# Project Milestone 2 – Algorithm Development

## Instructions

1. Read this document carefully. It provides you with all the requirements needed to complete the M2 Answer Sheet and any coding tasks. You are responsible for following all instructions in this document to complete your work.
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 |
|---|---|
| M2 Answer Sheet | M2_AnswerSheet_*SSS_TT*.pdf |
| M2 Algorithm | M2_main_*SSS_TT_login*.m<br>M2_sub2_*SSS_TT_login*.m<br>M2_sub3_*SSS_TT_login*.m<br>M2_sub4_*SSS_TT_login*.m<br>* If your team has multiple UDF algorithms, append a short description after “sub#” with appropriate descriptor. |

See submission requirements on the last page of this answer sheet.

6. Complete the Assignment Header before starting the answer sheet.

### Assignment Header and Role of Each Team Member

Complete the following on Page 1 of the M2 Answer Sheet.

### Assignment Header

The assignment header must contain the names and Purdue career account username for each team member.

### Programmer Assignment

In the project memo, you were told that each team member will need to be responsible for some aspect of the programming. You already selected Programmer 1 in M1. Everyone else will start programming this milestone.

### Milestone Work Report

In this section, put each team member’s name who worked on this milestone. In the Detailed Description of Work, each person on the team should write their own description of how they contributed to this milestone. Be very detailed here. Then in the last column, your team should estimate the percentage of the work that each team member did on this milestone. This column needs to add up to 100%. We know this will vary on any given milestone, but one person in the team should not be doing significantly more than the others throughout the whole project. Use this column as a way for you to make sure your workload is balanced throughout the project.

## Part 1: M1 Feedback Review

Reflect on your M1 feedback for the purpose of improvement. Your reflection should provide a clear, useful summary of your M1 feedback and provide a clear and practical plan to address the issues.

Complete Table 1 in the M2 Answer Sheet.

## Part 2: Algorithm Development

In Milestone 1, you developed ideas to manage data noise and errors and approaches for identifying each of the first-order parameters for the system. Now, you will take your best ideas from M1 and use them to create an algorithm (which will be four or more user-defined functions) coded in MATLAB to perform full parameter identification on the speed test data.

Your algorithm will contain 4 parts, each one programmed and managed by a different team member. In M1, you selected which roles each team member will fulfill. Your algorithm will consist of these 4 parts.

| Programmer | Programming task |
|---|---|
| 1 | Main function and data visualization |
| 2 | Clean up the data |
| 3 | Find acceleration start time and time constant |
| 4 | Find initial and final speed |

Programmer 1 started the main function in M1 and will continue to work on it in this milestone. The remaining programs will each be new subfunctions that are called within the main function.

### Algorithm Plan

As a team, develop a plan for your algorithm before you start coding. Outline your algorithm using pseudocode (i.e., plain English text, not MATLAB code). Remember, it is valuable to develop and organize your programming ideas and solutions *before* you code. A well-developed plan reduces coding frustrations. Address these items:

- What method(s) will you use to manage data noise and errors?
- What method(s) will you use to determine the acceleration start time, the time constant, the initial speed and the final speed?
- How will the subfunctions operate together? Who needs what values from which functions?
    - Decide how team members can start working on their programs while they wait for outputs from in-development programs.
- Do the data displays need to change or be updated to help the team understand how well the algorithm is working?

Complete your plans in Table 2 in the answer sheet.

## Programming your Algorithm

After you complete your plan, translate your plan into **at least 4** user-defined functions, one per programmer. Any programmer can create more than one subfunction to achieve their goal, but the final algorithm must have 1 main function and at least 3 subfunctions (1 per programmer 2, 3, and 4).

| | |
|---|---|
| Programmer 1: | update your function from M1 to be a main function. Coordinate function calling to use the subfunctions appropriately. |
| Programmer 2: | write at least 1 subfunction that will manage the noise and errors in the data. |
| Programmer 3: | write at least 1 subfunction that will determine the acceleration start time and the time constant. |
| Programmer 4: | write at least 1 subfunction that will determine the initial and final speeds. |

When run, your algorithm should use its main function and all its subfunctions to perform parameter identification on the speed data for each vehicle-tire combination. Your main function should load the data, segment the data, call the subfunctions as necessary, and display any relevant plots or text outputs. The subfunctions should clean the data and calculate the parameters themselves.

Every function in the algorithm (main or subfunction) must follow ENGR 132 Programming Standards and be clearly commented throughout. Any member of your team or of your section's teaching team should be able to easily read and understand what you are doing in your program.

Any team member can help any of their team members with their code, including paired programming if necessary. Teammates should not work alone nor in a vacuum. However, the primary programmer is expected to actually write the code in the program. In the header, the author should be the assigned programmer.

Name your files using this naming format:

| Programmer | Programming task | File naming convention |
|---|---|---|
| 1 | Main function and data visualization | M2_main_*sss_tt_login1*.m |
| 2 | Clean up the data | M2_sub2_*sss_tt_login2*.m |
| 3 | Find acceleration start time and time constant | M2_sub3_*sss_tt_login3*.m |
| 4 | Find initial and final speed | M2_sub4_*sss_tt_login4*.m |

Replace *sss_tt* with your *section_team* number; for example, Team 04 in Section 205 will use 205_04. Replace *login* with your career account login. The login and author name in the header need to match. Follow this pattern if you make extra subfunctions, appending a short descriptor after "*sub#*" to differentiate the subfunctions while also indicating which part of the algorithm it represents.

Do not submit any plots or figures with this milestone.

## Using Functions Not Taught in Class

This project can be completed using only MATLAB commands that have been taught or used in class materials; however, MATLAB has a huge library of built-in functions that can be very useful. You may find some built-in functions that might be useful for your algorithm. If you decide to use a function that was not taught in class, you must document the function name and how it works. Write several sentences to thoroughly explain the function, how it operates, and the underlying theory behind it. Replicating the MATLAB help documentation is neither appropriate nor sufficient.

Failure to demonstrate full understanding of a built-in function that you use in your algorithm will result in point deductions.

Complete Table 3 in the answer sheet. Add rows as needed. Leave blank if you do not use any new functions. Do not include commands from class if you learn new ways to use them (e.g., `plot(x,y,'LineWidth',1.5)`) does not need extra documentation because we use `plot` extensively in class).

If you are using built-in functions that you documented in M1, then you may reuse/adapt that documentation but you must include it in this Milestone.

## Part 3: Algorithm Reflection

In Table 4 on the answer sheet, discuss your choice of algorithm, your process for debugging your algorithm, and the strengths and limitations of your algorithm at this point. See the directions in each part of the table.

## How to Submit

1. Save the answer sheet as one PDF named **M2_AnswerSheet_*SSS_TT*.pdf**.
2. Gather all your submission files and deliverables in one folder.
3. Select one person to submit all files for the team. That person should
    a. Log into Gradescope and submit all these files together to the **M2** assignment:
        i. M2_AnswerSheet_*SSS_TT*.pdf
        ii. All functions (main and subfunctions). M-files only! No ASV or temporary files.
    b. Select all team members for the group assignment.
    c. Double-check that all team members are assigned to the submission.
4. Each team member should confirm that they are part of the submission. Everyone received an email when they were added. You will lose points if you do not include all teammates in the submission.
    a. You will see “Autograder” information when you view your submission. Select “Code” in the upper right. That will show all your submission files. The autograder feature is not enabled for this project.
5. After submission, distribute the submitted files to all team members. *Ensure all members of the team have copies of the submitted files.*
6. If you need to resubmit anything for any reason: you must **resubmit ALL files** for the assignment. For example, you realized that you did not properly name your m-file. You fix it. Then you resubmit the updated m-file, the answer sheet, and all other files that are part of your submission.

## Learning Objectives

**Teamwork (TW)**
Contribute to team products and discussions
TW02. Document all contributions to the team performance with evidence that these contributions are significant.

**Process Awareness (PA)**
Reflect on both personal and team's problem solving/design approach and process for the purpose of continuous improvement.
PA01. Identify strengths in problem solving/design approach.
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.
EB03. Clearly articulate reasons for answers with explicit reference to data to justify decisions or to evaluate 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.

**Engineering Professional Skills**
PC05. Fully address all parts of assignment by following instructions and completing all work.
EPS01. Use professional written and oral communication.

**Programming**
MAT01. Develop code that follows good programming standards
MAT08. Debug scripts and functions to ensure programs execute properly, perform all required tasks, and produce expected results.