Milestone M5 – Technical Brief

To: President Frank O. Simpson; FOS, Inc.

From: Evan Kelch, Jeff Nickels, Dante Sanaei, Cody Martin

RE: FOS Technical Brief

Date: April 25th, 2018

First Order Systems Inc. has requested that our engineering team design an algorithm capable of evaluating the performance of their five leading thermocouple designs. Our algorithm can be used to obtain thermocouple parameters and use them to analyze secondary data such as thermocouple response time to its respective cost (Figure 1).

Our main algorithm is capable of testing any first order thermocouple system to determine four parameters: the high temperature value (yH), the low temperature value (yL), the time step (ts), and the tau value (tau). Even when the data is varied (noisy) due to external stimuli, our model is able to determine the parameters needed. We are constrained by the data that we receive, the extent of its inaccuracy (noise), and the deadlines that needed to be reached by certain dates. The criteria for success will be how accurate our parameters model the real Table 1.

When drafting our algorithm, we considered multiple different engineering methods and made several keystone decisions, allowing us to improve the accuracy of our algorithm. We began by developing two different prototype algorithms that aimed to accomplish the same goal. We were able to determine the superior algorithm by comparing each of their generated parameters to the provided calibration data, considering both accuracy and efficiency. As supported by Table 3, we noticed that the figures generated by our second algorithm were more accurate; thus, justifying our decision to use algorithm 2 in further milestones.

Additionally, we made the critical decision of programmatically smoothing the noisy data that is provided by FOS. This decision was made with the evidence based rationale that it would be impossible to interpret the data when the temperatures were extremely varied and “noisy”. We tested this hypothesis by having our first algorithm not utilise a moving average, and having algorithm two use this kind of coding technique. The clean (non-noisy) data parameters can be noted in Table 3, and the noisy data is extremely similar. This is a direct result of the smoothing.

Lastly, during the refining phase of our project, we chose to specifically focus on greatly improving the efficiency and speed of the code. We had calculated that our earlier designs iterated millions of times, and we made it a priority to reduce the amount of looping. This was a vital decision due to the fact that our algorithm must run extremely quickly in order to utilize as little computer resources as possible and minimise compiling time (especially when utilized or looped in other executive functions). Our decrease in program time can be noted in Table 2.

First, our algorithm accepts time data in seconds and corresponding temperature data in degrees Fahrenheit. It smooths this data by averaging each temperature with the temperature immediately after and immediately before it, and this averaging action repeats exactly 150 times in order to ensure maximum accuracy of the data. Then, our algorithm calculates the slope of

each point with the point 200 time values in front. This allows for the code to find the maximum slope. The index in which the maximum slope occurs, is where the time step is located. After the time step is calculated, the yL and yH values are determined (the start and ending temperatures). These values are generated by calculating the average of the first 200 temperatures and the last 200 temperatures. Our code then compares the yH and yL values and determines which one is greater than the other in order to determine whether the data set represents a heating or cooling data set. With these values and the previously calculated time step, the code formulates the corresponding temperature for the time constant using the equation that corresponds to the type of data set, meaning our code uses the equation that corresponds to heating or cooling. The algorithm uses a looping structure in order to find the index of the closest temperature to the one calculated; thus, allowing it to obtain the resulting time constant.

Our algorithm is able to nearly emulate the calibration data provided. Tables 3 and 4 show that our tau and SSEmod values (measure of the model's accuracy) are nearly within an error of 0.1%. In the scope of FOS's applications, this miniscule variance is negligible.

The algorithm also works considerably well when analyzing the 100 test cases we were provided by FOS. Table 1 shows the time constant(τ) characteristics of the 5 different thermocouple designs determined from our algorithm. The significantly lower time constant of FOS-1/2 imply that these models are much more responsive as compared to cheaper designs, such as FOS-5. This is supported through non-linear regression, as displayed in Table 1. As time constant decreases, the model price exponentially increases.

The input of our algorithm is temperature changing over time, recorded by each thermocouple design. However, the data recorded by each thermocouple is 'noisy', and therefore is not perfectly linear in nature. The non-linear data sets allow different interpretations of how to correctly determine each of the parameters, with each method having particular strengths and weaknesses. Regardless, the error within our code is within an acceptable margin, and does not cause any significant discrepancies within our model's predictions.

With a mathematical algorithm to support them, FOS Inc may justify their products' performance and respective pricing. FOS may utilize the algorithm to accurately determine the correlation between cost and effectiveness in their designs. For each FOS thermocouple model, the SSEmod of each data point to its piecewise counterpart is approximately 0.4 ($degF^2$), as shown in Table 2. This means that each actual data point captured by the thermocouple was, on average, less than one degree from its predicted value using our algorithm. This model also portrays a high level of consistency performing across all thermocouple models. This is supported by the fact that our coefficient of determination is extremely close to a value of 1, shown in Table 2, and our standard deviation is close to the accepted range.

Table 1: Tau Statistics

Model Number	Model Pricing ($)	τ Characteristics		Mean SSE_{mod} ($degF^2$)
		Mean (sec)	Standard Deviation (sec)	
FOS-1	17.02	0.15	0.03	0.36
FOS-2	9.16	0.34	0.03	0.36
FOS-3	3.77	0.93	0.06	0.40
FOS-4	2.19	1.09	0.08	0.39
FOS-5	0.70	1.53	0.12	0.49

Table 2: Algorithm Properties

Parameter	**Value**	**Algorithm**	**Execution Time (sec)**
SSE ($\$^2$)	0.698	M3 Algorithm	.221
SST ($\$^2$)	23.330	M4 Algorithm	.022
r^2	.970		

Table 3: Algorithm Comparison Cooling

Cooling

Parameter	M2Calibration		M3Algorithm		M4Algorithm	
	Clean	Noisy	Clean	Noisy	Clean	Noisy
y_H (℉)	100.00	98.81	100.00	99.41	100.00	98.90
y_L (℉)	0.94	-0.21	0.93	-1.08	0.99	-0.75
t_s (*sec*)	1.50	1.50	1.57	1.55	1.52	1.52
τ (*sec*)	1.82	1.12	1.79	1.12	1.77	1.12
SSE_{mod}	0.51	1.30	1.99	2.73	0.24	1.29

Figure 1: Thermocouple Cost Compared to Efficiency