|  |  |
| --- | --- |
| **Team Member Names:** | Tyler Huter, Andrew Sartorio, Harith Kolaganti, Vedant Srinivas Raghavan |
| **Purdue Logins:** | thuter, hkolagan, raghavav,asartor |
| **Section Number:** | 5 |
| **Team Number:** | 12 |

|  |
| --- |
| **Submission Instructions:**   1. Rename this answer sheet to be **Project\_M4\_*sss*\_*tt*.docx**, where ***sss*** is your section number (e.g., 001 for section 001) and ***tt*** is your team number (e.g., 07 for team 7). 2. **BE SURE** to indicate in your MATLAB M-file, using as many comments as necessary, the specific details of your refinements. Please label them “Category 1” and “Category 2” or “Category 3”, and use comments to briefly describe the nature of the refinements. 3. Compress all deliverables into one zip file named **M4\_*sss\_tt*.zip**. Submit the zip file to the Blackboard drop box for M4 prior to Class 30.   Update all MATLAB m-file names to M4 and name the algorithm files as explained in this document. You will be submitting ***three*** m-files, the answer sheet, and the technical brief as part of this assignment:   * 1. **Project\_M4Exec\_sss\_tt.m**   2. **Project\_M4Algorithm\_sss\_tt.m**   3. **Project\_M4Regression\_sss\_tt.m**   4. **Project\_M4\_*sss*\_*tt*.docx**   5. **Project\_M4TechnicalBrief\_*sss.tt*.docx**   Notes:   * Only one submission is required per team (not one per team member). * Only the last submission will be graded, so make sure all files are submitted to the M4 drop box. * After submission, distribute the submitted files to all team members*. At no time during this project should only one member of your team be in possession of project documentation.* |

|  |
| --- |
| **Particular Learning Objectives are highlighted throughout the document. However, all LOs that you have encountered throughout the semester may apply where appropriate to your work on the Milestones.** |

**Part 0: M3 Feedback Review**

|  |
| --- |
| **Learning Objective (LO): 22.00 Reflect on feedback for the purpose of improvement**  ***Evidence of Proficiency Requires*:**   * Feedback summarization is clear and useful * Response plan is clear and practical |

1. In your own words, summarize the feedback you received on project milestone M3 that could lead to improvements in your work.

|  |
| --- |
| The feedback received on project milestone M3 presents the idea that our algorithm needs more **rethinking** rather than just small refinements. Our algorithm displays high mean values and high standard deviation values, which implicates “little robustness” in our code. To lead to improvements in our code, it is essential that every part of the code serves a crucial role, and some of these loops are unnecessary. |

1. Based on your feedback, what do you need to do to improve your parameter identification approaches? (Do not just reword your response to Part A. Do consider how you will incorporate your feedback into your work.)

|  |
| --- |
| To improve our parameter identification approach, we will need to make every loop more efficient and important. Major changes to how our code functions will need to be changed as well because to make it work for all kinds of data it encounters. It works with some data sets, as described in the feedback by Andres. But, it seems that a majority of the data will not work. The standard deviation for our tau values, as compared to the target values, are almost perfect for FOS-3, FOS-4 and FOS-5 but still off for FOS-1 and FOS-2. This indicates that our code works to some extent, and a few significant changes could largely alter the accuracy of our algorithm..  The SSEmod however are way off. Hence, we need to rework the section of our code that computes SSEmod ,entirely. |

**Part 1: Refinements Preview**

Consult the M4 memo from FOS, Inc. for the details concerning your task. Respond to each of the prompts below in the space provided. Your goal is to introduce ***two refinements*** to your original algorithm, and these refinements must improve your solution to the FOS parameter identification problem. ***Read the rest of this document*** ***carefully*** ***before you begin your work on this milestone***.

**Definition of “refinement”**

In this milestone, a refinement will fall into one of the following categories:

* **Refinement Category 1: Parameter Identification:** an improvement that changes the way you are doing parameter identification, and that improves your parameter identification results.
* **Refinement Category 2: Algorithm Efficiency:** an improvement that improves the efficiency of your code by (for example) removing un-needed looping structures, streamlining data handling, or otherwise reducing the execution time of your code.
* **Refinement Category 3: Algorithm Insight:** an improvement that involves analysis of your code and its limitations. For example, if you use any kind of thresholding in your code, you could determine the sensitivity of the solution to changes in that threshold parameter, and report how those changes affect your parameter identification and/or regression results.

In this milestone, you are ***REQUIRED*** to implement the parameter identification refinement (**Refinement Category 1)**. You must *also implement one of the other refinements*. You are therefore required to implement ***two*** total refinements.



Bottom of Form

|  |
| --- |
| **Learning Objective (LO): 21.02 Communicate ideas clearly and concisely**  ***Evidence of Proficiency Requires:***   * Purpose of communication is clear * Improvements are fully but concisely described   + All steps are included   + Appropriate technical language is used   + Clarifying images (e.g., sketches, graphs and/flow charts) are provided (as necessary) * External research is accompanied by an in-text citation and full reference |
| **Learning Objective (LO): 21.03 Evaluate model or algorithm development (e.g. ideas, work, functionality) using evidence-based rationales**  ***Evidence of Proficiency Requires:***   * Assumptions, claims, and critical decisions are clearly stated * An appropriate source of evidence is used to support assumptions, claims, and critical decisions * The evidence is clearly articulated * External research is accompanied by an in-text citation and full reference |

Briefly describe, in words (not code), the nature of the refinements you will implement in your MATLAB code. Provide a brief, but thoughtful, description of your refinement, *using evidence-based rationales for why the refinement is necessary and should improve your solution*.

|  |
| --- |
| **Refinement 1. Category 1: Parameter(s) Targeted: \_\_**Tau, Ts, & SSE**\_\_\_** |
| Description  Based on how our algorithm calculated the Tau, Ts, and SSE values before we can observe that as a model number gets more efficient, our algorithm gets less accurate. This is most likely due to the rate at which the data gets from yH to yL and vice-versa. Because our algorithm smooths 200 points at a time, and creates a vector of those smoothed chunks, the vector of smoothed data we have to work with is only about 51 points big. This means that even if you move one point at a time when finding the average slope in order to determine the location of ts, you will still likely be off by a large margin. This is because the distance between each point of the smoothed line of data, length-wise is equivalent to roughly 200 points of the non-smoothed data. This refinement will take smaller chunks of the data (about chunks of 20 points) and use them instead to eliminate any errors that could happen because of this. |
| Rationale for Refinement  By smoothing chunks of 20 points instead of 200 points, we increase the amount of points our smoothed curve has by a magnitude of 10. Meaning that we will be able to more accurately determine the exact point where Ts is located by using a moving average. There are two issues with this refinement however. The first issue is that when we make these changes, we will have to use the data we obtain when smoothing is all done for the rest of the algorithm, or else our index dimensions will not match. This means that we will have to alter a large portion of our variables in the algorithm to make the algorithm work. The second issue is that the run time will increase by a huge amount. This cannot be avoided because we are essentially making our algorithm smooth and compute ten times as many points as we have been before. meaning that our runtime will increase by roughly 10 or so times as well. (Probably a bit less than that due to small loops existing outside of our main smoothing loop.) |

|  |
| --- |
| **Refinement 2. Category: \_\_**2**\_\_\_** |
| Description  Our algorithm in its milestone 3 state actually ran pretty fast, however, with the addition of our first refinement, this is no longer the case. The algorithm for Milestone 3 runs fully (For a single set of M2 data) in about .08 seconds. However, because our refinement has our algorithm assess/smooth ten times the amount of data, it will run for longer than in its initial state. This code runs for around .5 seconds when run with one dataset. Due to the nature of the refined version of the algorithm, it isn’t possible to get below the milestone 3 version due to the fact that it is doing 10 times the amount of work as the milestone 3 algorithm. However, we can still speed it up a bit. It would be a good idea to do so because when running multiple sets of data, this .5 seconds really adds up. We can improve runtime by preallocating vectors that are constantly changing size before the processes with said vectors are actually run. Also, we might be able to shorten and even remove for loops in the algorithm. |
| Rationale for Refinement  Removing unnecessary for and while loops causes the algorithm to speed up considerably due to the fact that MATLAB doesn’t have to do the same processes multiple times in a row before moving on. Additionally lowering the amount of code that lies within a for or while loop will also make the code run faster because there is less code to repeat multiple times. Finally, preallocating vectors that constantly change size with the zeros command will decrease run time a lot because instead of creating an entirely new vector with the same size as the vector that has just changed size, MATLAB just has to replace a zero in the preallocated vector with the calculated value. |

**Part 2: Refinements**

Resave all M3 files as **Project\_M4Exec\_*sss*\_*tt*.m**, **Project\_M4Algorithm\_*sss*\_*tt*.m**, and **Project\_M4Regression\_*sss*\_*tt*.m** before starting to make refinements.

**Refinement Category 1: Parameter Identification (*Required*)**

Making all necessary refinements to your M3 algorithm in your **Project\_M4Algorithm\_sss\_tt.m** file.Refinements must be clearly commented in your code with the text “Category 1” AND an adequate description.Then evaluatethe improvement in your refined parameter identification algorithm. Use the clean and noisy calibration data from M2 and compare the parameters identified from the calibration data using the algorithm you submitted as your solution for M3 and your refined algorithm for M4. Report your results in Tables 1 and 2. Take care with units and decimal places when presenting results.

**Table 1. Algorithm performance comparison to HEATING calibration parameters**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | | **HEATING** | |
| Parameter | M3 Algorithm | | M4 Algorithm | | Actual | |
| Clean | Noisy | Clean | Noisy | Clean | Noisy |
| yL | 0.25 oC | -1.62 oC | 0.25 oC | -2.25 oC | 0.25 oC | -0.96 oC |
| yH | 99.38 oC | 97.61 oC | 99.38 oC | 98.42 oC | 99.38 oC | 98.75 oC |
| ts | 0.88 sec | 1.86 sec | 1.32 sec | 1.83 sec | 1.37 sec | 1.84 sec |
| τ | 0.70 sec | 1.23 sec | 0.26 sec | 1.30 sec | 0.21 sec | 1.35 sec |
| SSEmod | 78.25 sec2 | 3.78 sec2 | 2.29 sec2 | 3.42 sec2 | .0032 sec2 | 2.65 sec2 |
| ***Note:*** Verify your SSEmod calculation. Heating Actual Clean SSEmod should be between 0.00 and 0.05 sec2; Heating Actual Noisy SSEmod should be between 2.30 and 2.90 sec2 | | | | | | |

**Table 2. Algorithm performance comparison to COOLING calibration parameters**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | | **COOLING** | |
| Parameter | M3 Algorithm | | M4 Algorithm | | Actual | |
| Clean | Noisy | Clean | Noisy | Clean | Noisy |
| yH | 100.06 oC | 97.39 oC | 100.06 oC | 98.74 oC | 100.06 oC | 97.99 oC |
| yL | -0.87 oC | -1.52 oC | -0.93 oC | -2.37 oC | -0.93 oC | -0.01 oC |
| ts | 1.27 sec | 1.86 sec | 1.03 sec | 1.90 sec | 1.02 sec | 1.93 sec |
| τ | 1.27 sec | 1.85 sec | 2.07 sec | 1.03 sec | 1.92 sec | 1.03 sec |
| SSEmod | 28.92 sec2 | 98.36 sec2 | 5.47 sec2 | 4.36 sec2 | .53 sec2 | 4.54 sec2 |

Using your M4 algorithm, analyze the 100 time histories provided by FOS, Inc. to identify the four relevant first-order system parameters (yL, yH, ts, and τ) from each time history. In Table 3, copy your results from M3 for the M3 algorithm, and record your results for your M4 algorithm. Take care with units and decimal places when presenting results.

**Table 3. Algorithm performance comparison for FOS designs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model Number** | **M3 Algorithm** | | | **M4 Algorithm** | | |
| **τ Characteristics** | | **Mean SSEmod** | **τ Characteristics** | | **Mean SSEmod** |
| **Mean** | **Standard Deviation** | **Mean** | **Standard Deviation** |
| FOS-1 | 0.87 sec | 0.36 sec | 131.75 sec2 | 0.21 sec | 0.04 sec | 3.99 sec2 |
| FOS-2 | 0.93 sec | 0.37 sec | 86.09 sec2 | 0.41 sec | 0.04 sec | 2.29 sec2 |
| FOS-3 | 1.15 sec | 0.30 sec | 30.13 sec2 | 0.90 sec | 0.07 sec | 1.81 sec2 |
| FOS-4 | 1.27 sec | 0.22 sec | 74.54 sec2 | 1.28 sec | 0.08 sec | 1.70 sec2 |
| FOS-5 | 1.57 sec | 0.20 sec | 10.70 sec2 | 1.83 sec | 0.11 sec | 2.44 sec2 |

As necessary, make improvements to your price versus time constant (τ) regression model in **Project\_M4Regression\_sss\_tt.m**. Complete the price versus tau regression analysis on the 100 data sets using your M3 algorithm and your M4 algorithm. Generate a regression plot for your M3 algorithm and your M4 algorithm. Report the results of each model in Table 4.

|  |
| --- |
| **Learning Objective (LO): 12.00 Perform linear regression** |
| **Learning Objective (LO): 13.00 Perform function discovery and data transformations** |
| **Learning Objective (LO): 07.00 Create and evaluate x-y plots suitable for technical presentation (this includes all appropriate sub-LOs)** |

|  |
| --- |
|  |

|  |
| --- |
|  |

**Table 4. Algorithm performance comparison for**

**price versus time constant regression models**

|  |  |  |
| --- | --- | --- |
| **Regression Result** | **M3 Algorithm** | **M4 Algorithm** |
| General Equation | y = 7525.821 \* 10^3.046x | y = 34.997 \* 10^1.289x |
| SSE | 23.34 | 2.78 |
| SST | 35.12 | 35.46 |
| r2 | 0.34 | 0.92 |

**Refinement Category 2: Algorithm Efficiency**

***If you have refined the efficiency of your code***, complete Table 5 below to show the effects of your refinements. Use the MATLAB built-in functions **tic** and **toc** to measure how long it takes your code to execute. *Efficiency refinements must be clearly commented in your code with the text Category 2 AND adequate description.**Do not remove code; comment out unnecessary code and comment on the change. New code must be designated as such.*

**Table 5. Efficiency measurement results.**

|  |  |
| --- | --- |
| **Algorithm** | **Execution Time (sec)** |
| **M3 Algorithm** | .08 sec |
| **M4 Algorithm Before Refinement 2** | .50 sec |
| **M4 Algorithm After Refinement 2** | ..43 sec |

**Refinement Category 3: Algorithm Insight**

***If you have refined the robustness and performance of your algorithm*** in light of changes in a thresholding or other variable hardcoded in your algorithm, create one or more plots that illustrate the insights you have gained. The plot(s) should be suitable for technical presentation and clearly illustrate the effect of changes on the parameter identification and/or regression results. Write a paragraph that complements the plot(s). This paragraph must clearly describe changes to the thresholding or other variables hardcoded in your algorithm and the insights you gained. *The variables used in this analysis must be clearly commented in your code with the text Category 3 AND adequate description.*

*If you need guidance or other suggestions about how to execute this refinement, be sure to ask the teaching team*.

|  |
| --- |
| **Learning Objective (LO): 07.00 Create and evaluate x-y plots suitable for technical presentation (this includes all appropriate sub-LOs)** |

|  |
| --- |
| <*insert your ‘insight’ plot(s) here*> |

|  |
| --- |
| **Learning Objective (LO): 21.02 Communicate ideas clearly and concisely** |

|  |
| --- |
| **Description of Insights Gained**  <*write description here*> |

**Technical Brief Draft**

|  |
| --- |
| **Learning Objective (LO): 21.02 Communicate ideas clearly and concisely** |
| **Learning Objective (LO): 21.03 Evaluate model or algorithm development (e.g. ideas, work, functionality) using evidence-based rationales** |
| **Learning Objective (LO): 07.00 Create and evaluate x-y plots suitable for technical presentation (this includes all appropriate sub-LOs)** |

Resave ENGR132\_Sp17\_FOS-Project\_M4\_TechnicalBrief\_Template.docx as **Project\_M4TechnicalBrief\_*sss.tt*.docx**. Use this template to respond to the M4 FOS memo. You may find the original M1 FOS memo and project introduction materials useful when composing your technical brief.

M4 References Used in Evidence-Based Rationales

|  |
| --- |
| 1. 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> 2. Filtering and Smoothing Data. (n.d.). Retrieved April 6, 2017, from <https://www.mathworks.com/help/curvefit/smoothing-data.html> 3. UNSW : Goodness-of-Fit Statistics. (n.d.). Retrieved April 06, 2017, from <http://web.maths.unsw.edu.au/~adelle/Garvan/Assays/GoodnessOfFit.html> 4. Jacobs, D. (2013, August 27). CSS Architectures: Principles of Code Cleanup. Retrieved April 06, 2017, from <https://www.sitepoint.com/css-architectures-principles-of-code-cleanup/> 5. Techniques to Improve Performance. (n.d.). Retrieved April 6, 2017, from <https://www.mathworks.com/help/matlab/matlab_prog/techniques-for-improving-performance.html> 6. 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> 7. 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. |