Interpolation of Noisy Data

Jordan Rhoades

University of Central Arkansas

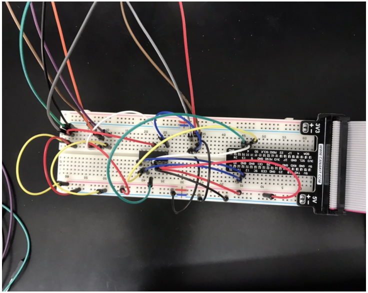
[redacted]

December 6, 2018

Overview

Introduction

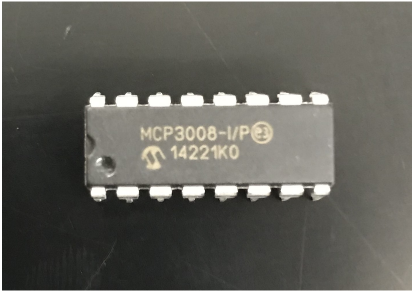
- My final project for Microcontrollers last year was to identify if a computer that was dusty ran at a higher temperature than that when it was dust free
- This project used a Raspberry Pi 3 and a circuit to be able to read and write data from three different locations on the computer and create graphs over a period of fifteen minutes when the computer was under a stress test

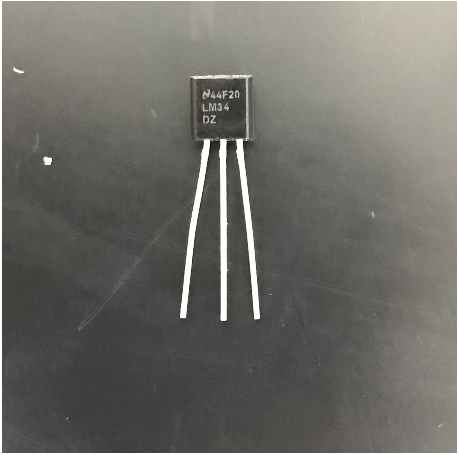


Introduction

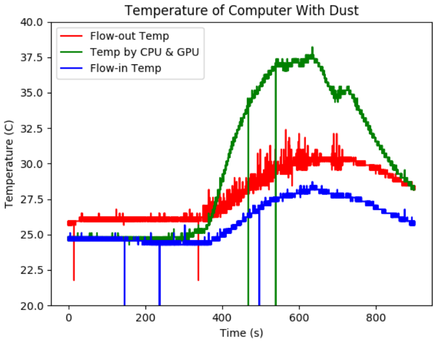
- The way the circuit read in information was with the help of three thermistors that depending on the amount of heat presented, would vary in resistance and could then have the temperature be calculated
- The other component used in the circuit was a MCP3008 which is an integrated circuit that reads in the voltage going across the thermistors and would send that data to the Raspberry Pi to then created plots in Python

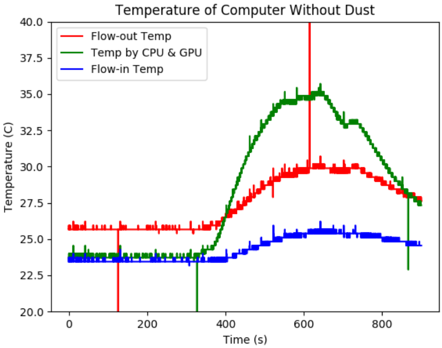
Introduction





Introduction





Problem Formation

- Due to the amount of noise gathered, it is rather difficult to take those two graphs and get a qualitative analysis
- You can see between the graphs that there is a difference in temperature but not by how much exactly
- I need to use a numerical method that we learned in this course to create a "best fit" polynomial that can give me a close approximation for these noisy graphs

Numerical Methods

- Since we learned about multiple interpolation methods in this course, I could use two methods and create good approximations and numerically compare my data
- The two methods that will be used in this project: Lagrange Interpolation and Least Squares
- I used Lagrange Interpolation because it does not create a polynomial that goes through each data point like the cubic spline method and instead goes through evenly spaced intervals following the data

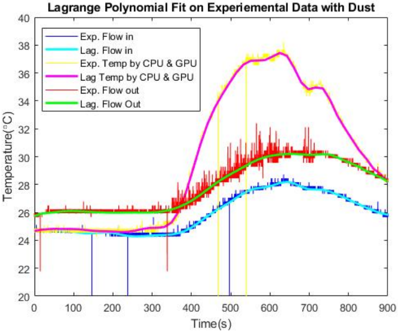
$$P(x) = \sum_{j=1}^{n} y_j \prod_{i=1; i \neq j}^{n} \frac{x - x_i}{x_j - x_i} \tag{1}$$

Numerical Methods

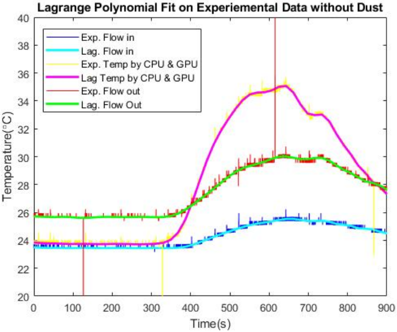
- I also used Least Squares because even with error in the experimental data, the approximated polynomial does not get distorted or taken over by error

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i \tag{2}$$

Results

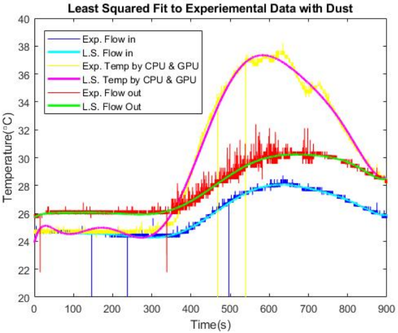


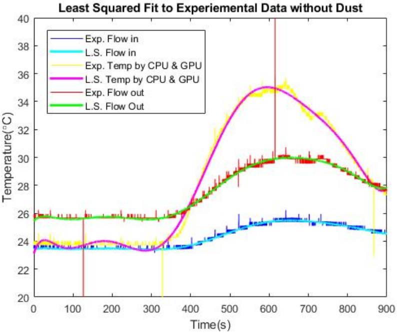
Lagrange Polynomial Fit on Experiemental Data with Dust
Exp. Flow in
Lag. Flow in
Exp. Temp by CPU & GPU
Lag Temp by CPU & GPU
Exp. Flow out
Lag. Flow Out
Temperature(°C)
Time(s)
40
38
36
34
32
30
28
26
24
22
20
0
100
200
300
400
500
600
700
800
900



Lagrange Polynomial Fit on Experiemental Data without Dust
Exp. Flow in
Lag. Flow in
Exp. Temp by CPU & GPU
Lag Temp by CPU & GPU
Exp. Flow out
Lag. Flow Out
Temperature(°C)
Time(s)
40
38
36
34
32
30
28
26
24
22
20
0
100
200
300
400
500
600
700
800
900

Results


Least Squared Fit to Experiemental Data with Dust
Exp. Flow in
L.S. Flow in
Exp. Temp by CPU & GPU
L.S. Temp by CPU & GPU
Exp. Flow out
L.S. Flow Out
Temperature(°C)
Time(s)
40
38
36
34
32
30
28
26
24
22
20
0
100
200
300
400
500
600
700
800
900


Least Squared Fit to Experiemental Data without Dust
Exp. Flow in
L.S. Flow in
Exp. Temp by CPU & GPU
L.S. Temp by CPU & GPU
Exp. Flow out
L.S. Flow Out
Temperature(°C)
Time(s)
40
38
36
34
32
30
28
26
24
22
20
0
100
200
300
400
500
600
700
800
900

Results

Thermistor	**5 min**	**10 min**	**15 min**
Exper. 1	24.3458	27.8791	25.8349
Exper. 2	24.3458	37.0771	28.1484
Exper. 3	25.9955	30.0495	28.2139

Table: Lagrange Interp. Data with Dust

Thermistor	**5 min**	**10 min**	**15 min**
Exper. 1	24.3519	25.3291	24.6429
Exper. 2	24.3072	34.6846	27.0372
Exper. 3	25.6667	29.7726	27.6429

Table: Lagrange Interp. Data without Dust

Conclusion

- Lagrange Interpolation surprisingly does a good fit of my noisy data and does exactly what I needed it to do
- Least Squares does a good job fitting two of the three sets of data but overall does not compare to the polynomials that the Lagrange Interpolation method makes

Future Work

- Test more interpolation methods to see if I can get another method to create a best fit polynomial
- I would need to run this experiment again using more accurate thermistors
- The thermistor reading the in-flow temperature needs to be on the outside of the computer
- Further testing needs to be done on better locations for reading the CPU and GPU temperatures

References

- Burden, Richard L., et al. Numerical Analysis. Cengage Learning, 2016.
- MathWorks - Makers of MATLAB and Simulink. Random Number Generation - MATLAB amp; Simulink, www.mathworks.com/.
- Montalvo, Dr. Carlos, director. YouTube. YouTube, YouTube, 17 Mar. 2015, www.youtube.com/watch?v=5usW7aqo-bU.

Questions?