KHWARIZMI SCIENCE SOCIETY

LAHORE SCIENCE FOUNDRY

PHYSLAB

Modeling the response of a thermistor

Wajahat Sami, Muhammad Sabieh Anwar

31 Dec, 2025

Experimental Design

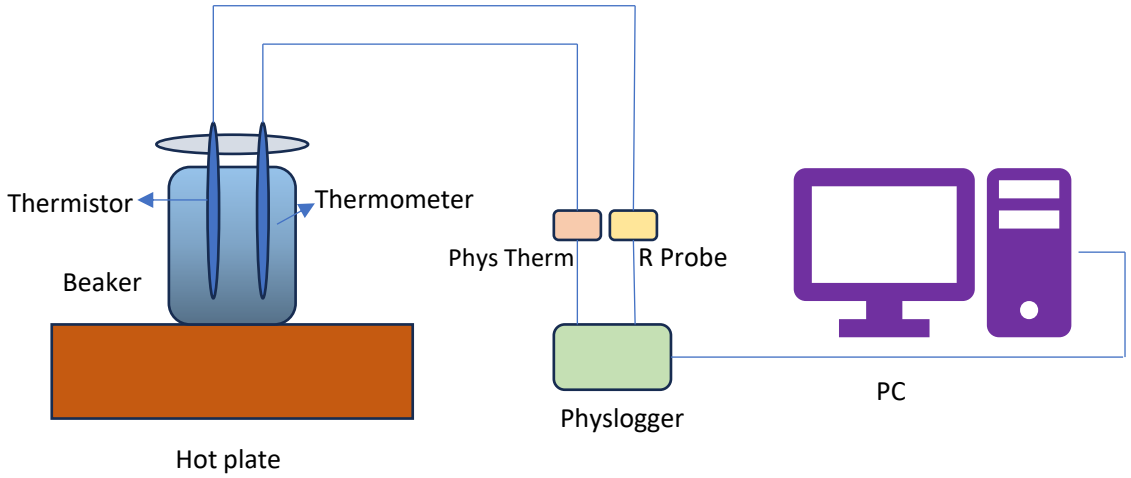

Figure 1: Schematic representation of the experimental arrangement used for thermistor calibration..

The experimental setup consist of a thermistor and thermometer placed in a beaker on a hot plate. The two sensors, an R probe (thermistor) and a Physthermm (temperature sensor), connected to Physlogger system. The processed data acquired from Physlogger is further analyzed in MATLAB to find coefficients of the Steinhart-Hart model equation.

Fitting data to Steinhart-Hart Equation (I)

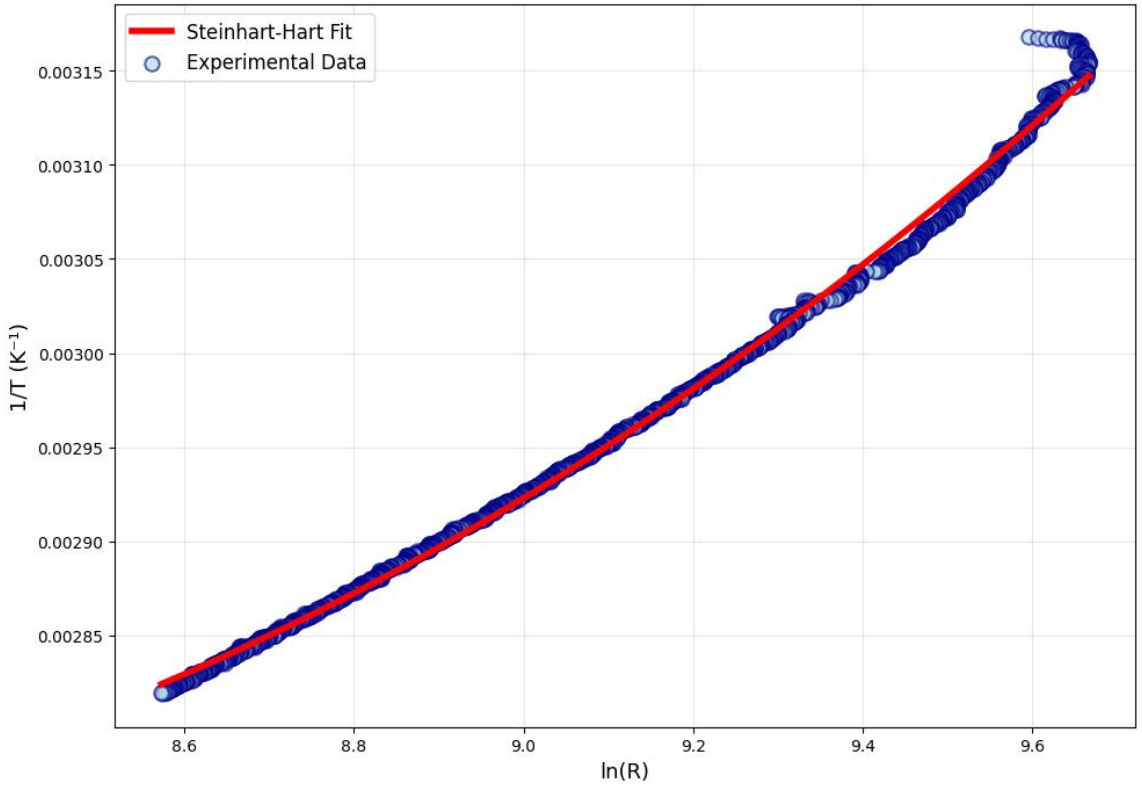

Figure 4: Transformed thermistor data plotted as 1/T vs. ln(R) for Steinhart-Hart equation fitting.

A (K^{-1})	5.90×10^{-3}
B (K^{-1})	5.90×10^{-4}
C (K^{-1})	53.54×10^{-6}

Table 1: The Steinhart-Hart coefficients.

$$\frac{1}{\mathrm{T}} = A + B\ ln(R) + C\ \left(ln(R)\right)^3$$

Fitting data to Steinhart-Hart Equation (II)

Python Code

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Load experimental data
# ======================================================================
data = pd.read_csv('Time-Resistance_Temperature 1.csv')
temperature = data.iloc[:, 3].values  # Temperature in °C
resistance = data.iloc[:, 2].values   # Resistance in kΩ

# Transform data for Steinhart-Hart fitting
# ======================================================================
T_kelvin = temperature + 273.15
X = np.log(resistance)  # ln(R)
Y = 1 / T_kelvin        # 1/T
# ======================================================================
# Fit: Y = A + B*X + C*X^3
# ======================================================================
X_matrix = np.column_stack([np.ones_like(X), X, X**3])
coefficients = np.linalg.lstsq(X_matrix, Y, rcond=None)[0]
A = coefficients[0]
B = coefficients[1]
C = coefficients[2]
# Calculate fitted curve
Y_fit = A + B*X + C*X**3
# ======================================================================
# Plot
# ======================================================================
plt.figure(figsize=(10, 7))

# Fitted curve (thick red line)
X_smooth = np.linspace(np.min(X), np.max(X), 200)
Y_smooth = A + B*X_smooth + C*X_smooth**3
plt.plot(X_smooth, Y_smooth, color='red', linewidth=4, label='Steinhart-Hart Fit', zorder=3
# Experimental data points (light color, semi-transparent)
plt.scatter(X, Y, s=80, color='lightblue', label='Experimental Data',
        zorder=2, edgecolors='darkblue', linewidth=1.5, alpha=0.6)
plt.xlabel('ln(R)', fontsize=13, fontweight='bold')
plt.ylabel('1/T (K⁻¹)', fontsize=13, fontweight='bold')
plt.legend(fontsize=12)
plt.grid(True, alpha=0.3)
# Add text box with coefficients
textstr = f'Steinhart-Hart Coefficients:\n'
textstr += f'A = {A:.6e} K⁻¹\n'
textstr += f'B = {B:.6e} K⁻¹\n'
textstr += f'C = {C:.6e} K⁻¹\n\n'
textstr += f'R² = {R_squared:.6f}'

plt.tight_layout()
plt.show()
```

Predicting R from the coefficients *A*, *B* and *C* determined by nonlinear fitting

Python Code

```
# --- read data ---
data = pd.read_csv("Time-Resistance_Temperature 1.csv")
T_C = data["Temperature_ºC"].to_numpy(dtype=float)
R_exp = data["Resistance_Ω"].to_numpy(dtype=float)
T_K = T_C + 273.15

# --- Steinhart-Hart coefficients ---
A = 5.64e-3
B = -5.89e-4
C = 3.54e-6

def R_from_T(T, lnR_guess):
    """Return (R, lnR) by numerically inverting Steinhart-Hart."""
    def f(lnR):
        return A + B*lnR + C*(lnR**3) - 1.0/T

    lnR_sol = fsolve(f, lnR_guess, xtol=1e-12, maxfev=200)[0]
    return np.exp(lnR_sol), lnR_sol

# --- compute fitted R for each experimental T ---
R_fit = np.empty_like(T_K)

# IMPORTANT: initialize the guess (use first experimental R if available)
lnR_prev = np.log(R_exp[0]) if np.isfinite(R_exp[0]) and R_exp[0] > 0 else np.log(1e4)

for i, T in enumerate(T_K):
    R_fit[i], lnR_prev = R_from_T(T, lnR_prev)

# --- plot experimental vs fitted ---
plt.figure()
plt.scatter(T_C,R_exp,color="red",marker="o", s=2,label="Experimental data")
plt.plot( T_C,R_fit, color="blue",linewidth=2.2,label="Steinhart-Hart fit")

plt.xlabel("Temperature (°C)")
plt.ylabel("Resistance (Ω)")

plt.grid(True)
plt.legend()
plt.show()
```

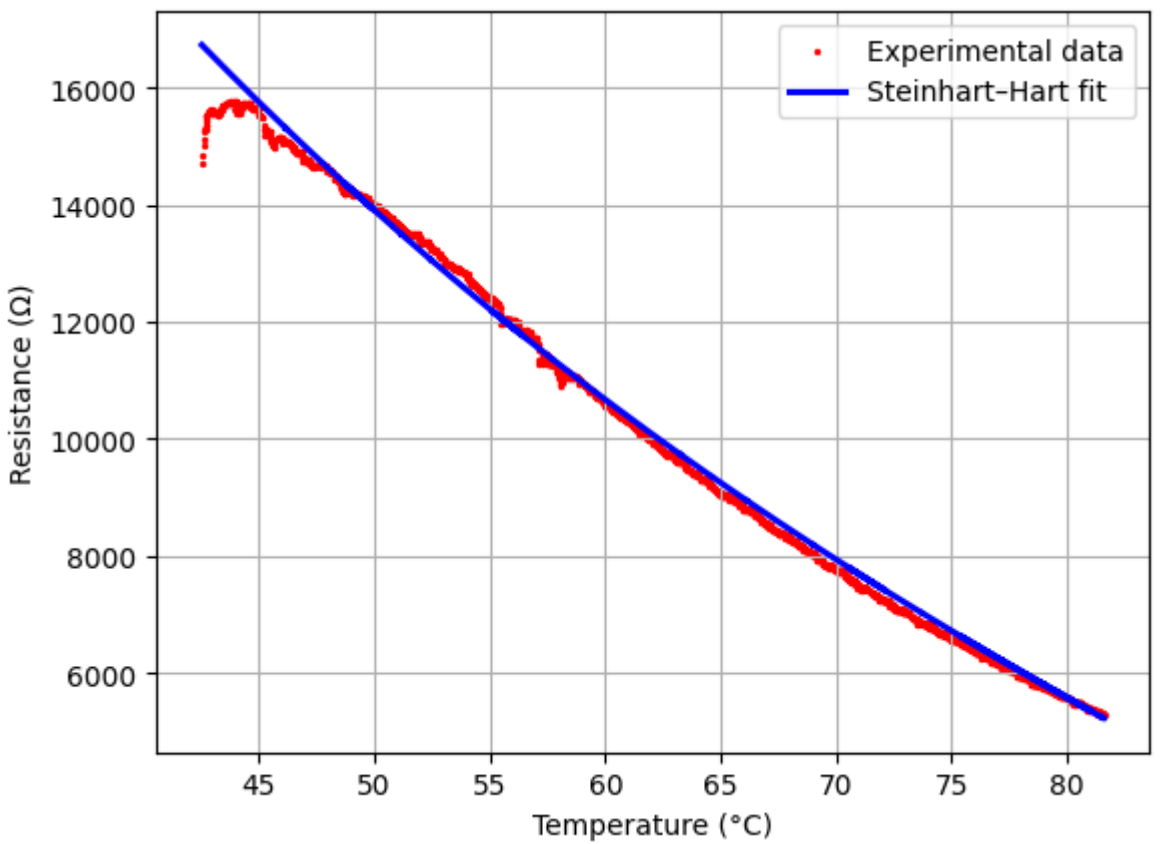

Figure 3: Comparison of experimental data with calculated values from the Steinhart-Hart equation using fitted coefficients A, B, and C.