PT-100 Hardware Assignment

Dhawal-ee24btech11015

Contents

Abstract—**This document contains a lab report on the modeling of the voltage-temperature characteristics of the PT-100 RTD (Resistance Temperature Detector) using least squares method.**

1 Training Data

The training data gathered by the PT-100 to train the Arduino is shown in Table 1.

Temperature (°C)	Voltage (V)
24.4	4.85
31.0	4.59
37.8	4.60
42.5	4.61
51	4.62
62.8	4.63
72.5	4.64
89.6	4.65

TABLE 1: Training data.

The C++ source `codes/data.cpp` was used along with *platformio* to drive the Arduino.

2 Model

For the PT-100, we use the Callendar-Van Dusen equation

$$V(T) = V(0)\left(1 + AT + BT^2\right) \tag{1}$$

$$\implies c = \mathbf{n}^\top \mathbf{x} \tag{2}$$

where

$$c = V(T),\ \mathbf{n} = V(0)\begin{pmatrix} 1 \\ A \\ B \end{pmatrix},\ \mathbf{x} = \begin{pmatrix} 1 \\ T \\ T^2 \end{pmatrix} \tag{3}$$

For multiple points, (2) becomes

$$\mathbf{X}^\top \mathbf{n} = \mathbf{C} \tag{4}$$

where

$$\mathbf{X} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ T_1 & T_2 & \dots & T_n \\ T_1^2 & T_2^2 & \dots & T_n^2 \end{pmatrix} \tag{5}$$

$$\mathbf{C} = \begin{pmatrix} V(T_1) \\ V(T_2) \\ \vdots \\ V(T_n) \end{pmatrix} \tag{6}$$

and $\mathbf{n}$ is the unknown.

3 Solution

We approximate $\mathbf{n}$ by using the least sqaures method. Using the pseudo-inverse method, the solution to (4) is

$$\mathbf{n} = \left(\mathbf{X}\mathbf{X}^\top\right)^{-1} \mathbf{X}\mathbf{C} \tag{7}$$

The Python code `codes/lsq.py` solves for $\mathbf{n}$.

The calculated value of $\mathbf{n}$ is

$$\mathbf{n} = \begin{pmatrix} 1.6547 \\ 3.199 \times 10^{-3} \\ -3.9599 \times 10^{-6} \end{pmatrix} \tag{8}$$

The approximation is shown in Fig. 1.

4 Validation

The validation dataset is shown in Table 2. The results of the validation are shown in Fig. 2.

5 Conclusion

This lab experiment demonstrates how machine learning methods can be used to model the behaviour of an unknown device, and find the right

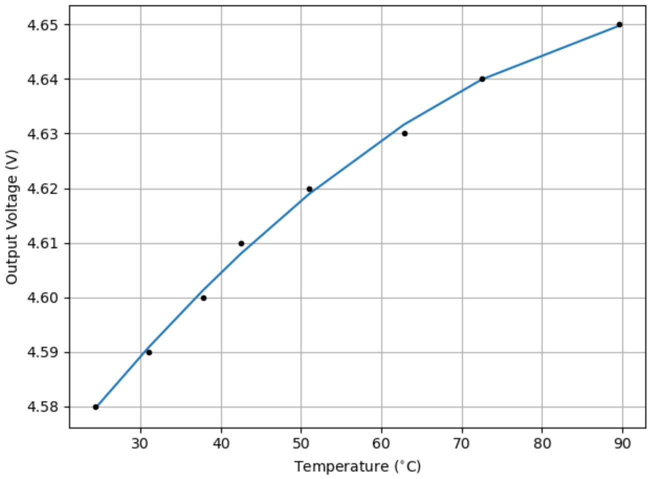

Fig. 1: Training the model.

Temperature (°C)	**Voltage (V)**
32.8	4.59
47.9	4.62
83.9	4.64
97.5	4.66

TABLE 2: Validation data.

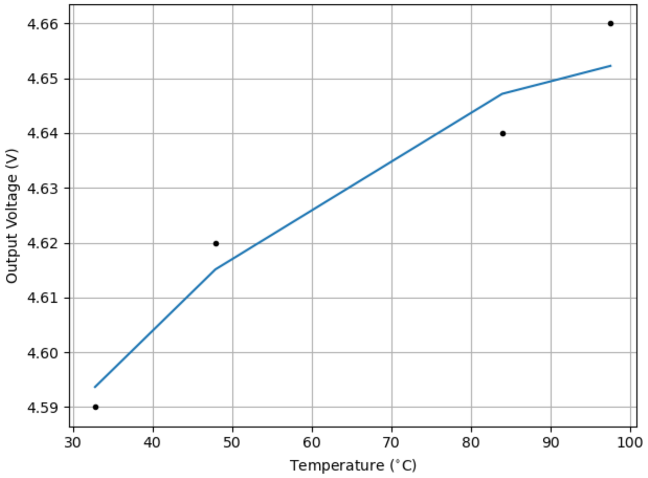

Fig. 2: Validating the model.

parameters that fit the model. It also shows how to use Python libraries and frameworks to collect data and perform optimization.