Project Milestone 4 – Algorithm Refinement Answer Sheet

Instructions

1. Read this document carefully. You are responsible for following all instructions in this document.
2. Read the Learning Objectives at the end of the document to understand how your work will be graded.
3. Use professional language in all written responses and format all plots for technical presentation. See EPS01 and EPS02 for guidelines.
4. Good programming standards apply to all m-files.
5. Submit deliverables to Gradescope and to Blackboard. Name your files to match the format in the table below, where *SSS_TT* is your section and team ID (e.g., 001_03 is Section 001, Team 3)

Item	Deliverables
M4 Answer Sheet	Project_M4_AnswerSheet_*SSS_TT*.docx
M4 Algorithm	M4_Algorithm_*SSS_TT*.m
M4 Regression model	M4_Regression_*SSS_TT*.m
M4 Executive Function	M4_exec_*SSS_TT*.m
Technical Brief Draft	M4_TechnicalBrief_*SSS_TT*.docx
Gradescope Submission	M4_*SSS_TT*.pdf

   See submission requirements on the last page of this answer sheet.
6. Complete the Assignment Header before starting the answer sheet.

Assignment Header

Section and Team ID (SSS_TT):	001_30

Team Member Name	**Purdue Career Account Login**
Alex Norkus	anorkus
Surya Manikhandan	smanikha
Julius Mesa	jmesa
Vincent Lin	lin971

Part 0: M3 Feedback Review

Reflect on your M3 feedback for the purpose of improvement. Your reflection should provide a clear, useful summary of your M3 feedback and provide a clear and practical plan to address the issues. Complete table 1 below.

Table 1. Feedback summary and plan

Part A: Summarize the feedback you received on M3 that could lead to improvements in your work.

As part of our M3 grading, we were given many potential avenues for improvements which we plan to implement in this particular milestone document. Each feedback here corresponds to the same number on the implementation

1. We needed significantly more elaboration when we are explaining the effects of our improvements on parameter identification (v0, Vmax, and Km).
2. We needed to elaborate how our executive function worked and also use appropriate technical language to describe the process.
3. A significant piece of feedback we received was to further explain how our improvements affected the specific parameters we are examining (v0, Vmax, Km).
4. Although we have followed the same method of citation in the previous milestones without any issues, we were counted off on the references in this milestone. This likely has something to do with the in-text citation according to a Peer Teacher's comment.
5. Regarding the M3 Regression Assignment, we had incorrect function call and we didn't have units in the explanation of the input data, which may cause mistakes when using the function.
6. We received multiple programming standards violation comments. These range from errors in the function header, having outputs on the executive function, not properly calling functions, etc.
7. We had incorrect values for SSE and SST on the regression function.
8. The header for the executive function needed to better explain how the regression function was implemented into the executive function.
9. The executive function produced an output when we were not suppose to put any input or output included in the function.
10. Linear regression equation was not displayed on the graph.

Part B: Explain how you will incorporate the M3 feedback to improve your parameter identification (do not just reword your response from Part A).

The methods for implementation shown below correspond to the feedback item in the cell above.

1. We will look over our document after we are done with it to ensure that we have connected how our changes have affected the parameter identification and explain in as much detail as possible. We can obtain 3d party verification by asking another team to peer-review our document for thoroughness. To show that we have improved our findings to v0, Vmax, and Km we will compare how much percent better our new SSE values compared to that of the PGO-X50 Reference Values AND the values we produced in the previous iteration of the program.
2. To change our technical language to be more suitable to class standards, we will use a thesaurus to better describe our thoughts for either changes or explanations we are trying to make in the document.
3. We will specifically state the parameter being improved, and elaborate on how the changes we made affect that specific parameter mathematically. We will make references to formulas where possible to better describe this.

4. To have a more formal citation, we will exclusively use APA in-text citations and references. We will ensure this by double-checking our citations with Purdue Online Writing lab, a fantastic resource for technical writing.
5. After inspecting the grading, it seems we were counted off for not including the pricing in our executive function, which led to the incorrect function call deduction. Therefore, we will address this by calling the regression function with our exec function
6. We will very closely refer to the course programming standards to check that our programs are in full compliance.
7. We will examine how the SSE & SST were calculated in our M3 submissions and attempt to find the mistakes we made in the calculations. We will fix these errors in calculations as we see fit. EDIT: The issue was the SSE was being calculated with the linearized data, which is incorrect. WE will change the code to calculate the SSE with the non-linearized data.
8. The regression function was NOT integrated in the executive function in our initial submission. We will fix this by ensuring the executive function will call the regression function appropriately.
9. The executive function will NOT use output arguments anymore. Instead, we will print out all of the relevant information directly to the command window, which we were told is the correct solution instead of the output arguments used before. This way, we can also make it more user friendly by stylizing the output.
10. We will read MATLAB documentation in order to find out how to display text on the graph. Then, we will implement a solution in order to display the proper equation on the chart.

Part 1: Algorithm Refinements Plan

Respond to each of the prompts below in the space provided. Your goal is to introduce **two refinements** to your M3 algorithm, and these refinements must improve your solution to the NovelEnzymes 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:

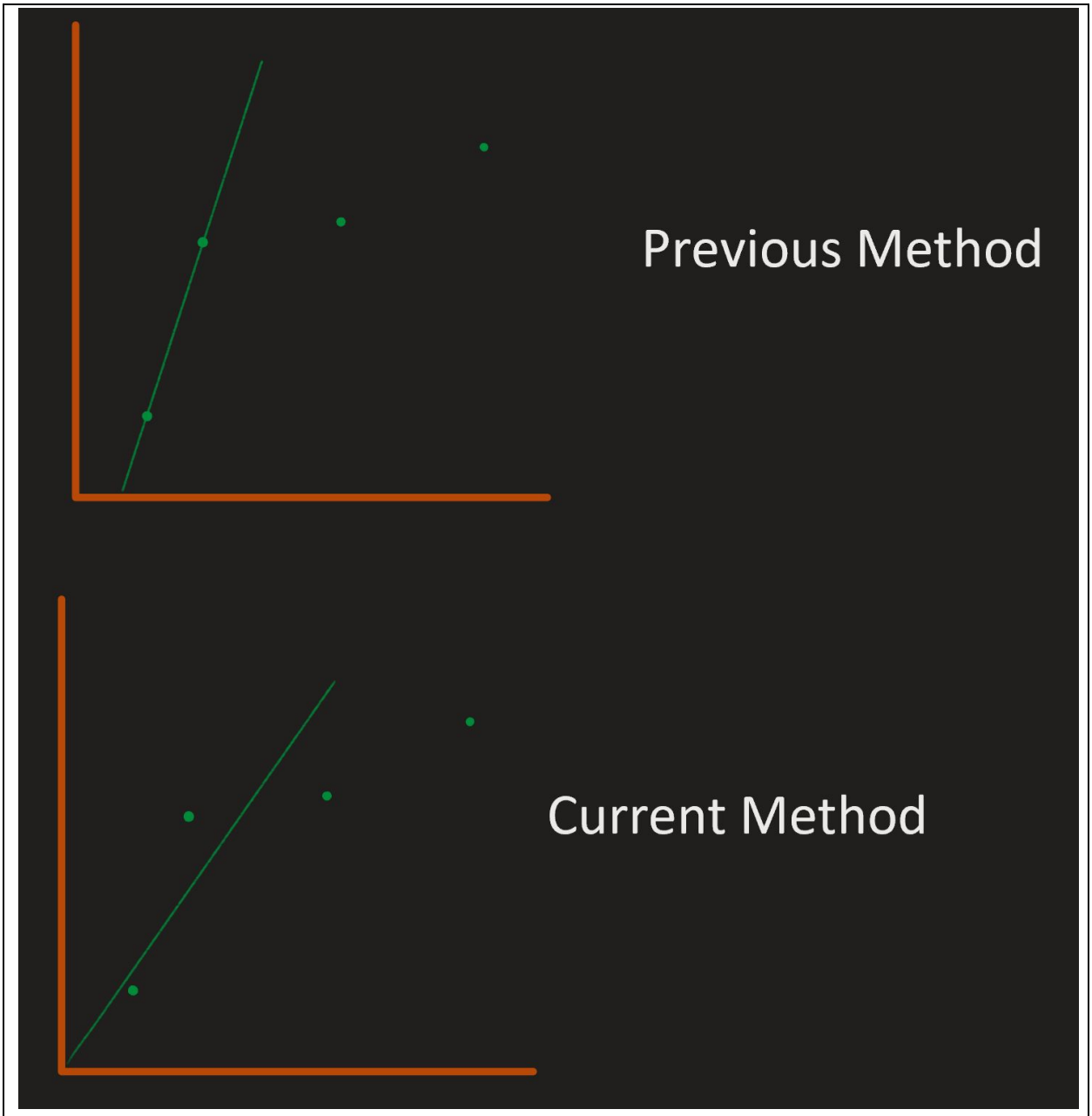
- **Refinement Category 1: Parameter Identification (Required)**
  An improvement that changes the way you are doing parameter identification, and that improves your parameter identification results.
- **Refinement Category 2: Algorithm Efficiency**
  An improvement that improves the efficiency of your code by (for example) removing unneeded 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 **two** refinements. One refinement must be in Category 1. The second refinement can come from either Category 2 or Category 3. Use your ideas from

Part 3 of M3 to help formulate ideas. 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.

Table 2. Algorithm refinement plans

Refinement 1: Category 1. Parameter Identification
Parameter(s) Targeted: *V0* Description *Our Algorithm uses the tangent line method in order to calculate v0 at each substrate concentration. Essentially, the algorithm starts by plotting product concentration [P] in uM on the Y axis and the time in minutes on the X-axis. Before this plotting step, the raw data for [P] vs time is smoothed using the moving-average method discussed in detail in the previous milestone documents.* *Once the smoothed data has been plotted, the algorithm will then calculate v0. Previously, this was done taking the first two points form the plot and using the formula rise/run in order to calculate the slope, which is our v0 value. Although this worked fairly well, the datasets given in this assignment had significantly more noise and sometimes systematic bias.* *Therefore, we will be implementing a new system for calculating the initial slope and therefore v0. M4 Algorithm will pick the first 3 smoothed data points and find the linear regression of those points. The intercept for the regression has been forced to the (0,0) origin point.*
Rationale for Refinement *This improvement is necessary because the datasets given to us in this assignment have significantly more noise and therefore bias. For example, some datasets had spikes in the middle, while others had increasing variability as time went on.* *For this particular assignment, it was clear that our method of choosing the first two points for the slope was not a great idea as it was being affected by the bias in the datasets, particularly at the lower substrate concentrations. The increased noise had started to affect the smoothness of the curve and therefore the integrity of the first two points. This means that the first two points being used to draw the tangent lines were not producing an acceptable tangent line, even though out smoothing function has done its best to iron out the noise in the data.* *Using three points allows us to be better resistant to bias in the dataset and produce a significantly better tangent line as it provides a buffer rather than just the first two points. Fitting a best fit line to these first 3 points allows us to make a much better tangent line than just the first two points. While it is hard to explain, the diagram below shows why we believe this is a great idea.*



As shown in the picture above, the two point method is easily fooled if either of the points are not perfectly on the tangent vector, which is an assumption we made for the previous milestone. Additionally, we noticed that the tangent lines do not pass through the origin in almost all instances when using this method, which produces an inaccurate tangent line as the raw data always starts off at the origin. Our improved 3-point method is able to account for any unexpected variation in the data points which would have been a significant threat to our previous algorithm, and is therefore superior.

Refinement 3: Category 2. Algorithm Efficiency

Parameter(s) Targeted: *Execution Time*

Description

Given the extensive amount of data needed to be imported in the code (>7000 unique lines with over 100 columns), increasing the efficiency of the data imports would shave off significant amounts of time. Therefore, we will have to explore alternative methods of importing data OR examine ways to cut down the amount of information we ae importing (just a few rows instead of the whole column, for example).
Rationale for Refinement *According to the MATLAB Profiler feature, the M3 code spent more than 90% of the time importing the data. Specifically, it took a significant amount of time to import small data sets using read matrix, which seems contradictory. For example, It took longer for the code to import the 100 substrate concentration [S] values than it did to read in all the thousands of [P] values. We ran tests of several import functions such as readmatrix, csvread, dlmread, and xlsread on a small dataset (100 elements to simulate the [S] values). This showed that for small datasets, the xlsread function was superior compared to the others in execution time. However, when trying to import the complete m4 dataset, readmatrix was the winner. Therefore, we will be using xlsread wherever we are importing small amounts of data such as [S], and readmatrix for importing large data such as [P].*

Part 2: Algorithm Refinements Implementation

Before you make any changes to your code, resave your M3 code files as

- M4_Algorithm_*SSS_TT*.m
- M4_Regression_*SSS_TT*.m
- M4_exec_*SSS_TT*.m

Category 1 Refinement (Required)

Implement your Category 1 refinements in M4_Algorithm_*SSS_tt*.m. Clearly comment the refinement changes within the code, using the text 'Category 1' and a concise, meaningful description of the change.

Evaluate the improvement in your algorithm by using the clean and noisy data for the reference enzyme PGO-X50 from M2. Compare the parameters identified for the PGO-X50 data using the algorithm you submitted in M3 and your refined algorithm for M4. Report your results in Table 3. Take care with decimal places.

Table 3. Algorithm refinement comparison

Parameter (µM/min)	PGO-X50 Reference Values		M3_Algorithm		M4_Algorithm	
	Clean	Noisy	Clean	Noisy	Clean	Noisy
v_{0_1}	0.028	0.028	.0283	.0305	.0281	.0282
v_{0_2}	0.056	0.055	.0567	.0572	.0562	.0570
v_{0_3}	0.110	0.11	.1121	.1022	.1112	.1111
v_{0_4}	0.193	0.19	.1959	.1657	.1944	.1979
v_{0_5}	0.360	0.338	.3674	.3546	.3650	.3591
v_{0_6}	0.6	0.613	.6059	.6395	.6026	.6060
v_{0_7}	0.883	0.917	.8915	.9439	.8878	.8790

v_{0_8}	1.212	1.201	1.2192	1.2364	1.2157	1.1796
v_{0_9}	1.376	1.282	1.3806	1.2812	1.3783	1.3269
$v_{0_{10}}$	1.584	1.57	1.5873	1.5274	1.5858	1.5788
V_{max}	1.72	1.61	1.7605	1.6782	1.7604	1.7375
K_m (µM)	226.92	214.28	235.8686	224.6648	237.8632	237.4837
SSE (µM/min)^2	0.0041	0.0251	0.0301	0.0224	0.02826	0.0079

* Verify your SSE values by comparing them to the provided SSE values for the reference parameters.

Next, use your M4 algorithm to analyze the full 100 enzyme test data sets and obtain the parameters V_{max} and K_m. In Table 4, copy your enzyme parameter and model goodness of fit results from M3 (i.e., the values from M3 Table 3) and record your results from your M4 algorithm. Take care with decimal places.

Table 4. M3 and M4 algorithm comparison of experimental data parameters

Enzyme	M3 Algorithm			M4 Algorithm		
	Enzyme Parameters		SSE (µM/s)^2	Enzyme Parameters		SSE (µM/s)^2
	V_{max} (µM/s)	K_m (µM)		V_{max} (µM/s)	K_m (µM)	
NextGen-A	0.9174	155.0399	.0003	.9737	150.3015	.00016
NextGen-B	0.8988	355.4164	.0025	.8763	337.2487	.00195
NextGen-C	1.2349	187.2996	.0010	1.2471	185.0924	.00002
NextGen-D	1.6574	304.0967	.0001	1.6277	288.6323	.00025
NextGen-E	1.7222	179.1636	.0010	1.7027	167.2828	.00065
% decrease in SSE from M3 to M4 (Average)						84.17%

Category 2 Refinement (Optional—Choose one of either Category 2 or 3)

Implement your Category 2 refinements to increase the efficiency of your algorithm. Run your code and document the effect of the refinements. Use the MATLAB functions `tic` and `toc` to measure how long it takes your program to execute. Efficiency refinements must be clearly commented in your M4 code with the text 'Category 2' and a concise, meaningful description.

Do not delete any code as you implement the refinements: comment out unnecessary code and comment on the change. New code must be designated as such.

Record the execution time of your M3 program and your refined M4 program in Table 5.

Table 5. Program execution times

Program	Execution Time (s)
M3 (before refinement)	9.321
M4 (after refinement)	2.760
% reduction in execution time	108.604%

NOTE: Time Measurements were done with laptopped unplugged and the graphics card disabled. Laptop was running in the 'balanced' power configuration to prevent CPU turbo boost.

Category 3 Refinement (Optional—Choose one of either Category 2 or 3)

After refining 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 other 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 a concise, meaningful description.

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

Table 6. Algorithm insight results

Refinement 3 Insight Plot(s)
<insert plot(s) here>
Description of Insights Gained
<write description here>

Part 3: Enzyme Pricing Recommendations

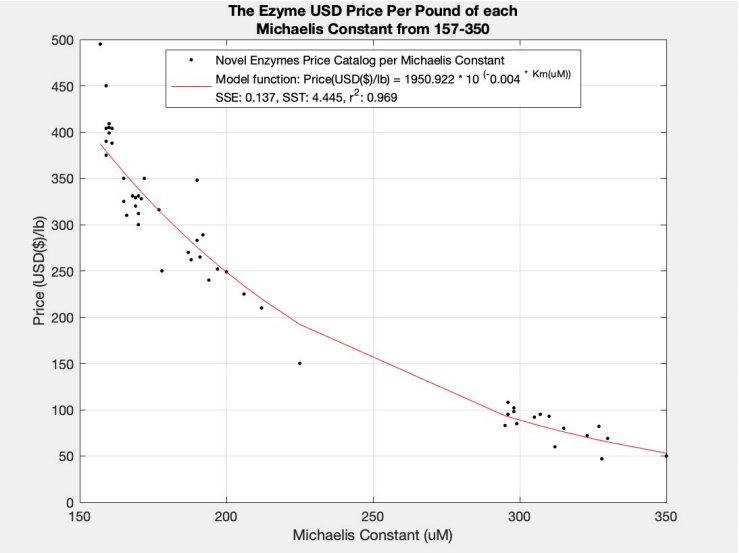
Now that you have implemented your algorithm refinements, you have a set of parameters with which to determine an initial price point for each of the five enzymes. Considering the parameters found by your M4 algorithm, decide what changes, if any, you need to make to your regression algorithm. These could be corrections, response to M3 feedback, or changes to model function. Implement changes as needed.

Report the regression model information in Table 7. Take care with decimal places.

Table 7. Regression results

Parameter	Value	Plot

Model	Exponential Function
SSE ($\$^2$)	0.137
SST ($\$^2$)	4.445
r^2	0.969



Using your model, provide a price for each enzyme. Report your values in Table 8. Add an indicator to any extrapolated value.

Table 8. Price prediction for each enzyme using regression model

Enzyme	**Price ($/lb)**
NextGen-A	414.94
NextGen-B	60.51
NextGen-C	289.99
NextGen-D	99.83
NextGen-E	348.36

In Table 9, justify your regression function selection and any extrapolation required to provide a using evidence-based rationales.

Table 9. Regression model justification

The regression function linearizes measured price using four different kinds of equations, and then calculates SSE value, find the lowest SSE value which indicates the best equation to model the relationship between price and Km value of the enzyme. After running the function, we got the SSE,SST and r^2 value as shown in the chart below.

- By using power equation and exponential equation, we can model the data with a low SSE value of 0.1397(power) and 0.1367(exponential), which indicates these two equations best fit the data.
- We also take r^2 value into consideration to make sure that the model can explain the percent of the variability in the data well. While r^2 value of power equation(0.9686) is very close to the r^2 value of exponential equation(0.9693), it's hard to decide which model to choose.
- What we consider further is that if the model is reasonable in reality and if the model predicts a reasonable price for extreme Km value. When the Km value is close to 0, the exponential function predicts that the price would be 1950.92$/lb, while the power function predicts the price to be ∞ (indefinite). After some research, the maximum price of bulk detergent enzyme is 575$/lb. (Thermo Scientific™, 2011) , so it is unreasonable that the price will reach unlimited. Thus, we chose the exponential equation to model the price.

No extrapolation was done in finding the price values for the enzymes. In the reference data, the minimum and maximum Km values being 150 and 350, respectively. While estimating the price, the minimum Km value is 150.3015 microMoles for Enzyme-A, and the maximum Km value is 337.487 microMoles for Enzyme-B. Since both values are within the range of 150 and 350, no extrapolation is needed while estimating the price for all 5 enzymes.

	SSE	r^2
Linear equation	**80307**	**0.903**
Power equation	**0.1397**	**0.9686**
Exponential equation	**0.1367**	**0.9693**
Logarithmic equation	**58649**	**0.929**

	SSE	r^2
Linear equation	**80307**	**0.903**

Power equation	**0.1397**	**0.9686**
Exponential equation	**0.1367**	**0.9693**
Logarithmic equation	**58649**	**0.929**

Part 4: Technical Brief Draft

Consult the M4 memo from NovelEnzymes, Inc. for the details concerning your technical brief. Use the provided template M4_TechnicalBrief_template.docx to respond to the memo. You may find the original introduction memo and the project background documents helpful when composing your technical brief.

Table 10. References used in evidence-based rationales (answer sheet only)

Thermo Scientific B-PER Bacterial Protein Extraction Reagents - BioPharmaceutical Production, Proteomics and Immunoassays. (n.d.). Retrieved from https://www.fishersci.com/shop/products/thermo-scientific-pierce-b-per-bacterial-protein-extraction-reagents-reagent-165ml/pi78243#?keyword=true.

How to Submit

1. Rename this answer sheet to be **Project_M4_AnswerSheet_*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). Save the answer sheet as a PDF.
2. Create a PDF of your technical brief document.
3. Publish each function you created in this milestone. Note: you may publish sub-functions (such as M4_Algorithm_*SSS*_*TT*.m) without filling in appropriate inputs or outputs – this will generate an error in the published code for that function that will not affect your grade. Note: these functions must run properly when called by your executive function.
4. Merge the answer sheet PDF, the technical brief PDF, and the published code PDFs into one PDF file named **M4_*SSS*_*TT*.pdf**.
5. Select one person to submit the PDF for the team. That person should
    a. Log into Gradescope and submit **M4_*SSS*_*TT*.pdf** to the **M4** assignment.
    b. Indicate which pages correspond to each part of the milestone.

    Failure to tag the items appropriately will result in your work receiving losing credit.

    c. Select all team members for the group assignment.
    d. Double-check that all team members are assigned to the submission.
6. Select one person to submit all m-files, data files, and answer sheet to the **M4 dropbox** on Blackboard.

    Failure to submit your files to Blackboard may result in every team member receiving a zero on this assignment.

7. Each team member should confirm that they are part of the submission and that all parts of the answer sheet were properly tagged.
8. After submission, distribute the submitted files to all team members. *Ensure all members of the team have copies of the submitted files.*

Learning Objectives

Process Awareness (PA)
Reflect on both personal and team's problem solving/design approach and process for the purpose of continuous improvement.
PA02. Identify limitations in the approach used.
PA03. Identify potential behaviors to improve approach in future problem solving/design projects.

Idea Fluency (IF)

Generate ideas fluently. Take risks when necessary.

IF03. Generate testable prototypes (including process steps) for a set of potential solutions.

Evidence-Based Decision Making (EB)

Use evidence to develop and optimize solution. Evaluate solutions, test and optimize chosen solution based on evidence.

EB01. Test prototypes and analyze results to inform comparison of alternative solutions.

EB03. Clearly articulate reasons for answers with explicit reference to data to justify decisions or to evaluate alternative solutions.

EB05. Present findings from iterative testing or optimization efforts used to further improve aspect or performance of a solution.

EB06. Clearly articulate reasons for answers when making decisions or evaluating alternative solutions.

Solution Quality (SQ)

Design final solution to be of high technical quality. Design final solution to meet client and user needs.

SQ01. Use accurate, scientific, mathematical, and/or technical concepts, units, and/or data in solutions.

Information Literacy (IL)

Seek, find, use and document appropriate and trustworthy information sources.

IL04. Include citations within the text (in-text citations) that show how the references at the end of the text are used as evidence to support decisions.

IL05. Format reference list of used sources that is traceable to original sources (APA or MLA are recommended)

Engineering Professional Skills

PC05. Fully address all parts of assignment by following instructions and completing all work.

EPS01. Use professional written and oral communication.

EPS02. Format plots for technical presentation.

Programming

MAT01. Develop code that follows good programming standards.

MAT03. Perform mathematical operations and calculations within MATLAB.

MAT05. Create and use MATLAB scripts and functions.

MAT08. Debug scripts and functions to ensure programs execute properly, perform all required tasks, and produce expected results.

MOD01. Create mathematical models to describe relationships between data.