Technical Brief

To: President Frank O. Simpson

From: Team 5, Section 12

RE: Technical Brief of the FOS Project

Date: 27th April, 2017

This problem tasked us with providing a quality analysis of thermocouples designed by First Order Systems (FOS) so that they may provide an ethical statement about their thermocouples to customers. FOS needs accurate information on the parameters and the functionality of their thermocouples. A deadline of April 27th, 2017 was established to ensure the algorithm is delivered on time. This information is supplied through our team's executive function, algorithms, and regression function which FOS can utilize to examine their thermocouples. The constraints that challenged us were the short time frame of the project we have been provided and the fact that we had to analyze noisy data. The algorithm is a deemed a success when the values that are outputted are within a set interval. This interval was provided by successful test cases.

Our algorithm makes use of MATLAB's smoothing function, and also averages sets of 20 points at a time to smooth the noisy data. The smooth function allows for sets of data to be replaced by a moving average (Filtering and Smoothing Data) which allows a smoother line to be generated by MATLAB's own system of noise reduction.By smoothing the data as much as possible, we are able to accurately detect when the thermocouple begins to read a change in temperature, and subsequently find the four parameters; τ (sec), t_s (sec), y_L (^oC), and y_H (^oC).

To create a functioning algorithm, our team first determined parameter identification methods generalized to fit any type of data set, then generated MATLAB algorithms based off our chosen methods, and worked on improving the algorithm that was most successful. During this process, our team made important decisions to improve the accuracy of parameter identification. We decided to compute y_H and y_L by taking the minimum and maximum of the smoothed data sets, and came to the conclusion that using the smoothed data set to compute τ and t_s was more accurate, rather than using the original data. After testing many other strategies, we determined computing yH and yL as simply as possible once we had the smoothed data set would be optimal. Taking the minimum and maximum of the improved data set proved to return the most accurate yH and yL values, as the smoothed data exists as a moving average of the original data, and therefore represents what the data would look like if truly smooth (Filtering and Smoothing Data). Furthermore, we discovered, after running many data sets through our algorithm, that computing the remaining parameters was also much more accurate when the algorithm computed based off of the smoothed data, for the same reason. Another important decision was calculating the moving average between 20 points of data at a time, rather than 200 (in our original algorithm). A moving average of every 20 points, compared to 200, is much more representative of the actual data (Han, K., & Jian-Pei, M. K.). However, we still wanted to average a set of points large enough to make sure that the averaged value between them accurately represents how the data is trending (Writer, L. G.). This decision caused the greatest increase in accuracy of our algorithm.

The first step of our algorithm makes use of the smooth function built into MATLAB. We embed the function in a loop to produce multiple smoothing iterations, and then took a moving average of the resultant time and temperature data. After this, we compute the y_H and y_L parameters simply by extracting the minimum and maximum temperatures from the now nearly smooth data. We then run the data through a loop to determine where the t_s value lies. By taking a moving average of the slope between adjacent data points, and having the code exit the loop when it detects a slope above a tolerance level, a position of this change is returned to pinpoint exactly where the slope increase occurred within the data. We then search the actual data with the position of change from the loop to find t_s. This tolerance was found through what would be considered a "reasonable" slope increase, as well as extensive trial and error for what tolerance value returned the most accurate parameters. Our tolerance works across all data sets and does not compromise the robustness of our code τ is found simply by finding the x-value where 63.2% of the temperature change occurs, and subtracting the t_S value.

Table 1 displays the τ characteristics that our algorithm is required to determine. The accuracy of our final parameter values are indicative of the refinements we made to our code throughout its development. The SSE, SST and r^2 values presented in Table 2 are in reference to the values displayed in Figure 1. The r^2 value measures how successful the fit is in explaining the variation of the data, or the correlation between the response values and the predicted response values (Goodness-of-fit Statistics). We have an r^2 value of 0.92. This means is that our algorithm's outputted best fit line accounts for 92% of variability in the data and hence, represents the data very well. SSE measures the total deviation of the response values from the fit to the response values and is also called the summed square of residuals. Our SSE value is 2.78 which is about 45% lower than the threshold recommended. SST is defined as the sum of the squared deviations and an SST value of 35.46 was calculated. Figure 1 represents the price of thermocouples compared to time constants regression model. A perfect set of data would be clean, making interpreting and analyzing the data would be easy. However, the data we received was noisy. As a result, interpreting and analyzing this on MATLAB becomes difficult. Consequently, this could be a potential error. One flaw in the quality of the experiment is mainly the short deadline provided.

In accordance to the difference in pricing, the higher priced products perform better. As displayed in Table 1, the most expensive model has the shortest time constant, while the least expensive model had the longest time constant. Also, seen in Figure 1, the manufacturing consistency is also evident in the spread of data at a certain time period. The higher priced models have a lower spread of data, or lowered standard deviation in comparison to the cheaper models. Therefore, the highest priced thermocouple performs the best, and the other models' performances (in terms of time constant) decreases exponentially with price.

References

1. Filtering and Smoothing Data. (n.d.). Retrieved April 6, 2017, from https://www.mathworks.com/help/curvefit/smoothing-data.html
2. Han, K., & Jian-Pei, M. K. (2012). *Data Mining Concepts and Techniques* [PDF]. Waltham: Elsevier. Stewart, J. (2016). *Calculus: Early Transcendentals* (8th ed.). Australia: Cengage Learning.
3. Hellerstein, J. M. (2008, February 27). Quantitative Data Cleaning for Large Databases. Retrieved April 6, 2017, from http://db.cs.berkeley.edu/jmh/papers/cleaning-unece.pdf
4. UNSW : Goodness-of-Fit Statistics. (n.d.). Retrieved April 06, 2017, from http://web.maths.unsw.edu.au/~adelle/Garvan/Assays/GoodnessOfFit.html
5. Writer, L. G. (n.d.). The Advantages of a Large Sample Size. Retrieved March 29, 2017, from http://sciencing.com/advantages-large-sample-size-7210190.html

Table 1: Performance comparison for FOS designs

Model Number	τ Characteristics		Mean SSE_{mod}
	Mean	Standard Deviation	
FOS-1	0.21 sec	0.04 sec	3.99 sec^2
FOS-2	0.41 sec	0.04 sec	2.29 sec^2
FOS-3	0.90 sec	0.07 sec	1.81 sec^2
FOS-4	1.28 sec	0.08 sec	1.70 sec^2
FOS-5	1.83 sec	0.11 sec	2.44 sec^2

Table 2: Final Regression Values

SSE	2.78
SST	35.46
r^2	0.92

Figure 1: Final Representation of Regressed Data