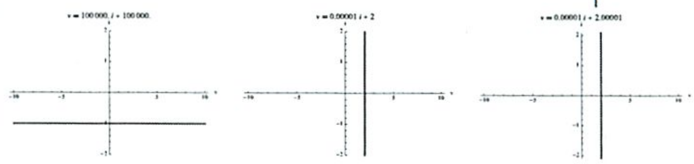
$v_2 = V_2^0$

Voltages $v_1 = v_2 = v_p$, so

$v_p = V_2^0$

A vertical line at V_2^0.





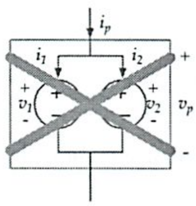
$i_1 = -1$ $V_2^0 = 2$ $V_p^0 = 2$

Parallel: Voltage Source and Voltage Source

$v_1 = V_1^0$

$v_2 = V_2^0$

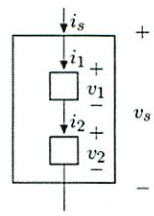
Voltages $v_1 = v_2 = v_p$, so unless $V_1^0 = V_2^0$, we have a problem!

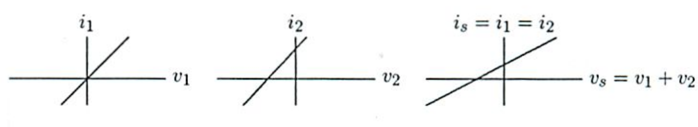


Series One-Ports

The parallel combination of two linear one-ports is a linear one-port.

The proof follows from the current-voltage relations.





The "horizontal sum" of two straight lines is a straight line.

Series combination

We know components have a linear description:

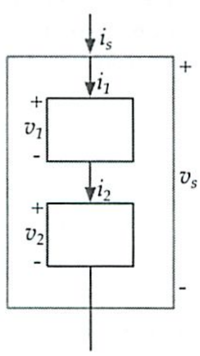
$v_1 = i_1 R_1 + V_1^0$

$v_2 = i_2 R_2 + V_2^0$

Currents $i_1 = i_2 = i_s$, so

$$v_s = v_1 + v_2$$
$$= i_s R_1 + V_1^0 + i_s R_2 + V_2^0$$
$$= i_s(R_1 + R_2) + V_1^0 + V_2^0$$

Combination is describable as a line with parameters $R_1 + R_2$ and $V_1^0 + V_2^0$.



Series: Resistor and Resistor

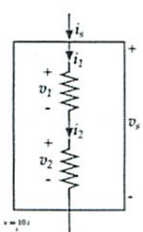
$v_1 = i_1 R_1$

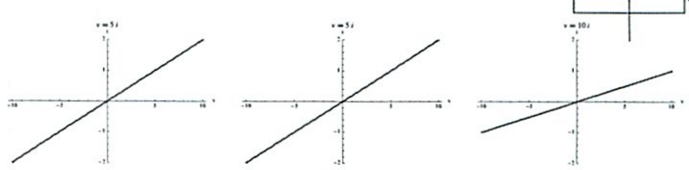
$v_2 = i_2 R_2$

Currents $i_1 = i_2 = i_s$, so

$v_s = i_s(R_1 + R_2)$

A line with parameters $R_1 + R_2$ and 0.





$R_1 = 5$ $R_2 = 5$ $R_s = 10$

Series: Voltage Source and Resistor

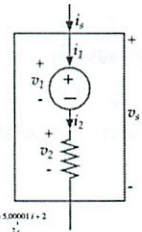
$v_1 = V_1^0$

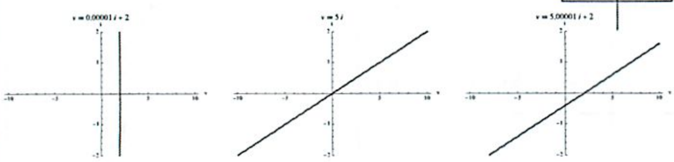
$v_2 = i_2 R_2$

Currents $i_1 = i_2 = i_s$, so

$v_s = i_s R_2 + V_1^0$

A line with parameters R_2 and V_1^0.





$V^0 = 2$ $R = 5$ $R_s = 5, V_s^0 = 2$

Series: Voltage Source and Voltage Source

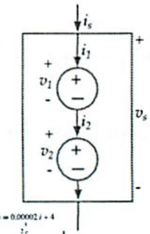
$v_1 = V_1^0$

$v_2 = V_2^0$

Currents $i_1 = i_2 = i_s$, so

$v_s = V_1^0 + V_2^0$

A vertical line at $V_1^0 + V_2^0$.



$V^0 = 2$ $V^0 = 2$ $V_s^0 = 4$

Series: Current Source and Resistor

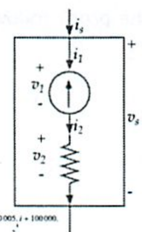
$i_1 = -I_1$

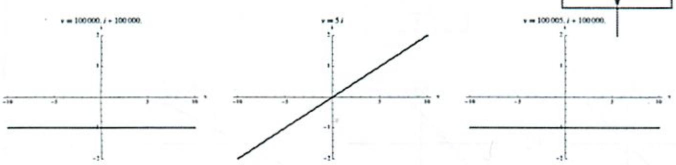
$v_2 = i_2 R_2$

Currents $i_1 = i_2 = i_s = -I_1$, so

$I_s = -I_1$

A horizontal line at $-I_1$.





$i_1 = -1$ $R = 5$ $i_s = -1$

Series: Current Source and Voltage Source

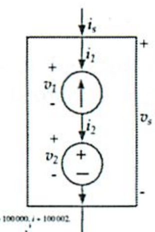
$i_1 = -I_1$

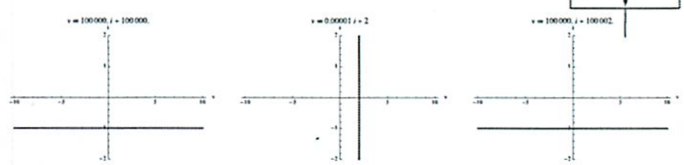
$v_2 = V_2^0$

Currents $i_1 = i_2 = i_s = -I_1$, so

$I_s = -I_1$

A horizontal line at $-I_1$.





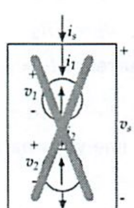
$i_1 = -1$ $V = 2$ $i_s = -1$

Series: Current Source and Current Source

$i_1 = -I_1$

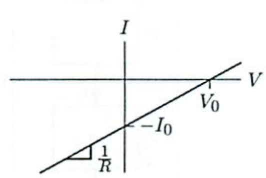
$i_2 = -I_2$

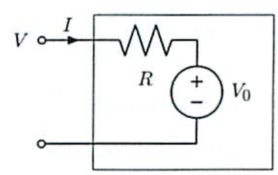
Currents $i_1 = i_2 = i_s = -I_1 = -I_2$, so unless $I_1 = I_2$, we have a problem!



Thevenin Equivalents

If the relation between terminal voltage and current can be represented by a straight line, then the terminal behavior of the one-port can be represented by a voltage source in series with a resistor.





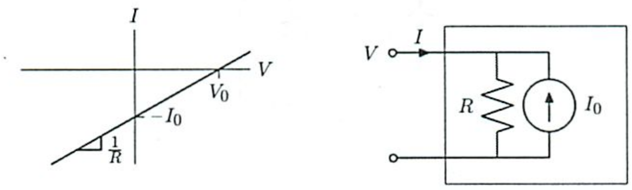
From the circuit, $I = \frac{V - V_0}{R}$.

If $I = 0$, then $V = V_0$ (the x-intercept of the plot).

The rate of growth of $I = \frac{V}{R} - \frac{V_0}{R}$ with V is the slope $1/R$.

Norton Equivalents

If the relation between terminal voltage and current can be represented by a straight line, then the terminal behavior of the one-port can be represented by a current source in parallel with a resistor.



From the circuit, $V = (I + I_0)R$.

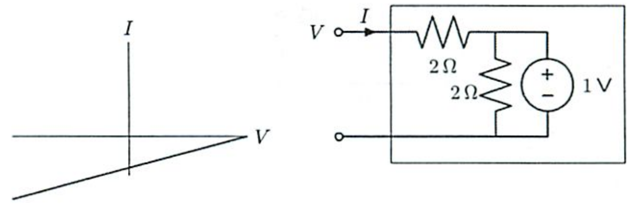
If $V = 0$, then $I = -I_0$ (the negative of the y-intercept of the plot).

The rate of growth of $I = -I_0 + V/R$ with V is the slope $1/R$.

Open-Circuit Voltage and Short-Circuit Current

If a one-port contains just resistors and current and voltage sources, then its terminal behavior can be characterized by determining just two points on its v-i curve.

Example: open-circuit voltage and short-circuit current.

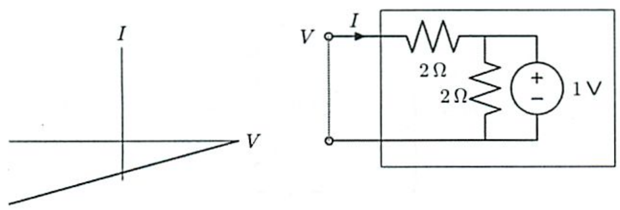


$V = V_0$ when $I = 0$, i.e., $V_0 = 1\,\text{V}$.

Open-Circuit Voltage and Short-Circuit Current

If a one-port contains just resistors and current and voltage sources, then its terminal behavior can be characterized by determining just two points on its v-i curve.

Example: open-circuit voltage and short-circuit current.

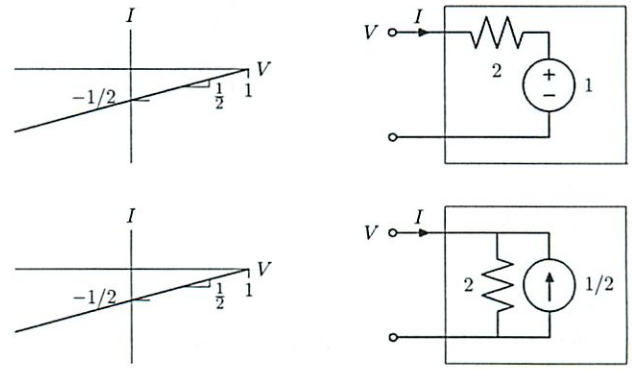


$V = V_0$ when $I = 0$, i.e., $V_0 = 1\,\text{V}$.

$I = -I_0$ when $V = 0$ (i.e., add the red wire), i.e., $I_0 = 1/2\,\text{A}$.

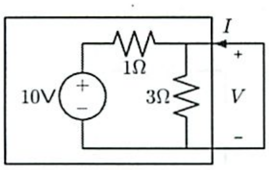
Thevenin and Norton Equivalents

Given the open-circuit voltage $V_0 = 1\,\text{V}$ and short-circuit current $I_0 = 1/2\,\text{A}$, find the Thevenin and Norton equivalent circuits.



Thevenin Example

Find the Thevenin equivalent of this circuit.



Open-circuit voltage (i.e., $I = 0$)

$$V_0 = V = \frac{3}{3+1} \times 10 = 7.5\,\text{V}$$

Short-circuit current (i.e., $V = 0 \to$ add a wire!)

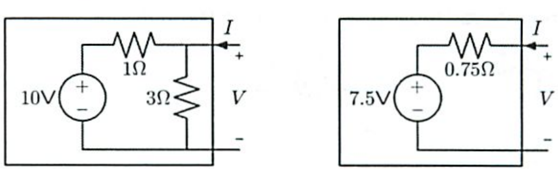
$$I_0 = -I = \frac{10\,\text{V}}{1\,\Omega} = 10\,\text{A}$$

Equivalent resistance

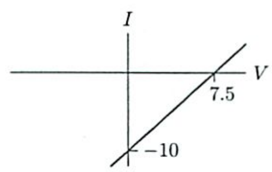
$$R = \frac{V_0}{I_0} = \frac{7.5\,\text{V}}{10\,\text{A}} = 0.75\Omega$$

Thevenin Example

This is the Thevenin equivalent.

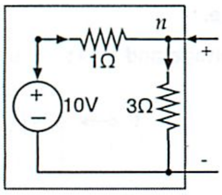


Do these circuits have the same current-voltage relations?



Thevenin Example – Another solution

Find **open-circuit voltage** (v, when $i = 0$):



$$\frac{10-n}{1} = \frac{n-0}{3}$$

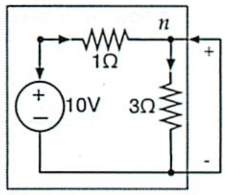
$$30 - 3n = n$$

$$n = 7.5$$

So, one point on our graph is $(7.5, 0)$.

Thevenin Example – Another solution

Find **closed circuit current** (i when $v = 0$). Add wire, so that $n = 0$.

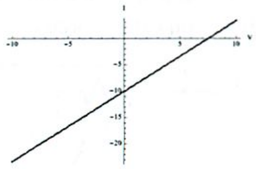


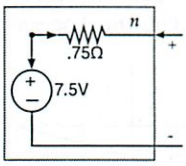
$$\frac{10-n}{1} = -i$$

$$i = -10$$

So, another point on our graph is $(0, -10)$.

Thevenin Example – Another solution





$$i = mv + b$$

$$0 = m(7.5) + b$$

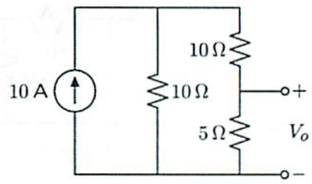
$$-10 = m(0) + b$$

$$i = 1.333v - 10$$

So, equivalent source is $7.5V$ and equivalent resistance is $\frac{1}{1.333} = 0.75\Omega$.

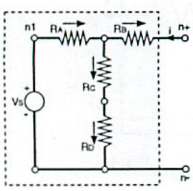
Check Yourself

Determine the Thevenin voltage and resistance as seen from the V_o port.

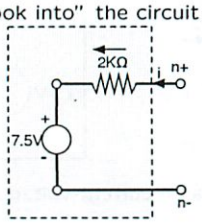


Thevenin depends on how we view the circuit

Assume $R_A = 2K\Omega$ and $R_B = R_C = R_D = K\Omega$. What is the equivalent combination if we "look into" the circuit through the shown ports.



Example circuit

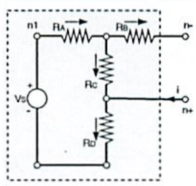


Thévenin

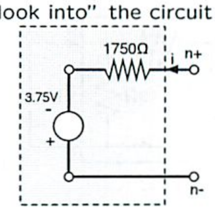
Equivalent source is $7.5V$ and equivalent resistance is $2.0K\Omega$.

Thevenin depends on how we view the circuit

Here is the same circuit. We want to determine the equivalent combination if we "look into" the circuit through the shown ports.



Same circuit, different ports

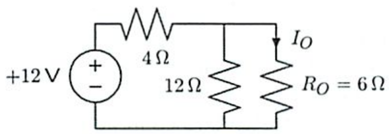


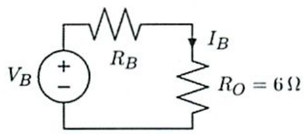
Its Thévenin equivalent

Here, equivalent source is $-3.75V$ and equivalent resistance is 1750Ω.

Check Yourself

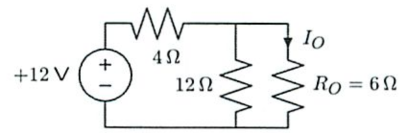
Find V_B and R_B so that $I_B = I_O$.
Choose values so that $I_B = I_O$ even if $R_O \neq 6\,\Omega$.

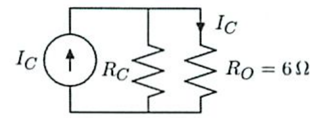




Check Yourself

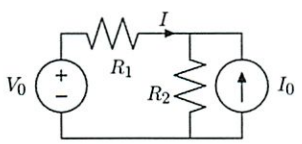
Find I_C and R_C so that $I_C = I_O$.
Choose values so that $I_C = I_O$ even if $R_O \neq 6\,\Omega$.

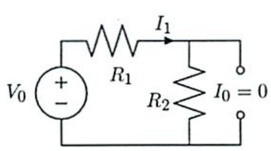


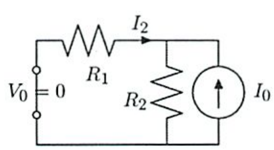


Superposition

If a circuit contains only linear parts (resistors, current and voltage sources), then any voltage (or current) can be computed as the sum of those that result when each source is turned on one-at-a-time.

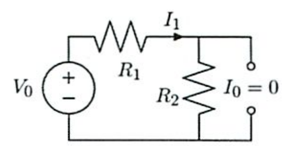






Superposition

First component, with $I_0 = 0$

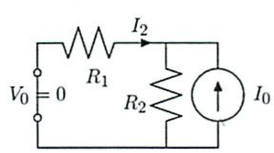


Resistances just add, since no current flow through open element

$$I_1 = \frac{V_0}{R_1 + R_2}$$

Superposition

Second component, with $V_0 = 0$



Use KCL at nodes:

$$I_2 + I_0 + i_2 = 0$$

Voltage drop across both resistances must be same, since parallel resistances

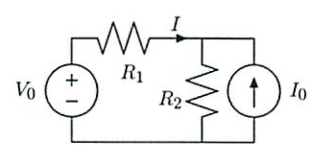
$$i_2 R_2 = I_2 R_1$$

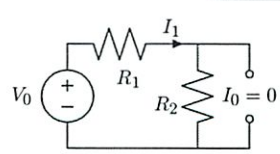
Combining gives:

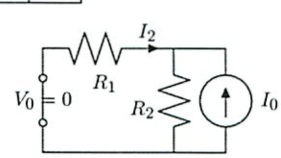
$$I_2 = \frac{-R_2}{R_1 + R_2} I_0$$

Superposition

If a circuit contains only linear parts (resistors, current and voltage sources), then any voltage (or current) can be computed as the sum of those that result when each source is turned on one-at-a-time.



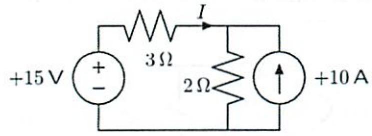


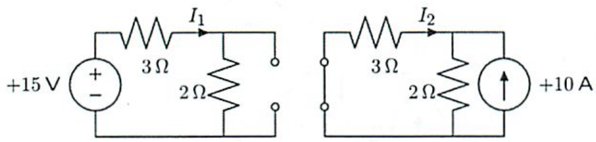


$$I = I_1 + I_2 = \frac{V_0}{R_1 + R_2} - \frac{R_2}{R_1 + R_2} I_0$$

Superposition

For many circuits, superposition is even easier to apply than the node or the loop methods.





$$I = I_1 + I_2 = \frac{15}{2+3} - \frac{2}{2+3}10 = 3 - 4 = -1\,\text{A}$$

Summary: PCAP for Circuits

Python:

- **procedures** are abstractions that combine primitive operations to capture common patterns
- **classes** are abstractions that associate data (attributes) and procedures (methods) that are related

Signals and Systems:

- **signals** are abstractions that collect all of the samples at different times into a single object
- **system functions** are abstractions that combine operations (delays, gains, and adders) to capture common patterns

Circuits

- **one-ports** are abstractions that combine any number of primitive elements (e.g., resistors, current sources, and voltage sources) into generalized circuit elements
- **Thevenin and Norton equivalents** provide a common framework for analyzing linear one-ports

This Week

Software lab: Work on HW3. Good time to get checkoffs.

HW 3: Due at the beginning of design lab.

Design lab: Building one of your circuit designs from HW3, to make the robot head turn to the light.

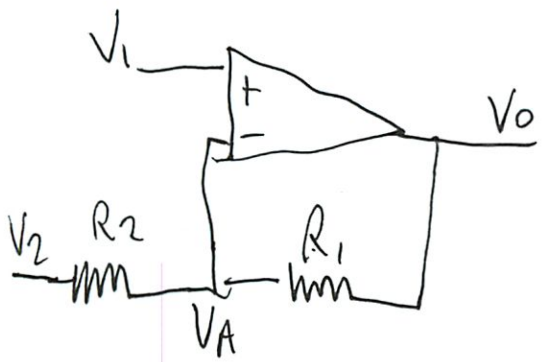
Midterm 2:

- Tuesday, November 9, 7:30–9:00PM, 32-141 or 32-155
- Any printed material okay

Conflict exam:

- Wednesday, November 10, 8:00–9:30AM, 34-501
- Email `welg@mit.edu` before Monday, November 8 if you need to take this exam.

Profs having trouble w/ daylight savings



$V(\emptyset)$ ← letter not zero

a) $V_a = V_1$

b) $V_2 = 0$

write $V(\emptyset)$

$\frac{1}{=} V_2$

So just non inverting amp w/ $R_1 \to R_2$ flipped

$$V_o = V_1 \underbrace{\frac{R_1 + R_2}{R_2}}$$

just want k ↑ R_1

c) V_2 is some value (this is not centering around 0)

$V_o - V = k(V_1 - V)$

just V_2 and k = same

②

d) Assume $V_2 = 5$ V

V_1 is b/w 0 - 10V

How does VO behave?

(remember output must be b/w 0 → 10 V)

round to an int

$R_2 = 10000$

given V_1	given V_2	given R1	find Vo
	does not matter		$V_0 - V_2 = \frac{R_1 + R_2}{R_2}(V_1 - V_2)$
			$V_0 = \cancel{\frac{R_1+R_2}{R_2} V_1}$ no v2 does matter
			$\frac{R_1+R_2}{R_2}(V_1 - V_2) + V_2$
10	5	100	10.05 → round 10 ✓
7	5	100	7.02 → 7 ✓
5	5	100	5 ✓
3	5	100	2.98 → 3 ✓
0	5	100	-.05 → 0 ✓
10	5	10,000	15 ✗
7	5	10,000	9 ✓
5	5	10,000	5 ✓
3	5	10,000	1 ✓
0	5	10,000	-1 ✗ -5 copy error

← Oh so almost the same! ~~Cutof Just~~

here it is the difference

depends on resistance

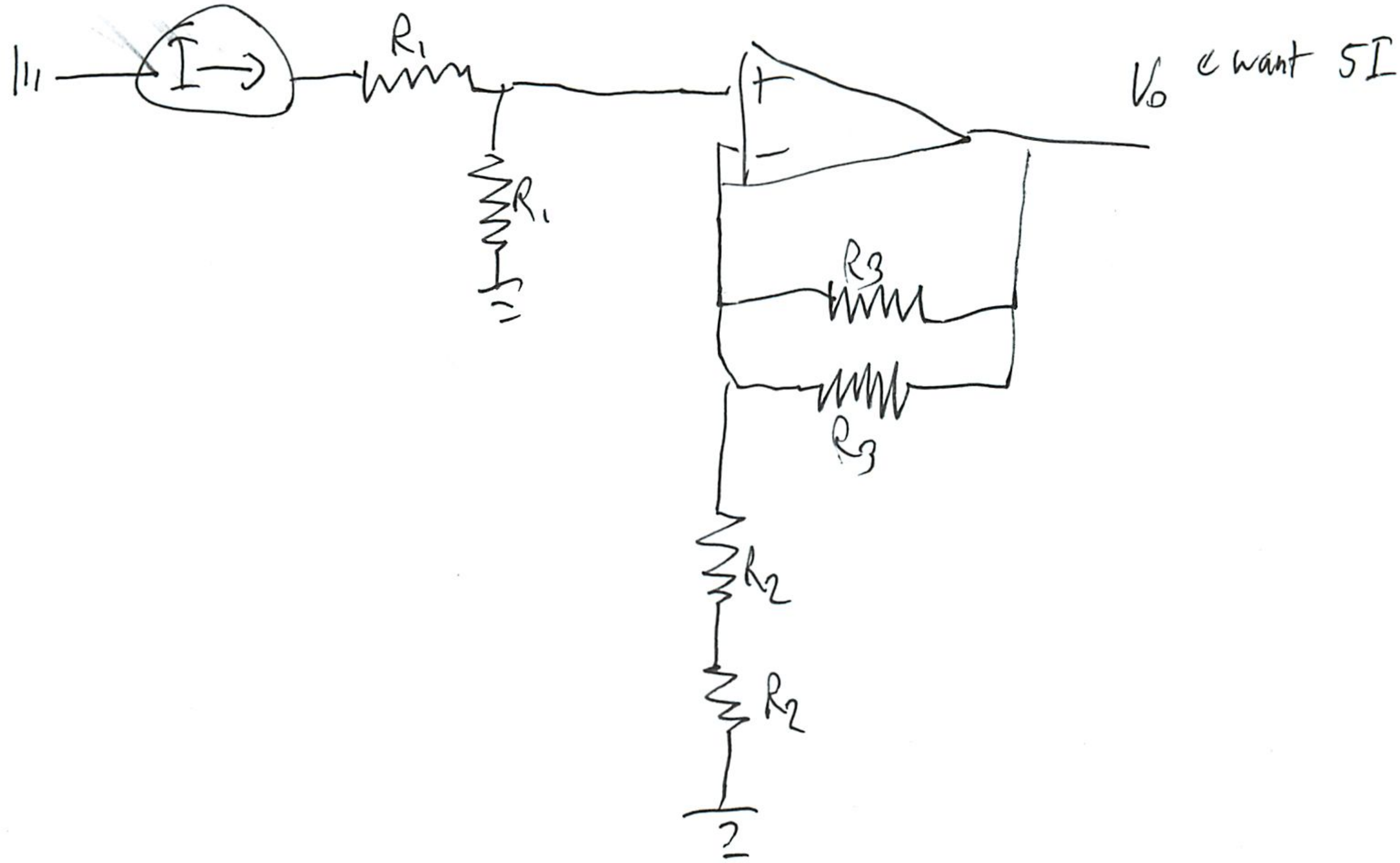
③

* Real world can't go > 10 !
< 0 !

So thats it for the lab
just need to write up HW — since went to OH Sunday

nano 9

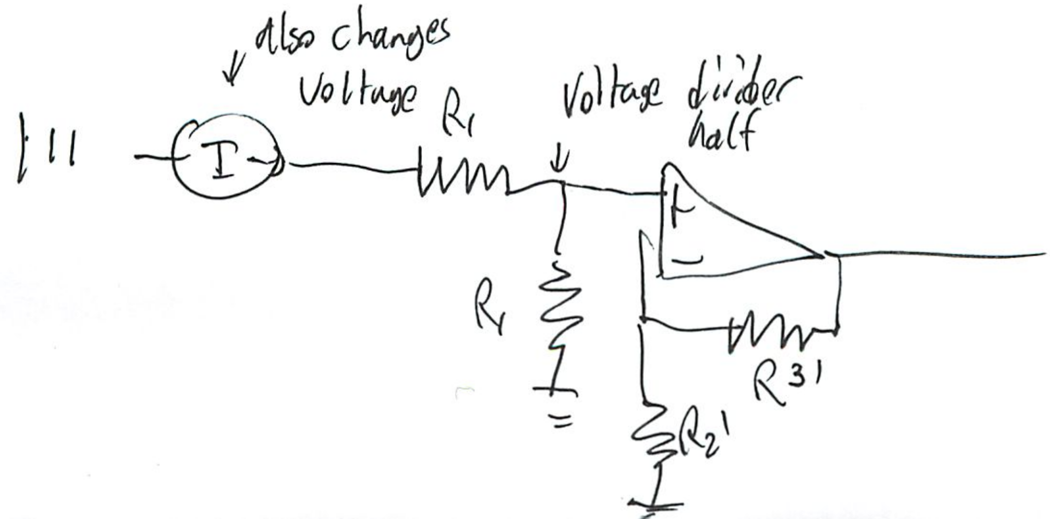
11/4



Define R_1, R_2, R_3

$R_2' = R_2 + R_2 = 2R_2$

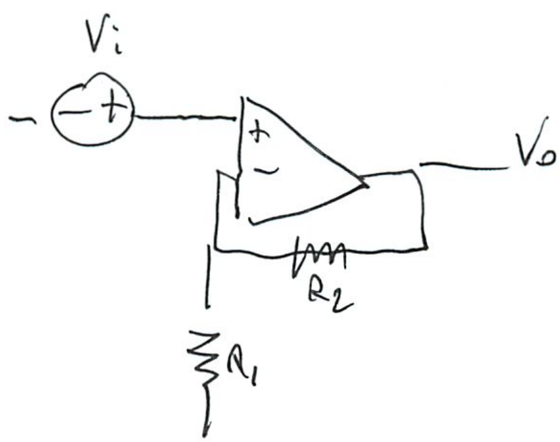
$R_3' = \frac{R_3 R_3}{R_3 + R_3} = \frac{R3^2}{2R^3} =$



(2)

Non inverting amp:

$V_o = V_i \frac{R_1 + R_2}{R_1}$



: but what is starting voltage?

- half what current sorce is outputting

but it defines output in current that is throwing me off

- prob makes this eaiser

or it says $V = 5 I$

5 current

: so aim for 5

define current source at 1

$V = I \frac{R_1}{2}$

$V_o = \frac{1}{2}$

then $V_o = \frac{1}{2} \frac{5 + 45}{5}$

our R_1, R_2

So $R_1 = 1$

$R_2' = 5$

$R_3' = 45$

③

$R_2 = 2.5$

$R_3 = 45 = \frac{R_3^2}{2R_3}$

$90 R_3 = R_3^2$

$90 = \frac{R_3^{\cancel{2}}}{\cancel{R_3}}$

$90 = R_3$ ⊗

~~Double all to remove decimal~~

I think start must be wrong

$V_i = I \frac{R_1}{2}$

$1 \frac{2}{2}$ $\quad R_1 = 2$

$V_i = 1$

$V_0 = 5 = 1 \cdot \frac{1+4}{1}$

$R_2 = .5$

$R_3 = 8$ ①

Double all ⊗

4

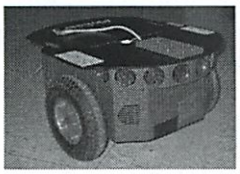
$R_1 = 1000\ \Omega$

$R_2 = 1000\ \Omega$

$R_3 = 1000\ \Omega$

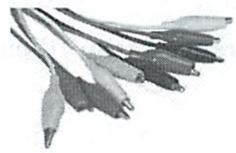
Design Lab 9: Visually Attractive

Each partnership will need a lab laptop. Do `athrun 6.01 update`. Files will be in `Desktop/6.01/lab9/designLab/`. In addition, you will need:



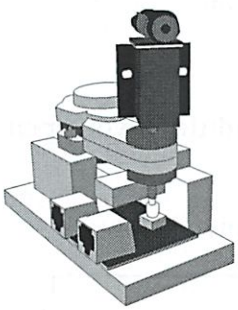
Robot

Lamp

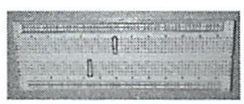


Two clip leads

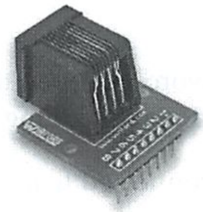
Multimeter



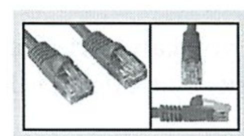
Robot head



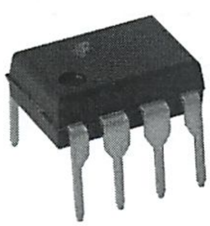
Proto board



Two eight-pin connectors



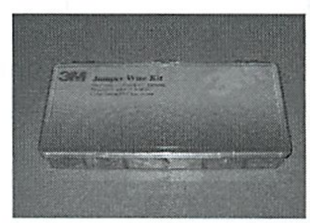
Red cable



Op-amp package

Potentiometer

Resistors, as needed



Wirekit

The relevant files in the distribution are:

- `CMax.py`: Used to start up the CMax layout tool.
- `lib601/eyeServo.py`: Input signal for simulator.
- `turnToLightAnalogBrain.py`: Brain that will plot the value of the neck potentiometer as a function of time.
- `roverBrainSkeleton.py`: Brain file for implementing your pet robot controller.

See the back page of this handout for the pin-outs of the connectors.

Last week, you designed and analyzed circuits to build a "head" that will allow the robot to sense and track light. Today, you and your partner will construct, debug, and demonstrate the head.

1 Pointing Circuit

Step 1. **If you completed the reference and alternate designs** (Checkoffs 3 and 4 from week 8), discuss your circuit designs and simulations with your partner. Pick which circuit you are going to build.

If you only completed the reference design, then use that design for this lab session (and complete the alternate design later).

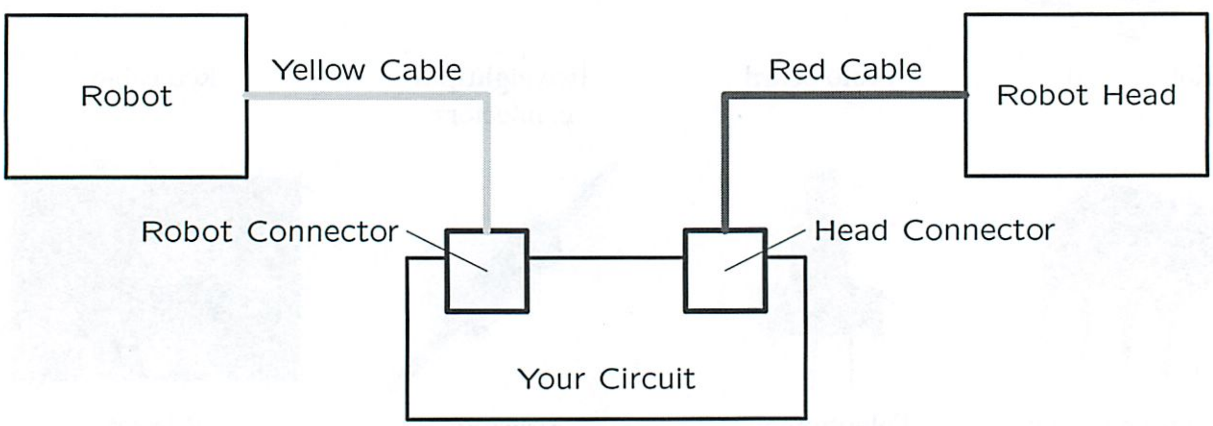
If you did not complete either design, then finish the reference design, get Checkoff 3 from Week 8, and use the reference design for this lab session (complete the alternate design later).

Step 2. Lay out the design you chose using CMax (see documentation near the end of this handout). You can run CMax by going to a Terminal window, navigating to `lab9/designlab` and typing

```
> python CMax.py
```

Or, you can start Idle with keyword `-n`, open the file `CMax.py` and do **Run Module**. If you open `CMax.py` through Idle, close it down by killing Idle.

Your circuit will ultimately be connected to a robot and to a robot "head" as shown below.



Include a **Robot Connector** and a **Head Connector** in your layout. The robot connector will accept the yellow cable from the robot, which provides power and ground for your circuit (do not use a separate power supply). The head connector connects through the red cable to a robot head, and provides connections to the sensors and to the motor.

> Keep your layout simple (so it will be easy to build). Use short wires where possible. Use only vertical or horizontal wires (not diagonal). Do not run wire over components. **Do not cross wires.**

Step 3. In CMax, click the **Simulate** button and select the file `lib601/eyeServo.py`, which specifies the input signals for the simulation.

Make sure that the result of the simulation matches your results from homework 3. Remember that you can use the probes to see the values of voltages on the board; this is useful in debugging.

Step 4. **Make sure that there is no black wire connecting your proto board to the motor.** This is a safety precaution for the equipment, in case your circuit is incorrect, and accidentally causes the head to turn out of control.

- Build the circuit on a proto board.

> **Make your circuit match your CMax layout EXACTLY.**
> Then your circuit will work, since your similation already worked!

Notice that the wire colors in CMax correspond to the wire colors in the wire kits.

- Start with a low gain (1 or 2), but be sure your circuit can be simply modified to obtain higher gains.
- Connect one 8-pin connector (the robot connector) to the yellow cable coming from the robot.
- Connect the other 8 pin connector (the head connector) to the front connector (near the eyes) of the head using a red cable.
- Turn on the robot to provide power to your circuit.

> *Check Yourself 1.* Verify that your circuit works by measuring the voltages across the motor, being sure that they behave appropriately as you change the light levels on the eyes. Demonstrate the correct behavior to a staff member and they will give you a black cable to connect the motor.

Step 5. Turn off the robot power (which will also turn off power to your circuit). Plug in the black cable. Turn on the robot power **for just 15 seconds** and then turn it back off. If the circuit was wired correctly, the head should turn toward the light. However, if the circuit is wired incorrectly, the head will slam against a stop, and the op amp will overheat. If the latter happens, then remove the black wire and retest the the head, as in the previous Check Yourself.

Step 6. Connect the center pin of the neck potentiometer (pin 2 on the head connector) to the first analog input (pin 1 on the robot connector). Also, connect the top and bottom pins of the neck potentiometer to power and ground on your protoboard.

Step 7. Use the brain `turnToLightAnalogBrain.py` to plot the neck potentiometer voltage as a function of time, and report the settle time of the signal. Follow these steps:

1. Unplug the motor (black cable).
2. Turn the head 45 degrees from the light (in either direction).
3. Start the brain and wait a second or two.

4. Plug in the motor.
5. Stop the brain a second or two after the head has stopped moving, or if it is clear that it will not stop moving.

You should see a graph appear, and an estimated settle time will be printed on the Soar window. Verify the printed settle time by examining your graph closely. (Sometimes our settle-time estimator is confused if there are minor variations at the end of your graph). The settle time is the number of steps it takes for the head to converge on its final direction (ignoring the initial period in which it is constant, before the cable is plugged in).

Step 8. Pick a good gain for your circuit so that you get as fast a response as possible (it shouldn't have significant oscillations, but a little overshoot is fine) over a range of distances from the light. Gather data with at least two different gains in your circuit and at least two distances. Save the graphs and settle times for each.

> *Checkoff 1.* Illustrate your circuit and its performance at two different distances with the two gains you investigated. How does the settle time behave with gain and with distance? **Keep your plots and measurements to discuss in your interview.**

2 Pet Robot

We would like our robot to follow a bright light around the room. The head you just built is capable of turning much more quickly than the (heavy) robot, so we will contruct a two-level control system in which the head turns to track moving light and the robot body turns so as to keep the head pointing forward relative to the body. This is analogous to your visual system, where your eyes move quickly to track motion and your head turns in the direction of gaze.

Use the head as configured in the previous part, so that it automatically turns toward a bright light. Mount the head on the robot body facing forward (same direction as the sonar array).

Step 9. Design a robot behavior that uses signals from its head to turn the robot toward a bright light.

> *Check Yourself 2.* What control variable from the head is important for turning the robot toward the light? Explain.

Step 10. You can read the voltages from pins 1, 3, 5, and 7 of the robot connector from a soar brain as the list of four values `inp.analogInputs`, where `inp` is an instance of the `io.SensorInput` class.

Step 11. Write a `soar` brain to implement your controller. The output of the controller should be an `Action` that specifies the rotational velocity `rvel` of the robot. We have provided a skeleton in `roverBrainSkeleton.py`

Hint: You can debug the robot behavior by tilting the robot backwards so that the wheels do not touch the ground and watching to see that the behavior is reasonable before unleashing your robot on the world. Also, start with the black cable disconnected, so that you can manually turn the head and observe if the wheels turn correctly.

Step 12. Demonstrate that your robot turns toward a bright light. What is the highest gain (in the software control loop) for which you get stable responses?

Step 13. We would like to rework the brain and circuit to make the robot's behavior depend on its proximity to the light. If the light is off, the robot should stand still in obedience. If the light is on, the robot should approach the light, positioning itself approximately half a meter from the bulb.

> *Check Yourself 3.* What control variable(s) from the head is/are important for determining proximity to the light? Explain.

For this behavior, the `soar` brain will need access to not just the neck pot but also some measure of the light intensity. Figure out how to make your circuit provide this information, and make whatever connections are needed, using one or more of pins 1, 3, 5, and 7 on the robot connector. **Your solution to this part will depend on which design you implemented for your pointing circuit. Talk to an instructor for ideas if you don't see how to do it for your circuit.**

> *Checkoff 2.* Demonstrate your pet robot's behaviors, including facing the light, approaching the light, retreating from it, and patiently waiting when there is no light. For extra brownie points, try parallel parking.

> Save all code and plots, and mail them to your partner (for the next interview).

3 Analog Bull's Eye (optional: do this if you have time)

Characterize the speed and precision with which the head tracks light.

Step 14. Turn off the power to the robot. Plug the laser connector (small black wire with round connector near the yellow cable coming out of the robot) into the laser on your robot head.

Now turn on the power to the robot, and measure head tracking as you did in Step 7. The laser light should strike the shade, ideally in its center! Tune your circuit to make the head tracking as accurate as possible. Recall from lab 7 that the photodetectors may not be matched perfectly. Consider how you could add one potentiometer to your circuit to improve pointing accuracy.

Checkoff 3. This checkoff is optional. Demonstrate the pointing accuracy of your head. Describe the fundamental limitations to its accuracy.

Turn off your meter, disassemble your board, and put the wires, op-amp, pot, and connectors back in the appropriate places. Throw away the resistors.

CMax Documentation

Adding components and wires

- You can place components on your board (resistors, op-amps, etc.) by clicking the associated button. They will appear in the lower left corner of the board, and then you can drag them to where you want them to appear on the circuit. Note that CMax allows you to place components in locations on the board that don't have any holes; this is to give you room to maneuver, but be careful not to leave any components disconnected.
- To obtain a component in a different orientation, hold down the **Shift** key when you select it from the menu.
- Resistors are rectangles with three color bands. You can change the value of the next resistor you place by clicking on the color stripes of the *prototype* resistor icon (the one with the text in it); it will cycle through the colors. **Shift**-clicking a resistor band will cycle through the values in the other direction.
- You can connect any two locations on the board with a wire by clicking on the first spot and dragging to the second.
- Wire ends and component leads must be at one of the gray dots that repesent a hole. Only one wire or component lead can occupy a hole.
- You must connect your board to power and ground by adding the `+10` and gnd components to it. You must have exactly one of each.

Modifying your circuit

- You can delete a component or wire by holding down the control key (you see a skull/crossbones cursor) and then clicking on the body of the component or on a wire.
- You can move the endpoint of a wire by clicking and dragging it; you can move a whole wire by dragging the middle.
- Moves of components and component creation can be undone using **Undo**. The undo operation is only one level deep, so hitting **Undo** again will re-do the operation.
- To read the value of a resistor in your layout (in case you forget what the color bands mean), **shift**-click the resistor. The value of that resistor will be shown in the prototype resistor button.

File management

- The **Quit, Save, Save As, New** and **Open File** commands should do what you expect. Make sure that the files you create to save your circuits have a `.txt` extension. The **Revert** button will erase the changes you have made since the last time you saved the file.

Running tests

There are several ways to see what happens when you run your circuit. The **Simulate** button will run your circuit; it needs to use an input file that specifies time sequences of inputs to the potentiometers in your circuit. You won't ever need to write an input file; we will specify them for you, to run particular tests.

When you click **Simulate** the first time, you pick a test file. It will use the same test file thereafter. If you **Shift**-click **Simulate**, it will re-prompt you for a test file, so you can select a different one.

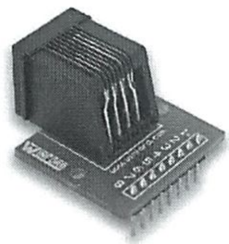
- You can measure the voltage between two points in your circuit by placing a `+probe` and a `-probe` (exactly one of each) in your circuit, hitting the **Simulate** button, and selecting the file `noInput.py`; it will print the voltage across the probed locations in the window from which you started Python. If there is a component with temporal dynamics (a potentiometer or a motor) in your circuit, then when you simulate, it will also pop up a window showing the signal generated at the probe.
- If there is a *motor* in your circuit, when you hit **Simulate**, a window will pop up that shows a time sequence of the motor's speed, in radians per second.
- If you want to see how your circuit behaves as a function of an input signal, you can add a *potentiometer*. If there is a potentiometer in your circuit, when you hit **Simulate**, a window will pop up that shows a time sequence of the potentiometer alpha values, so you can see what the input is that your circuit is reacting to.

Debugging

Here are some common problems:

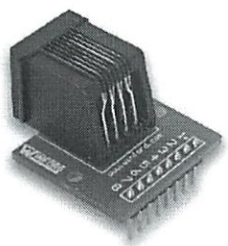
- `Failed to solve equations! Check for short circuit or redundant wire` This can be caused by connecting power to ground, for example. Examine your wiring. Maybe you inadvertently used the same column of holes for two purposes. At worst, you can systematically remove wires until the problem goes away, and that will tell you what the problem was.
- `Element ['Wire', 'b47', 'b41'] not connected to anything at node b41` The name `'b41'` stands for the **b**ottom group of five holes, in column 41. If you get a message like this, check to see what that element should have been connected to. You know that there should be something else plugged into the bottom section of column 41, in this case.
- `Illegal pin` means that you have a wire or component that has an end or a pin in position on the board that does not have a hole.

Head Connector Pin-out



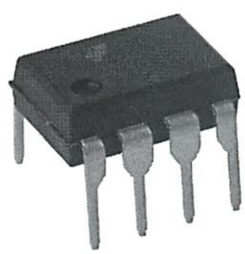
pin 1:		neck pot (top)
pin 2:		neck pot (center)
pin 3:		neck pot (bottom)
pin 4:		photoresistor (left)
pin 5:		photoresistor (common)
pin 6:		photoresistor (right)
pin 7:	V_{M+}	Motor drive +
pin 8:	V_{M-}	Motor drive −

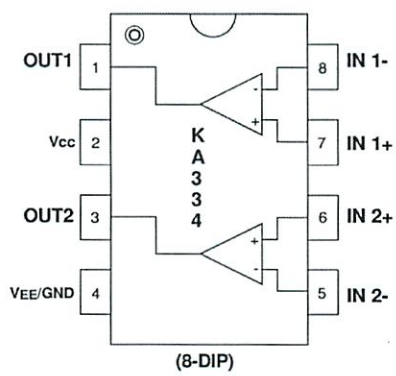
Robot Connector Pin-out



pin 1:	V_{i1}	analog input #1
pin 2:	$+10\,\text{V}$	power (limited to $0.5\,\text{A}$)
pin 3:	V_{i2}	analog input #2
pin 4:	ground	
pin 5:	V_{i3}	analog input #3
pin 6:	V_o	analog output
pin 7:	V_{i4}	analog input #4
pin 8:	$+5\,\text{V}$	power (limited to $0.5\,\text{A}$)

Op-Amp Pin-out





desLab 4

11/4

Build layout in CMax

Such a mess!

- but works out

Then wire up

Straighted some things out

Power up....

- need black cable head board -> head motor
  - I wired ~~head~~ motor to 8+9 - not 7+8
    - opps! - hard to read

Works really well!

Now need to measure

H1 -> 10

H2 -> R1

H3 -> gnd

Use TurntoLightAnalogBrain.py

(2)

Get a graph + settle time

Settle time = 4

Step 8

Pick a good gain

- this is pretty good, but need to try other things
- also try at 2 distances

Gain = 1 further distance

Settle = 6

Now new gain = 5

- need to identify what resistor to change!
- change R_2 to ~~4~~ 4 kΩ resistor

· Opps I never built the op amp for gain!,

- thats why I had those extra wires

Gain 5

Short

Long 34

n

6.01 Exam 2 Review Session

11/8

- exam only Tue
- no lab, no OH Tue
- no desLab Thur holiday

Concepts

1. Circuits 40-50%
2. System Functions
3. State Machines
4. Software

↑ importance

review

- could be trick - implement a SM w/o needing state
- converge, etc, poles, O picture

guess where poll is when have ossilations

---

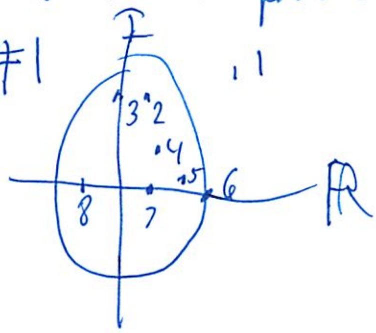
his take taking approach

1. Read every problem on exam
   - 5 min
   - know what about
2. Tackle easy problems
3. If stuck "mark" and move on
4. Ask for clarification
5. profit

2

going to do practice problems

Spring10#1



- all we ~~a~~ care about
  - along real line
  - inside 1st quad

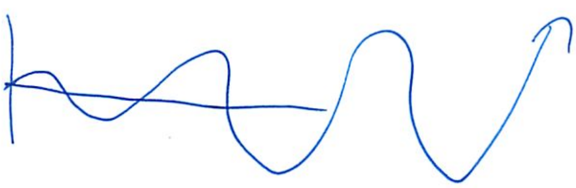
#1

- ossilitory divergant

has some imaginory

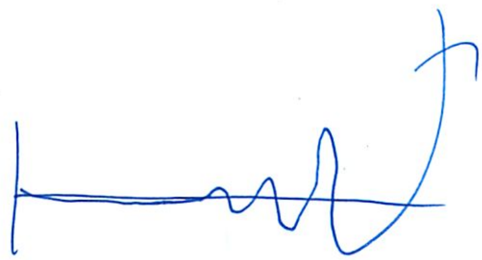
outside circle

pole may > 1



A

or



depends on scale of graph

#2

- inside, still complex

not divergant convergant

still ossilates

(but remember we are doing discritazation of it)

#3

- purly imagionary
- no real component

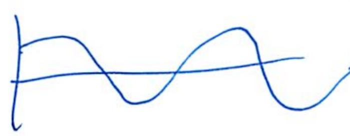
E

exactly on the line - fastest occilation at sample rate
- faster would not capture right

3

#4

- how #2, #4, # 5 different
  - speed of / freq of oscilation
  - higher imaginary than real → faster ossilation

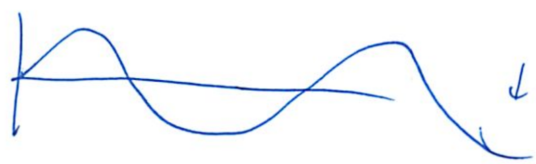


↓converges

ossilatets medium ~~slow~~ speed

#5

Ossilates very slowly



↓ converges

#6

Outside circle, on real line

↓ diverges ↓ monotically ↑ no ossilate

B

#7

~~[illegible]~~ on real, inside

↓ monotonic ↓ conversion

- sometimes too slow above

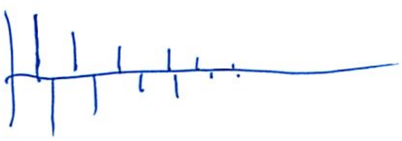
F

#8

differs by — sign from 7

- every other value is negitive

D



no passing through 0!

- so can tell difference

(4)

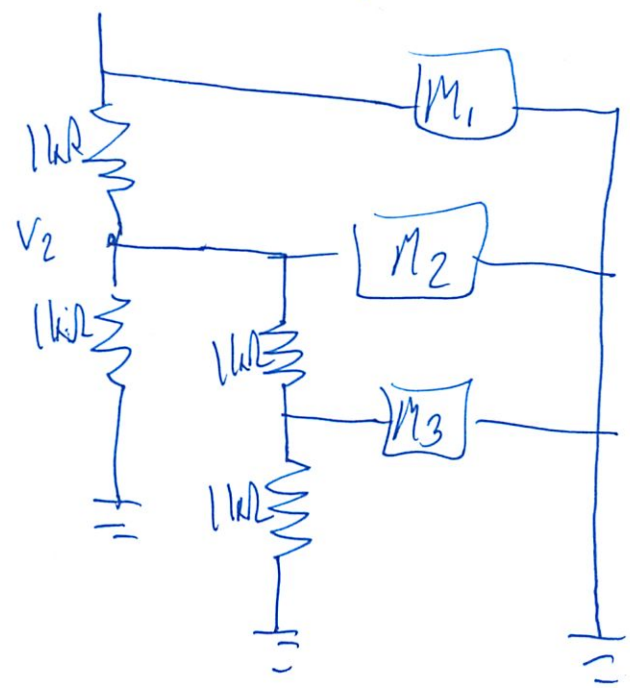
the larger the magnitede (near 1) takes a long time to converge

smaller " " (near 0) converges quickly

0 - converges immediatly

Spring 10 #4 Motor Control

- engineers have a bunch of solutions
- Voltage drop accross motors

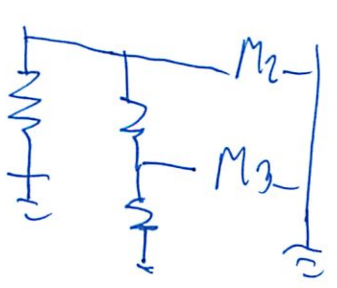


$M_1 = 10V - 0V = 10V$

$M_2 = V_2 - 0V$

$V_2 =$ main too voltage divider
but also resistors to right
ground on right - so not ideal

need eq resistance

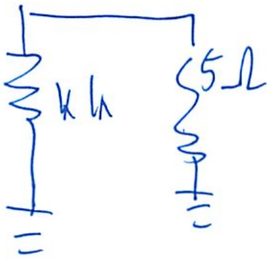


(5) know resistors = 5Ω

1k − 5Ω parallel = 5Ω

" " "

Could also do KCL



again ~~eff resistance~~ 5Ω

1k − 5Ω = 5Ω

$$M_2 = \frac{5}{1005}(10\,V) - 0\,V = .05 \approx 0$$

- remember the lab
- I thought of that!
- yeah building circuit intuition

M_3 = super 0 ~~also~~ − starting at .05

4.2 just change Resistances

10 V up top same

100 Ω in series w/ blah

− but just made resistances even larger

~~M_2, M_3 even closer to 0!~~

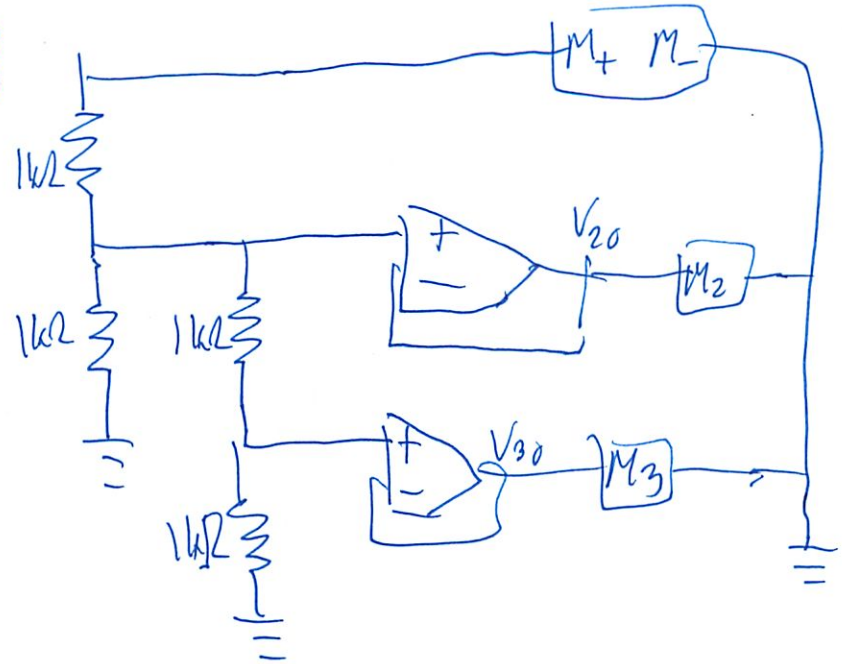
$M_2 = \left(\frac{5}{105}\right) 10\,V = .5$ − actually makes our scenario better!

$M_3 \approx$ basically 0

loading − power supply can't provide enough current to maintain voltage − so voltage drop

⑥

4.3



$= 10V$

our ideal in op amp $V_+ = V_-$

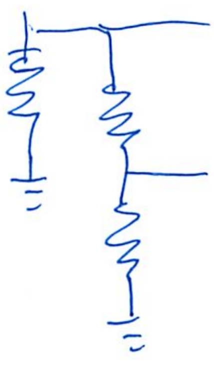
when we have to wire $V_- \to V_0$

$V_+ = V_- = V_0$

Voltage follower

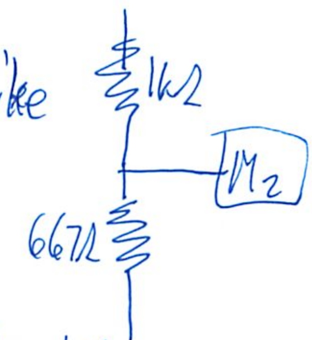
but does op amp help
- buffer / isolation
  - They don't draw current

- So only care about



$R_{eq} = \frac{1000 \cdot 2000}{1000 + 2000} = 667\Omega$

so like

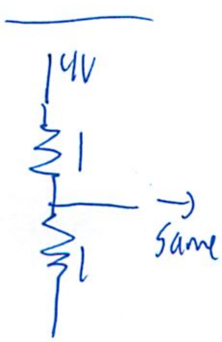


remember $\frac{\text{bottom}}{\text{sum}}$

$M_2 = \left(\frac{667}{1000 + 667}\right) 10V - 0V$

$\approx 4V$

Power supply must source at least 2 Amps



so $\frac{1}{2}(4) \cdot = 2V$

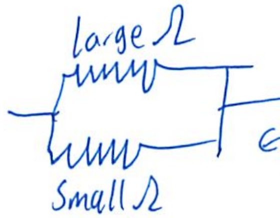
(1)

4.4 Kim (not redrawing)

$M_1 = 10$

Now need eq resistance of bottom

~~the~~ remember assumption



← just the small

$$M_2 = \left(\frac{100}{100+100}\right) \cdot 10V = 5V$$

$$M_3 = \frac{1}{2} 5V = 2.5V$$

$\left(\frac{100{,}000}{200{,}000}\right)$

– what we wanted! (but kinda hard to do – relies on resistor trick)

4.5 Jamie (this is how I would build it – safest)

$M_1 = 10V$

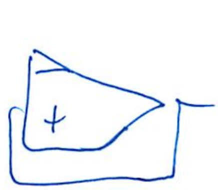
$$M_2 = \frac{1000}{2000} \cdot 10V - 0 = 5V$$

$M_3 =$ (: actually will this work at?
– yeah op amp puts out whatever current needed within reason)
– assumption we made before ~ same source)

$$\left(\frac{1000}{2000}\right) 5V - 0V = 2.5$$

---

if



will slam to rails – ie 10V or 0V

very unstable

$V_o = k(V_+ - V_-)$

internal $k \approx 10^6 \approx \infty$ assumption

$V_o = k(V_+ - V_-)$

here

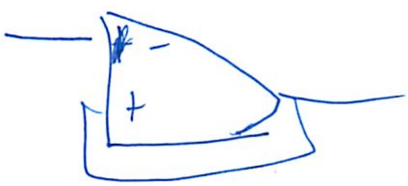
$V_o = k(V_+ - V_o)$

⑧ $(k+1)V_0 = k \cdot V_{in}$

$V_0 = \frac{k}{k+1} \cdot V_1$

in real life k varies widly
- L so don't design for a certain k
- L heat errors would be large

if



$V_0 = k(-V_{in} + V_0)$

$(1-k)V_0 = -k \cdot V_{in}$

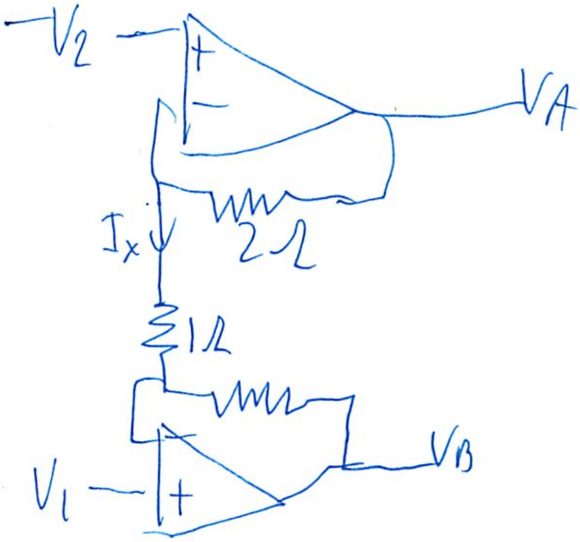
$V_0 = \frac{k}{k-1} \cdot V_{in} \rightarrow V_{in}$

? basically a <u>proportional controller</u> - like his golumn story

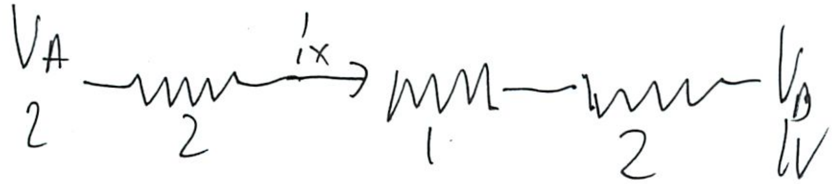
---

~~nano quiz 10 → on paper~~

~~Spring 08 → #3 → on paper~~



ha - I remember this one
can I remember?

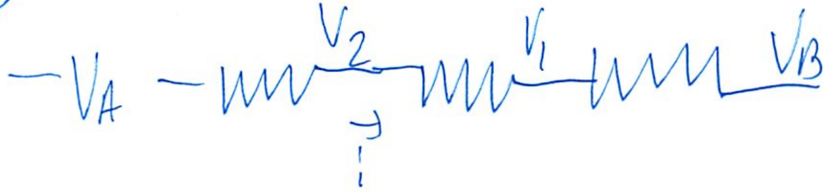


$2-1 = 5i$

- he liked my trick

oh V_A and V_B not realy known

(9)



$V_2 - V_1 = i \cdot 1$

$1 = i1$

$i = 1$

$\frac{V_2 - V_1}{R} = I_x$

b) $V_{\cancel{A}A} - V_2 = iR$

$V_A - 2 = 1 \cdot 2$

$V_A = 4$

$V_A = iR + V_2$

c) but where V_1?

$\frac{V_A - V_1}{3} = \frac{V_2 - V_1}{1}$ current same on both so =

$VA - V_1 = 3V_2 - 3V_1$

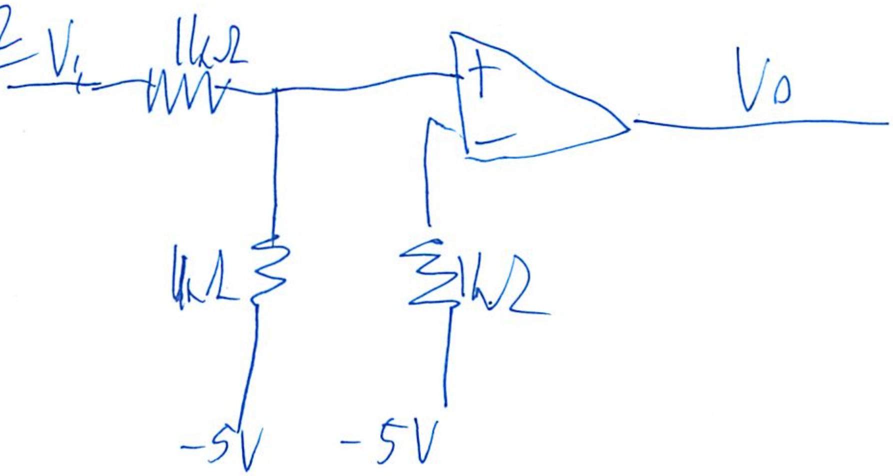
$V_A = 3V_2 - 2V_1$

~~A~~ don't rely too much on tricks!

10

Part 2

$V_+ = V_-$



a) both voltage dividers

$$\frac{V_1 - -5V}{2k} = \frac{V_+ - -5V}{1}$$

$4 - 5 = V_+ = -1$

$V_o = 3V$ ← must be = in b/w 4V drop across both resistors

b) $V_1 = 7$ solving a bit different

$$V_+ = \frac{V_1 - 5}{2} = V_- = \frac{V_o - 5}{2}$$

? must be $\frac{1}{2}$ since voltage divider

$V_1 - 5 = V_o - 5$

$V_1 = V_o$

So $V_1 = 5V$

$V_o = 5V$

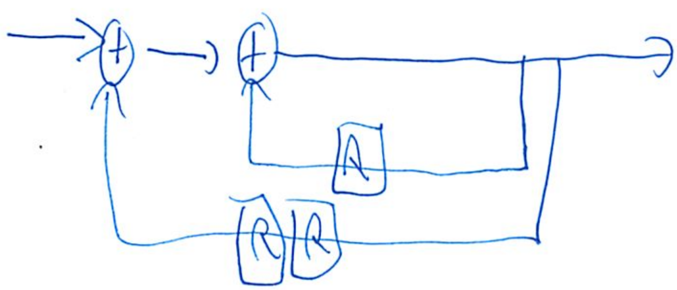
and same for $V_1 = 4V$

(11)

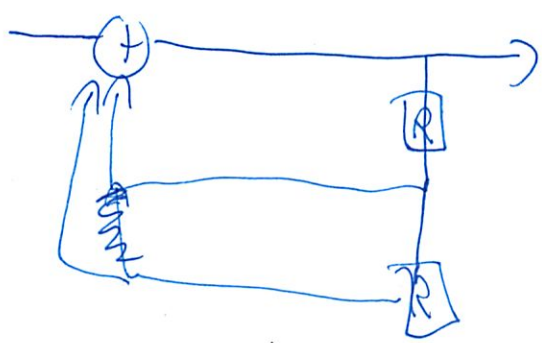
Spring 09 #3 State Machine Behaviors

first make diff eq

$$y[n] = x[n] + y[n-1] + y[n-2]$$



or



note $x[n]$ - so can do anything to top line

---

transduce 1, 0, 0, 0, 0, 0, 0, 0

1	1	2	3	5
↑	↑	↑	↑	↑
1, 0, 0	0, 1, 0	0, 1, 1	0, 2, 1	0, 3, 2
new state	new state	new state	new	
(1, 0)	(1, 1)	(2, 1)	(3, 2)	(5, 3)

... fibinici seq.

(12)

<u>Part c</u> mag dom pole

have diff eq already
now need system function

$$Y = X + YR + YR^2$$

$$Y(1 - R - R^2) = X$$

$$\frac{Y}{X} = \frac{1}{1-R-R^2}$$

So long since I did this last

now solve for roots of denom

- need to convert w/ $z = \frac{1}{r}$

$$\frac{1}{1-\left(\frac{1}{z}\right)-\left(\frac{1}{z}\right)^2} \cdot \frac{z^2}{z^2}$$

$$\frac{z^2}{z^2-z-1}$$

quadratic formula

$$= \frac{-1 \pm \sqrt{(-1)^2-4(-1)(1)}}{2}$$

$$= \frac{1 \pm \sqrt{5}}{2}$$

know $1+\sqrt{5} > 1-\sqrt{5}$

$\frac{1+\sqrt{5}}{2}$ thats the answer

13

part d (totally unrelated)

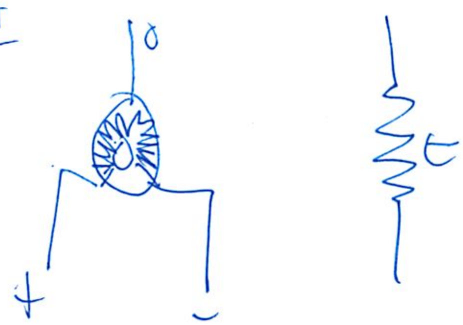
$H = \frac{Y}{X} = 1 - R^3$

so $y = x - xR^3$

$y[n] = x[n] - x[n-3]$

```
def getNextValue(self, state, inp):   ← remember exact syntax
    state = (x1, x2, x3)
    yo = inp - x3
    return ((inp, x1, x2), yo)
```

Pot



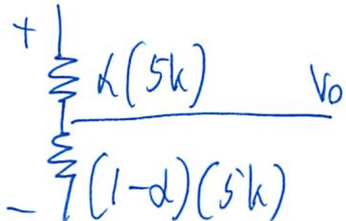
Symetric
- does not matter which terminal you choose

at start → no resistive tape covered - full voltage

as slide → more resistance tape - so voltage drop

α = normalization of angle

5V = end



basically a voltage divider