Experiment 3: Temperature measurements

City College of New York

Professor: Ioana Voiculescu

ME 31100

Fundamentals of Mechatronics

Group: 4

Serigne Mbaye (23391419)
Yuehua Li (23754131)
Catherine Lu (24187472)
Sarah Liu (24311220)

Due Date: 20 March 2024

Table of Contents:

Abstract

This lab focuses on the use of the three different first-order temperature sensor instruments (thermocouples, resistance temperature detectors (RTD), and thermistors) to generate data showing heating and cooling curves. The way this is done is through using a medium kept at a constant temperature lower than room temperature. The sensors transmit data that is recorded through a MATLAB program to generate multiple samples for analysis. The data is then used to plot charts of the curves and find the speed differences of the provided sensors.

Introduction

Through the conduction of various lab works, the idea of sample rate has been established. It can be seen that in the previous lab which worked with free fall, a high sampling rate is necessary in order to compile the data needed to find the damping coefficient of the receptive samples. In this lab, sensors are also used to detect the process of heating and cooling the sensors from and to room temperature. The data is run through MATLAB and saved into final thats are converted into data charts for interpretation and dissection. The temperature sensor used in this experiment is a first-order instrument and the three types being used here include: thermocouples, resistance temperature detectors (RTD), and thermistors.

Objective

This lab looks at the various steps of thermocouple, RTD and thermistor sensors to observe the change in temperature from an ice bath to room temperature. The time constant of the different sensors is found by applying curve fitting onto experimental data to effectively determine the speed differences of the sensors.

Nomenclature

q = the rate of heat transfer to the sensor by convection
h = the convective heat-transfer coefficient
A = the surface area of the sensor through which heat passes,
$\theta_m(t)$ = the temperature of the surrounding medium at time t
$\theta T(t)$ = the temperature of the sensor at time t
m = the mass of the sensor
c = the specific heat capacity of the sensor
T_m = The temperature of the fluid medium
T_0 = The initial Temperature of the Sensor
τ = The time constant of the first-order measurement

Theory

The following step function equation is used to describe the feedback received from the temperature sensors:

$$\frac{\theta(t)-\theta_m}{\theta_0-\theta_m}=e^{-t/\tau} \qquad (1)$$

θ_0 is the initial temperature which in this case is the room temperature, θ_m is the temperature of the provided medium that is kept constant, $\theta(t)$ is the function of the sensor temperature at a certain time, t and τ is the time constant of the first-order measurement.

In order to find the settling times for the cooling and heating portions of the sensors, the time constants are multiplied by a value of 3.912, corresponding to when the 98% of the time frame that the data is collected. Note that the settling time would not occur at the end time (100%) as the temperature would settle prior to the end.

$$t_{sc} = 3.912 * \tau_{cooling} \quad (2)$$

$$t_{sh} = 3.912 * \tau_{heating} \quad (3)$$

Experimental Apparatus

1. Omega Engineering exposed type J thermocouple (**Channel 0**),
2. Omega Engineering $\frac{1}{8}$"-dia. inconel sheathed type J ungrounded thermocouple (JTIN-18U-12, **Channel 1**),
3. Omega Engineering $\frac{1}{4}$"-dia. inconel sheathed type J ungrounded thermocouple (JTIN-14U-12, **Channel 2**),
4. Omega Engineering $\frac{1}{8}$ "-dia. inconel sheathed type K ungrounded thermocouple (KTIN-18U-12, **Channel 3**),
5. Omega Engineering $\frac{1}{8}$ "-dia. inconel sheathed type J grounded thermocouple (JTIN-18G-12, **Channel 4**),
6. Omega Engineering $\frac{1}{8}$ "-dia. thermistor (TJ36-44033-1/8-12, **Channel 5**),
7. Omega Engineering $\frac{1}{8}$ "-dia. platinum RTD probe (R_0 = 100Ω at 0℃, α_1 = 0.00385Ω/℃,) (PR-11-2-100-1/8-12-E, **Channel 6**),
8. Omega Engineering Model 450-AJT handheld type J thermocouple thermometer,
9. Cooled ice bath
10. PC with National Instruments data acquisition board with LabVIEW and MATLAB® software,
11. National Instruments 6034E data acquisition board,
12. National Instruments SC2345 signal conditioning connector block, with associated signal conditioner blocks.

Procedure

1. Prepare an ice bath and assemble the thermocouples. Leave one thermometer out to read and record the room temperature.
2. Open MatLab7 (Figure A1) in order to start data acquisitions of the sensors.
3. Load the code for cooling data.
4. Run the code and insert all the thermocouples into the ice bath (Figure 1). Leave the sensors in the ice bath for 12 minutes.
5. Once the timer is up, save the data and load the heating data code.
6. Run the code and remove all the thermocouples from the ice bath. Lay the sensors apart from one another in the configuration shown in (Figure 2).
7. Leave all the thermometers out and await another 12 minutes.
8. Save the data.

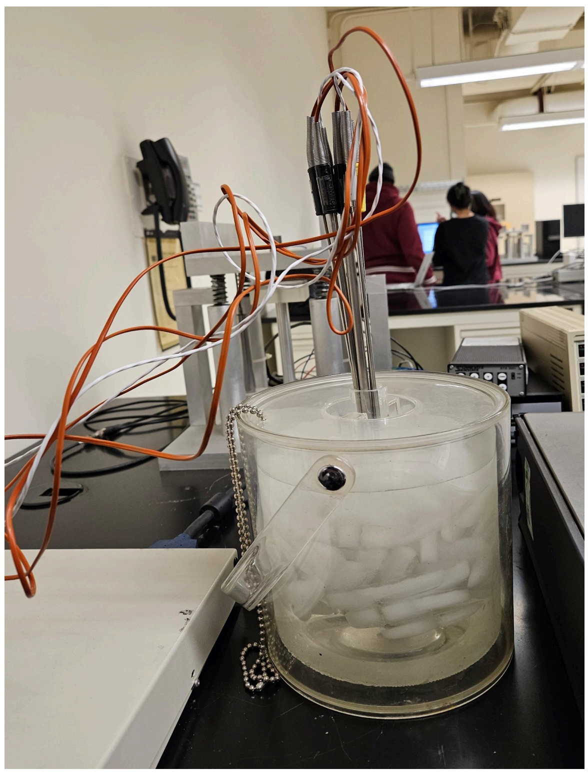

Figure 1: Setup of sensor 2-7 during cooling cases.

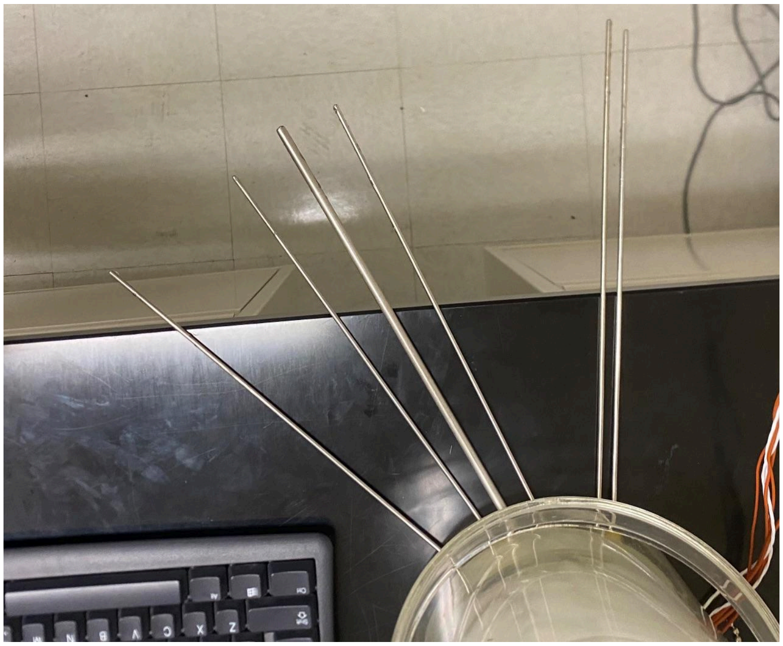

Figure 2: Setup of sensor 2-7 during heating cases.

MATLAB Processing

1. Starting off with the cooling data, import data from the experiment into the MATLAB code.
2. Define your minimum and maximum temperature through the ‘min’ and ‘max’ functions.
3. The $T_0$of the cooling case will be the max temperature and T_m is your minimum. Prior to plotting, make sure to fit the data in the range of when the temperature measure first starts. That means cutting off the time in which the program began running but was not placed in the temperature medium.
4. To find the time constant, input equation (1) to the code and change the range of time value until the fitted curve matches overlaps with the data.
5. Repeat steps 1-4 with the 7 different sensors, notting that for sensor 1, the time constant is neglected as it was not placed in the temperature medium
6. Plot all fitted curves on the same graph by using the ‘hold’ function. Make sure to not plot experimental data
7. Repeat the steps 1-7 with the heating data, making note that The T_0 of the heating case will be the minimum temperature and $T_m$is your maximum temperature.

Results

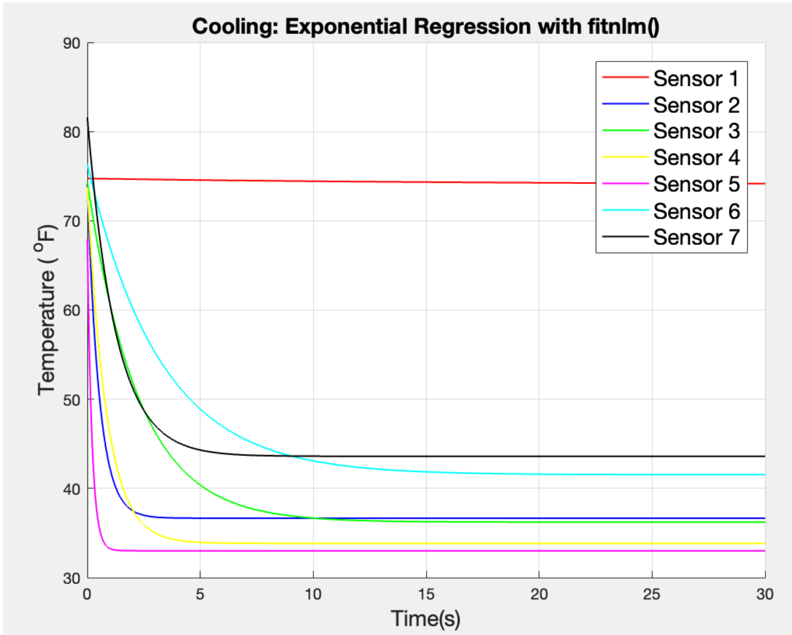

Figure C1: Plotted Temperature vs Time of the medium under cooling for the 7 sensors

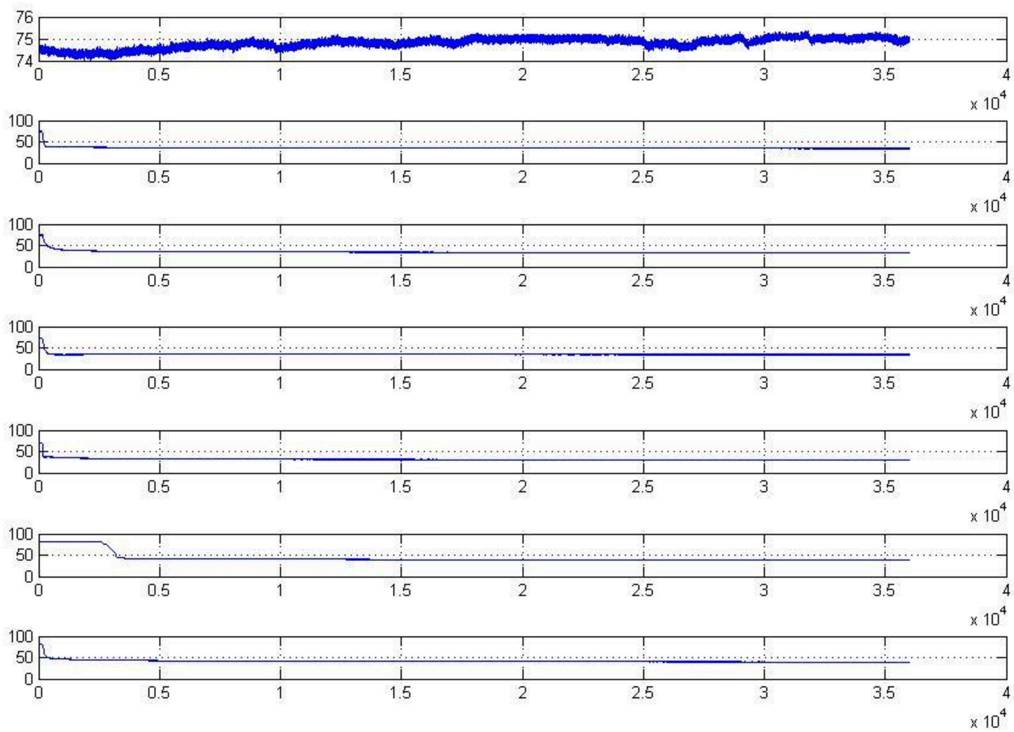

Figure C2: Step response of the 7 sensors during cooling

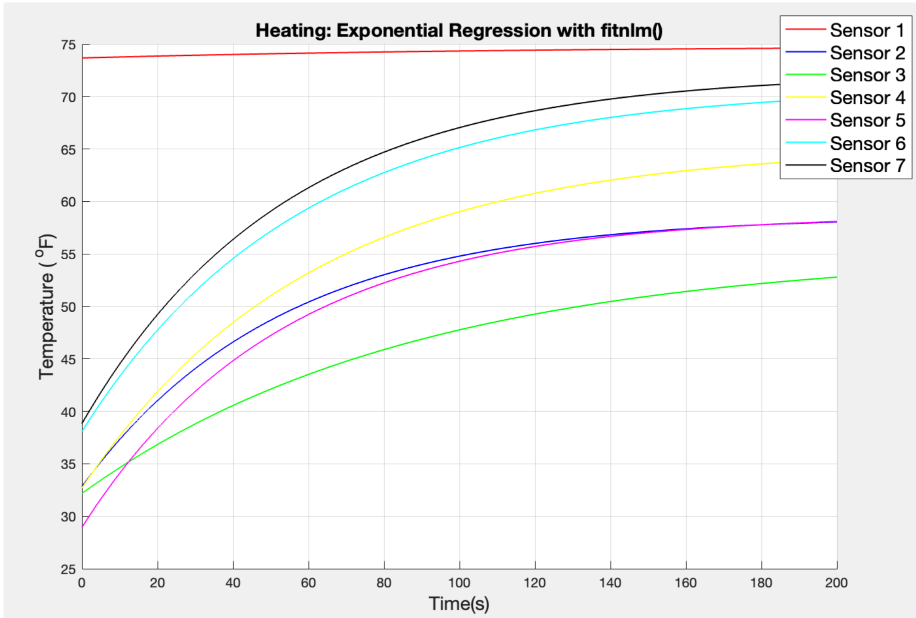

Figure H1: Plotted Temperature vs Time of the medium under heating for the 7 sensors

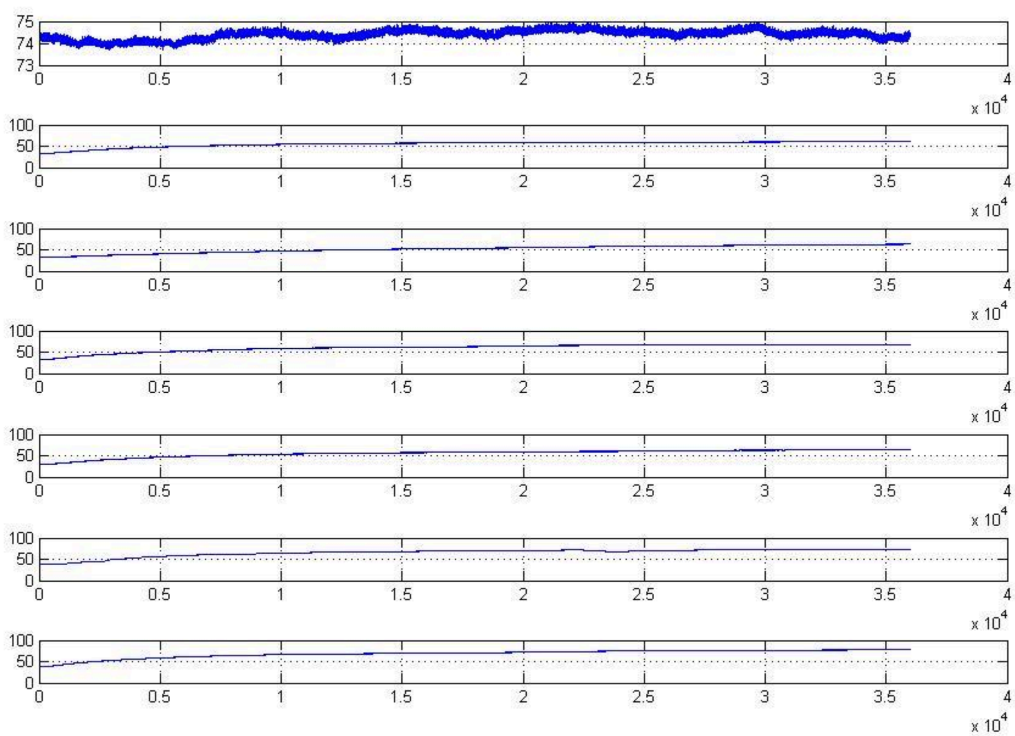

Figure H2 : Step response of the 7 sensors during heating

Sensors	$\tau_{cooling}$	$\tau_{heating}$	$t_{cooling}$	$t_{heating}$
Sensor 2	0.527	52.39	2.0616	204.950
Sensor 3	2.256	88.33	8.825	345.547
Sensor 4	0.849	60.372	3.321	236.175
Sensor 5	0.203	52.63	0.794	205.889
Sensor 6	3.217	57.27	12.585	224.040
Sensor 7	1.253	53.69	4.902	210.035

Figure 3: Table of time constant and settling time of sensors 2 - 7. Note the units for all these values are in seconds.

Discussion of results

The experiment revealed a notable disparity between the time constants observed during heating($\tau_{heating}$) and cooling($\tau_{cooling}$) phases for all sensors. Specifically, the time constant for heating was consistently larger in magnitude compared to that for cooling across equivalent sensors. This observation suggests that the sensors exhibited notably slower reactions to sudden changes in ambient temperature during the heating phase as opposed to the cooling phase.

The experiment also highlighted significant differences in the 98% settling times between the heating ($t_{cooling}$) and cooling ($t_{heating}$) cases for all sensors. Notably, the 98% settling times during heating were notably larger than those during cooling, consistent with the variances observed in the sensors' time constants between the two scenarios. This underscores the fact that the duration for the sensors to approach within 2% of their steady-state values during heating was considerably longer compared to the time required during cooling. Such value obtained within the experiments utilizes equations (2), (3) to obtain the table of values of settling time for both the cooling and heating cases in (Figure 3).

The best fitted curve for the experimental regression is plotted all in Figure C1. Some notable remarks of the graphs: Sensor 1 was not placed into the ice bath and is set to measure the room temperature. The graph plots a 30 second time for cooling. The graph is noted as an exponential function. Such values fitted for the individual sensors are seen from Figure AC1-6.

The best fitted curve for the experimental regression in heating is plotted in Figure H1. In the same remarks as in the cooling case, sensor one was placed in the room and therefore does not experience any heating. . Such values fitted for the individual sensors are seen from Figure AH 1-6.

The sensor exhibiting the largest time constant $\tau_{cooling}$ at 3.217 s, along with the longest 98% settling time, was identified as the ⅛ inch diameter thermistor, designated as sensor 6 during cooling. This outcome is likely attributable to the sensor's larger mass, which is proportionally related to its time constant when compared to sensor 1 with a similar composition but a smaller diameter (⅛ inch). The increased mass had a more pronounced impact on the time constant than the sensor's surface area, which is inversely proportional to the time constant.

In the case of heating the largest time constant $\tau_{heating}$at 88.33 s. was observed in the 3rd sensor, a ¼ diameter inconel sheathed type J ungrounded thermocouple. This suggests that the specific heat of the sensor material significantly influences its time constant.

The value of the settling time are calculated with equations (2), (3). Such values depend solely on the constant 3.912 and the time constant. It is noted previously that the time settling is the time at 98% of the total data collection time as the value will settle prior to the end of the experiment As seens in Figure C1 and H2. The different $t_{cooling}$ and $t_{heating}$ are found in Figure 3.

In this experiment, where rapid temperature changes of the sensor were monitored, a single-ended sensor was preferred. The higher sampling rate offered by a single-ended thermocouple enhanced the resolution of the temperature measurements and facilitated more accurate temperature change modeling compared to a differential sensor. However, this came at the expense of noise cancellation typically provided by differential alternatives. For applications involving gradual temperature changes, a differential sensor would be preferred for its noise cancellation capabilities.

A Thermocouples will not require a differential input as it only has one low wire. There is no need for a differential which would be needed there is a possibility that the voltage will be induced on both the wires. Built in Engineering function eliminates the need to convert data received into the proper time and temperature units needed directly, eliminating the need for separate conversions.

Conclusion

In conclusion, this experiment provided valuable insights into the performance of first-order temperature sensors—thermocouples, RTDs, and thermistors. Key findings included variations in sensor response times between heating and cooling phases, with sensors exhibiting slower reactions during heating. Additionally, specific sensors, like the ¼ inch diameter inconel sheathed type J ungrounded thermocouple, demonstrated longer time constants and settling times, likely due to their larger mass. The experiment highlighted the importance of sensor selection based on application needs, with single-ended sensors favored for rapid temperature changes and differential sensors for gradual changes. Overall, this study enhances our understanding of temperature sensing technologies and their practical implications in engineering and scientific applications.

References

[1] "Understanding Thermocouple Time Constants & Response Times." *HGSI*, https://www.hgsind.com/blog/understanding-thermocouple-time-constants-response-times.
[2] "Omega. (n.d.). Differential or Single-Ended? - Understanding the advantages and disadvantages of differential and single-ended measurements".https://www.omega.com/en-us/resources/differential-or-single-ended
[3] Experiment #3: Temperature Measurements Manual

Appendix

```
%Skin Temperature Measurement
% This program is divided into several key components:
% [1] Setting up the device object along with the Engineering Units % [2] Setting up the plot for real-time display of data
% [3] Acquiring the data and plotting it
% [4] Cleaning up

close all clear clc

% [1] Set up the data acquisition
% Initialize handle for analog input - board is device 1 on computer ai = analoginput('nidaq',1);

%Setup hardware channels - 7 thermocouples - 7 channels

ch = addchannel(ai, [0:6]);
set(ai, 'InputType', 'SingleEnded');

% Setup the engineering units
% J-type
set(ch(1), 'SensorRange',[-0.1154 0.0541]); %Expected range of voltage from the sensor
set(ch(1), 'InputRange', [-5 5]); %which part of the range that we are interested in
set(ch(1), 'UnitsRange', [33.4530 92.5732]); %the corresponding real world values in Fahrenheit
set(ch(1), 'Units', 'F'); %temp scale

% J-type

set(ch(2), 'SensorRange',[-0.1206 0.0506]); set(ch(2), 'InputRange', [-5 5]);
set(ch(2), 'UnitsRange', [32.8526 92.3341]); set(ch(2), 'Units', 'F');

% J-type

set(ch(3), 'SensorRange',[-0.1207 0.0478]); set(ch(3), 'InputRange', [-5 5]);
set(ch(3), 'UnitsRange', [32.8526 92.3341]); set(ch(3), 'Units', 'F');

% K-type

set(ch(4), 'SensorRange',[-0.0954 0.0399]); set(ch(4), 'InputRange', [-5 5]);
set(ch(4), 'UnitsRange', [32.8526 92.3341]); set(ch(4), 'Units', 'F');
```

```
% J-type grounded

set(ch(5), 'SensorRange',[-0.1242 0.0457]); set(ch(5), 'InputRange', [5 10]);
set(ch(5), 'UnitsRange', [32.8526 92.3341]); set(ch(5), 'Units', 'F');

% Thermistor

set(ch(6), 'SensorRange',[2.5 10]); set(ch(6), 'InputRange', [-10 10]); set(ch(6), 'UnitsRange', [25
100]); set(ch(6), 'Units', 'F');

% Platinum RTD probe

set(ch(7), 'SensorRange',[2.5393 2.8621]); set(ch(7), 'InputRange', [-5 5]);
set(ch(7), 'UnitsRange', [32.8526 92.3341]); set(ch(7), 'Units', 'F');

%Set up sampling duration in seconds duration = 30; %30 seconds

% Sample rate of sensor is 100 samples per second

set(ai, 'SampleRate', 100);

ActualRate = get(ai,'SampleRate'); TotalSamples = duration*ActualRate;
set(ai,'SamplesPerTrigger', TotalSamples)

% [2] Setting up the plot to preview the data
scrsz = get(0, 'ScreenSize');
figure('Position',[1 scrsz(4) scrsz(3) scrsz(4)]) set(gcf,'doublebuffer','on'); %Reduces plot flicker
subplot(7,1,1), P1 = plot(zeros(TotalSamples,1)); subplot(7,1,2), P2 =
plot(zeros(TotalSamples,1)); subplot(7,1,3), P3 = plot(zeros(TotalSamples,1)); subplot(7,1,4), P4
= plot(zeros(TotalSamples,1)); subplot(7,1,5), P5 = plot(zeros(TotalSamples,1)); subplot(7,1,6),
P6 = plot(zeros(TotalSamples,1)); subplot(7,1,7), P7 = plot(zeros(TotalSamples,1));

%Start the data acquisition

start(ai)

while ai.SamplesAcquired < duration*ActualRate data = peekdata(ai, ai.SamplesAcquired);
set(P1, 'ydata', data(:,1));
set(P2, 'ydata', data(:,2));

set(P3, 'ydata', data(:,3));
set(P4, 'ydata', data(:,4));
set(P5, 'ydata', data(:,5));
set(P6, 'ydata', data(:,6));
```

```
set(P7, 'ydata', data(:,7));
drawnow; %Forces MATLAB to update plot immediately

end

%------------------------------------------------------------------------ % [4] GETDATA from analog
input object and plot the data. data = getdata(ai); %Convert to Fahrenheit
save 'thermo.dat' data -ASCII -tabs

%------------------------------------------------------------------------ % [5]Cleaning up
delete(ai)
clear ai
```

Figure A1: MATLAB 7 code used for data acquisitions of the sensors for both heating a cooling

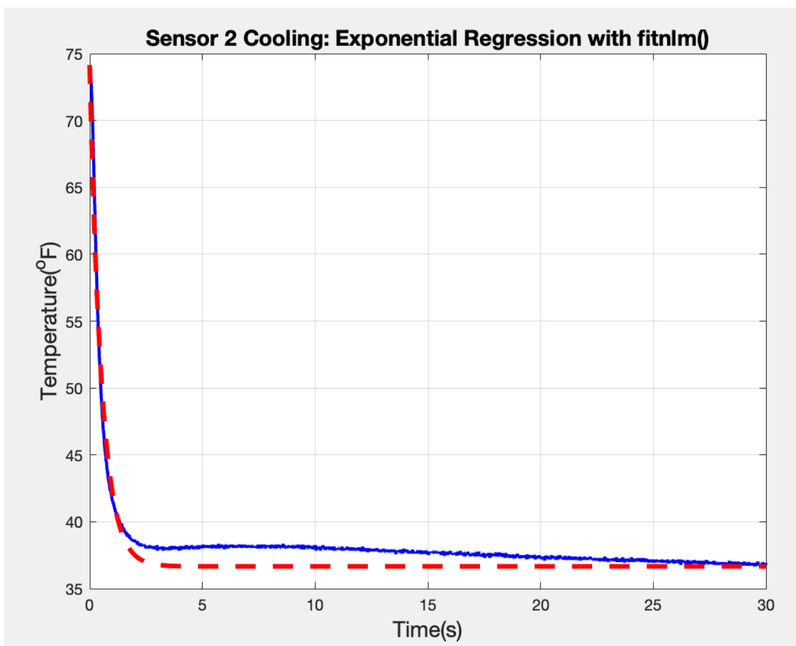

Figure AC1:Graph comparing time vs temperature of Experimental regression fitting of sensor 2 in cooling case

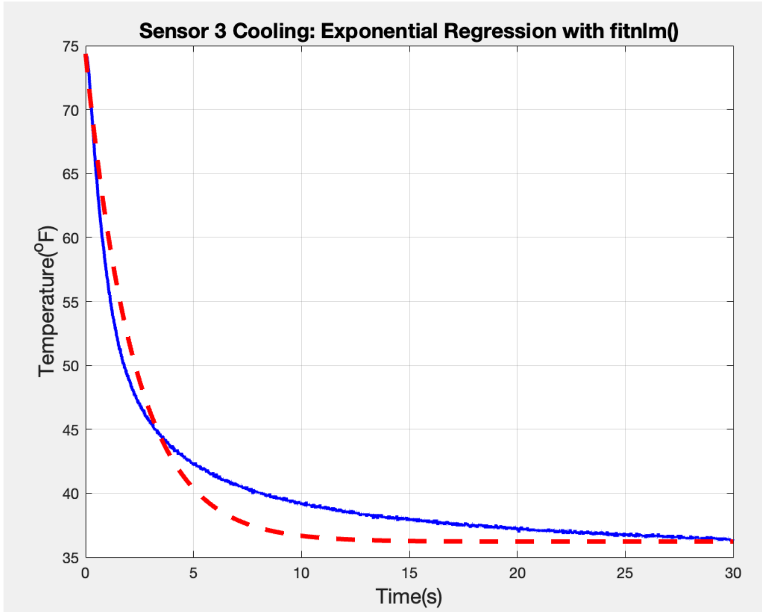

Figure AC2: Graph comparing time vs temperature of Experimental regression fitting of sensor 3 in cooling case

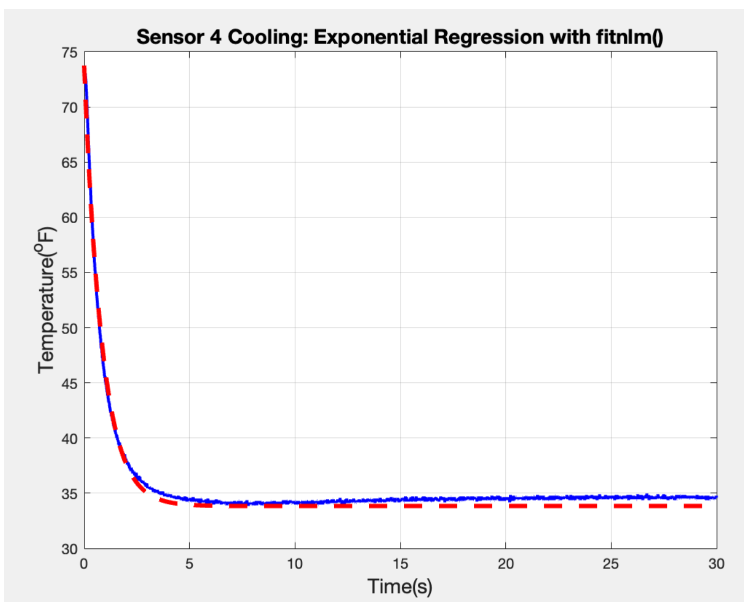

Figure AC3: Graph comparing time vs temperature of Experimental regression fitting of sensor 4 in cooling case

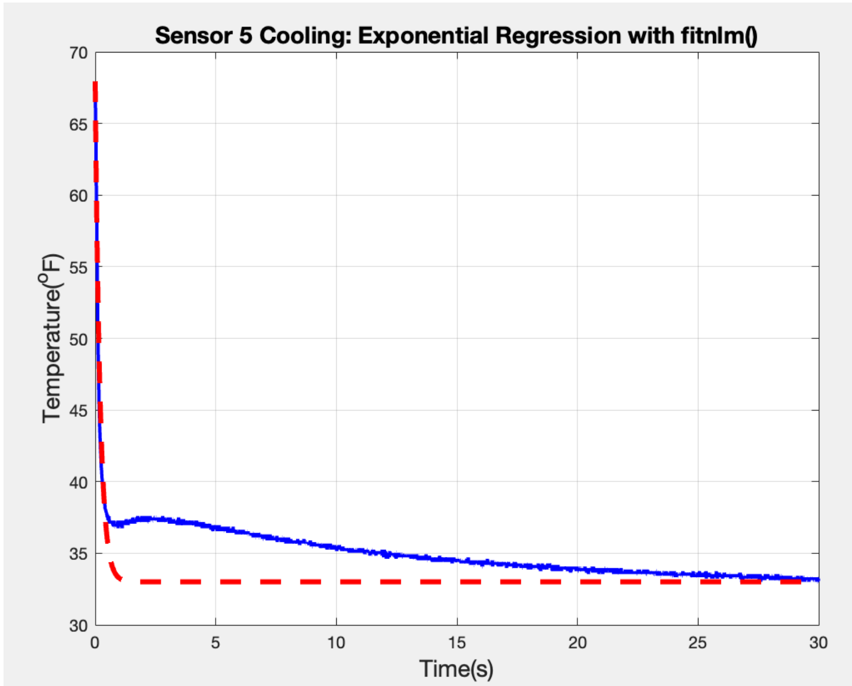

Figure AC4: Graph comparing time vs temperature of Experimental regression fitting of sensor 5 in cooling case

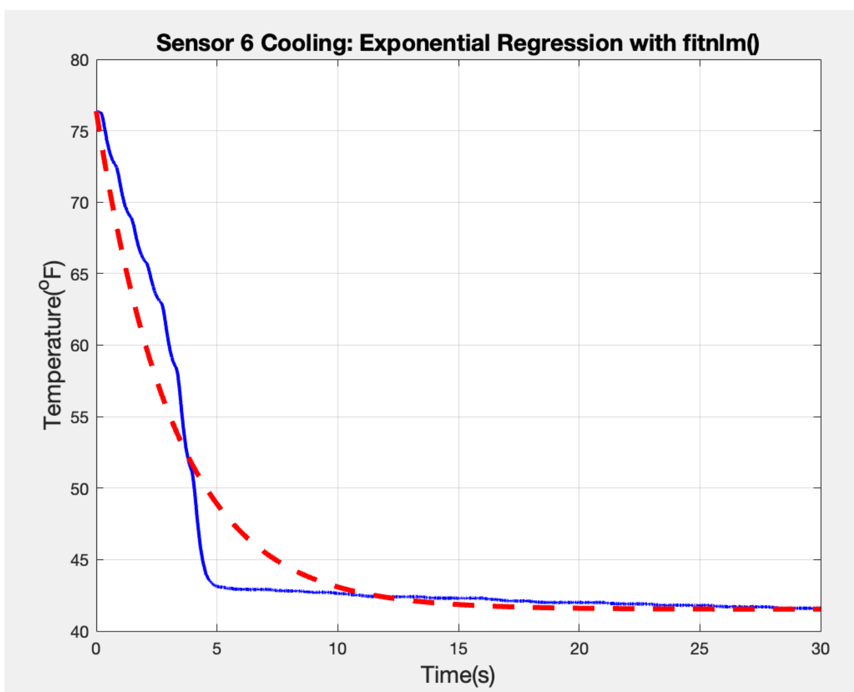

Figure AC5: Graph comparing time vs temperature of Experimental regression fitting of sensor 6 in cooling case

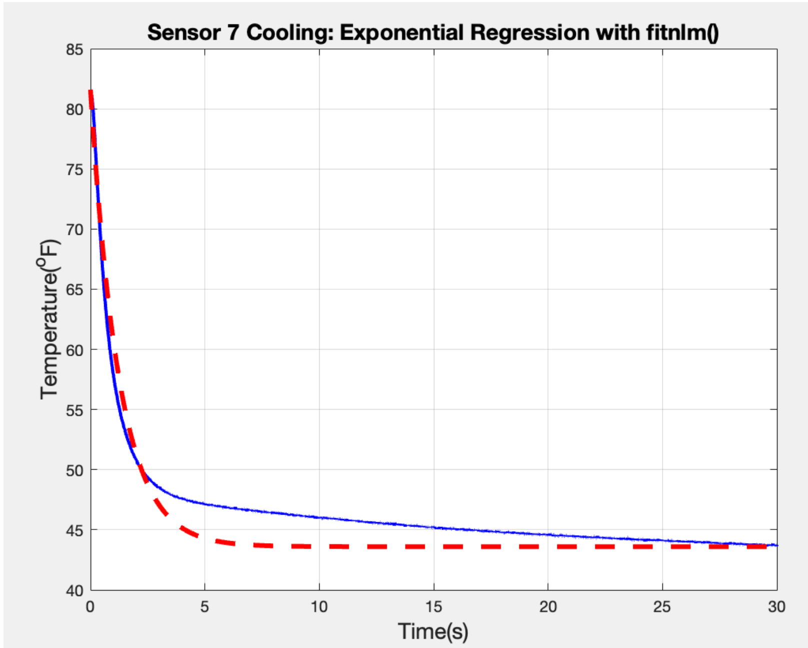

Figure AC6: Graph comparing time vs temperature of Experimental regression fitting of sensor 7 in cooling case

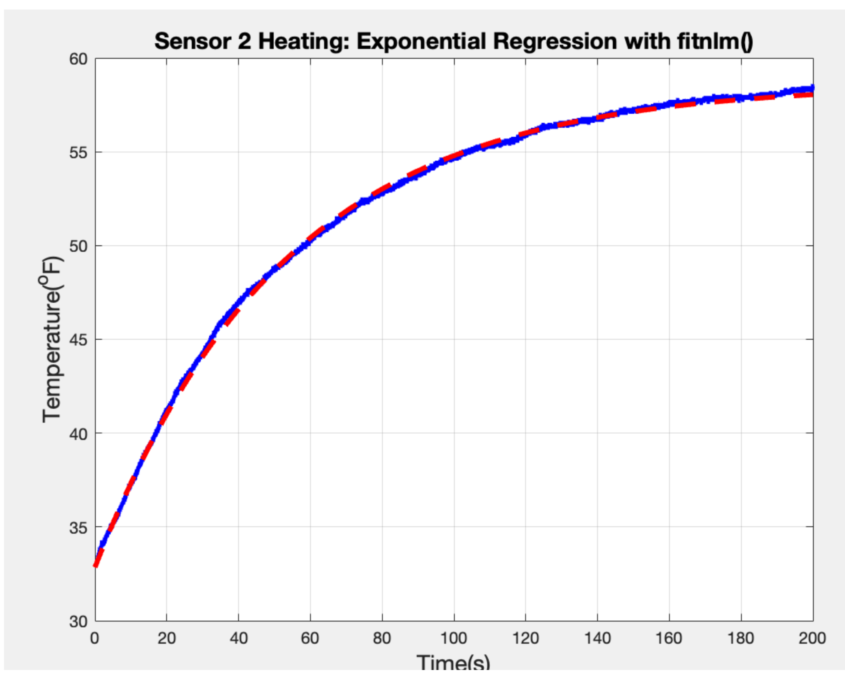

Figure AH1: Graph comparing time vs temperature of Experimental regression fitting of sensor 2 in heating case

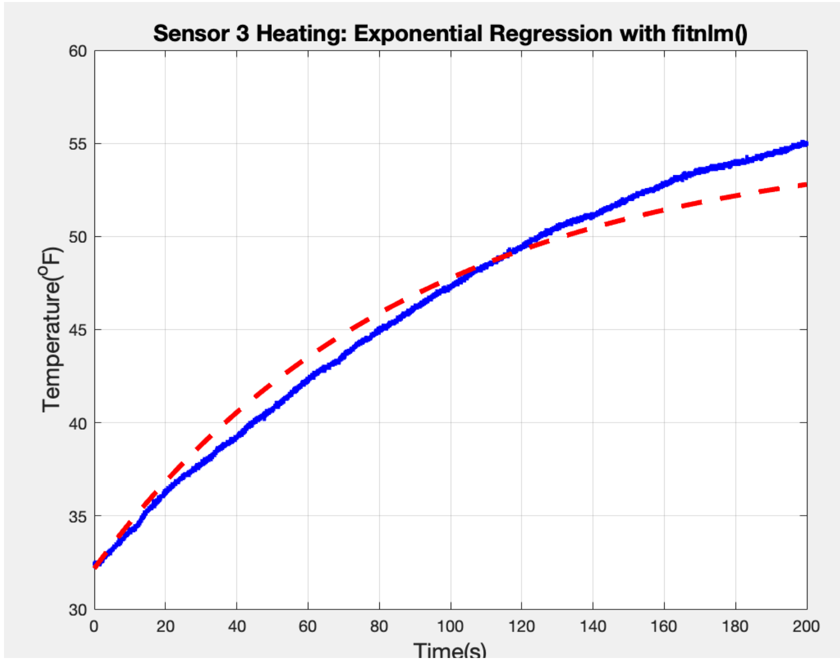

Figure AH2: Graph comparing time vs temperature of Experimental regression fitting of sensor 3 in heating case

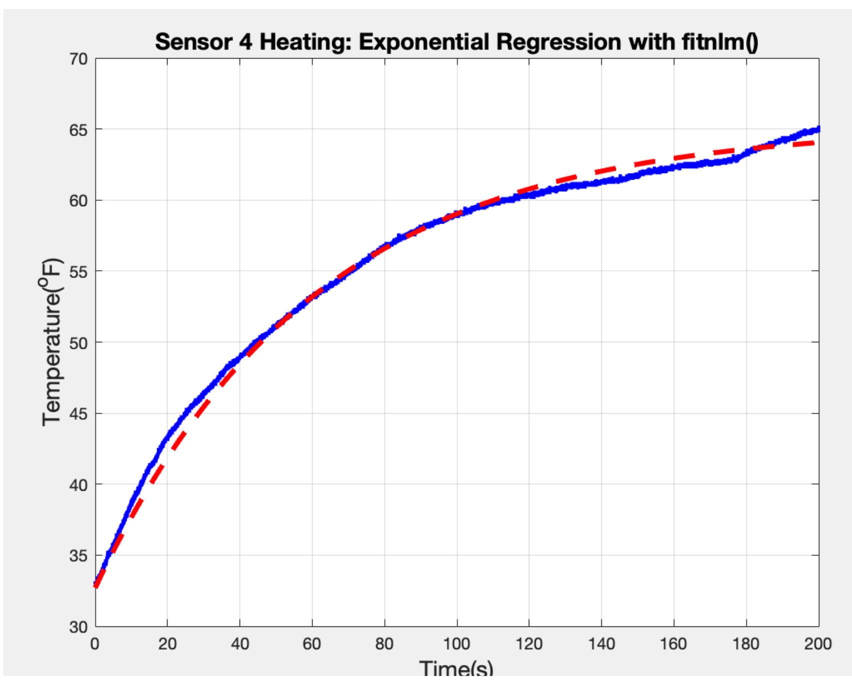

Figure AH3: Graph comparing time vs temperature of Experimental regression fitting of sensor 4 in heating case

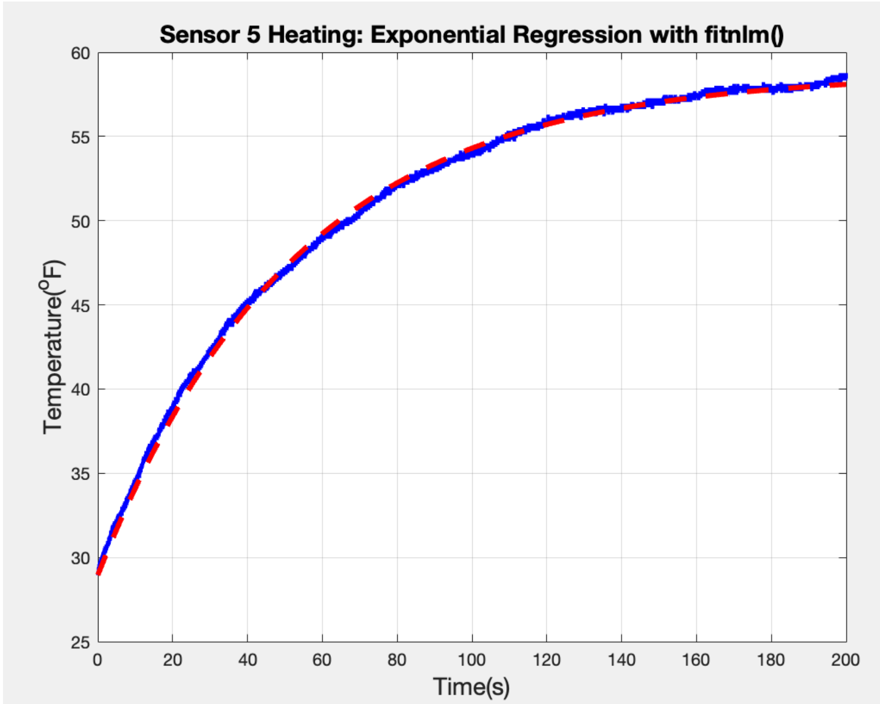

Figure AH4: Graph comparing time vs temperature of Experimental regression fitting of sensor 5 in heating case

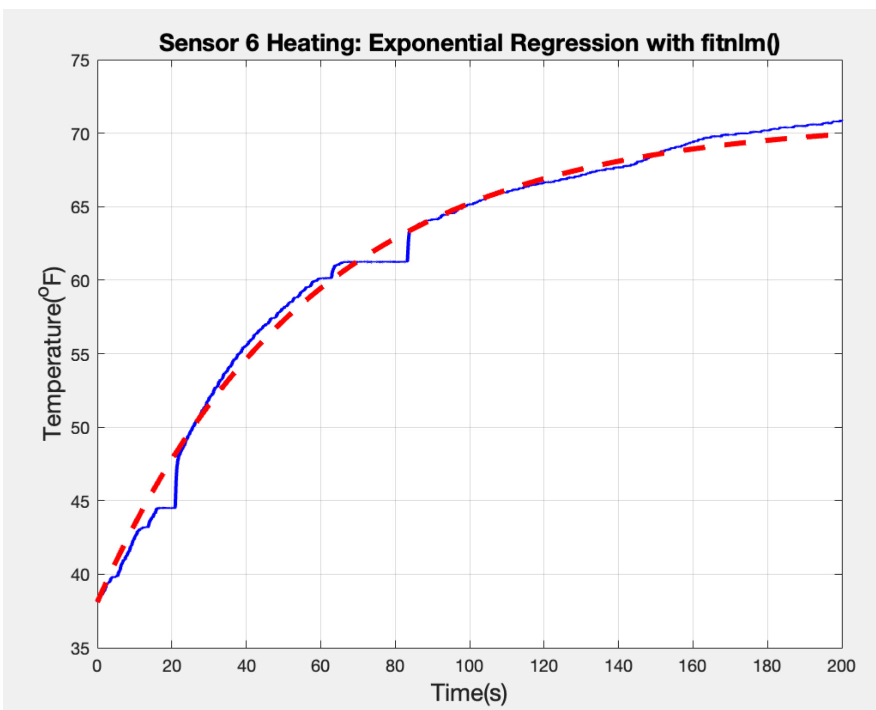

Figure AH5: Graph comparing time vs temperature of Experimental regression fitting of sensor 6 in heating case

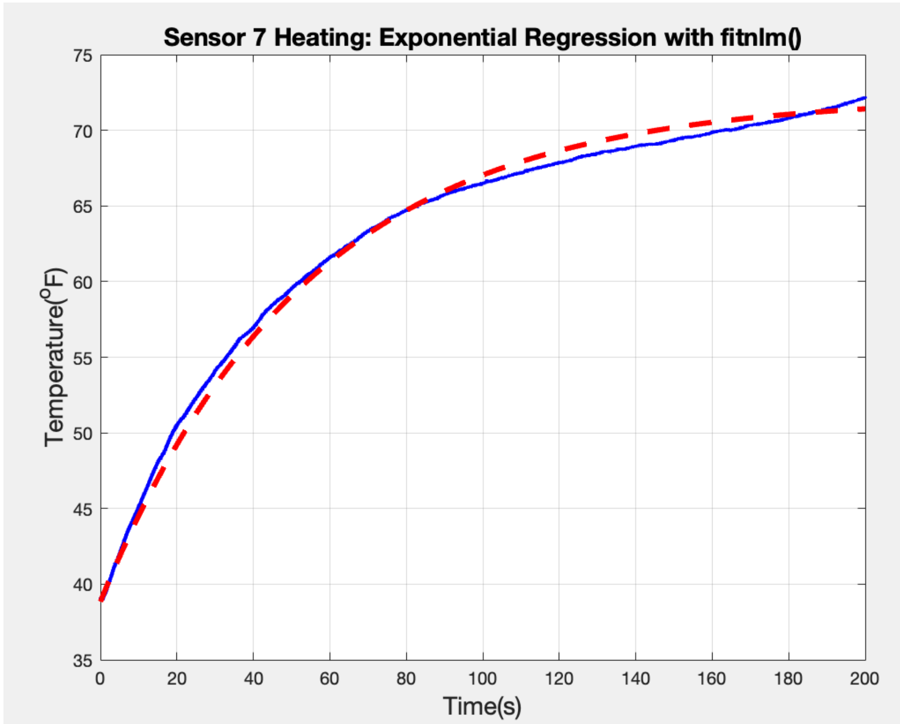

Figure AH6: Graph comparing time vs temperature of Experimental regression fitting of sensor 7 in heating case