Physics 364, Fall 2014, Lab #16 **Name:** _____

(PID motor-control lab)

Monday, October 27 (section 401); Tuesday, October 28 (section 402)

Course materials and schedule are at `positron.hep.upenn.edu/p364`

Today, we will build and try out an opamp-based P-I-D controller (Proportional, Integral, Derivative). PID controllers are useful in a wide range of control applications that involve feedback. For instance, you may want to output a voltage that controls the speed of a motor such that an elevator stops at the desired floor (that's **control**), in a system where one of your circuit inputs tells you how far away the elevator currently is from the desired position (that's **feedback**). The key idea is that there is an error function $E(t)$ that tells you at a given time how far away you are from the desired state (e.g. how many millimeters your elevator is from the 3rd floor, where you want it to go). The output that you send to the motor at time t has three terms: one that is **proportional** to the present $E(t)$, one that is proportional to the present **derivative** $\mathrm{d}E(t)/\mathrm{d}t$, and one that is proportional to the recent **integral** $\int_{t-\Delta t}^{t} E(t')\mathrm{d}t'$.

The PID controller can be a useful addition to your toolkit, if you work in a research lab. It is also a pretty sophisticated application of opamps, combining the opamp difference amplifier, the opamp inverting amplifier, the opamp differentiator, the opamp integrator, and a push-pull transistor follower into one big circuit. So it provides an opportunity to review several opamp circuits. PID controllers are often implemented using computer programs instead of opamps. While it is not surprising to see a computer perform a sophisticated control task, it is fun to see this job done by hardware as "unintelligent" as a few opamps.

Today's circuit is even more complicated than Lab 15! Neat and systematic work will speed up your debugging.

There will not be too much for you to write down today, as we want you focus on building and testing the controller. We'll sprinkle in a few questions here and there just to make sure you're thinking things through.

This lab is borrowed from Tom Hayes's *Physics 123,* Lab 10. We have taken his lab and made minor adaptations to fit our course and the available materials. Where possible, we've used Tom's figures and wording. Tom was kind enough to help us find the linear motor-potentiometers that are at the center of today's lab; unfortunately, the rotary motor-pots used in the original *Physics 123* lab are not currently available from any vendor.

Part 1 **Start Time:** _____

introduction (time estimate: 10 minutes)

1.1 (Theoretical background) Today's circuit looks straightforward: a potentiometer sets a *target position* of a sliding lever that moves on a conveyor belt; a DC motor turns the conveyor belt to try to achieve that position, which is measured by a second potentiometer. This is directly analogous to trying to send an elevator to the desired floor, with feedback telling you the elevator's current position. There are two complications.

The first complication is that the voltage output by your controller does not directly set the quantity that you want to control (the slide position). Instead, the voltage output sets the motor torque, which (along with friction) determines the rate of change of the slide position. In the language of Fourier analysis, if you were to represent the slide position as $\sin(\omega t + \phi)$, the slide position would be phase-shifted by $-90°$ w.r.t. the voltage sent to the motor, because speed is the rate of change of position. When designing a system to use negative feedback, one needs to be aware of phase shifts, because a phase shift of 180° would turn *negative* feedback into *positive* feedback ($e^{i\pi} = -1$), which would transform a stable system into an unstable system. This is the reason why an opamp intentionally reduces its own gain at high frequency ("*frequency compensation*"): so that the gain is $\ll 1$ at high frequencies, where stray capacitance within the opamp causes large phase shifts.

The second complication is that systems using negative feedback have a tendency to oscillate. Even a simple thermostat will cause the temperature to oscillate somewhat about the set point. Imagine a mechanical system designed to keep an object fixed at a point of stable equilibrium: if the object is too far to the left, a restoring force pushes the object to the right; if the object is to far to the right, a restoring force pushes the object to the left. You know that in the absence of adequate damping, such a system will oscillate at its natural frequency. A feedback system needs a damping mechanism to prevent oscillation.

To recapitulate: First, you need to make sure that the feedback is always negative, never positive, so that the equilibrium is stable, not unstable. Second, you need sufficient damping to prevent oscillations about the stable equilibrium.

We will see that the **P**roportional term provides the "restoring force" that you would find in a mechanical oscillator. If the present setting is too far to the left, you want to turn the motor toward the right, and vice versa.

The **P**roportional term gives a differential equation like that of a simple harmonic oscillator, so in fact it is possible to oscillate around the equilibrium position, if the proportional term is large enough to overcome the mechanical friction in the motor. Therefore, the **D**erivative term is there to act like the damping term in a damped harmonic oscillator, proportional to the present rate of change of the displacement from equilibrium. In the absence of a **D** term, one needs to keep the **P** coefficient quite small to avoid oscillations. If the **D** term is present, one can afford to make the **P** term big enough to move more quickly toward the equilibrium position. A properly tuned **D** term prevents the controller from overshooting

the target, as in the case of a critically-damped oscillator.

Finally, the **I**ntegral term is useful because sometimes you find that there is a small difference between the present position and the target position, but the **P** term is not large enough to overcome mechanical friction for such a small error. The **I** term gets rid of this long-term-average error. One often sees elevators do this in the instant before the doors open: the elevator stops very close to the right place, then after a second or two makes a final adjustment, then opens the door. Presumably, that final adjustment is due to the **I** term taking out a small error that persists over a long time.

1.2 (Practical background) It is much more fun to build circuits that *do* something — in contrast to circuits that just produce images on a scope screen. Today's circuit qualifies: it will be like building your own little elevator controller. You can imagine yourself as a very unhappy elevator passenger when your underdamped motor causes the elevator to oscillate violently around the desired height, and as a much happier passenger when the elevator smoothly and quickly reaches its destination.

As we noted on the front page, this circuit is also great review and practice with opamp circuits. Opamps are the Phys 364 tool you're most likely to use when this course is long over.

This PID circuit is by far the most complex we have built yet. That makes it a good setup for improving your debugging skills. (In other words, you're likely to make some wiring errors today.) Electronics courses often boast that *bugs are their most important product*, as well-honed debugging skills are useful in a wide range of other contexts.

1.3 Today's goal, most simply stated, is to use a feedback loop to get one DC voltage to match another. Since both voltages come from potentiometers, the goal can be restated as making the position of one potentiometer mimic that of another. We want to be able to use our fingers to turn a potentiometer by hand and see a motor-driven pot mimic our action. Such controls are sometimes used on fancy audio equipment (e.g. a recording studio's giant sound-mixing panel), so that the equipment can be controlled either by twisting a knob, or by using a remote that controls the knob from across the room (or from settings saved in a computer).

1.4 The motor-potentiometer assembly

Look over the motor-pot gadget to get an idea for how it works. The motor turns at a speed proportional to the applied voltage, which can be positive or negative. A little gear attached to the motor causes a grooved belt to move when the motor turns. The slide lever (the "elevator cab") is attached to the belt (the "elevator cable"). A clutch mechanism allows the belt to slip, if the force between belt and motor becomes too large. This clutch serves two purposes: first, it permits a human hand to move the slide directly, when the motor is stopped. (That's useful for the audio-mixer-booth application.) Second, it protects the motor against stalling and overheating, if the motor drives the pot to one of its limits.

If you have time before class, you might enjoy watching this three-minute video showing several of these motor-pots being used in a kind of ping-pong game.

`https://www.youtube.com/watch?v=E2Stni6W7Vc`

1.5 The motor control loop

Here's an overview of today's circuit. One potentiometer sets the "target" position. The motor-potentiometer reports its actual position. A differential amplifier subtracts the two position readings, amplifies this "error" signal, and drives a motor, which in turn changes the motor-potentiometer's setting.

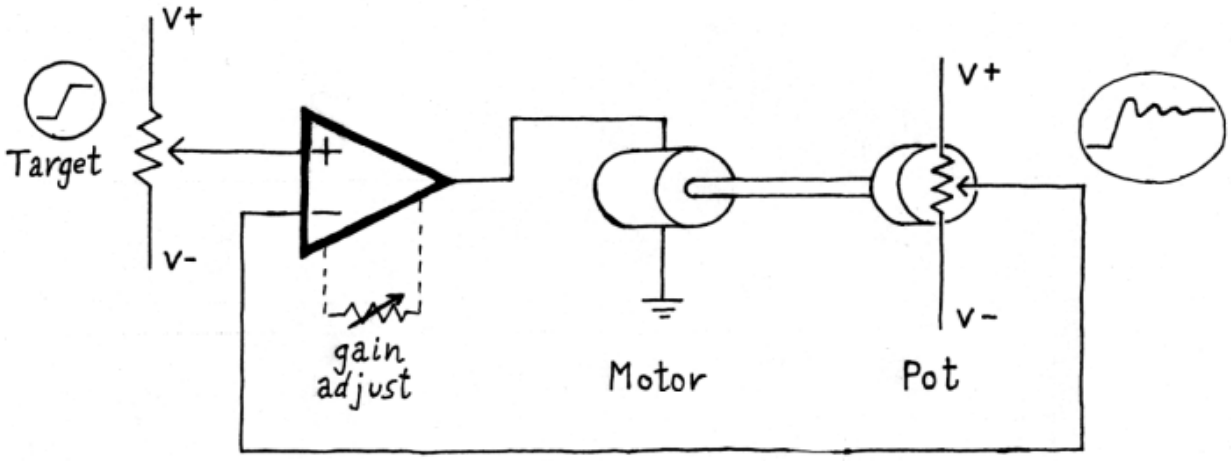


Figure 3: Proportional-only drive will cause some overshoot; gain will affect this

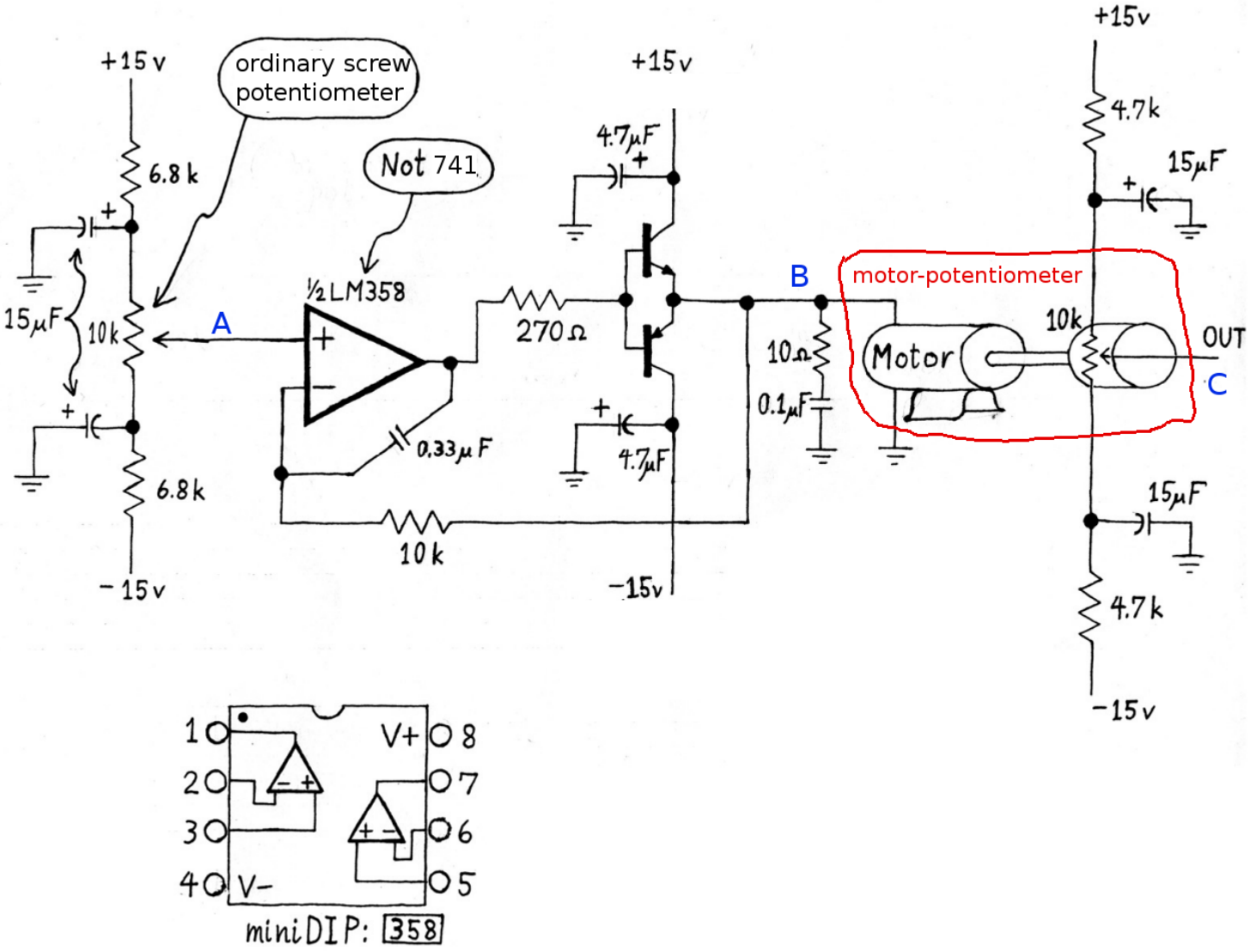
We will first try this circuit with its gain adjusted *low,* and we expect to find the circuit fairly immune to oscillation, because of mechanical friction. Then, as we increase gain, we should begin to see overshoot and ringing. If we push on to still higher gains, we should see the circuit oscillate continuously.

Two cautionary suggestions, to reduce the risk of damaging these (somewhat expensive) motor-pots. First, turn off the power-supply outputs while adding components and wires to your circuit. Second, keep an eye on your power-supply's reported power, and switch off your circuit (and ask us for help) if the total exceeds 5 watts.

Part 2 **Start Time:** _____

motor driver (time estimate: 45 minutes)

To drive the motor, let's start with a familiar circuit: a high-current push-pull follower, capable of driving a substantial current — up to a couple of hundred milliamps. We'll use the power transistors you've met before: MJE3055 (npn) and MJE2955 (pnp). A feedback loop using an opamp follower will remove the push-pull's crossover distortion. Because driving inductive loads, such as motors, tends to provoke amplifiers to oscillate, this motor-driver circuit takes several precautions: decoupling capacitors at power supplies, a high-frequency *snubber* between the push-pull and the motor, and an 0.033 μF *high-frequency feedback* capacitor to reduce the opamp's high-frequency gain.



Wire up the two potentiometers (the ordinary 10 kΩ screw potentiometer and the big motor-pot), as well as the motor-driver itself. The resistors at the ends of the two potentiometers — 6.8 kΩ resistors on input, 4.7 kΩ resistors on the motor pot — restrict input and output to a range of about ±7 V, to keep all signals well within a range that keeps the opamps happy.

The difference in R values makes sure that the input range cannot exceed the achievable output range.

Wiring diagram for your motor-pot. The motor-pot has two wires for driving the motor: red=motor, black=ground. Of the two wires twisted together at the motor, the red one is point **B** on the circuit diagram, and the black one should be grounded. The "potentiometer" half of the motor-pot has three wires, just like an ordinary potentiometer: red=positive, green=negative, yellow=wiper. So the yellow wire is the "output" (point **C** on the circuit diagram) that reads the pot position, while red goes to the upper 4.7 kΩ resistor, and green goes to the lower 4.7 kΩ resistor. The total resistance of the motor-pot's potentiometer is 10 kΩ.

The opamps are LM358 opamps (which come two-to-a-package), not your usual 741 opamps. You can use both halves of each LM358 package, to save space, or you can use a separate LM358 package for each opamp, if you find that easier for making your circuit layout better resemble the schematic diagram. It's up to you.

When you look at the face side of the MJE3055 (npn) or MJE2955 (pnp) transistor, with pins downward, the pin order, from left to right, is **B**ase, **C**ollector, **E**mitter.

A few notes before you start building:

- Note that the power transistors may get hot, so watch your fingers. **Ouch!**
- But **do** touch the motor-pot's motor every now and then to make sure that the motor is not becoming too warm.
- **Don't let the power transistors' tabs touch one another,** as each tab is connected to the corresponding transistor's **collector.** So touching the two tabs would short +15 V to −15 V. **Zap!**
- Also, keep an eye on the total power and make sure it stays below about 5 watts.

Test the motor driver by varying the input voltage (with the input pot) and watching the voltage out of the motor-driven pot. You should be able to vary its speed and direction by dialing the input potentiometer. While doing this, you should watch points **A**, **B**, and **C** simultaneously with your oscilloscope, to see what is happening. Make sure that a positive V_{in} (point **A**) causes V_{out} (point **C**) to increase, and that a negative V_{in} causes V_{out} to decrease.

(Blank page.)

Part 3 **Start Time:** _____

multi-stage amplifier (time estimate: 30 minutes)

Now we do a strange thing: we use three opamps to make a multi-stage amplifier circuit. The first stage is an opamp "difference amplifier," whose purpose is to compute the "error" by subtracting the actual motor-pot position from the "target" position, i.e. by subtracting the two potentiometer outputs. It has unity gain. The second stage simply inverts, for reasons that will become clear once we add the **D** and **I** stages to this **P**-only circuit. The third stage seems to be simply *undoing* the inversion of stage 2. That is true, for now, but soon we will use stage 3, fed by two more inputs, as a *summing* circuit to combine **P**, **I**, and **D** signals. In the present **P**-only circuit, we also use stage 3 to vary the overall gain of our multi-stage amplifier.

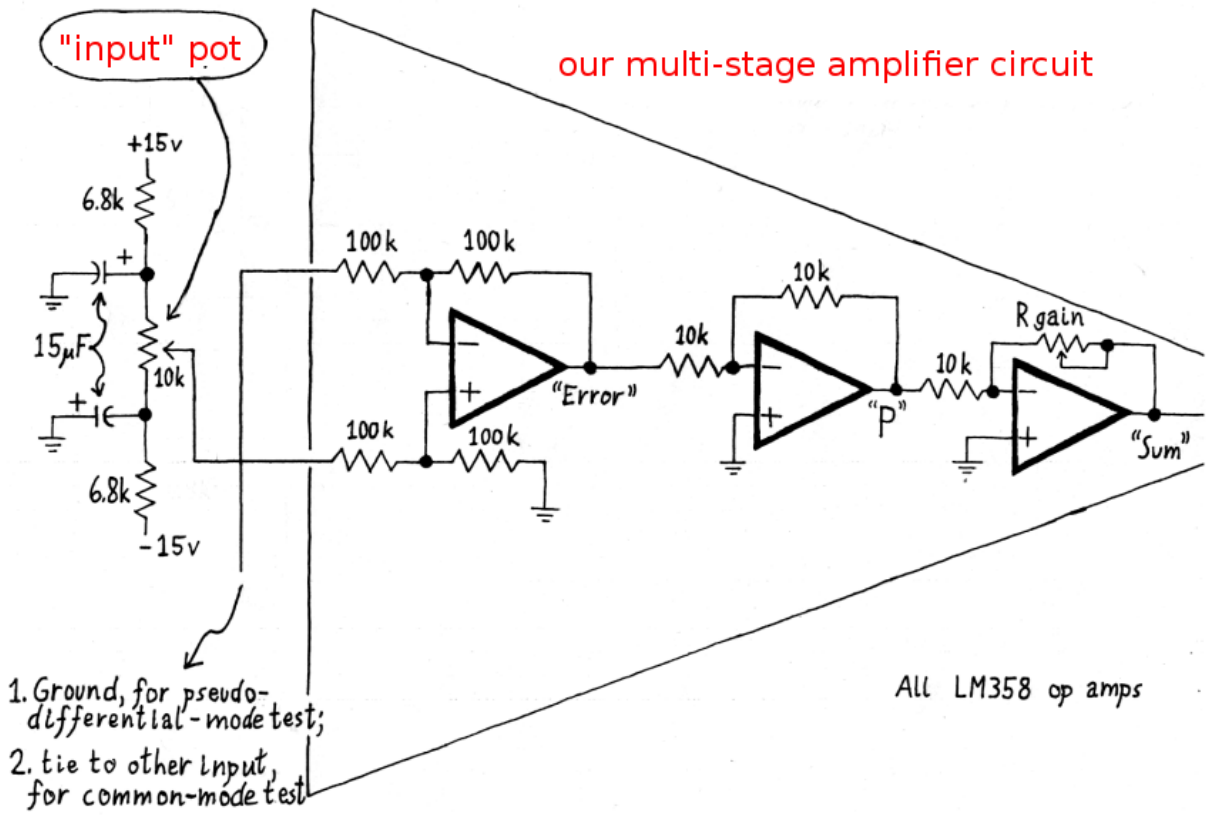


Figure 7: Differential Amp Followed by Gain Stage and an inversion

The entire multi-stage amplifier circuit is simply a differential amplifier with adjustable gain. And this gain is always low relative to the very-high values we are accustomed to in opamps. We will need this modest gain (and lack of appreciable phase-shift between input and output) to keep our PID controller operating stably and without making the motor-pot oscillate wildly.

Note that you have already built the "input pot" on the left side of the diagram. You are

simply inserting the triangular amplifier block between the existing "input pot" and the rest (to the right of point **A**) of the existing "motor driver" schematic.

3.1 Build the multi-stage amplifier circuit shown above, and connect it to the ordinary potentiometer that is your "input" pot. Be careful to follow the suggested capacitor polarities! Use $R_{\text{gain}} = 100\ \text{k}\Omega$ for an overall gain of 10. Make a **very quick** check of the common-mode and differential gains, to check that your amplifier is working.

Common-mode gain. Using a 1 V_{pp} signal at about 10 Hz from one channel of the function generator and a wire to send that signal into both amplifier inputs (i.e. into both inputs of the opamp difference amplifier), see whether the output of your multi-stage amplifier is indeed very small for this common-mode signal.

Differential gain. Then ground one input (the 100 kΩ that feeds the first opamp's *inverting* input), using the level from the input potentiometer as input. Use the scope to watch that input, and the circuit output, with R_{gain} set to 100 kΩ. See if you get the expected gain of +10.

When you finish testing, leave the potentiometer set such that the output voltage is close to zero volts.

You might find it quicker and more convenient to use the function generator instead of the "input potentiometer" to provide the "target" input to your difference amplifier. The choice is yours.

Part 4 **Start Time:** _____

drive the motor (time estimate: 45 minutes)

4.1 You have already tested the motor driver. Now let's check the three new stages — those that form the multi-stage amplifier — by letting their output feed the motor driver. Confirm that you can make the motor spin one way, then the other, by adjusting the input pot (or function generator) slightly above and then below zero volts. (The motor-driven pot fortunately can take the pot to its limit without damaging the pot or the motor.)

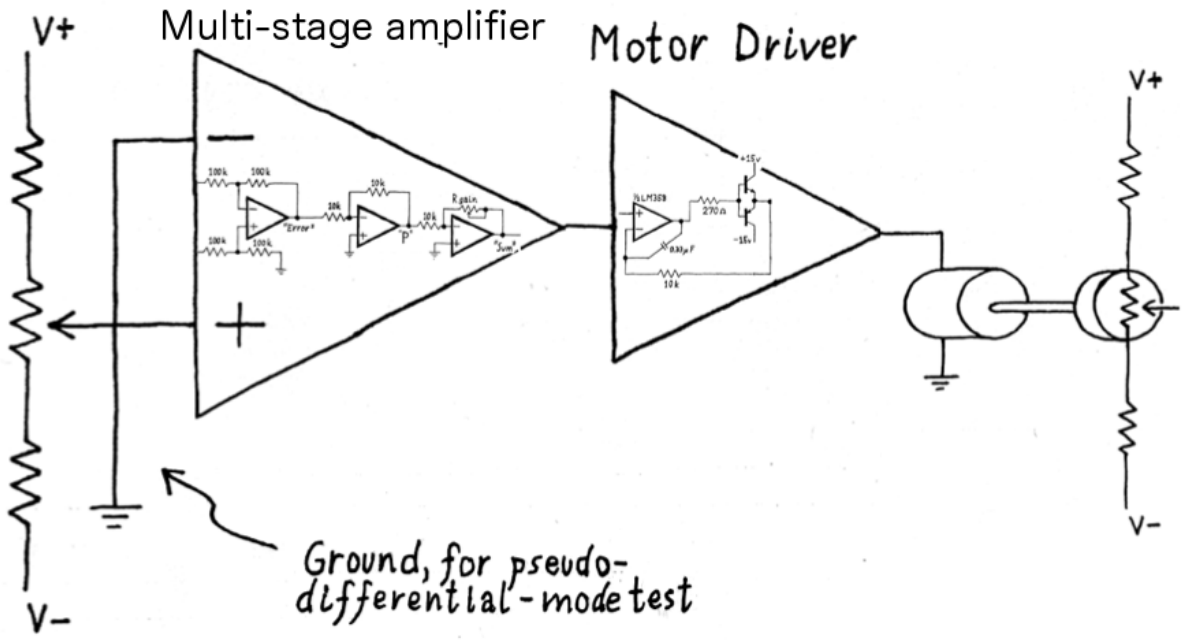
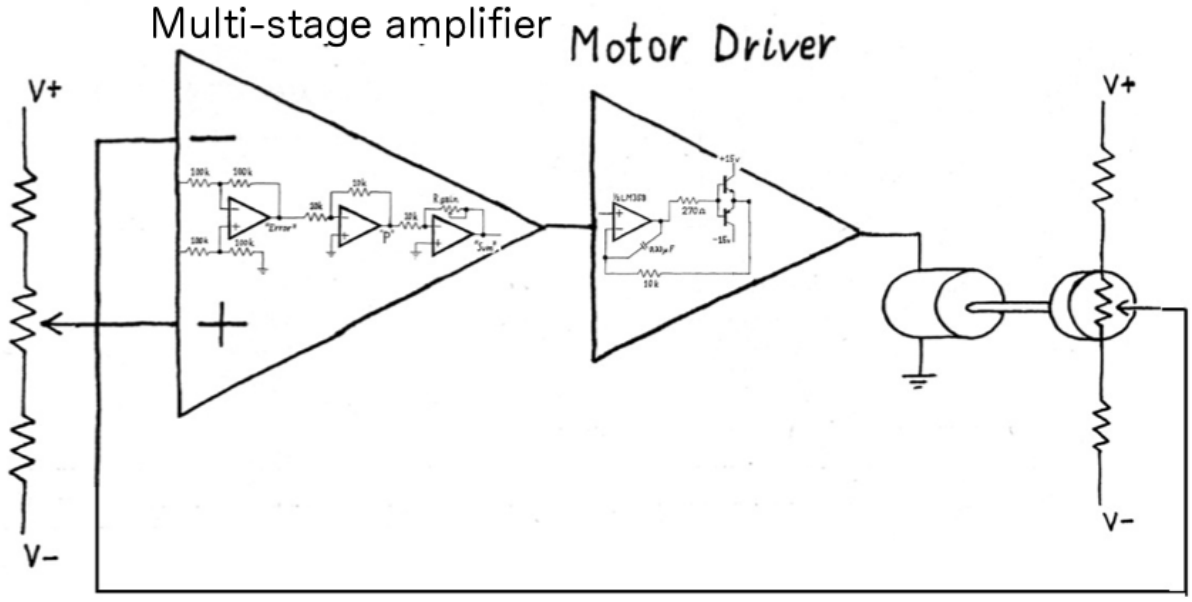


Figure 8: Try making motor spin, to test the diff amp, gain stage, sum and motor drive

4.2 Next, we'll **close the loop.** Reduce the gain to about 1.5, by setting R_{gain} to about 15 kΩ. Replace the *ground* connection to the inverting input of our multi-stage amplifier with *the voltage from the output potentiometer,* i.e. with the motor-pot output. Watch V_{in} on one channel of the scope (i.e. the signal from your input pot or from your function generator), and watch $V_{\text{output-pot}}$ on another scope channel (i.e. the signal coming out of the motor-pot). Use a very slow sweep rate on the scope, e.g. maybe 0.4 seconds per division.



To test your feedback loop (now running in **P**-only mode), apply voltage steps to V_{in} (the "target" setting), either by dialing the input potentiometer by hand or by using the function generator. It is almost certainly easier to use the function generator.

Two ways to drive the input:

- **square wave, from function generator:** The FG can provide a square wave (e.g. 5 V_{pp}) at a very low frequency (e.g. 0.2 Hz). This input can replace the *manual* input potentiometer, temporarily. This is probably the best choice, since it provides consistency that you cannot achieve by hand.
- **manual step input:** You may prefer the simplicity of *manually* applying a "step input" from the *input* pot: a step of about a volt.

An alternative test: disturb the output, and watch recovery: We highly recommend trying out this alternative test, as it is fun! Leave the input voltage constant, then manually force the motor-pot away from its resting position, simply by moving the slide with your finger. Let go, and watch the slide return to its initial position — showing some overshoot and oscillation, as when the change was applied at the input pot.

(Blank page.)

4.3 You started out with a very low gain (1.5), which should keep the circuit from oscillating, even in this **P**-only form. Now increase R_{gain} to increase the gain. At a gain of 20 or so, you should see some overshoot and transient oscillation. If your motor were controlling, say, the rudder of an airplane, the overshoot would be pretty unsettling. The circuit works — but it would be nice to get it to settle faster and to overshoot less.

As you increase the gain to ≈ 50, you should be able to make out several cycles of oscillation. You probably need to set the scope now to about 100 ms per division, as the oscillation frequency should be on the order of 40 Hz.

If you increase the gain even further (e.g. $\gtrsim 70$, using $R_{\text{gain}} \gtrsim 700$ kΩ), you should be able to provoke a continuous oscillation, merely by giving the system a mild jolt, either by hand or by changing the target voltage. **Find the gain at which your circuit starts oscillating, and then measure the period of oscillation** at the lowest gain that will give sustained oscillation. We will call this the period of "natural oscillation," and soon we'll use it to scale the remedies that we'll apply against oscillation. For comparison, Jose found that his PID controller started to oscillate continuously for a gain of 60–70; at this gain, he found an oscillation frequency of about 50 Hz. To see this effect, he used the FG to provide, as a "target" input, a 5 V_{pp} square wave at 0.2 Hz.

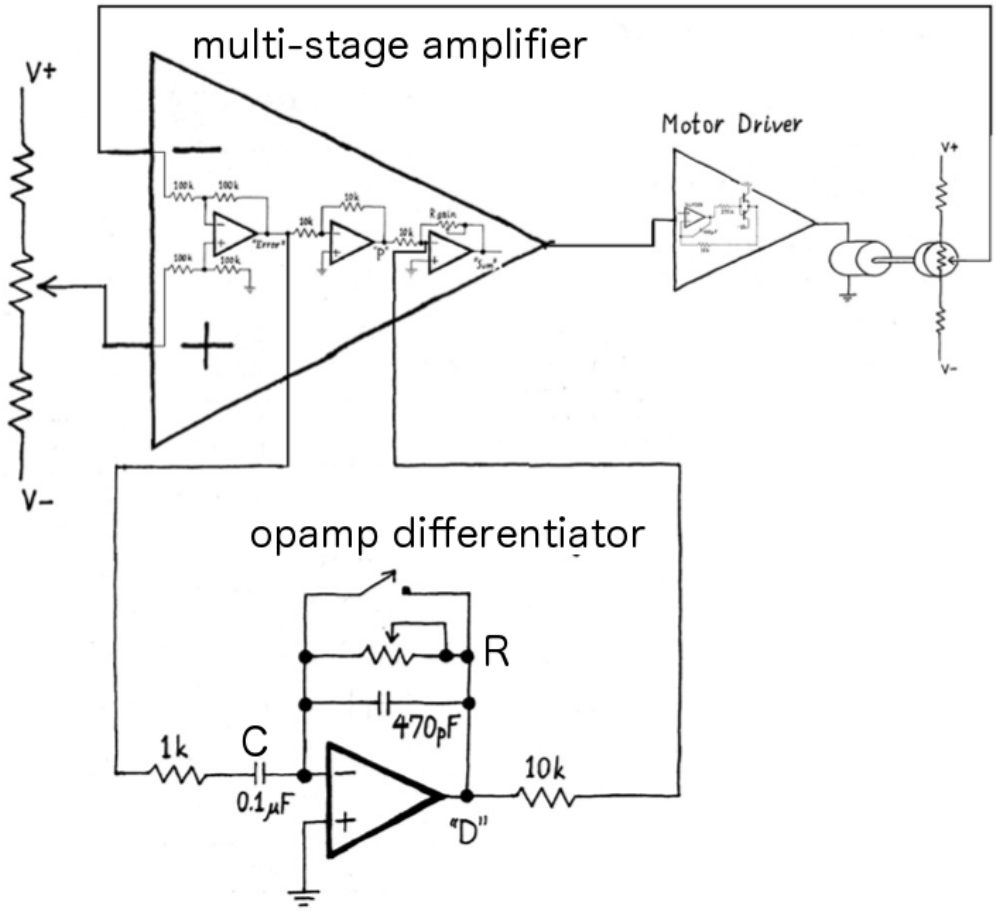
(Blank page.)

Part 5 **Start Time:** _____

add derivative term (time estimate: 40 minutes)

5.1 We can get the motor-pot to move quickly to its target position, without overshoot or oscillation, by adding the **D**erivative term, which acts like the damping term in a mechanical oscillator. We will aim for what would be called "critical damping" in an oscillator or in a car's shock absorber. The analogy with a mechanical oscillator is pretty good: a term proportional to the rate of change of the "error" signal plays the role of the oscillator's velocity-dependent damping term.

5.2 The opamp differentiator shown below will contribute its output to the Stage 3 summing circuit. Here, we show the entire circuit, with differentiator added. Note that an idealized opamp differentiator **would not** include the 1 kΩ input resistor, the 470 pF feedback capacitor, or the switch. The purpose of the input resistor and the feedback capacitor (which you might expect to form an *integrator,* not a differentiator) is to kill the differentiator's gain for high frequencies; the naive opamp differentiator circuit (without these added components that kill the high-frequency gain) tends to oscillate at high frequencies. The purpose of the switch is to allow you to disable the **D** term in your PID controller, by pressing a button. The big **R** and **C** on the differentator form the *RC* discussed below.



For the switch that appears in the feedback loop, you can choose either a push-button momentary-contact switch, or a pushbutton DPDT switch, or a simple piece of wire when needed. For the differentiator's feedback resistor R, use a 100 kΩ potentiometer.

5.3 How to calculate the needed Derivative gain: You may remember (or at least find it plausible) that the expression for an opamp differentiator's output is $V_{\text{out}} = -RC\frac{\mathrm{d}V_{\text{in}}}{\mathrm{d}t}$. So $-RC$ is the "gain" of the differentiator.

To prevent the PID controller from oscillating at the "natural frequency" you measured in Part 4.3, we want the **D** term to cancel out the **P** term at that frequency. Calling the measured "natural frequency" f_0, we want a **P** signal of the form $\sin(2\pi f_0 t)$ to be to be opposed by a **D** signal of the same magnitude.

The derivative of $\sin(2\pi f_0 t)$ is $2\pi f_0 \cos(2\pi f_0 t)$. The opamp differentiator scales this derivative by a factor $-RC$. To make these two magnitudes equal, we want $2\pi f_0 RC = 1$, or

$$RC = \frac{1}{2\pi f_0}.$$

In other words, RC should be about $\frac{1}{6}$ of the period of natural oscillation.

If you have studied "critical damping" of oscillators in mechanics, then this result should make perfect sense to you. For critical damping, the decay time equals $1/\omega_0$. So leave $C = 0.1$ μF in the differentiator and adjust R for the differentiator's feedback potentiometer such that $RC \approx 1/(2\pi f_0)$.

(Blank page.)

5.4 Wire in the **D** term as shown in the schematic, and give it a try. We hope you find this **D** to be strong and effective medicine. Once you've confirmed that adding **D** has tamed the overshoot and ringing in your controller circuit, continue to crank up the **P** gain (using the usual R_{gain} adjustment at Stage 3) to 100 or more (e.g. $R_{\text{gain}} = 1$ MΩ).

Does an excess of **D** cause trouble? The scope image of the circuit's response will let you judge whether you have too much or too little **D**. Too little, and you'll see remnants of the overshoot you saw with "**P** only." Too much **D**, and you'll see an RC-decay-like curve as the output approaches the target: it chickens out as it gets close to the target. (Yet more **D**, and the circuit becomes unstable, because of large phase shifts adding up to turn negative feedback into positive.)

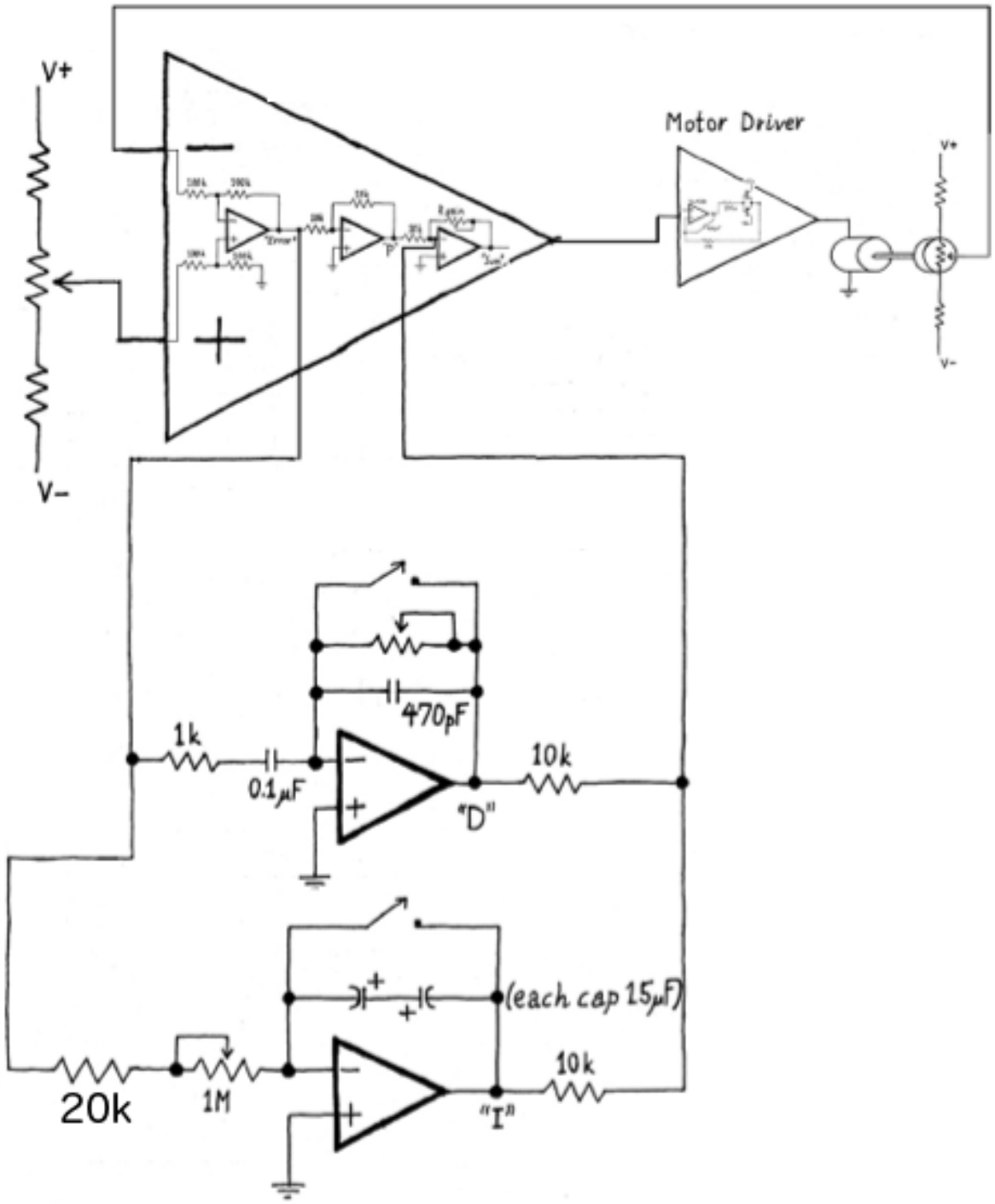
The **push-button switch** across the differentiator's feedback resistor lets us cut **D** in and out, in case you want to try momentarily disabling/enabling the **D** term. You should give it a try to see the effect that **D** has.

(Blank page.)

Part 6 **Start Time:** _____

optional — add integral term (time estimate: 20 minutes)

6.1 Adding the third term, **I**, can drive residual error (a small, persistent difference between the target voltage and the output pot voltage) to zero. Here is a diagram of the *full* PID circuit, with the integrator added.



Two details of the integrator to note:

two polarized caps placed end-to-end: This odd trick works to permit use of *polarized* capacitors in a setting that can put either polarity across the capacitance. The effective capacitance is, of course, only half the value of each capacitor. We use polarized caps only because large-value caps like these 15 μF parts are hard to find in non-polarized form.

seeming absence of DC feedback: At first glance, this integrator seems doomed to drift to saturation, since the integrator includes neither of our usual protections against such drift — feedback resistor or momentary-discharge switch. But neither is necessary here, because overall feedback — all the way around the large loop, from input pot to output pot — makes such unwanted drift impossible. In short, there *is* DC feedback, despite appearances to the contrary.

6.2 Watching the effect of *I*: In today's circuit, the residual error is hard to see on the scope, so adding **I** will not be as rewarding as adding **D** was. Your best hope is if you cut the **P** gain very low: try $R_{\text{gain}} = 20$ kΩ, so that the circuit feedback ought to tolerate a noticeable residual error, when not fed an **I** of the error.

If you patiently turn the input pot to slowly move the target position, you may even be able to make out the effects of the motor pot's "sticktion" (a cute term for "static friction"). The motor and pot do not move smoothly in response to a slowly-changing input. Instead, the motor fails to move until the **I** voltage reaches some minimal level. Then the output voltage jumps to a new level, and waits for another shove.

Jose recommends that to check out the integrator, it is best to have a gain of 2 ($R_{\text{gain}} = 20$ kΩ) and a signal of 0.2 Hz, 5 V_{pp} square wave, because for gain > 10, the long-term error is too small to be able to see easily. He also recommends disabling the **D** term while you first check out the **I** term, because there are no oscillations when the gain is so low. He finds that the "stiction" effect is much easier to see than the "small residual error" (e.g. elevator is "almost" at the right place) effect. Jose used $R = 20$ kΩ for the integrator, in place of Tom Hayes's 1 MΩ potentiometer: this gave a time constant $RC = 0.3$ s. When he tried too small a time constant, e.g. $RC < 0.15$ s, the motor-pot oscillated so violently that the gear-drive loop came off. That's why we have the fixed 20 kΩ resistor in front of the 1 MΩ variable resistance.