Experiment A9
First Order Transient Response
Procedure

Deliverables: checked lab notebook, tech memo, demonstration of working device to Lab TA
Recommended Reading: Sections 6.2, 14.2, and 18.4; Chapter 21 in textbook

Overview

In this lab, you will examine the heating and cooling of a metal sample. An electric heater will supply a known power, and the temperature vs. time will be measured. You will examine two cases:

- **Response to a Step** – The heater will be abruptly turned on for some amount of time, which will cause the temperature to rise. Then it will be abruptly turned off, which will cause the temperature to decrease.
- **Response to an Oscillating Heater Power** – The heater power will oscillate sinusoidally at a specific frequency, which will cause the temperature to oscillate at the same frequency.

Theoretical Background

Most thermal systems exhibit a 1st order transient response to some heat input. That is, the temperature of the system $T(t)$ is related to the transient heater power $q(t)$ via the 1st order differential equation

$$mc_p \frac{dT}{dt} = Ah\left(T_{Air} - T\right) + q, \tag{1}$$

where h is the convective heat transfer coefficient, m is the mass of the sample, c_p is the specific heat of the sample, A is the surface area exposed to air, T_{Air} is the ambient temperature of the air, and $q = iV$ is the heater power.

This mathematical approach to heat transfer is known as the "lumped thermal mass" model, which assumes that the temperature is approximately the same everywhere in the sample as it heats or cools with time. This assumption is only valid if the Biot number is small, $Bi \ll 1$. The Biot number is a non-dimensional (unitless) parameter given by the formula

$$Bi = \frac{hL}{k}, \tag{2}$$

where h is the convective heat transfer coefficient, L is the thickness of the sample in the direction of heat flow, and k is the thermal conductivity of the sample.

Equation (1) can be rewritten as

$$\tau \frac{dT}{dt} = T_{Air} - T + \frac{q}{Ah}, \tag{3}$$

where

$$\tau = \frac{mc_p}{Ah} \tag{4}$$

is the thermal time constant. The parameters m, c_p, A, and h are difficult determine individually, especially for composite structures with complex geometry. Thus, the time constant τ is typically measured via the response to a step.

Response to a step

Consider a case where the heat source is initially OFF, such that the heater power $q = 0$ and the initial temperature is T_0, then the heater is turned ON such that $q = iV$ and allowed to warm up to an equilibrium temperature T_{max}. For this case, the solution to Eq. (3) for this type of heating is

$$T(t) = T_{\max} + (T_0 - T_{\max})\exp\left(-t/\tau\right), \tag{5}$$

where T_0 is the initial temperature when the heater is turned ON, τ is the thermal time constant, and T_{max} is the maximum temperature that the sample will asymptotically approach.

Now, consider a heated sample abruptly having the heat source switched OFF, causing the sample to cool down from its new initial temperature is T_0 (warmer than the previous case) down to T_{Air}. The solution to Eq. (3) is then

$$T(t) = T_{Air} + (T_0 - T_{Air})\exp\left(-t/\tau\right). \tag{6}$$

The temperature vs. time for the subsequent heating and cooling of a sample, as predicted by Eqs. (5) and (6), is plotted in Fig. 1.

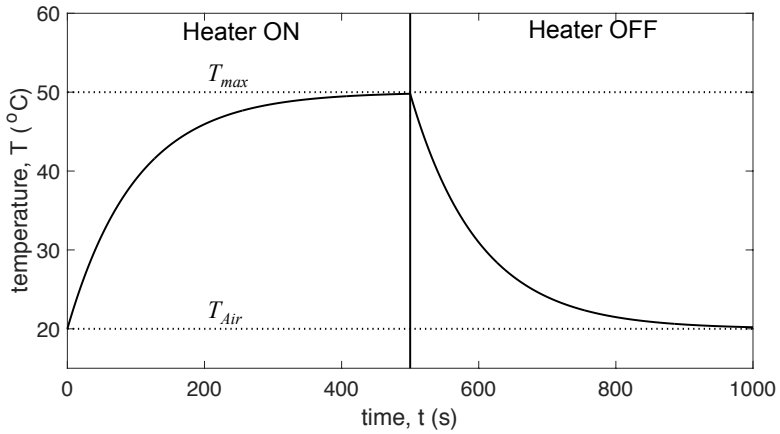

Figure 1 - The theoretical temperature vs. time traces given by Eqs. (3) and (5) are plotted for an unknown sample. The theoretical heater is turned ON at time $t = 0$s and turned OFF at time $t =$ 500s.

Response to Oscillatory Heater Power

In many cases, a system is subject to an oscillating heat load, which causes the temperature to oscillate at the same frequency. This is common for power plants or HVAC systems, where the power is routinely cycled ON and OFF.

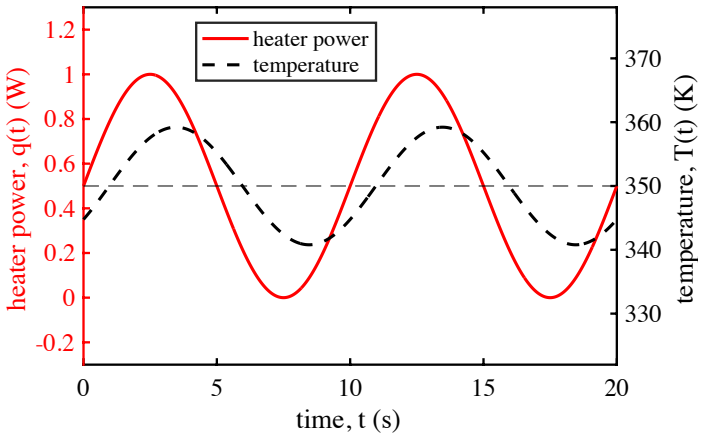

Figure 2 - The heat load (heater power) in a thermal system is cycled ON and OFF in a sinusoidal fashion, causing the temperature to oscillate at the same frequency.

In this lab, we will cycle the power of a small electric heater, in a sinusoidal fashion, between 0 and q_{max}. In this case, Equation (3) becomes

$$\tau \frac{dT}{dt} = T_{Air} - T + \frac{q_{max}}{2Ah}[1 + sin(\omega t)] , \qquad (7)$$

where ω is the "driving frequency" of the heater oscillations. For our daily on earth, there are two driving frequencies affecting temperature—the daily cycle of sunrise and sunset, and the yearly cycle between summer and winter.

The solution to Eq. (7), which is valid for times $t >> \tau$, is

$$T(t) = \frac{1}{2}(T_{max} + T_{Air}) + \frac{1}{2}\frac{(T_{max} - T_{Air})}{\sqrt{1+(\omega\tau)^2}} sin(\omega t + \phi), \qquad (8)$$

where τ is the thermal time constant of the system and T_{max} is the maximum temperature, both of which can be measured via the response to a step. Equation (8) is plotted along with the heat input $q(t)$ in Fig. (2) above. Note that the temperature oscillates at the same frequency as the heater, but is phase shifted by an angle

$$\phi = arctan(-\omega\tau). \qquad (9)$$

The phase angle can be written as a time lag by dividing by angular frequency, yielding,

$$\Delta t = \frac{arctan(-\omega\tau)}{\omega}. \qquad (10)$$

You have likely observed this phase shift in your everyday life. The hottest part of the is in the afternoon, a few hours after the brightest part of the day. Similarly, the hottest days of the year usually occur in late July and early August, several weeks after the brightest day of the year.

Experimental Procedure

You will now build the thermal system shown in Fig. 3. This is a fairly complex system, so we will simplify things by breaking it up into smaller subsystems.

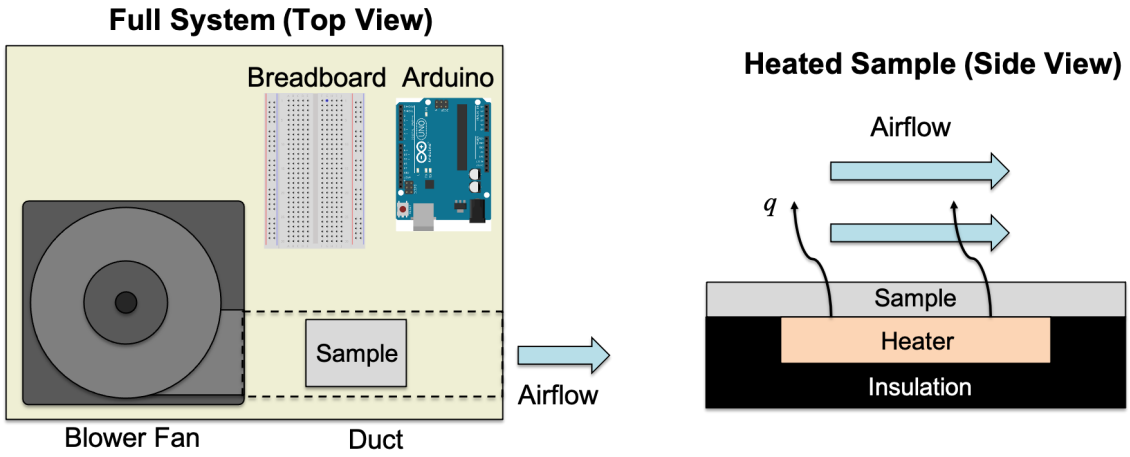

Figure 3 – A heated sample is cooled by airflow from a blower fan. The temperature and heater power are recorded via an Arduino microcontroller.

Part I: Response to a step

Subsystem A: Thermistor

A thermistor will be used to measure the temperature of the sampleThis particular thermistor has a negative temperature coefficient (NTC), which means that its resistance decreases as it gets hotter. The thermistor will be wired up in the voltage divider circuit shown in Fig. 4. The Arduino microcontroller will use its analog-to-digital convertor (A/D) to read the voltage output V_{out} from this circuit.

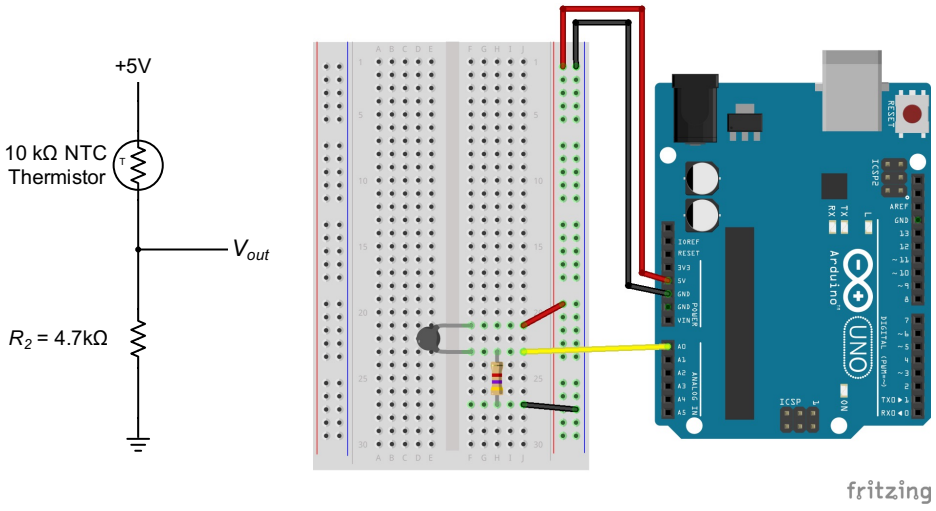

Figure 4 – A thermistor is wired up in a voltage divider circuit. (The actual thermistor will be embedded in the sample.) Its output voltage V_{out} is read by the A0 analog input of the Arduino.

Procedure

1. Sketch the complete thermostat system in Fig. 3 in your lab notebook.
2. Locate the sample. It is an aluminum plate with black and red wires hanging off the side. Measure the dimensions of the rectangular aluminum surface, and record the values in your lab notebook.
3. Sketch the circuit drawing on the left side of Fig. 4 in your lab notebook.
4. Connect the USB cable to the Arduino and lab computer. The green LED on the Arduino should light up.
5. Use red and black jumper wires to connect the +5V and GND pins on the Arduino to the vertical bus lines on the right side of the breadboard, as shown in Fig. 4. Use the orange handheld DMM to verify that it is providing +5V of power.
6. Select 4.7 kΩ resistor from the bin. Measure its resistance, and write the value down in your lab notebook as R_2.
7. The thermistor is already mounted to the sample. Measure the resistance of the thermistor via the black wires hanging from the sample. It should be around 10 kΩ at room temperature. Clasp the sample in the palms of your hands, and the resistance should gradually decrease.
8. Construct the thermistor voltage divider circuit pictured in Fig. 4, and connect the voltage output V_{out} to analog input pin A0 on the Arduino.
9. Download the A9 "Thermistor Template" code template from the lab webpage, read the comments, and modify it.
    a. Right-click the link and "Save as…" with an intelligent file name (i.e. "A9_thermistor_yourName.ino").
    b. Open the file in the Arduino IDE software.
    c. Replace the *** in the variable declaration with the correct analog input pin for the thermistor.
    d. Replace the other *** with the value of you measured for the 4.7 kΩ resistor.
    e. Save the code file.
10. Use the Arduino IDE software to compile the code and send it to the microcontroller.
    a. Go to "Tools" > "Board" and make sure either "Arduino/Genuino" or "Arduino UNO" is selected.
    b. Go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.
    c. Click the check mark at the top of the Arduino program to check the code for errors.
    d. Press the arrow button to compile the program and send it to the Arduino.
    e. Go to "Tools" > "Serial Monitor" (or press "Ctrl + Shift + M") to view the output from the "Serial.print()" commands at the bottom of the screen. (Make sure the Baud rate is set to 9600.)

11. You should see two columns of data separated by a comma printed to the serial monitor.
    a. The first value is the time in seconds.
    b. The second value is the temperature in degrees Kelvin.
12. Clasp the sample between the palms of your hands. You should see the temperature slowly increase.
13. **Demonstrate the working subsystem to the TA.**

Subsystem B: 12V DC Power Supply, Heater, and Fan

You will now connect the 12V DC power supply, heater, and fan

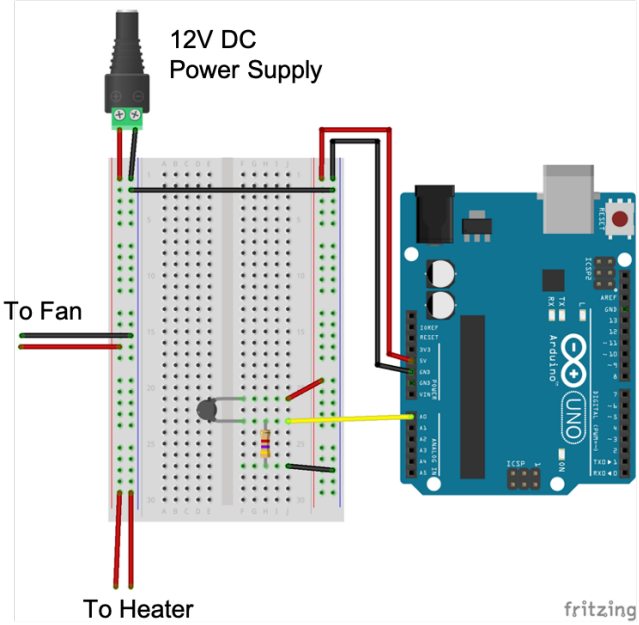

Figure 5 – The vertical bus lines on the left side of the breadboard are powered by the 12V DC power supply. Both sides share a common ground.

Caution: Hot! Be careful handling the sample. It will become very HOT if the heater is left on.

Procedure

1. Locate the 12V DC power supply. (It looks like a laptop charger.)
    a. Connect the 5.5mm barrel connector on the 12V DC power supply to the inline kill switch. Make sure the kill switch is in the OFF position.
    b. Connect the other end of the kill switch to the screw terminal adapter.
    c. Connect red and black male pin jumper wires to the + and – screw terminals.
2. Use the handheld DMM to verify that the power supply works by measuring its voltage output. Plug in the DC power supply. Flip the kill switch ON and OFF.

3. Switch the DC power supply OFF, and make the following connections illustrated in Fig. 5.
    a. Connect the DC power supply to the vertical bus lines on the left side of the breadboard.
    b. Connect the negative bus line on the right side to the "common ground" bus line on the right side, as shown in Fig. 5. That is, both negative vertical bus lines should be connected to GND on the Arduino.
4. Use the handheld DMM to measure the resistance of the heater R_H. Record the value in your lab notebook.
5. Use the appropriate formula to calculate the power that will be dissipated in the heater (in units of Watts) when it is connected to 12V DC.
6. Calculate the surface area of the sample (units of m^2). Then calculate the total heat flux through the sample $q'' = \dot{q}/A$ (units of W/m^2).
7. Use male-male jumper wires to connect the blower fan to the 12V DC bus line, as shown in Fig. 5.
8. The red wires coming out of the sample are the heater wires. Connect the heater wires to the 12V bus lines, as illustrated in Fig. 5.
9. Place the sample in the duct area on the wood board with the exposed aluminum side facing up. Cover it with the duct, such that the wires pass through the notch in the duct.
10. Run the thermistor program on the Arduino, and pull up the serial monitor.
11. Use the inline kill switch to turn on the DC power supply and power the heater. You should see the temperature values (in units of kelvin) begin to increase.
12. Unplug the red heater wires, and the temperature should decrease.

System Identification and Characterization – Response to a step

You will now perform an experiment to determine the parameters T_{Air}, T_{max}, and τ. With these parameters, the behavior of the system can be predicted, and its temperature can be controlled. This procedure is often referred to as *System Identification and Characterization*.

1. Take your lab notebook over to the mercury thermometer located on the cart in the middle of lab. Record the air temperature T_{Air} from the thermometer in your lab notebook.
2. Make sure the heated sample is cooled down to within a degree of T_{Air}.
3. Clear the output on the serial monitor, then connect the red heater wires to the 12V bus line. Turn ON the DC power supply, if it's not already ON. Make sure the fan is ON. You should see the temperature values in the serial monitor begin to rise.
4. Allow the sample to heat up to its stead steady state value T_{max}. This will take at least 180 seconds. Record the final value of T_{max} in your lab notebook.
5. Unplug the read heater wires, and allow it to cool down from T_{max} back to T_{Air}. The fan should still be running. This should take another 180 seconds.

6. After the temperature cools down close to T_{Air}, disconnect the USB cable from the computer and save the data.
    a. Highlight all of the data in the serial monitor up to time $t = 0$. (Make sure you are only selecting data from the most recent drop.)
    b. Press "Ctrl + c" to copy the data.
    c. Paste the data in a blank .txt file using notepad. Make sure you only have one complete data set beginning with. Delete any data from previous tests that may have still been hanging around in the serial monitor.
    d. Save the data as a .txt file in the folder with your Arduino code.
7. Import the data into Matlab or Excel, make a quick plot of temperature vs. time, and **show it to the TA**. **The data should look similar to Fig. 1.** If it does not look like this, you must repeat the measurement.
8. Email the data to yourself and your lab partner, or transfer it to you laptop with a USB flash drive.

Part II: Response to an Oscillating Heat Source
Subsystem C: Power Modulation

The heater will be sinusoidally cycled ON and OFF via a MOSFET switch. Recall that a MOSFET is a transistor that only allows current to flow between the source and drain when a sufficient voltage is applied to the gate. Shown in Fig. 6, the MOSFET controls the flow of current through heater. The average amount of heater current is proportional to the duty cycle of the PWM signal driving the MOSFET gate.

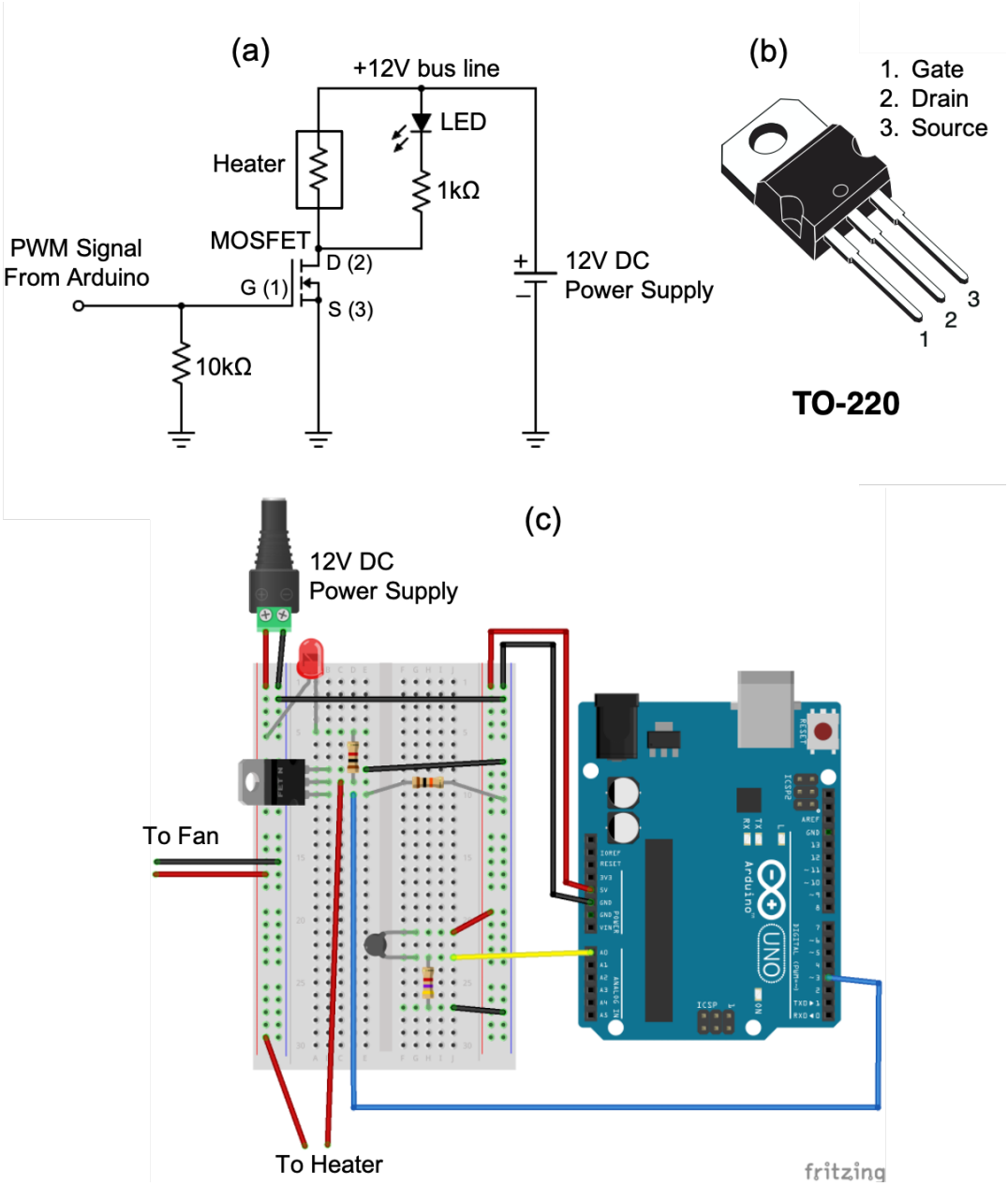

Figure 6 – An N-channel MOSFET is used to modulate the heater power. A PWM signal from the Arduino is used to drive the gate of the MOSFET and modulate the flow of current through the heater.

Procedure

1. Make sure the heater is disconnected from the breadboard and the DC power supply is OFF.
2. Focus your engineering awareness on Fig. 6. Sketch the pinout for the TO-220 N-channel MOSFET shown in Fig. 6b in your lab notebook.
3. Use the breadboard to construct the circuit shown in Fig. 6.
   a. The blue vertical bus lines on the breadboard are common ground. Anything with a ground symbol in the circuit should be connected to the common ground.
   b. The fan should still be connected to the +12V and ground bus lines.
   c. Connect the heater between the +12V bus line and the drain of the MOSFET.
   d. Connect a 10 kΩ "pull-down" resistor from the gate to the common ground.
   e. Connect the source of the MOSFET to ground.
   f. Connect the LED in series with a 1 kΩ from the +12V bus line to the drain of the MOSFET. Recall that LEDs must be wired up in the correct polarity.
4. Run your thermistor code to monitor the temperature, then switch on the 12V DC power supply. The heater should *not* turn ON. If the temperature rises, you did not construct the circuit correctly.
5. Test the MOSFET circuit by manually connect the gate to 5V. Disconnect the gate of the MOSFET from I/O pin 3 and connect it directly to the 5V bus line. This will charge the gate and allow current to flow from the source to drain, thus turning ON the heater. You should see the temperature gradually rise. The red LED should illuminate.
6. Disconnect the gate from 5V. The heater should turn OFF and the temperature should begin to decrease.
7. Reconnect the gate to digital I/O pin 3 on the Arduino.
8. Download the A9 "Power Modulation" code template from the lab webpage, read the comments, and save the code file to the desktop.
9. Use the Arduino IDE software to compile the code and send it to the microcontroller.
   a. Go to "Tools" > "Board" and make sure either "Arduino/Genuino" or "Arduino UNO" is selected.
   b. Go to "Tools" > "Port" and select the COM port that says "(Arduino/Genuino Uno)" next to it.
   c. Click the check mark at the top of the Arduino program to check the code for errors.
   d. Press the arrow button to compile the program and send it to the Arduino.
10. Switch ON the DC power supply. The fan should be running.
11. In the serial monitor, you should see 4 columns of data printed: time (s), PWM duty cycle (0-255), heater power (W), and temperature (K). The LED brightness should vary proportional to the PWM duty cycle, and the temperature values should change by at least a degree or two.

Data Collection – Response to an Oscillating Heat Source

You will now record the heater power and temperature as a function of time. In your tech memo, you will compare the data to Eqs. (7) and (9)

1. Turn OFF the 12V DC power supply.
2. Examine the Arduino code (read the comments) and determine the driving frequency in Hz. Record the value in your lab notebook.
3. Press the reset button on the Arduino UNO, then quickly clear the output on the serial monitor and turn ON the DC power supply.
4. Allow the system to run to steady state, where the temperature oscillates between the same two values. This should take at least 300 seconds (5 minutes).
5. After you have at least 300 seconds of data, disconnect the USB cable from the computer and save the data.
   a. Highlight all of the data in the serial monitor up to time $t = 0$. (Make sure you are only selecting data from the most recent drop.)
   b. Press "Ctrl + c" to copy the data.
   c. Paste the data in a blank .txt file using notepad. Make sure you only have one complete data set beginning with. Delete any data from previous tests that may have still been hanging around in the serial monitor.
   d. Save the data as a .txt file in the folder with your Arduino code.
6. Import the data into Matlab or Excel, make a quick plot of heater power and temperature vs. time, and **show it to the TA**. **The data should look similar to Fig. 2.** If it does not look like this, you must repeat the measurement.
7. Email the data to yourself and your lab partner, or transfer it to you laptop with a USB flash drive.

Clean-up

To receive full credit, you must return the lab bench to its initial state:

- Unplug the DC power supply. Remove the kill switch and screw terminal adapter.
- Disassemble the circuit. Remove the wires and circuit components from the Arduino and breadboard.
- Return all resistors to the appropriate bins.
- Line up the Dupont pin jumper wires in a neat little pile on the lab bench.
- Disconnect the USB cable from the computer.

Data Analysis and Deliverables – Using the format and style outlined in HW1, write a brief tech memo containing the following items.

1. A plot of the temperature (kelvin) vs. time (seconds) for the "response to a step" with both heating and cooling.
2. A single plot of *both* the heater power (watts) and measured temperature (kelvin) vs. time (seconds), similar to Fig. 2, for the "oscillating heat source".
   a. Show 2 or 3 periods of steady-state oscillation toward the end of the data set. (Do not show the initial transient.)
   b. Use the "yyaxis left" command to plot heater power on the left axis.
   c. Use the "yyaxis right" command to plot measured temperature on the right axis.
   d. Make sure it is clear to the grader which time trace corresponds to which y-axis.
   e. Make sure both signals are vertically centered, such that they both oscillate about the middle of the plot. (Similar to how you centered the sine waves on the oscilloscope, which made it easier to determine the phase.)
3. A table containing the following parameters.
   a. The measured air temperature T_{Air}.
   b. The maximum temperature T_{max} measured in the response to a step.
   c. The time constant τ determined from the response to a step data.
   d. The non-dimensional parameter $\omega\tau$ for the oscillating heat source
   e. The measured time lag Δt.
   f. The theoretical time lag Δt calculated using Eq. (10).
   g. The measured peak-to-peak amplitude of temperature oscillation.
   h. The theoretical peak-to-peak amplitude of temperature oscillation.

Talking Points – Please discuss the following in your lab report.

- How does the measured time lag Δt compare the theoretical value you computed?
- How does the measured peak-to-peak amplitude of temperature oscillation compare to the theoretical value you computed?

Appendix A

Equipment

- 12V DC power supply w/ inline kill switch and screw terminal adapter
- Arduino UNO Microcontroller
- 6ft USB cable
- Blower fan mounted to small plywood sheet – Amazon Part # B08P1S5DBN
- Breadboard mounted to small plywood sheet
- 2” × 2.5” × 1/16” aluminum sample w/ polyimide film heater and thermistor
    - Polyimide film heater – Amazon Part # B09X16XCVS
    - 10k NTC Thermistor, adhesive mount, Amphenol – Digikey Part # 235-1457-ND
- N-Channel MOSFET TO-220AB (Digikey Part # 497-2765-5-ND)
- Red LED
- Male-Male DuPont pin jumper wires
- 3M Velcro command strips
- 6” ruler
- 1/8” flathead screwdriver
- Mercury or Alcohol thermometer (on cart in center of room)