Newton's law of cooling: Procedure

ME 240: Fundamentals of Instrumentation & Measurement • D. H. Kelley • I. Mohammad

Introduction

Thermocouples and thermal cameras are often used to measure temperatures. Thermocouples are inexpensive and produce a voltage proportional to the temperature at one location. Thermal cameras produce thermal images by sensing infrared radiation. In this exercise, you will perform two tasks, one using thermocouples and one using an infrared camera.

Your first task is to predict and test which of the three objects shown in Fig. 1a will cool fastest. All have the same mass and are composed of copper, but their shapes differ. You will measure the temporal variation of the temperature of each object with a thermocouple, then fit an exponential curve to each set of data. Solving Newton's law of cooling (also known as the heat equation) leads to the prediction that objects cool according to

$$T(t) = (T_i - T_a)e^{-t/\tau} + T_a, \tag{1}$$

where $T(t)$ is the instantaneous temperature of the object, t is time, T_i is the initial temperature of the object, T_a is the ambient temperature, and τ is the characteristic cooling time which represents the amount of time it takes for an object's temperature to decay to 37% of its initial temperature. Your goal is to determine τ for each object.

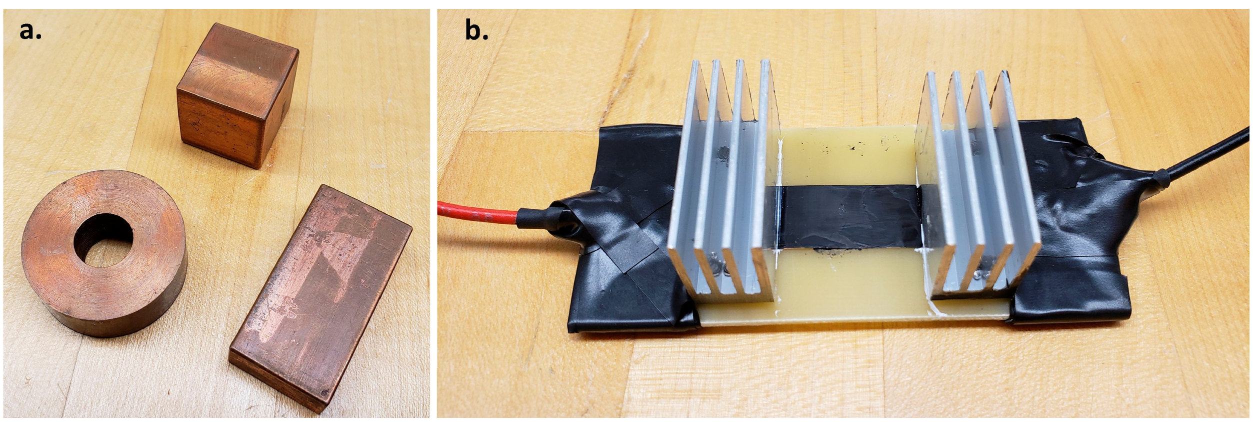

Figure 1: **a,** Three copper objects whose cooling rates you will predict and measure. **b,** A metal ribbon to be heated electrically and kept cool at its ends by heat sinks.

Your second task is to determine the composition of a metal ribbon from its temperature profile. The ribbon will be heated throughout by an electrical current and will be connected to heat sinks that keep its ends cool, as shown in Fig. 1b. You will measure the spatial variation of the temperature of the ribbon with an infrared camera, then fit a polynomial curve to the temperature profile. For such an arrangement, Newton's law of cooling predicts the ribbon temperature to vary spatially according to

$$T(x) = \alpha\left(\left(\frac{L}{2}\right)^2 - x^2\right) + T_a, \tag{2}$$

where $T(x)$ is the local temperature, L is the exposed length of ribbon between the heat sinks, x is the spatial coordinate along the length of the ribbon(with $x = 0$ at its center), T_a is the ambient temperature, and

$$\alpha = \frac{I^2}{2W^2H^2}\frac{\rho_e}{\rho_m C\kappa}, \tag{3}$$

where I is the electrical current, W is the width of the ribbon, H is the thickness of the ribbon, ρ_e is the electrical resistivity, ρ_m is the mass density, C is the specific heat capacity, and κ is the thermal diffusivity.

Learning goals

- Measure temperatures, varying over time and space, using thermocouples and an infrared camera.
- Predict and test which of the provided three objects will cool fastest.
- Determine the composition of a metal ribbon from a measured temperature profile.
- Fit experimental data to known analytic forms using Matlab.
- Gain familiarity with image processing techniques.

Materials

- **Cooling rate measurement:** calipers, copper objects, thermocouple acquisition equipment with cable, K-type thermocouple, 3 K-type thermocouple cables, cooling block, furnace
- **Ribbon temperature measurement:** thermal camera, calipers, ribbon assembly connected to power supply

Safety

Closed-toed shoes and eye protection are required. Avoid burns by handling hot objects only when necessary and safely (e.g., with tongs and/or heat resistant gloves).

Cooling rate measurements

This task involves the three copper objects shown in Fig. 1a: a cube, a plate, and a tube. Use calipers to measure the dimensions of a room-temperature set of objects (for the plate, length is longest, then width, then height), recording your measurements in the Deliverables document. All three objects have mass 100 g. How long do you predict the cooling time τ to be (seconds?, minutes?, hours?) and why?. Which object do you predict will cool fastest? Why? Record your predictions and explanation in the Deliverables document.

Next, set up instruments for recording temperatures from thermocouples. Connect the thermocouple acquisition equipment to a lab computer and to power (if necessary). Three

K-type thermocouple cables[1] are already connected to the first three pairs of terminals, with yellow wires going to the positive terminals and red to the negative terminals, as shown in Fig. 2. Connect a K-type thermocouple to the first cable.

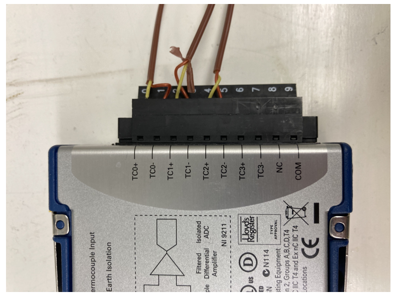

Figure 2: K-type thermocouple wires connected to a National Instruments thermocouple module for data acquisition.

Now set up to acquire thermocouple measurements in Matlab. Start with these commands:

```
dl = daqlist; % list available data acquisition devices
dl{1,'DeviceInfo'} % see info about the device
```

Note the DeviceID, which is probably something like `cDAQ1Mod1`; you will need it later. Continue with these commands:

```
dq = daq('ni'); % create National Instruments DataAcquisition object
dq.Rate = 1; % set sampling rate (Hz)
ch0 = addinput(dq,dl.DeviceID,'ai0','Thermocouple'); % add channel
ch0.ThermocoupleType = 'K';
dq.Channels % see info about the channels
```

If something goes wrong, you can use `daqreset` to start over. You can always use `doc` to learn more about a Matlab function (e.g., `doc daqlist`). Acquire and plot a few temperature measurements by using these commands:

```
[d,t0] = read(dq,seconds(10)); % read one thermocouple for 10 s
figure; plot(d.Time,d.DeviceID_ai0); % here use your DeviceID + _ai0
```

Label the axes (including units!), add a title, save the figure as an image file, then include it in the Deliverables document. (It's good practice to save all figures as `.fig` files as well, in case you need them later. You might also want to save variables in a `.mat` file, using either the `save` command or `File → Save`.) Calculate the mean ambient temperature using the command

```
Ta = mean(d.DeviceID_ai0) % here use your DeviceID + _ai0
```

[1]Common thermocouple types include K, J, and T. Each type involves a different pair of joined metals and has a different operating range.

and record it in the Deliverables document. Are your measurements reasonable? If not, stop and troubleshoot before continuing.

Disconnect the thermocouple and set up two more acquisition channels:

```
ch1 = addinput(dq,dl.DeviceID,'ai1','Thermocouple'); % add channel
ch2 = addinput(dq,dl.DeviceID,'ai2','Thermocouple'); % add channel
ch1.ThermocoupleType = 'K';
ch2.ThermocoupleType = 'K';
```

Test your setup by reading and plotting data from all three channels:

```
[d,t0] = read(dq,seconds(10));
figure; plot(d.Time,d.DeviceID_ai0); % here use your DeviceID + _ai0
hold on; plot(d.Time,d.DeviceID_ai1); % here use your DeviceID + _ai1
plot(d.Time,d.DeviceID_ai2); % here use your DeviceID + _ai2
```

Without thermocouples connected, the temperature values will be meaningless, but make sure you see three curves, each with a duration of 10 s.

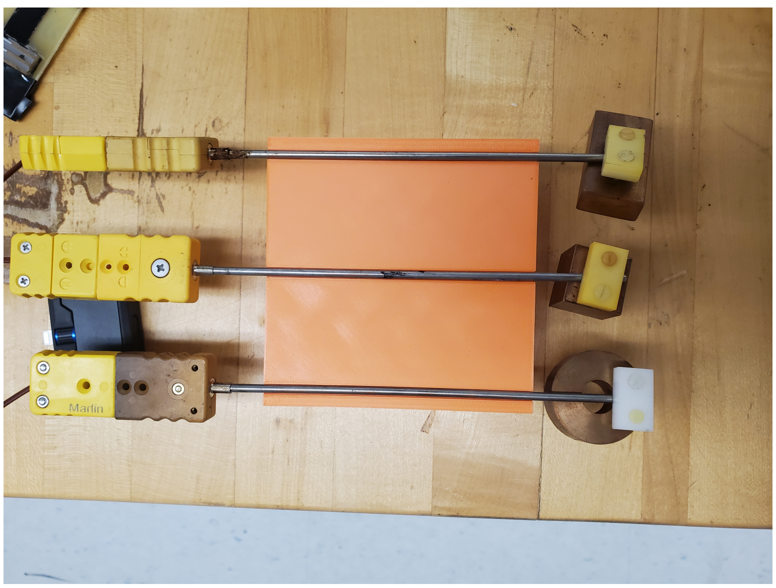

Figure 3: Three copper objects with the thermocouples connected, placed on plastic platform.

Using tongs and/or gloves, carefully move one cube, one plate, and one tube from the oven to the cooling block. Each has an attached thermocouple which you should connect, quickly but carefully, to one of the thermocouple cables. A plastic platform is provided so you can easily place the objects with thermocouples connected as shown in Figure 3. Be sure to note which object is connected to which channel (`ai0` or `ai1` or `ai2`). Once all three are connected, use commands like the ones above to record their temperatures for 10 s and make plots, to ensure that the connections are good and the system is collecting data. Then, record for 12 minutes. While the objects are cooling and the computer is logging data, proceed with ribbon temperature measurements (below).

Ribbon temperature measurements

Start this task by verifying that both ends of the metal ribbon are connected to the power supply and that current is flowing. Then, connect a thermal camera to your phone, using either an iOS or Android version as appropriate. Install the FLIR ONE app when prompted, or else use one of these links (iOS, Android) or one of the QR codes in Fig. 4. Using the app, determine the temperature at a point near the center of each ribbon and the temperatures at the ends of each ribbon. Write your measurements in the Deliverables document. Then, capture a thermal image of the entire metal ribbon (without heat sinks) between the two heat sinks, orienting the camera so that the ribbon is aligned with the horizontal direction of the image and appears rectangular in the image. Record the number indicated on the back of the ribbon setup on the Deliverables document. You may find it interesting to view other objects (or people) with the thermal camera as well! Transfer your thermal image of the ribbon (which is saved in `.JPG` format) to the lab computer, being sure that your phone does not resize or otherwise modify the original image file, then return the thermal camera to its case. Include the thermal image in the Deliverables document.

FLIR ONE iOS | FLIR ONE Android | resistivity | density | specific heat | diffusivity

Figure 4: QR codes for the FLIR ONE app and for material properties.

Turn off the current, then use calipers to measure the length (largest dimension) and the width of the ribbon. The thickness of the ribbon is 0.001 inches. Read the displays of the power supplies to determine the current running through each ribbon. Record your measurements in the Deliverables document.

Cooling rate analysis

Your mission is to determine the cooling rate τ, which appears in Eq. 1, for each of the three objects. Matlab's library of built-in fit functions does not include Eq. 1, but it does include $y = ae^{bx}$, which you can use to fit your data, with y corresponding to $T(t) - T_a$, a corresponding to $T_i - T_a$, x corresponding to t, and b corresponding to $-\tau^{-1}$. Starting with your measurements from the cube, perform a linear least-squares fit using a command like

```
t = seconds(d.Time); % extracting time vector
Tcube = ....; % add code here to grab data from your thermocouple data
structure
cubeFit = fit(t,Tcube-Ta,'exp1') % rename variables as appropriate
```

This command forces the inputs to be column vectors, as required. The coefficient values are listed when the fit is calculated and can also be accessed using

```
cubeFit.a, cubeFit.b
```

Write your estimate of the cube's characteristic cooling time τ and its 95% confidence bounds in the Deliverables document. Plot your measurements of the temperature of the cube, and on the same axes, plot the curve that fits those measurements (remembering that y corresponds to $T(t) - T_a$). A fit curve can be plotted with a command like

```
plot(t,cubeFit(t)+Ta)
```

Add plots of your measurements and fit curves for the plate and tube to the same axes. Add a legend that identifies all plotted curves, then include your plot in the Deliverables document. Write your estimates of the characteristic cooling times (with confidence bounds) for the plate and tube in the Deliverables document. How closely were you able to predict τ for each object?

Ribbon temperature analysis

For this part of the exercise, you'll use Matlab plugins from the manufacturer of the infrared camera, which are installed on computers in the lab and can be downloaded from the course website if you want to work at home[2]. Read the temperatures from the thermal image of the ribbon using commands like

```
obj = FlirMovieReader('your_file.JPG'); % e.g., 20220728T154710.JPG
obj.unit = 'temperatureFactory'; % Celsius
[T, info] = read(obj);
```

The variable `T` contains the thermal image, a two-dimensional array whose elements are temperatures in °C. Plot it:

```
figure; imagesc(T); colormap hot
hcb = colorbar; hcb.Label.String = 'temperature (^\circ C)';
xlabel('x (pixels)'); ylabel('y (pixels)');
```

Next, determine the pixel size. To do so, use the Data Tips tool to click on the two ends of the ribbon, recording their coordinates in pixels (`X`) in the Deliverables document. Using the pixel coordinates and the length of the ribbon, calculate the pixel size and record it in the Deliverables document. Also save it as a variable, for use later:

```
pixelSize = 0.000; % (mm) replace with your value
```

Now, use the Data Tips tool to determine the coordinates of the corners of a rectangle that spans as much of the ribbon as possible but includes nothing beyond the ribbon. Note that for images, the origin is placed in the top left corner, and the y coordinate increases *downward*. Take that into consideration when you pick your top and bottom corners. To check your numbers, draw the rectangle using commands like

[2] Put the files in your Matlab home folder (typically `C:\Users\your_username\Documents\MATLAB` or similar) so the Matlab app knows where to find them.

```
hold on
hr = rectangle('position', [left bottom right-left top-bottom], 'linewidth',
2, 'edgecolor', 'b');
```

In place of `[left bottom right-left top-bottom]`, enter coordinates, all as integers measured in pixels. If your rectangle looks wrong, delete it using `delete(hr)` and try again. Save the figure as an image and include it in the Deliverables document.

The next step is to estimate $T(x)$ (see Eq. 2) from your thermal image of the ribbon. To do so, use only the temperature measurements inside the rectangle, and average in the vertical direction, so that only horizontal variation remains and noise is reduced. Also, create a vector of corresponding positions. You can use commands like these:

```
Trect = T(bottom:top,left:right); % thermal image within rectangle
Tx = mean(Trect,1); % average over vertical direction
x = (left:right)*pixelSize; % corresponding positions (mm)
x = x-mean(x); % move origin to center of ribbon
```

On a new figure, plot the spatial temperature variation and label the axes.

Next, determine the value of α. Matlab's library of built-in fit functions does not include Eq. 2, but it does include $y = p_1x^2 + p_2x + p_3$, which you can use to fit your data, with y corresponding to $T(x)$, p_1 corresponding to $-\alpha$, and p_3 corresponding to $\alpha L^2/4 + T_a$. We expect $p_2 = 0$. Perform a linear least-squares fit using a command like

```
ribbon1fit = fit(x(:),Tx(:),'poly2') % rename variables as appropriate
```

Plot your fit curve on the same axes and add a legend that identifies the two curves. Save the figure as an image and include it in the Deliverables document. Determine the value of α from the fit coefficients and record it in the Deliverables document. Using α and your values for the current, width, and height of the ribbon, calculate the quantity $\rho_e\rho_m^{-1}C^{-1}\kappa^{-1}$ by using Eq. 3. Record your value in the Deliverables document. Finally, use tabulated values of material properties to calculate the quantity $\rho_e\rho_m^{-1}C^{-1}\kappa^{-1}$ for four common metals: copper, stainless steel, carbon steel, and aluminum. Wikipedia offers lists of electrical resistivity, mass density, specific heat capacity, and thermal diffusivity (see Fig. 4); you may use other sources as well. Record all four values in the Deliverables document. Of which of the four metals is the ribbon composed? Record your answer in the Deliverables document.