Engineering 13300
MATLAB 3

This assignment has three sections.

I) The first section has Team Exercises that are intended to be done with your team. Work the problems together and answer the questions. Type the solutions into MATLAB to check them AFTER you have worked them out by hand. After you find the correct answers, you can move to the second section

II) The second section has five team tasks (Tasks 1-5) that you may work together and develop code together as long as you document who contributed.

III) The third section has two individual tasks (Tasks 6-7). You can work together to conceptually help each other but each person should work on their own code.

Section I

These are to be worked together as a team. There are no files to be uploaded to Brightspace or Gradescope. Rather, work the problems and come up with the answers and be prepared to present them to a member of the instructional team. After you get the right answers, move on to the next problem. The intent is to understand the concepts before we begin to write code ourselves.

Consider the user-defined functions, fn1 and fn2, as shown below.

```
function [x,y]=fn1(x,y)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Help description
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = x - 4;
[y,x] = fn2(x,y);
y = x^2 - 10;
end
```

```
function [out1,out2]=fn2(in1,in2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Help description
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
out1 = in1 + in2;
out2 = in2 - in1;
end
```

The following codes are executed in the MATLAB Command window.

```
>> x = 3;
>> y = 5;
>> [x,y] = fn1(x,y)
```

What values do the variables x and y have?

Recall the guidelines for team activities:

1. You should work as a team; **all** team members will be held responsible for all material. You may work together and contribute to one program and submit similar codes as long as the contributors to the development of the solution are documented.
2. Each student is responsible for submitting their own assignment.

Team Task 1 (of 7) Security Camera Placement

An astronomical society restored their historic observatory and converted it to a museum. They hired your engineering firm to design a security system for the updated building. The building's layout, with pertinent measurements but not drawn to scale, is shown in Fig. 1.

The society wants the ability to place security cameras wherever they choose in each of the rooms.

Each camera will communicate its coordinates to a central receiver. Code must be written to confirm the location of the camera based on its reported coordinates. The building's coordinate system has its origin in the center of the observatory. **Note that the vestibule is part of the observatory room for camera placement.**

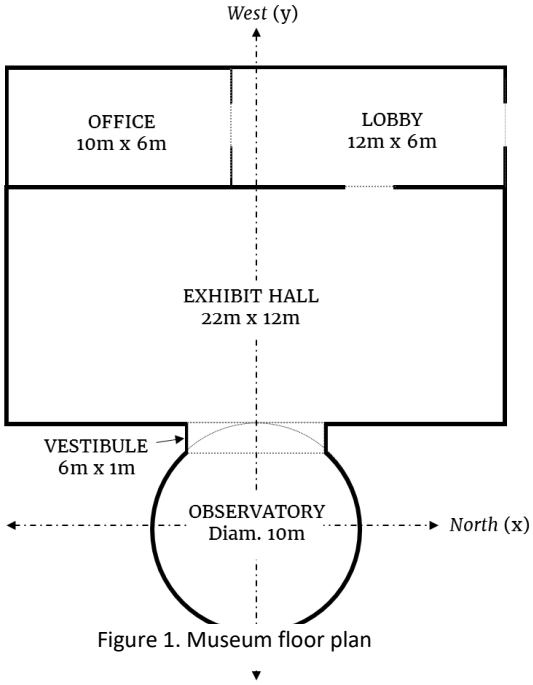

Figure 1. Museum floor plan

Write a function file *Ma3_Task1_username.m* starting with ***ENGR133_MATLAB_UDF_Template.m*** that has two inputs, (x,y) where x is the horizontal distance shown in the figure in meters (m) and y is the vertical distance, also in meters. The output should be an `fprintf` statement that displays the message in the command window: coordinates and the confirmed location (room) of the security camera should be stated. For example, for the input (3,20) "Lobby" would be printed.

Run your function for the following coordinates as inputs

A. (x,y) = (3,20)
B. (x,y) = (-2,4)
C. (x,y) = (-7,22)
D. (x,y) = (-6,-2)
E. (x,y) = (15,10)

Paste the `fprintf` output from the command window into the comment statements in the output section of your script.

Publish the *Ma3_Task1_username.m* function as a PDF for the case (x,y)=(3,20) and name it *Ma3_Task1_username*.pdf. Upload *Ma3_Task1_username.m* also.

Task 1 Files:

- `Ma3_Task1_username.m`
- `Ma3_Task1_username.pdf (published file)`

Team Task 2 (of 7) Length of a line

It is not uncommon that you would need to be able to calculate the length of a line given two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ in the *xy*-plane. The length of the line (*d*) can be computed by the following equation:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

A. Create a user-defined function that calculates the length of the line.
   1. Open ***ENGR133_MATLAB_UDF_Template.m*** and re-save as **Ma3_Task2_line_*username*.m**
   2. Write the code necessary to calculate the length of a line given two points using the equation shown above.
      - The input arguments are the x and y coordinates of two points in inches.
      - The outputs are the length of the line in inches and centimeters.
      - The unit conversion from inches to centimeters must be completed by calling another user-defined function (see step B below)

B. Write a user-defined function called **Ma3_Task2_INtoCM_*username.m*** that converts the length of the line from inches to centimeters. Use ***ENGR133_MATLAB_UDF_Template.m***
   - The input argument is length in inches.
   - The output is length in centimeters.

C. Call your **Ma3_Task2_INtoCM_*username.m*** function in your **Ma3_Task2_line_*username.m*** function to perform the necessary conversion calculations. (*Note: Both function files must be saved in the same folder for this computation.*)

D. Develop two test cases to demonstrate that your functions work properly. Use an alternative method to compute the line length in inches and centimeters to ensure your MATLAB solution is correct. Copy and paste the results of the call to your function **Ma3_Task2_line_*username*** from the command line and the outputs to the Command Window as comments in your **Ma3_Task2_line_*username*** function.

Publish the *Ma3_Task2_line_username.m* function as a PDF for one of the test cases and name it *Ma3_Task2_line_username*.pdf. Upload *both of your .m files also.*

Task 2 Files:

- `Ma3_Task2_line_username.m`
- `Ma3_Task2_INtoCM_username.m`
- `Ma3_Task2_line_username.pdf (published file)`

Team Task 3 (of 7) Exercise Planner

Open the template file ***ENGR133_MATLAB_Template.m***. Modify the file to create a script file that solves the following problem:

You are planning a workout schedule for a 4-week workout schedule. You want your schedule to meet the following requirements:

- The schedule must start on Monday.
- The schedule should initially alternate between a day with a 30-minute run and a day with a 45-minute run every day. Start the schedule with a 30-min run on the first Monday and continue alternating running days for the rest of the month.
- On Sundays, every 7th day, you want a rest day and should replace the run time with 0 minutes.
- On Tuesdays, you play soccer (football), so replace the time with 60 minutes instead. Do not reset the running schedule.

Problem Steps

1. Draw a flowchart or diagram before you start to program and save it as `Ma3_flowchart_username.pdf`
2. Initialize the calendar array in the `INITIALIZATION` section. Create an array for the 4 weeks where each row represents one week (like a calendar). Hint: lookup linear indexing
3. In the `CALCULATIONS` section, create a for loop that will set 30min run and 45min run a day.
4. In the `CALCULATIONS` section, set run time on Sunday's as 0 and set run time on Tuesday's as 60min using conditional statements. (Hint the "rem" command may be helpful. It calculates the remainder of a division between two inputs)
5. Create a function within the `OUTPUTS` section that will display the calendar in "week x day" format using `fprintf` and `disp` function in. The function will have one input that is the array with the values for each day and no output. Name the function "disp_fun_task3" and call it from the `CALCULATIONS` section.
6. **Publish** your script as a PDF and name it Ma3_Task3_*username*.pdf. [refer the 'how to publish' demo video and resource webpage]

Sample output:

```
The exercise plan is:
     M     T     W    Th     F    Sa    Su
    30    60    30    45    30    45     0
    45    60    45    30    45    30     0
    30    60    30    45    30    45     0
    45    60    45    30    45    30     0
```

Task 3 Files:

- `Ma3_flowchart_username.pdf`
- `Ma3_Task3_username.m`
- `Ma3_Task3_username.pdf (published file)`

Team Task 4 (of 7) Energy Conversion

Problem Setup

Energy conversion is an area of architectural (in civil engineering at Purdue) and mechanical Engineering and a concern of a number of other engineering disciplines. This area is often concerned with the heating, ventilation, and air conditioning (HVAC). To determine the heating and cooling loads of buildings, the engineer needs to determine the heat flows in and out of buildings. Solar radiation comes through windows. You will create a set of interacting user-defined functions to determine the transmission, absorption, and reflection of solar energy through horizontal venetian blinds.

Horizontal venetian blinds transmit, absorb, and reflect solar radiation, and these radiation values can be calculated. The calculations require knowledge of the operational geometry of a venetian blind, as shown in Fig 2.

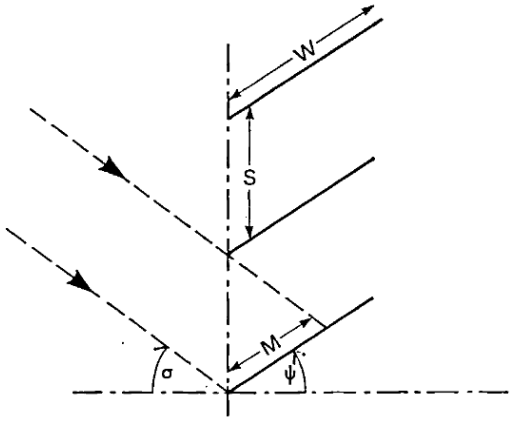

Fig 2. Operation of a blind

The geometric parameters of a blind are:
ψ = slat angle with the horizontal
W = slat width (mm)
S = slat spacing (mm)
σ = vertical shadow angle
M = width of slat illuminated

Additionally, the blind surface material will have an absorptivity constant, α (unitless).

There are three fractions that are used in the transmission and absorption equations:

- **F1:** fraction of total radiation reflected by the lower slat that passed into the room when the whole width is illuminated;
- **F2:** fraction of total radiation reflected by the lower slat that is intercepted by the upper slat when the whole width is illuminated;
- **F3:** fraction of radiation reflected by the upper slat that passes into the room;

$$F_1 = \frac{1}{2}\left(1 + \frac{S}{W} - \sqrt{1 + \left(\frac{S}{W}\right)^2 + 2\frac{S}{W}\sin\psi}\right)$$

$$F_2 = \frac{1}{2}\left(\sqrt{1 + \left(\frac{S}{W}\right)^2 + 2\frac{S}{W}\sin\psi} + \sqrt{1 + \left(\frac{S}{W}\right)^2 - 2\frac{S}{W}\sin\psi} - 2\frac{S}{W}\right)$$

$$F_3 = \frac{1}{2}\left(1 + \frac{S}{W} - \sqrt{1 + \left(\frac{S}{W}\right)^2 - 2\frac{S}{W}\sin\psi}\right)$$

When the whole of the lower slat is lit and some radiation passes directly into the room, then the total fraction of incident radiation transmitted by the blind is

$$T_D = 1 - \frac{W}{S}\frac{\sin(\sigma+\psi)}{\cos\sigma}\left(1 - F_1(1-\alpha) - \frac{F_2(1-\alpha)^2[F_3 + F_1 \cdot F_2(1-\alpha)]}{1 - {F_2}^2(1-\alpha)^2}\right)$$

And the total fraction of radiation absorbed is

$$A_D = \frac{\alpha W \sin(\sigma+\psi)}{S\cos\sigma\,[1 - F_2(1-\alpha)]}$$

You are given the parameters for one type of blind (Table 1). You will use user-defined functions to calculate the transmission and absorption values for three settings of the blind (Table 2).

Table 1. Blind parameters

Parameter	Blind 1
Slat Spacing	50 mm
Slat Width	60 mm
Absorptivity Constant	0.76
Shadow Angle	45 deg

Table 2. Blind settings

Parameter	Setting 1	Setting 2	Setting 3
Angle with horizontal	30 deg	45 deg	60 deg

Problem Steps

1. Use the ***ENGR133_MATLAB_UDF_Template.m*** for each function. Be sure to fill out all the information in the header.
2. Create a user-defined function named **Ma3_Task4_fractions_*username*.m** that calculates and returns the three fractional values F_1, F_2, and F_3.
3. Create a user-defined function named **Ma3_Task4_transmission_*username*.m** that calculates and returns the total fraction of incident radiation transmitted by the blind.
4. Create a user-defined function named **Ma3_Task4_absorb_*username*.m** that calculates and returns the total fraction of radiation absorbed.
5. Create an executive function named **Ma3_Task4_exec_*username*.m** that
    a. Initializes the given values of the blind parameters (Table 1),
    b. Calls the fraction UDF, the transmission UDF, and the absorption UDF, and
    c. Prints the results from the transmission and absorption UDFs to the command window.

i. The print command should say something similar to
`The transmission value for Blind 1 at <setting> is <value>.`
`The absorption value for Blind 1 at <setting> is <value>.`

6. Run your code for each of the blind settings provided in Table 2.
7. In the **COMMAND WINDOW OUTPUTS** section of your **executive function**:
    a. Copy and paste the transmission and absorption results for each setting from the command window.
8. Clear all variables from the workspace and the command window.

The UDFs get necessary parameter values (such as F1, F2, F3 or the parameters from Tables 1 or 2.) as inputs from the executive function. There should be nothing printed to the command window except from your executive function.

Publish the Ma3_Task4_exec_username.m function as a PDF for the case of 30 degrees and name it Ma3_Task4_exec_username.pdf. Upload all four .m files along with the .pdf

Reference: http://bse.sagepub.com/content/1/2/83.full.pdf

Task 4 Files:

- `Ma3_Task4_fractions_username.m`
- `Ma3_Task4_transmission_username.m`
- `Ma3_Task4_absorb_username.m`
- `Ma3_Task4_exec_username.m`
- `Ma3_Task4_exec_username.pdf (published file)`

Team Task 5 (of 7) Control Systems

Problem Setup

Designing control systems for electronic devices requires an engineer to understand the relationships between power and device output, such as heat, light, or sound. You work for audio company that is designing two new headphone prototypes that need volume control systems. The company has collected experimental data in their testing lab for the two prototype designs (Design OEP4 and Design IEP3). The company sent you their data and their model for the power-volume relationship for the two designs.

The data is in a file named **Data_volume_power.csv**. The model equations are as follows:

OEP4: $v_{OEP4} = 67.1 \log_{10} P - 1.3$

IEP3: $v_{IEP3} = 77.7 \log_{10} P - 7.3$

Where *v* is the headphone's volume in decibels (dB) and *P* is the power necessary to produce the sound, in milliwatts (mW).

Use the data and the model equations to complete the steps below:

- Open the script **volume_template.m** file. Complete the header information. Save your script with the name format required by the deliverables list (`Ma3_Task5_volume_username.m`).
- Open Data_volume_power.csv and review the information it contains.
- In the `INITIALIZATION` section of the script, import the data into the script using the appropriate MATLAB built-in function for the provided data file format. Copy each data column into a separate variable.
- In the `CALCULATIONS` section of the script, use the data obtained in the previous step to calculate the *predicted* volumes for OEP4 and IEP3, using the equation provided above.

In the `FORMATTED FIGURE` section of the script, write the code to

- Create one figure that contains a single plot of the original data for both headphone designs.
- Overlay the model predictions for each headphone design.
- Format the plot with a descriptive title, useful axes labels with units, and gridlines. Each set of data points must have a different marker style and color. Each model line must be a different style but match the color of its corresponding data markers. Add a legend that has appropriate labels and control the location of the legend to ensure it does not cover plotted information.

In the `ANALYSIS` section of the script, answer the following questions. Justify your answer using the plots.

Q1. Which model best fits its data?

Q2. Headphone sensitivity is a measure of how the volume level changes as the power level changes. The headphone with the largest volume gain for the same power gain is the most sensitive. Using the model plots, which headphone design do you believe is more sensitive?

Q3. Battery life is a function of power output over time. Which headphone design do you predict will have the best battery life if a listener consistently uses the headphones to produce a volume of 60 dB? At 30 dB?

Publish your script as a PDF and name it Ma3_Task5_volume_username.pdf.

Task 5 files to submit:

- `Ma3_Task5_volume_username.m`
- `Ma3_Task5_volume_username.pdf (published file)`

Individual Tasks

Guidelines for Tasks 6-7:
Tasks 6-7 are individual tasks. You may seek help from classmates, the instructional team or others but the work you submit should be your own. If you collaborate with others and use information developed together or by someone else, ALWAYS document and reference that material.

Individual Task 6 (of 7) LiDAR

Background
LiDAR, light detection and ranging, is a surveying system that measures distance by illuminating a target with laser light. Once data is obtained it is used to generate 3D point clouds of the objects or areas surveyed (see images 2a and 2b). LiDAR can be used for many applications, one of which is to measure lane widths in construction work zones. To collect lane width data, LiDAR equipment is mounted on top of a vehicle (see image 1) and driven through construction areas.


Image 1 –LiDAR equipment on vehicle

A traffic study was performed on a portion of interstate where construction was taking place. Lane width (obtained by LiDAR data) and the traffic speed data along I-74 from milepost 146 to 145 (the driving direction is from milepost 146 to 145) was obtained. Traffic speed data was collected from approximately 6:15 am to 5:45 pm, from the 1st to the 7th day, which adds up to approximately 80 hours in total. The data shows the lane width as well as the number of hours (within the 80 hours) that the traffic travels at different speed intervals at each mile marker location.

In this problem you will be looking at the affect lane width has on traffic speed. The research observed that when lanes are narrowed it affects vehicle speed. Reduced vehicle speed will affect traffic flow capacity and might lead to congestion, which would increase the drivers' stress as well as the probability of traffic accidents.

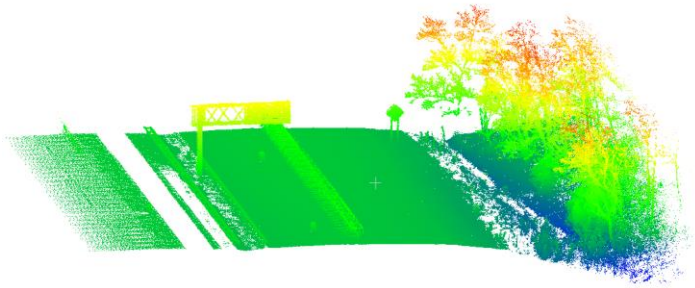

Image 2a – LiDAR data colored by height

Image 2b –corresponding image

Problem Steps

1. Observe the data file to see what each column represents before you remove the headers.
   Hint 1: *Consider ways to parse the loaded data into useful pieces.*
   Hint 2: *Some useful functions that may or may not be needed are:* ***min****(),* ***max****(),* ***find****(),* ***numel****(),* ***sum****(),* ***disp****(). If you have any questions about what these are for or how to use them, look them up within MATLAB help or on MATLAB's website.*
2. Load the data (***Lanewidth_TrafficSpeed.csv***) with the MATLAB `csvread` command into the script file to be saved as ***Ma3_Task6_ConstrolSys_username.m.*** Modify the template file – ***ENGR133_MATLAB_Template.m***. as a start for the script.
3. In the CALCULATIONS section, use relational and logical operators and associated built-in functions, as appropriate, to answer these three questions:
   a. Find the maximum and minimum lane width, and the corresponding mile markers.
   b. Find the mile marker range where the lane width is less than 10 ft. Let P represent the first mile marker in this range, and Q represent the last mile marker in this range. Show P and Q.
   c. Using your results from question B, calculate the average number of hours for different speed intervals based on the following mile marker ranges: 145 to P, P to Q, and Q to 146.
   d. Create a plot of lane width versus mile markers. Notice that within your PQ range, it looks like the width may go above 10 feet a couple times. Do you think this is a problem?
   e. To find out exactly how much of the PQ range is above 10 feet, make a vector that is composed only of the lane width data points between P and Q. Use logic statements, commands, and (if necessary) additional vectors to figure out how

many of the data points between P and Q have lane widths above 10 feet. Use that result to calculate the percentage of data points between P and Q which are greater than 10 feet.

f. Now that you have a more precise indicator of how many times the lane width is greater than 10 feet between P and Q, does this change the answer that you provided in Part D? Why or why not? Type a comment in your code with the answer to this question.

4. Write `fprintf` commands to print the results of the above question. Make sure you frame your results as sentences, not just the numbers. Your `fprintf` commands should include the following:
    a. Your lane min and lane max as well as their corresponding mile markers
    b. The mile markers for both P and Q.
    c. The percentage of data points between P and Q where the lane width is greater than 10.
5. Publish your script as a PDF and name it Ma3_Task6_ControlSys_username.pdf.

Task 6 files to submit:

- `Ma3_Task6_ControlSys_username.m`
- `Ma3_Task6_ControlSys_username.pdf (published file)`

Individual Task 7 (of 7) Launch vehicle tracking

Problem statement

You are testing a new propellant for space flight using a model rocket. On board the rocket is an R-DAS (Rocket Data Acquisition System) that measures the rocket's altitude and acceleration over the course of the fight. Your goal is use that information to find the velocity of the rocket. One data file is available: **Data_RDAS.csv** contains time of the flight in seconds and the acceleration of the rocket in g's (Remember 1 g = 32.2 ft/s^2).

To determine the velocity throughout the flight, you will need the area under the acceleration curve. The Trapezoidal Rule gives a formula to find a graphical approximation of that area:

$$V = \int_{t_0}^{t_f} a(t)dt \approx \sum_{k=2}^{N} \left(\frac{a_k + a_{k-1}}{2}\right)(t_k - t_{k-1})$$

Where N is the number of data points, a is the acceleration of the rocket, and t is the flight time.

Part A

Open **launchVel_tracking_template.m** and write a script that does the following:

- Imports the data and assigns the columns to unique variables.
- Uses a for loop to create a vector of the approximated velocity for each time entry in the data set.
    - Hint: you will need to initialize the velocity value to 0 ft/s for the first time entry.
    - Hint: notice the initial value of k in the summation is 2.
- Creates one figure with 3 subplots. The subplots must be in one column, with acceleration (in ft/s^2) as the top plot, velocity (in ft/s) as the middle plot, and altitude (in ft) as the bottom plot. All three plots must be plotted with respect to time. Format for technical presentation.
- Displays the maximum launch velocity (i.e., not falling) and the time it occurs to the Command Window using professional formatting.

Part B

Follow instructions in **Ma3_launchVel_tracking_template.docx** to complete the tracking table. Save the file as **Ma3_Ind_Task7_username.pdf.** Does your tracking table result match your script result? Debug your script and/or tracking table as necessary.

Publish your script as a PDF and name it Ma3_Task7_launchVel_tracking_username.pdf.

Task 7 files to submit:

- `Ma3_Ind_Task7_username.pdf`
- `Ma3_Task7_launchVel_tracking_username.m`
- `Ma3_Task7_launchVel_tracking_username.pdf (published file)`