

Hardware Assignment - Report

AI24BTECH11031 - Shivram S

1 Introduction

This report details the process of determining the voltage across a PT-100 RTD (Resistance Temperature Detector) as a function of temperature. The least squares method was used to estimate the parameters of the Callendar-Van Dusen equation.

2 Collecting Data

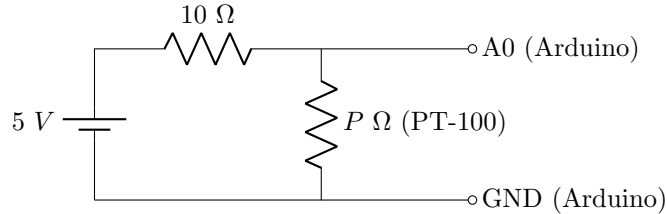
We were provided with the following components

- PT-100 Resistance Temperature Detector
- Arduino Uno and USB Cable
- $10\ \Omega$ resistor
- Breadboard
- Wires

We also made use of the following items from the EE Lab to control and monitor the temperature

- Electric kettle
- Digital Thermometer

The $10\ \Omega$ resistor and the PT-100 were connected in series between the 5V output pin and the ground pin of the Arduino to create a voltage divider. The other pin of the PT-100 is connected to the A0 pin of the Arduino to measure the voltage across the PT-100.



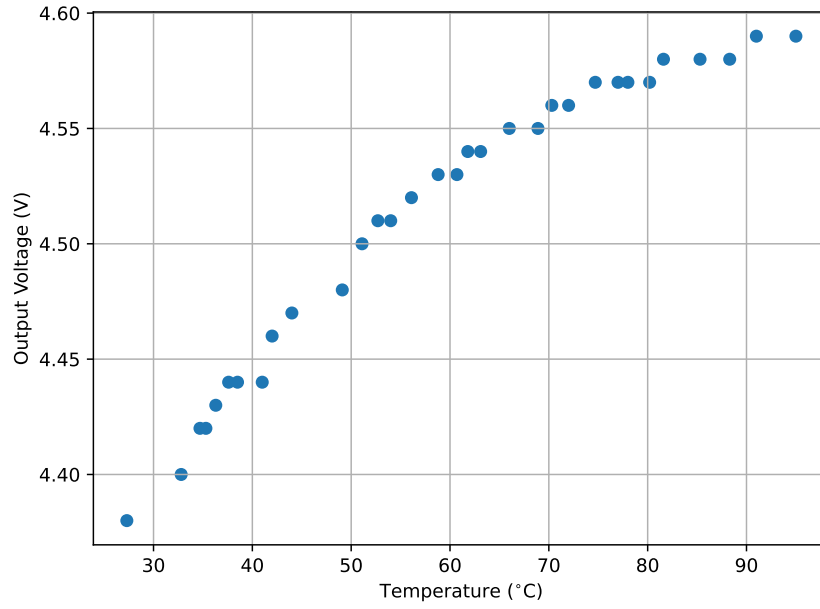
The PT-100 was immersed in an electric kettle filled with water, and a digital thermometer was used to measure the temperature. The C++ code used to drive the Arduino can be found in the appendix.

The kettle was turned on for some time, and then turned off. Once the reading of the digital thermometer became stable, a reading was taken, and the Temperature was increased again. A total of 32 readings were taken over a range of temperatures from $27.3\ ^\circ\text{C}$ to $95\ ^\circ\text{C}$.

7 points were randomly chosen from the data set to form the validation set. The remaining 25 points were put in the testing set.

3 Model

We use the Callendar-Van Dusen equation to model the voltage across the PT-100 as a quadratic function of temperature.



$$V(T) = V(0)(1 + AT + BT^2)$$

$$c = \mathbf{n}^\top \mathbf{x}$$

where

$$c = V(T), \mathbf{n} = V(0) \begin{pmatrix} 1 \\ A \\ B \end{pmatrix}, \mathbf{x} = \begin{pmatrix} 1 \\ T \\ T^2 \end{pmatrix}$$

We can write the equation in matrix form as

$$\mathbf{X}^\top \mathbf{n} = \mathbf{C}$$

where

$$\mathbf{X} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ T_1 & T_2 & \dots & T_n \\ T_1^2 & T_2^2 & \dots & T_n^2 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} V(T_1) \\ V(T_2) \\ \vdots \\ V(T_n) \end{pmatrix}$$

In the least squares method, we try to minimize the squared loss function

$$L(\mathbf{n}) = \|\mathbf{X}^\top \mathbf{n} - \mathbf{y}\|^2$$

where \mathbf{y} is the vector of voltage readings. This can be done by finding the point at which the gradient of the loss function is zero.

$$\frac{dL(\mathbf{n})}{d\mathbf{n}} = \mathbf{0}$$

The solution which minimizes the loss function is given by

$$\mathbf{n} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

The solution may also be calculated using singular value decomposition. This method reduces rounding errors caused by smaller numbers, and also handles under-determined systems better.

In singular value decomposition, we write \mathbf{X} in terms of three matrices - two orthogonal matrices \mathbf{U} and \mathbf{V} , and a diagonal matrix $\mathbf{\Sigma}$.

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

We can then write \mathbf{n} as

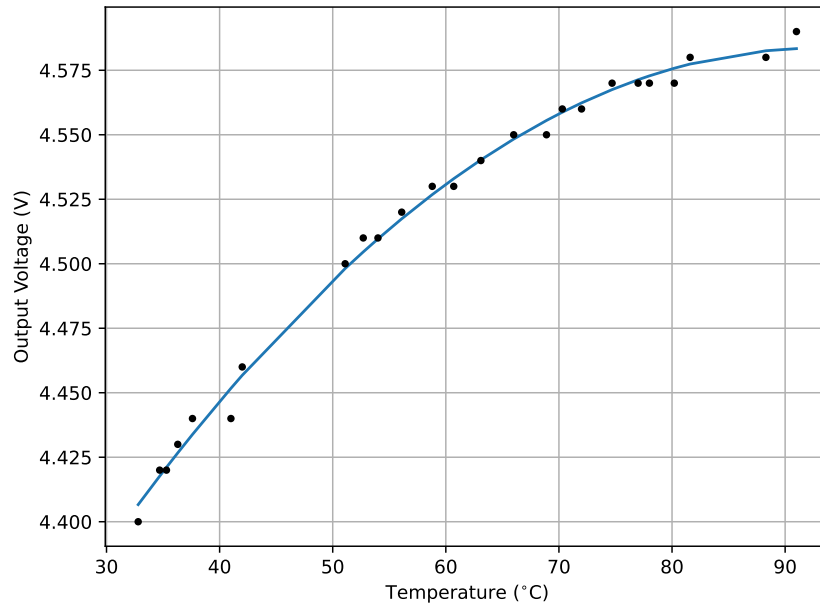
$$\mathbf{n} = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^\top\mathbf{y}$$

The Python program in the appendix computes the value of \mathbf{n} using the least squares method and also plots the voltage as a function of temperature.

$$\mathbf{n} = \begin{pmatrix} 4.1600 \\ 9.132 \times 10^{-3} \\ -4.9232 \times 10^{-5} \end{pmatrix}$$

The approximate model is given by

$$V(T) = 4.1600 + (9.132 \times 10^{-3})T + (-4.9232 \times 10^{-5})T^2$$



The model can be validated by using the validation data set.

