**Total Points: 100 Due:** Week 2 in Lab

The project must be completed independently.

**Objective**

This project contains a set of programming problems, aimed to introduce the students to the general functionality of MATLAB.

Upon completion of the project, the students are expected to be reasonably comfortable with MATLAB data plotting, Command Window computations, basic array operations, creating and running script files in MATLAB Editor, interpreting workspace information, loading data files into MATLAB workspace, using basic MATLAB built-in functions.

**Part I: Command Window and Workspace**

The following problems are to be completed in MATLAB Command Window. When completing the report for this project, use a snipping tool (or any screen capturing tool) to capture a section of the Command Window, such that all commands used and the MATLAB command window (or workspace) outputs are clearly visible. The implementation captured in the report must NOT contain any previous unsuccessful attempts – make sure to clear those from your command window prior to capturing it.

**Problem 1 – 8 pts**

* **2 pts** **–** Create an array named **MyAngle** of size 1x5 containing the phase angles .
* **2 pts** **–** Use the appropriate MATLAB function to compute the tangent of all angles in the array **Angle**, and assign the output to the new array **Tangents**.
* **1 pts –** Without using any designated display commands such as *disp*, *fprintf*, display the calculated tangents in the command window.
* **1 pts –** The implementation must be optimized as much as possible (e.g. no unnecessary commands, as few as possible commands used). Also, the command window may NOT contain any unrelated variables, calculations, commands, or error messages.
* **2 pts –** In the Word Document, under the heading “Problem 1,” display the screen capture of your command window with all implemented commands clearly visible.

**Problem 2 – 12 pts**

Prior to starting this problem, make sure to clear the contents of the workspace and command window. Close all previously created figures.

The displacement as a function of time of a mass in the 2nd order mechanical system is described by the following equation.

Use Dot Multiply!

* **2 pts –** Create an array of time **t**containing time values from 0 to 20 seconds in increments of 0.1 sec.
* **2 pts** **–** Compute **y** using the given equation.
* **1 pts** **–** Plot **y** as a function of **t** in **Figure 1**.
* **1 pts** **–** Create a title to Figure 1 “2nd Order System Response.”
* **1 pts** **–** Create x-axis label “Time (sec).”
* **1 pts** **–** Create y-axis label “Displacement (m).”
* **1 pts** **–** Add a grid to Figure 1 using a designated MATLAB command.
* **1 pts –** The implementation must be optimized as much as possible (e.g. no unnecessary commands, as few as possible commands used). Also, the command window may NOT contain any unrelated variables, calculations, commands, or error messages.
* **2 pts –** In the Word Document under the heading “Problem 2,” display the screen capture of your command window with all implemented commands clearly visible, and the generated figure. The figure should be copied (Edit/Copy Figure) and pasted.

t=0:0.1:20

y = 0.101.\*exp(-0.25\*t).\*cos(1.98\*t-0.125)

figure(1)

plot(t,y)

title('2nd Order System Response')

xlabel('Time(sec)')

ylabel('Displacement(m)')

grid on **Problem 3 – 8 pts**

Do **NOT** clear the contents of the MATLAB workspace from Problem 2. However, make sure to clear your command window and close all previously created figures. To do this problem, MATLAB workspace must contain variables **t** and **y** from Problem 2.

* **2.5 pts –** Change the layout of the Workspace window to contain **Name, Value, Mean, Size,** and **Bytes** fields.
* **2.5 pts –** Directly from the workspace, plot **y** as a function **t** using an **area** plot function.
* **1 pts –** The implementation must be optimized as much as possible (e.g. no unnecessary commands, as few as possible commands used). Also, the command window may NOT contain any unrelated variables, calculations, commands, or error messages.
* **2 pts –** In the Word Document under the heading “Problem 3,” include the screen capture of the workspace clearly displaying the new layout. Include the screen capture of the command window displaying all the automatically generated commands used to create the plot. Include the generated figure. Briefly describe the steps you performed to generate the plot.

**Problem 4 – 14 pts**

Clear the contents of the workspace and the command window. Close all previously generated figures.

* **2 pts –** From workspace, import two data files “EEG1.txt” and “EEG2.txt.” When importing, make sure that the data columns in each file are separated by spaces.
* **2 pts –** Create a time array **time** containing time values from 0 to 0.998 seconds in steps of 2 msec.
* **1 pts –** In command window, using a designated command, display the size of **time** array, and make sure that the number of time values in the array corresponds to the number of data values in EEG1 and in EEG2.
* **2 pts –** Plot data in EEG1 as a function of **time** in the top subplot of **Figure 15**.
* **3 pts –** Plot data in EEG2 as a function of **time** in the bottom subplot of **Figure 15**. Change the color of the line to red.
* **0.5 pts –** Create an x-axis label “Time (sec)” for both subplots.
* **0.5 pts –** Create a y-axis label “EEG Amplitude (mV)” for both subplots.
* **1 pts –** The implementation must be optimized as much as possible (e.g. no unnecessary commands, as few as possible commands used). Also, the command window may NOT contain any unrelated variables, calculations, commands, or error messages.
* **2 pts –** In the Word Document under the heading “Problem 4,” include the screen capture display the screen capture of your command window with all implemented commands clearly visible, the screen capture of the workspace, and the generated figure.

**Part II: MATLAB Editor and Scripts**

The following problems are to be completed in MATLAB Editor and to be saved as an M-file or script. Each problem must have a designated script file. Each script is to be executed, and the results are to be displayed in the command window and/or in the figure form.

Each script file is to be properly commented as described in the lecture. Please see Slide 23 of BE-2200 Chap 1 and Intro ppt for more information on commenting your MATLAB code.

Publish each script file to a pdf file directly from MATLAB Editor (will be shown in lab), and attach it to your report from Part I.

**Problem 1 – 25 pts**

You are tasked with writing a MATLAB script to automatically compute Estimated Blood Volume (EBV) and acceptable Average Blood Loss (ABL) as part of the pre-surgical planning for a patient of interest. The EBV and ABL can be computed as follows.

*W* = weight of the patient in kg

*ABV* = average blood volume in ml/kg.

*HCTi* = initial (pre-surgical) hematocrit in percent

*HCTf* = final hematocrit in percent

Your patient is an adult female with a weight of 50 kg, and an initial hematocrit of 45%. Assuming the ABV of an adult female of 65 ml/kg and the final acceptable hematocrit of 30%, compute the patient’s **EBV** and **ABL** in **liters**. Furthermore, your script must meet the following specifications.

* **2 pts –** Your program will incorporate sufficient descriptive commenting and will be named *lastname\_Part2\_Prob1.m*.
* **5 pts –** EBV and ABL will be properly computed using the equations listed above. The unit conversion will be properly incorporated if needed.
* **6 pts –** EBV and ABL will be computed using variables rather than actual values. This means that you will be defining the necessary variables with their assigned values in the first few lines of your script file (following comments).
* Upon execution, your program must display the following using one of the designated MATLAB commands.
  + **2 pts –** “Patient: adult female weighing **[weight variable]** kg with initial hematocrit **[HCTi variable]** %.”
  + **2 pts –** “Average blood volume of an adult female is **[ABV variable]** ml.”
  + **2 pts –** “Final allowable hematocrit is **[HCTf variable]** %.”
  + **2 pts –** “Estimated EBV = **[EBV variable]** liters.”
  + **2 pts –** “Estimated ABL = **[ABL variable]** liters without necessary blood transfusion.

Note that you cannot simply hard-code the known numerical values. Instead, you must use the pre-defined variables, such that the values are placed in the corresponding locations automatically upon the execution of your program.

* **2 pts –** The script and the results of its execution will be published to a pdf file directly from the editor and attached to your report.

**Problem 2 – 33 pts**

You are tasked with writing a program, which will compute the total power in the Electroencephalographic (EEG) signal recorded from the conscious and unconscious patient, respectively. All calculations must be done in the script file, while the results of the script execution are to be displayed in the command window and/or in the figure form. Your detailed design specifications are as follows.

* **2 pts –** Your program will incorporate sufficient descriptive commenting and will be named *lastname\_Part2\_Prob2.m*.
* **4 pts –** Using **load** or **importdata** MATLAB function, import two data files into MATLAB workspace.
* **3 pts –** Using MATLAB help (or other sources) locate the appropriate function to compute an average (mean) of a dataset, and implement it to compute the mean of each of the two data sets.
* **2 pts –** Subtract the mean from its corresponding dataset. This is a common practice in data processing, and it ensures that the baseline drift is eliminated.
* **5 pts –** Compute the total power in each of the two data sets using the following equation. *Hint*: look up the function **sum** to perform this calculation easily.

🡪 This equation squares individual data values in the EEG signal, then sums all the squared values (*hint:* look up how summing is done in MATLAB in one line of code), and then divides the sum by the total number of data values in the EEG signal *N*.

* Using an appropriate display command, display the results of the total power computation as follows.
  + **2 pts –** “Total power in the EEG1 (conscious) signal is **[EEG1 power variable]** Watts.”
  + **2 pts –** “Total power in the EEG2 (unconscious) signal in **[EEG2 power variable]** Watts.”
* **3 pts –** Create a simple two-column **bar** graph displaying two power values side by side for easy comparison.
* **2 pts –** Create appropriate axis titles with the units properly specified.
* **2 pts –** Create a meaningful title for the bar chart.
* **3 pts –** The program will be optimized as much as possible. Variables, rather than hard-coded numerical values will be used in all automated calculations (e.g. power, mean, mean subtraction). The variables will be used in the display statements rather than hard-coded numerical values.
* **3 pts –** The script and the results of its execution will be published to a pdf file directly from the editor and attached to your report.

**What to Submit?**

The following three files are to be submitted to me via email by the beginning of Week 3 lab.

1. Submit a Word Document containing all required screen shots for Part I under Problem 1, Problem 2, etc. headings, respectively. Name the document *lastname\_Part1.doc.*
2. PDF file of the script file from Part II Problem 1. The name of the file should be *lastname\_Part2\_Prob1.pdf*.
3. PDF file of the script file from Part II Problem 2. The name of the file should be *lastname\_Part2\_Prob2*.*pdf.*