

ME 461 Laboratory #5

Characterization and Control of PMDC Motors

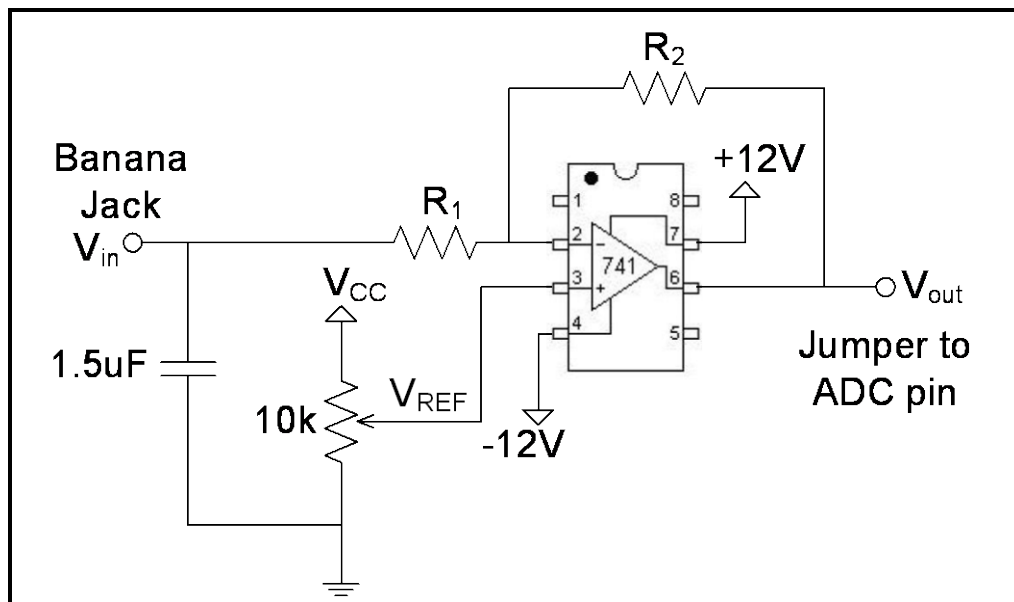
Goals:

1. Build an op-amp circuit and use it to scale and shift an analog voltage.
2. Calibrate a tachometer and use it to determine motor speed.
3. Control a PMDC motor using two methods (by varying voltage and duty cycle).
4. Understand the function of an H-Bridge motor amplifier.
5. Use motor current to estimate angular position in static equilibrium.
6. Perform open-loop control on a PMDC motor and evaluate performance.

NOTE: It is very important that you follow the directions in this lab carefully. You will be working with voltages outside the operating range for your microcontroller, so an incorrect connection could damage the microcontroller. Each exercise has different wiring, so follow each exercise description carefully to setup the correct wiring. If you are unsure of anything, **ask your TA before guessing. Damaging the hardware will obligate you to re-solder an entirely new microcontroller setup.**

Exercise 1:

In this exercise, you are going to build the op-amp gain and shift circuit you designed as part of Prelab 5 using the LM741. The goal is to linearly map a voltage in the range of $[-12V, +12V]$ to a voltage in the range $[0V, 3V]$. Solder the gain and shift circuit on the given op-amp board located in the parts boxes by the soldering irons. Use the schematic below, the values of R_1 and R_2 you found in the homework assignment, and the demo board in lab as a guide.



Op-amp circuit schematic

Your TA will show you where the bipolar power supply is located and give you a cable that easily connects $\pm 12\text{V}$ to your op-amp board. The op-amp board connects -12V to the LM741's pin #4, $+12\text{V}$ to pin #7 and the power supplies common to ground. V_{cc} and GND of the op-amp board should already be soldered to V_{cc} and GND of the microcontroller. DO NOT connect VOUT of the op-amp board to one of the microcontroller's ADC pins until you are told to later in the lab. If the op-amp board's offset is not yet set, the output of the op-amp could be large and damage the ADC of the microcontroller.

There are three vertical potentiometer connections to the left of the motor. Connect the top and bottom terminals to $+12\text{V}$ and -12V . Then, connect the middle terminal (potentiometer output) to the V_{in} banana jack of the op-amp board. Ground is already connected through the three wire $\pm 12\text{V}$ cable. **Show your wiring to your TA.**

In Exercise 2 you will use the potentiometer to provide input voltages between $\pm 12\text{V}$ in order to verify the gain and shift of the op-amp circuit.

Exercise 2:

First, set $V_{in} = 0\text{V}$ by rotating the motor and potentiometer while measuring the potentiometer output with the digital multimeter (DMM) or oscilloscope. Then, without disturbing the V_{in} setting, measure the output voltage of your gain and shift circuit, V_{out} . Now, adjust V_{ref} by adjusting the 20-turn $10\text{K}\Omega$ pot on your op-amp board so that $V_{out} \approx 1.5\text{V}$.

Next, set V_{in} according to the table below by rotating the motor shaft-mount potentiometer. Record V_{out} for each case. Use the DMM to measure the input and output voltages.

V_{in} (V)	V_{out} (V)
-12	
-8	
-4	
0	
4	
8	
12	

Plot the data (V_{out} vs. V_{in}) in Excel and fit a linear trend line to the data. Record the exact gain and offset below.

Gain: _____ Offset: _____

Show your results to your TA. Once you TA ok's your value, disconnect your wiring and solder V_{out} of the op-amp board to ADC channel 1 (A1) of the MSP.

Exercise 3:

Now you are ready to determine the relationship between motor RPM and the tachometer's output voltage. As asked at the end of Exercise 2, you should have wired ADC10 channel 1 (A1) to the op-amp board's output. Write a program to sample A1 every 1 ms and display the reading in mV to the terminal window every 250 ms. As always, leverage previously written code.

To minimize wiring mistakes we are going to start back at the beginning. Disconnect all the wires you connected in exercise 2. Then as before connect the top and bottom terminals of the motor's potentiometer to +12V and -12V. This time connect the output of the motor's potentiometer to only the oscilloscope with the oscilloscope's ground connected to common (either at the $\pm 12V$ COM terminal or the banana jack connected to your MSP's GND). The potentiometer's output will be used to calibrate the tachometer's gain. Also make sure the special power cable for the Op Amp board is connected to the +/- 12v supply and to the Op Amp board. Connect the motor's motor+ (red) and motor- (black) terminals to the HP programmable power supply. Finally, connect the tachometer ground (grey terminal) to common ground (the MSP430 board's ground) and the tachometer output (orange terminal) to the V_{in} banana jack of the op-amp board.

Show your TA your wiring and your code for sampling the connected ADC channel.

By varying the voltage across the motor terminals, it is possible to drive the motor at different speeds. You can reverse the direction of rotation by reversing the polarity of the voltage applied to the motor power terminals (by swapping the + and - leads at the motor). **Ensure that there are no wires laying over any rotating parts.** Set the programmable power supply's current to 2A and the voltage to 0 V to start. Make use of the "Output On/Off" button on the programmable power supply to start and stop your motor. With the motor stationary, the output of the gain and shift circuit should be roughly 1.5V. With positive velocity it should decrease toward 0V and with negative velocity it should increase toward 3V.

Observe the waveform generated by the spinning potentiometer on the oscilloscope. You can measure the rotational speed of the motor by measuring the frequency of the sawtooth waveform on the oscilloscope. Select *Measure* \rightarrow *Time* \rightarrow *Frequency* to display an automatic reading of the motor's speed in Hz. For the motor voltages listed below, record the gain and shift output voltage as well as the motor speed in RPM.

Motor Voltage (V)	V_{out} (mV)	Motor Speed (RPM)
-12		
-10		
-8		
-6		
0		
6		

8		
10		
12		

Plot the data (RPM vs. $V_{out}(mV)$) in Excel or Matlab and fit it with a linear trend-line. Record the slope of the line (in RPM per millivolts) and y intercept below.

$$\text{RPM} = \text{_____} (\text{RPM/mV}) * \text{Millivolts} + \text{_____}$$

The published gain for this tachometer is 3V/1000rpm. Does your gain (the slope above) make sense given that it has gone through your gain and shift circuit? **Check the values with your TA.**

Modify your program to print the speed of the motor in RPM to the terminal window using integers/longs only. **Demonstrate to your TA** for several positive and negative applied motor voltages.

Exercise 4:

First, disconnect all the wires connecting the $\pm 12V$ supply to your board and to the motor's potentiometer. You are done using the $\pm 12V$ supply and the op-amp board so cut the wires connecting op-amp board to your microcontroller board. Solder a banana jack to your board that connects to V_{cc} (you could use the same wire that connect V_{cc} to the op-amp board). You are going to be powering the motor's potentiometer with your board's V_{cc} in this section and following sections. Also make sure a banana jack is soldered to ADC channel 2 (A2) and a black banana jack is connected to ground (from Lab 4). Wire the pins of a 1-by-5 male header to your board in the following fashion: 1 – Digital I/O pin; 2 – Not connected; 3 – Timer B1; 4 – ADC channel 6 (A6); 5 – GND. ADC channel 6 must also be connected to GND through a 2.2-3.0 k Ω resistor.

In the previous exercises, you learned how to calibrate and use the tachometer to measure angular speed of the motor. Now, you are going to calibrate the potentiometer to measure angular position. Connect the output of the potentiometer to ADC channel 2's banana jack, but this time power the potentiometer with the board's V_{CC} and GND. Write a program to display the millivolts read on ADC10 channel A2. Ask your TA to attach a weighted link to the motor's output shaft. Rotate the link to three angles: $+90^\circ$, 0° , and -90° relative to the vertical and record the potentiometer voltage.

Link Angle (degrees)	V_{pot} (mV)
-90	
0	
90	

Plot the data (Link Angle vs. V_{pot}) and determine the best-fit line that relates angle to potentiometer voltage. Fill in the parameters in the equation below.

$$\theta = \text{_____} \times V_{\text{pot}} + \text{_____} \text{ (degrees)}$$

Change your MSP430 program to display the angular position (in degrees) of the link attached to the motor. **Demonstrate to your TA.**

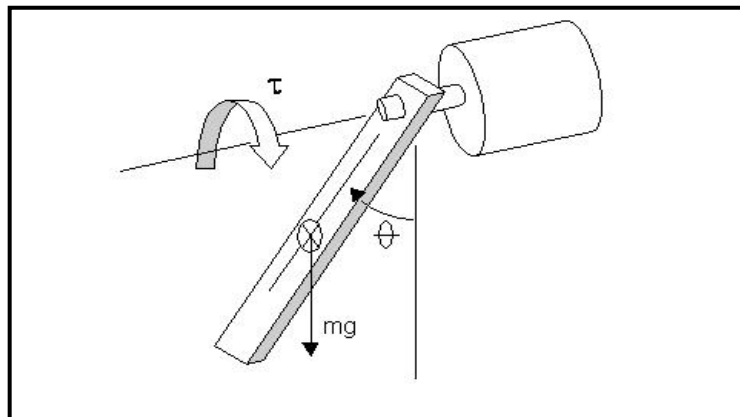
Exercise 5:

Until now, you have driven the DC motor by varying the supply voltage. However, as you learned in lecture, it is possible to exploit the baseband characteristics of PMDC motors to control them. In this exercise, you are going to use the National Semiconductor LMD18200 H-bridge module to control the motor with a PWM signal. Connect the HP programmable power supply to the power supply jacks on the H-bridge but **do not turn it on yet**. Connect the OUT1 and OUT2 leads on the H-bridge to the motor+ and motor- jacks. Connect your 5-pin header to the female jack on the H-bridge (GND to GND). Keep the potentiometer connected to ADC channel 2.

Leverage the code you wrote in Lab 2 to develop a program for outputting PWM signals to the H-bridge. Set the PWM frequency to 20kHz. Use two of the four switches to control the duty cycle in the following manner: both off – 0% duty cycle; both on – 50% duty cycle; switch 1 on – 17% duty cycle; switch 2 on – 33% duty cycle. Use a third switch to control the state of the digital I/O pin wired to the *Direction* input on the H-bridge. You do not need to use pin interrupts for the switches in this case; simply read P2IN in the TimerA0 ISR or elsewhere in your program. Maintain your code from exercise 2 sampling A2 and displaying the angular position in degrees. Load the program onto your microcontroller. **After you verify your wiring with your TA**, turn on the HP power supply and set the current to 4A and the voltage to 12V. Press the switches in different combinations and observe as the link moves to each static equilibrium state. Also observe the angle reading in the terminal window. **Demonstrate to your TA.**

Exercise 6:

In this exercise, you will develop another method for estimating the static position of the linkage. You determined the equation for static equilibrium of the motor plus linkage in Prelab 5 and found that the torque applied to the motor by the linkage arm (gravity) was balanced by the torque applied to the linkage arm by the motor.



Linkage free-body diagram

Recall that motor torque $\tau_m = K_m i_m$. As you know from prelab 5, the LM18200 H-bridge makes measuring motor current easy because the H-bridge outputs a current on the *Sense* pin that is proportional to the motor current. Thus, $i_{sense} = K_{sense} i_m$. If a resistor is connected between the *Sense* pin and GND, the voltage across the resistor can be measured by the ADC on the MSP430F2272 and used to determine the *Sense* current: $V_{sense} = R_{sense} i_{sense}$. You used this knowledge in the homework to show that the *Sense* voltage is linearly proportional to the sine of the linkage angle:

$$V_{sense} = K \sin \theta,$$

where K is a constant that lumps together the relevant physical parameters (i.e., K_{sense} , R_{sense} , mass m , acceleration due to gravity g , length L , and motor constant K_m) responsible for the static response of this system. As you might have guessed, you are going to estimate the lumped parameter in the above equation using data measurements.

Attach the DMM or oscilloscope to A6 to measure the *Sense* voltage. Configure the pin as an input (digital or analog, for now) so that it is high-impedance, but do not worry about sampling the channel yet (you have not yet learned how to sample multiple ADC channels and we still need to sample the potentiometer). Load the program to your MCU. Send several duty cycle commands to your motor using the three switches and record the resulting angles from the terminal window and *Sense* voltages from the DMM or oscilloscope.

Direction	Duty Cycle	Link Angle (degrees)	V_{sense} (V)
Forward	50%		
	33%		
	17%		
	0%		
Reverse	17%		
	33%		
	50%		

Plot $\sin \theta$ vs. V_{sense} and θ vs. V_{sense} on the same plot. Notice that for small angles, θ (in Radians) is a good approximation of $\sin \theta$. Determine the slope of the $\sin \theta$ vs. V_{sense} plot and record it below. Also plot V_{sense} vs. duty cycle and record the slope below. **Check with your TA.**

$$\sin \theta = \text{_____} \times V_{sense}$$

$$V_{sense} = \text{_____} \times \text{duty cycle (V)}$$

The equation you just found for link angle is called a *grey-box model* because it is based on physical laws and parameters, but it does not explicitly identify all of the physical parameters.

Modify your microcontroller program to sample the voltage across the *Sense* resistor (comment out the previous configuration; you will need it again). Use the model you found and the assumption that θ is small to estimate the link angle from the motor current. Print the link angle to the terminal. **Demonstrate to your TA** as you vary the input with the switches.

Exercise 7:

Now, modify your program so that instead of specifying a duty cycle with the switches, you specify an angle in $[-40^\circ, +40^\circ]$. Then, in your program, calculate the duty cycle required to produce that angle by using the equations you found in exercise 6. Send that duty cycle to the H-bridge. Additionally, revert back to sampling the potentiometer reading. Print the angle command from the switches next to the angle measured from the potentiometer reading.

With your hand, simulate the effect of adding extra weight to the linkage by **gently** pressing downward on the suspended link. What happens to the angle reading and the command? Set the angle command to 0° and move the linkage. What happens to the angle reading and command? **Explain this discrepancy to your TA.**

You have just observed the motivation for *closed-loop control* and *sensor feedback*. Open-loop control is sensitive to both disturbances and parameter uncertainty. Also, if you compare your results to those of your lab mates, you will likely recognize that the identification procedure must be performed on an individual basis. This is another drawback of open-loop control; closed-loop control allows for some model variation. You will learn more about closed-loop control in future labs. What do you think some *advantages* of the current-sensing configuration might be? *Hint*: think cost, size, and sensor reliability.