Curve fits of thermistor data in MATLAB EAS 199B
Review of screencast

Introduction

This screencast provides a tutorial on curve fitting of thermistor calibration data. The measurements are assumed to be stored in a plain text file.

The first step is to investigate the raw data by plotting histograms of the analog output readings made during the calibration experiments. Next, the mean of the readings at each temperature are computed. A polynomial curve fit is performed for the thermistor temperature as a function of the mean of the voltage divider readings at each calibration temperature.

The quality of the curve fit is examined by plotting the residuals

The calibration data set

Figure 1 is a sketch of the calibration experiment. Data was obtained by recording the raw analog values from a voltage divider with a thermistor probe in one leg. Readings were made at several temperatures of a water bath in an insulated coffee mug. The output of the voltage divider was read and averaged with an Arduino.

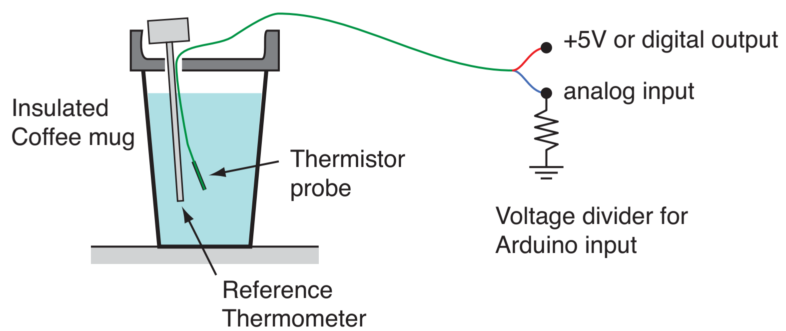

Figure 1: Schematic of the calibration experiment.

For each temperature, many (approximately 100) averaged readings were recorded in a spreadsheet. The data set of averaged readings were exported to a plain text file having the structure depicted in Figure 2.

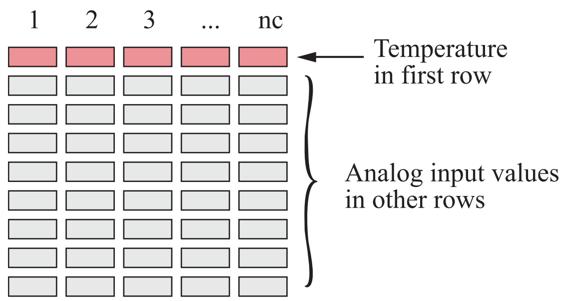

Figure 2: Layout of data in the tab-delimited file exported from Excel.

1. Reading the data into MATLAB variables.

Assume the data is stored in a tab-delimited file called `thermistor_data.txt`. Read the data into a MATLAB matrix with the load command

```
D = load('thermistor_data.txt');
```

Copy the first row into a vector called `T` and the remaining rows into a matrix called `V`.

```
T = D(1,:);
V = D(2:end,:);
```

Compute the average of each column in V

```
Vave = mean(V):
```

Plot both the calibration data.

```
plot(T,Vave,'o')
xlabel('Temperature of water bath (C)')
ylabel('Raw analog readings of V (10-bit values)')
```

2. Plot histograms and compute standard deviation of the data set.

Use histograms to check the distribution of the raw data. For data in the first column, i.e at the first temperature

```
hist(V(:,1));
```

The colon wildcard (:) gives values in all rows. Therefore, `V(:,1)` is the first column of the `V` matrix. Likewise, the histogram of data in the second column is obtained with

```
hist(V(:,2));
```

For the sample data, the histograms reveal negligible variability in voltage at each temperature. The numerical variability at each temperature can be quantified with the standard deviation. The `std` command computes the standard deviation of each column in `V`

```
Vstd = stdev(V)
```

A plot of `Vstd` versus `T` shows negligible variation for the given data set.

3. Curve fit of the V = f(T) data

Use the built-in polyfit command to obtain the coefficients of polynomial curve fit of the average V as a function of temperature. Here we obtain the fit of a quadratic polynomial

```
c = polyfit(Vave,T,2);
```

Remember that Vave is a vector of averages for each of the columns in the data set. Each Vave corresponds to one calibration temperature in the T vector. The values in c are the coefficients of the polynomial in order of decreasing powers of T. For example

$$V_{\mathrm{ave}} = c_1 T^2 + c_2 T + c_3$$

The following command prints the coefficients in scientific notation

```
fprintf('%12.7e\n',c);
```

The curve fit is evaluated over the range of T data in the original data set.

```
Tfit = linspace(min(T),max(T));  %  100 values in range of T data
Vfit = polyval(c,Tfit);          %  Curve fit eval at each Tfit
```

The original data and curve fit are plotted on the same axes with the following commands

```
plot(T,Vave,'o',Tfit,Vfit,'r--')
xlabel('Temperature of water bath (C)')
ylabel('Analog readings (10-bit scale)')
legend('Data','Polynomial fit','Location','Northwest');
```

4. Evaluate quality of the fit from the residuals

For a curve fit of the form

$$y \approx f(x)$$

where (x_i, y_i) are the given data, and f(x) is the fitting function, the residuals are

$$r = y - f(x)$$

In general, the residuals will not be zero because the fit function does not pass exactly through the data. In other words $y_i \neq f(x_i)$.

Given the previously defined curve fit coefficients, the residuals are computed with

```
rfit = Vave - polyval(c,T);
```

A plot of rfit as a function of T reveals any trend in the residual. The curve fit is deemed to be good when the residuals are small and appear to be random over the range of T.