To: President Homer J. Simpson of FOS, Inc.

From: Engineering Team 19

RE: Final Technical Brief

Date: April 20, 2017

The problem originally proposed to our team consisted of constructing an algorithm in MATLAB that would be able to analyze a series of provided time histories for 5 different types of thermocouple models. Based off our analysis we were then instructed to provide feedback in a comprehensive manner. This included graphical representation and a concise summary of our findings. We were also expected to provide an error analysis that would sow how accurately our algorithm determined the time constant. To do this we found the SSE, SST, and R squared values for our data as well as the mean of the time constant value, tau, and the standard deviation of tau. We were able to compare our algorithms results with provided target values of mean tau, the standard deviation of tau, and the mode of the SSE. If our data was within the target value range the algorithm would be considered successful.

We smooth the data set and then use a slopes method to determine ts; the ts value is then used to find yl and yh using the index of ts and either the maximum or minimum of the data set. We then use the provided equations and the found parameters to calculate tau.

1. The first decision we made was to choose Algorithm 2 as our major model for future programming and improvement. We carefully compared every single result by A1 and that by A2 with the actual values, and found that the results produced by A2 to be more accurate than those produced by A1. Therefore, we decided to use A2 as our future programming model.
2. We decided to do curve smoothing twice. We had known that the build-in commands movmean() which functions by moving the mean value and smooth() which functions by smoothing the data can help to smooth the noisy data and make the curve cleaner. In our early algorithms, we either use movmean() or use smooth() to smooth the noisy data. We now combine these two together so that we can get a smoother curve for the noisy data. Although we had results which were close to the actual values, after the improvement we got even more accurate results. For example, before the improvement, the SSEmod for heating data smoothed by movmean() was 5.25 and by smooth(), 2.25, and after combining them together, we got a SSEmod of 2.1664 which shows that the results are more accurate than before.
3. Since the τ value is more important and more relative to the final price-τ regression, we changed our algorithms to find a more accurate τ value: we chose to compute yl in heating data and yh in cooling data by directly find the Temperature values with the same index of ts throughout Time values. By doing this we got better τ values and thereby make the price-τ regression more accurate: the r-squared raised to 0.9215 from 0.9037.

Our algorithm sorts through the input data to find out whether it is a cooling or heating curve. For the cooling curve, we first smooth the data. Then we find the minimum and maximum points on the curve. We then run a for loop to find the index of ts for the temperature data. We assign ts from the index value returned. We then take the index of ts in the time vector and apply that index to our smoothed temperature vector to get the yh value. To find yl we take the minimum of the data set. We then calculated ytau using the provided equation and the variables found by our algorithm. For the heating curve, we smooth the data. Then we find the max and min of the data set again. We again used a for loop to find the index of ts then assigned the value for ts using the index. To compute yl we take the index of ts in the time vector and apply that index to our smoothed temperature vector to get the yh value. We then take the mean of the vector to find yl. We compute yh is the max value of the data set. We then compute tau using the equations provided.

Our parameter identification algorithm could be better if we were able to find more accurate values for all parameters not just tau. We sacrificed the quality of of yl for our heating data and yh and yl for our cooling data that we found in milestone 3 to improve the quality of our tau for milestone 4. For example, in m3 for one cooling data set, the yh is 97.2087 and in m4 the yh is 95.8919. The actual yh is 97.99. The tau for the cooling data was 1.0068 in M3 in M4 it was 1.0478 and the actual tau is 1.03. This shows that although our tau increased in accuracy it was at the expense of our yh data. This show there is an error in our algorithm.

Based on Figure 1, for a given thermocouple at price x, all the time constants that we got from our analysis were within 0.2 seconds of each other. FOS can say that their products are consistent in the sense that for each trial run for each model the time constants found are relatively consistent. In terms of price and accuracy, the more expensive the thermocouple the small the time constants are. This can be seen in figure one where thermocouple model shows the more expensive the thermocouple the small the time constant is.

Reference:

Thermocouple Measurements (2017).  Retrieved 19 April 2017, from https://www.newark.com/ Making-Good-Thermocouple-Measurements-in-Noisy-Environments.pdf

|  |  |  |
| --- | --- | --- |
| **M4 Algorithm** | | |
| ** Characteristics** | | **Mean SSEmod** |
| **Mean** | **Standard Deviation** |
| 0.21 sec | 0.0360 | 4.0151 |
| 0.39 sec | 0.0370 | 1.8760 |
| 0.88 sec | 0.0385 | 1.2768 |
| 1.26 sec | 0.0291 | 1.3383 |
| 1.82 sec | 0.0303 | 2.0340 |

Figure 2: This figure shows the Mean, Standard Deviation, and Mean SSE for our M4 algorithm.



Figure1: This figure shows the final regression results for our data analysis