Project 1 - CSC 113 Introduction to MATLAB, 2017 Fall ver3

Engine Control Unit Analysis Project

In this project, you will be writing code to analyze and visualize information captured by the engine control unit (ECU) of a high performance vehicle. This data is logged while the vehicle is being driven on a professional track. The data is utilized by vehicle manufacturers and professional racing companies to analyze and improve the performance of their vehicles. The data provided for this project is taken from actual test laps from a professional racing vehicle.

1 Project Requirements

Your code must first load an input ECU dataset file into MATLAB which contains all the data about the vehicles. Next, your code will analyze the data and generate the following types of output: 1) *printed output to the Command Window* and 2) *three plots in Figure Windows*.

Example Run

The following example run shows the output for ONE particular dataset input file. Note, however, that your script is expected to work with other input files of the same format.

The script begins by asking the user for the name of the dataset input file. In this example, the user picked the *test01.csv* input file. Next, the input data is analyzed and the following *printed output* is seen in the *Command Window*.

```
Command Window
Input the name of the data file you would like to analyze: test01.csv

The maximum RPM was 7627.1187, and it occured at 110.80 seconds.
The maximum coolant temperature was 175.8931 degrees Fahrenheit, and it occured at 348.20 seconds.
The duration for the test was 5.80 minutes.
The average battery voltage was 14.4247 volts.
The minimum oil pressure was 29.8697 psi, and it occured at 187.39 seconds.
The average coolant temperature was 157.0955 degrees Fahrenheit.
>>
```

In addition, the following three plots in *Figure Windows* are generated:

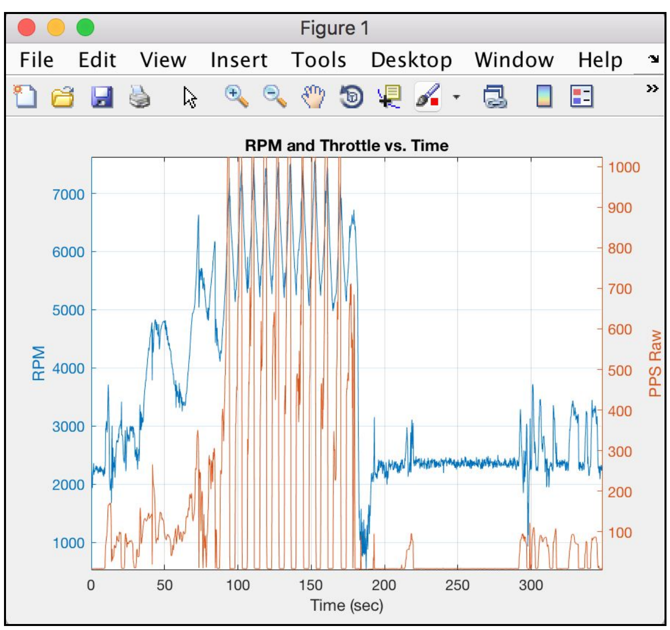

Figure 1

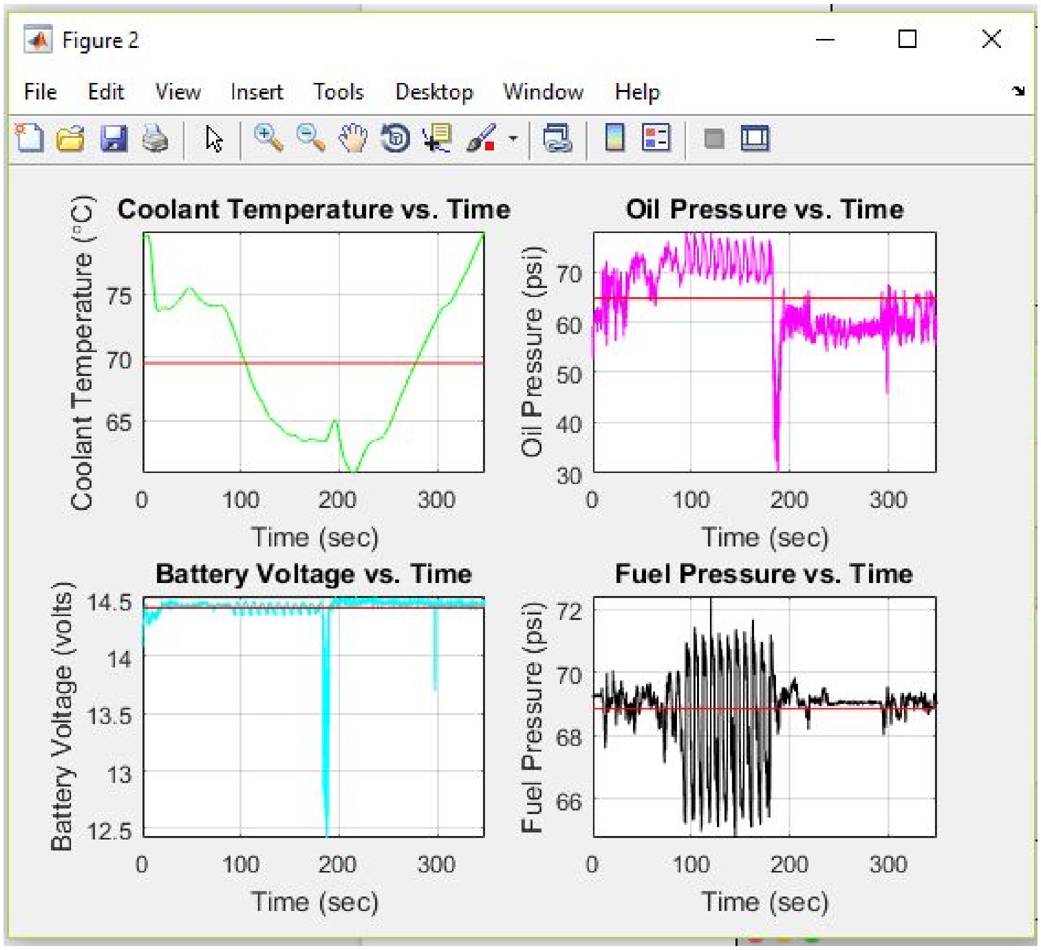

Figure 2

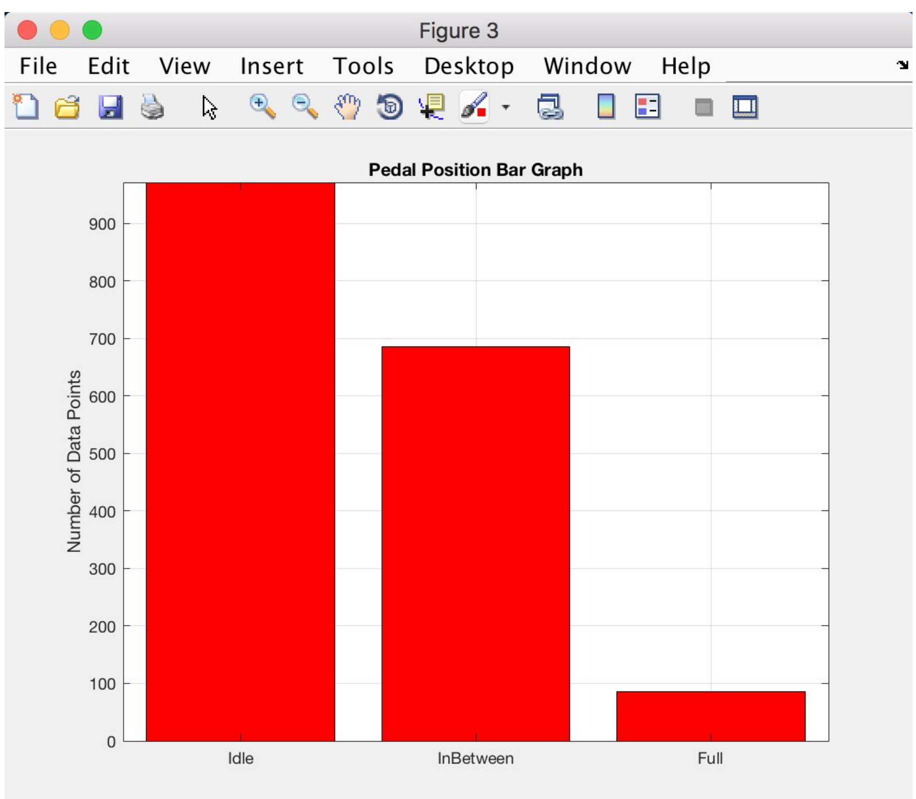

Figure 3

2 Design

Here are the detailed steps to help you complete this project in MATLAB.

2.1 Script Name

Your code for this project shall be in a single script named **ECUanalyzer.m**. You will lose points if you script is named differently. Capitalization is important too.

2.2 Load the Input Data

Your script shall work for any input dataset file specified by the user. The datasets will be stored in files with a Comma Separated Value (CSV) format. Three different input datasets are provided on Moodle. The format of the three input files is the same. Figure 4 shows the data in the input dataset file called **test01.csv** for the the first testing session.

	A	B	C	D	E	F	G	H
1	Time	RPM	Fuel Flow	Coolant Temp	Battery	Oil Pressure	Fuel Pressure	Pedal Position
2	[sec]	[RPM]	[lb/hr]	[°F]	[Volts]	[psi]	[psi]	[%]
3	0	3054.575195	21.089233	174.944504	14.079217	59.176651	69.217857	0.911181
4	0.2	2320.544434	20.959568	174.95459	14.243073	55.59544	69.269142	0.895884
5	0.4	2132.499268	21.22575	174.987549	14.283377	52.712982	69.289833	0.874245
6	0.6	1931.662109	19.990753	175.011078	14.27717	53.324123	69.246391	0.849265
7	0.8	1957.543091	19.470047	174.980957	14.288706	54.934582	69.291267	0.877552
8	1	2195.336182	20.033222	175.024857	14.313591	59.065411	69.220047	0.873655
9	1.2	2216.967285	20.382889	175.000824	14.320278	60.198975	69.220024	0.873921
10	1.39	2264.037109	19.816944	175.003967	14.291214	60.186554	69.272514	0.880939
11	1.6	2359.974854	21.117689	175.004578	14.360198	60.153675	69.228905	0.890866
12	1.79	2159.309082	18.46512	175.017853	14.432972	61.214676	69.313362	0.903358
13	2	2243.046631	19.193081	175.011215	14.350362	60.143044	69.319168	0.908284
14	2.2	2206.098633	19.076279	175.014542	14.443615	59.827709	69.273323	0.898453
15	2.4	2215.657227	19.083754	175.06134	14.446705	61.863144	69.244896	0.909496
16	2.59	2263.581543	19.644264	175.122665	14.462564	60.356075	69.258949	0.929035
17	2.79	2275.485352	19.54748	175.112503	14.460859	62.371773	69.25576	0.940664
18	3	2257.676025	19.60445	175.124313	14.451703	60.648632	69.27282	0.900499
19	3.2	2321.981445	20.728014	175.169113	14.426682	60.306805	69.209648	0.907865
20	3.4	2300.613525	19.931345	175.246292	14.433872	62.589275	69.32679	0.915199
21	3.59	2234.803223	19.202211	175.200531	14.432603	61.273373	69.291527	0.89434
22	3.79	2264.264893	19.863548	175.23642	14.423777	59.702362	69.235519	0.914456
23	4	2309.587402	20.35688	175.266449	14.463901	61.980553	69.219193	0.929479
24	4.19	2244.388916	19.152197	175.284515	14.353769	62.650303	69.285263	0.896747
25	4.4	2236.802979	19.508207	175.338837	14.362067	57.949406	69.243286	0.908857
26	4.59	2181.924072	18.774752	175.35202	14.387369	60.597443	69.250282	0.903591
27	4.8	2243.270264	19.247429	175.329788	14.381795	61.33662	69.325684	0.921245
28	5	2278.019531	19.799557	175.320786	14.368338	61.665485	69.269501	0.925653
29	5.19	2210.43335	18.634697	175.302185	14.276844	63.121315	69.223366	0.882769
30	5.4	2216.748779	18.976698	175.301132	14.281901	61.551037	69.274506	0.908123
31	5.59	2216.09375	19.182016	175.313416	14.272912	60.999992	69.267456	0.921295

Figure 4

So first, your code shall obtain **input** from the user to determine which of the three input datasets to analyze. Then, you shall load that specific input file into MATLAB. The input files has the following columns:

- *Time* ***(In seconds)***
- *RPM (Revolutions per Minute)*
- *Fuel Flow* *(In pounds per hour)*
- *Coolant Temp* ***(In degrees Fahrenheit)***
- *Battery* *(In volts)*
- *Oil Pressure* *(In psi)*
- *Fuel Pressure* *(In psi)*
- *Pedal Position (Throttle)* *(Percentage)*

2.3 Analyzing the Data

Utilizing built-in MATLAB functions, find and print to the *Command Window* the following values:

A. The **maximum RPM** recorded for the dataset and the time in seconds at which it occurred.
B. The **maximum coolant temperature** recorded for the dataset and the time in seconds at which it occurred.
C. The **duration** of the entire test **in minutes**.
D. The **average battery voltage** of the dataset in volts.
E. The **minimum oil pressure** recorded for the dataset and the time in seconds at which it occurred.
F. The **average coolant temperature** of the dataset in degrees fahrenheit.

2.4 Calculations

Next, calculate the following values:

A. The **raw pedal position** sensor values. The pedal position values given in the dataset are given in percentage. We would like to calculate the raw values recorded from the 10-bit sensors. The formula to convert the percentages to raw values is given below:

$$\text{Raw Pedal Position} = \text{Round}\left(1024 * \left(\frac{\text{Pedal Position Percentage}}{100}\right)\right)$$

B. The **coolant temperature in degrees Celsius**. The formula for converting from Fahrenheit to Celsius is given below:

$$\text{Celsius} = (\ \text{Fahrenheit} - 32\ ) * (5/9)$$

2.5 Plotting

Next, generate the following plots:

A. A plot as shown by **Figure 1** that contains the **RPM vs time** and the **raw pedal position values vs time**. Utilize the `yyaxis` command. Keep the MATLAB default colors when plotting the values.

B. A plot as shown by **Figure 2** that contains various subplots:
- Subplot 1 will have a **green plot of coolant temperature (IN CELSIUS) vs. time**.
- Subplot 2 will have a **magenta plot of oil pressure vs. time**.
- Subplot 3 will have a **cyan plot of battery pressure vs. time**.
- Subplot 4 will have a **black plot of fuel pressure vs time**.

Each subplot shall have a **red line at the mean value of the dataset**.

C. A **red bar graph** as shown by **Figure 3** that counts the number of data points the throttle is Idle, Full, or InBetween. The threshold for these values is given below:

Idle = **Pedal Position Percentage ≤ 5%**
Full = **Pedal Position Percentage ≥ 95%**
InBetween = **Pedal Position Percentage > 5% and Pedal Position Percentage < 95%**

3 Implementation Details

- To load the input dataset file into MATLAB, use the built-in command [`csvread`](csvread).

- For **Figure 3,** in order to change the labels on the x-axis, utilize the **set** command with the axis property **XTickLabel.**

```
set(gca, 'XTickLabel', {'Idle','InBetween','Full'})
```

- For all of your plots, limit your x-axis by the **minimum** and **maximum x values**, and limit your y-axis by the **minimum** and **maximum y values.** **Do NOT hardcode the axis limits, obtain those limits from the dataset.**

- For all of your plots, have a **title**, **x-axis label, y-axis label**, and **grid on**.

4 Testing your code

Make sure that your code behaves as specified in this document. Here are a few ways that you can test it:

- Look at the Example Run on page 1. For the input dataset file **test01.csv,** your printed values in the Command Window and the three generated Figures shall exactly match.
- Make sure that you **suppress ALL unnecessary output** in your script. No additional printing should be seen in the Command Window.
- Your script shall work with ALL the input datasets provided in Moodle and any other dataset given that the format is the same. The teaching staff will use different input files to grade your code.

5 Submission

This project will be **due Monday, September 11th , 2017 at 2:00pm (one hour before lecture).**

Before you submit your work, make sure the program:

- Is thoroughly tested.
- Satisfies all criteria on the Grade Sheet for this project .

5.1 Project Partners

To complete this project you are encouraged (but not required) to work in groups of two. Partners for this project are to be chosen independently. Should you want to work in a group but are not able to find a partner, please fill out the **Project 1 Partner Request Google Form** on Moodle before Friday (8/25).
You can also use the Piazza forums to find a partner for your project. Please note that you must choose a different partner on future projects.

5.2 Styling Directions

For this assignment you will create one m-file called **ECUanalyzer.m.**
At the top add:

```
% Name(s): (of both students if working in a group!)
% Date:
% Lab Section # (make sure to include both section #s if different)
% Project 1: Engine Control Unit Analysis
% Description of Assignment
clc ; clear ; close('all');
```

5.3 Online Submission.

You need to submit online through the course's Moodle site. The name of the assignment is **Project 1 Submission.** **ONLY ONE student must submit in Moodle.** When we return the grading feedback ONLY the student that submitted will get feedback in Moodle in the form of a pdf file. **Make sure that you have both students name in the comments of your file.**

You need to submit **one** file: **ECUanalyzer.m**

5.4 Early Submissions

Early submissions will receive extra points on the assignment depending on the date in which they were submitted to the Moodle submission locker. **Note that no grade above 100 will be given for this project and the extra points are not cumulative.** In other words, if you lost any points during grading, the early submission points will help you get these points back. The early submission points given are as follows:

- Submissions by 11:55pm on 8/28/2017 receive 5 extra points.
- Submissions by 11:55pm on 9/04/2017 receive 3 extra points.

5.5 Late Submissions

You have up to 24 hours to submit your project late, which results in automatic 20 points deduction. After 24 hours, you will receive a 0 for this project.

6 Grade Sheet

This is how your project will be graded:

Points	Earned	Description
Input		
10		Correctly obtaining user input and importing the file into MATLAB
Matrix Manipulation		
24		Finding and Printing values from Section 2.1.2
Plot Generation		
18		Plot 1 generated and correctly annotated
32		Plot 2 generated and correctly annotated
16		Plot 3 generated and correctly annotated

Points	Earned	Description
-2		Name(s), date, section, etc. not included at the top of each .m file
-5		Incorrectly Named Files (**ECUanalyzer.m**)
-5		Code is not well commented
-5		Code is not cell blocked
+5		Submitted by 11:59pm on 08/28/2017
+3		Submitted by 11:59pm on 09/04/2017
-20		Submitted LATE by 2pm on 09/11/2017