HOMEWORK - 0

This MATLAB homework is intended to improve your computational software utilization skills which will be helpful not only in solving the homeworks in EE361 & EE362, but also for your engineering career. This homework will not be graded.

This homework is prepared considering a student who sees MATLAB for the first time in his life. You may skip the parts that you feel you do not need.

Note: You can write the command "**help** function_name" to the command window to investigate how a function operates and is used.

INITIALIZATION

(1) Open the MATLAB command window, workspace and editor.

(2) Create an m-file called **"my_project.m"** on a specific directory.

(3) On that file, using editor, create two variables with numbers 23 and 32. Run the program by pressing **F5**. Write name of those variables to the command window to display them.

(4) Go to **workspace** and investigate the variables. Apart from the **value,** you can view **the size, byte and class** information of any variable from the workspace. You can also list the variables and their properties by writing **whos** command.

(5) From this point on, keep your code in the m-file you created. You can seperate your subroutines using **%%.** Note that, if you press F5, all the lines in your m-file will be executed. To run only a fraction which is seperated, you can use **Ctrl+Enter.**

(6) If you do not want to display the variables on command window when running the m-file, you can use **semicolon (;)** at the end of the line.

VECTORS AND MATRICES

(7) Create a vector composed of the natural numbers from 1 to 20 using **linspace** function. Now repeat this operation using only **colons**. Now, create another arbitray vector with the same size. **Sum, subtract, multiply and divide** these vectors. Note that, these operations should be **element by element**. For multiplication and division, use a **dot (.)** before the operator.

(8) Create two empty 5x2 matrices by using **zeros** function. Assign the 1st matrix such as; the odd numbers from 1 to 10 on the 1st column and the even numbers from 1 to 10 on the 2nd column by using **linspace**. Create another arbitray matrix with the same size. **Sum, subtract, multiply and divide** these matrices. Same element by element principle is valid for the matrices. Actually, a vector is also a matrix with number of rows (or columns) 1.

(9) Take transpose of the second matrix you created by using **apostrope (')**. Now, **multiply** the two matrices (not element by element). Investigate the size of the resultant matrix. Assign and display the value on the 3rd row and 3rd column of this matrix as a separate variable.

(10) Create a 2x2 matrix with arbitrary numbers and take the inverse of the matrix by using **inv** function.

MATH OPERATIONS

(11) Create a single element variable with an arbitrary natural number. Take the square, squareroot, power of 5 of this real number.

(12) Create a single element variable with an arbitrary real number with two decimal points. Now, round this number to the nearest integer using **round, floor** and **ceil** functions separately. Investigate the difference of these functions. Now, try to round this number to not an integer, but to one decimal point.

(13) Create a single element variable with an arbitrary complex number with non-zero real and imaginary parts. Display the parts of this number using **real** and **imag** functions. Take the complex conjugate of this number using **conj** function. Calculate the magnitude and phase of this complex number using **abs** and **phase** functions. Convert the phase from radians to degrees.

FOR, IF AND CASE

(14) Using the **rand** function, obtain an array of size 100 with random real numbers between 1 and 2. Now, implement a subroutine with **for loop** and **if condition** to find the minimum number in this array. Repeat the same operation by using **min** function. Using **tic** and **toc**, observe the elapsed time for each method and compare them.

Note: *Computation time is especially important in embedded and real time systems. It depends not only the quality of the software, but also the speed of processors.*

(15) Add comments to the routine in (13) by using **%**.

(16) Write a MATLAB routine which asks for an input in [1,5] using **input** function, and returns an answer using **fprintf** function according to the following table:

Input	**Output**
1	Very Poor
2	Poor
3	Average
4	Good
5	Very Good

If an input other than these is entered, the program should return “wrong input!”. Use **IF** and **Switch**/**Case** statements together. Also note that, the default **string** input must be converted to **double**.

(17) Implement the following situation using **input** function, **IF** statement and **logical/relational** operators:

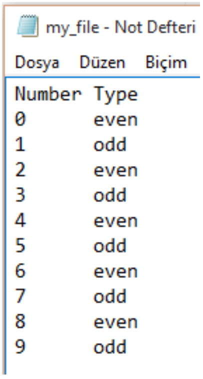
Ahmet, Deniz, and Selin attempt an exam. The instructor says; “If all of your grades are above 50, I will give you AA, BA, and BB according to your grade order. If only two of your grades are above 50, the order will be BA, BB and CB. If only one of your grades is above 50, the order will be BB, CB and CC. If all of your grades are below 50, the order will be CC, DC, DD. If at least one of your grades is below 30, I will pull down all your letter grades by one letter.”

Test your code for various inputs.

(18) Try to write a MATLAB routine which finds all the prime numbers between 1 and 1000.

DATA INPUT & OUTPUT

(19) To a text file “my_file.txt”, write the numbers from 0 to 9 and their types (even/odd) using the functions **fopen**, **fprintf** and **fclose**. The file should look like this:



```
Number Type
0      even
1      odd
2      even
3      odd
4      even
5      odd
6      even
7      odd
8      even
9      odd
```

(20) Now, read the data in the file “my_data.txt” on the homework directory which is a temperature data within a week using **fopen**, **fscanf** and **fclose**. Find and print; maximum tempreture and its day, minimum temperature and its day, average temperature of the week and root-mean-square of the temperature of the week.

(21) Create a time array with sampling frequency of 100 kHz and length of 1 seconds. Create a unit sinusoid against this time array with frequency of 100 Hz. Write this daha with time information on the 1st column and waveform information on the second column to an excel file “my_file2.xlsx” using **xlswrite**.

(22) Now, read the data (and record) in the file “my_data2.xlsx” on the homework directory which is another sinusoidal waveform data, using **xlsread**.

FIGURES AND PLOT

(23) In editor, open a **figure**. **Plot** the sinusoidal data you created in (21) against time in the range of [100 msec – 140 msec]. Using **hold on** function, plot the data you recorded in (22) on the same figure. Add **grid** to your figure. Adjust the color and linewidth of the figures. Add **title**, **xlabel** and **ylabel**. Now add **legend** to the figure as “waveform-1” and “waveform-2”.

(24) Notice that, once you have a figure opened, you may edit it as you want from the interface. However, it is a good practice to create graphs with low level techniques.

(25) Using the same time array in (21), create a three-phase utility grid voltage set with magnitude 400 V. *Beware that the magnitude is given as a **line-to-line** and **rms** quantity*. The frequency should be 50 Hz. Also note that the three-phase system is balanced. Now, on another figure, plot this voltage set for two fundamental cycles. Apply all the figure properties asked in (23).

(26) Now, suppose that only the phase-A of the voltage set in (25) is applied to a load and the resultant current (which is also 50 Hz) has $20A_{rms}$ magnitude, and lagging the phase-A voltage by 30 degrees. Create this current waveform and plot these two variables on the same figure. Be careful with the sign of the phase on the current waveform expression!

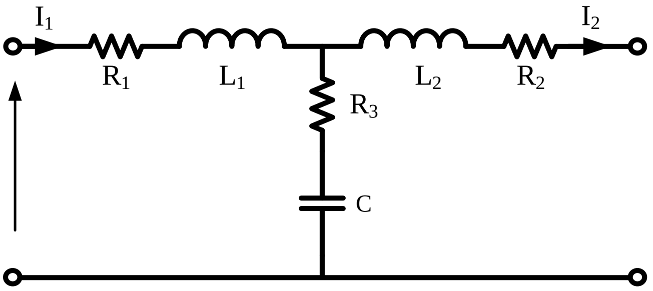
(27) What **type of load** is connected in (26)? (Capacitive, purely inductive etc.). Calculate the **power factor** of the system in (26) using MATLAB. Also calculate, **active power, reactive power** and **apparent power** delivered/injected to/from the load.

(28) Is the load linear or nonlinear? Calculate the values of the elements on the load (resistance etc.) using MATLAB.

(29) Try to obtain the phasor diagram showing the voltage, current and load vectors of this system using MATLAB.

FREQUENCY RESPONSE

(30) Consider the RLC circuit shown in the figure below:



R_1 = 0.1 Ω, L_1 = 220 uH
R_2 = 0.1 Ω, L_2 = 80 uH
R_3 = 0.07 Ω, C = 60 uF

Mathematically model this circuit in s-domain and obtain the transfer function between V_1 and I_2.

(31) Create this system (transfer function) using **tf** function. Find its **pole**s and **zero**s. Obtain a bode plot of this circuit using **bode** function.

(32) Investigate the frequency response of this circuit. What may the main objective of this circuit be? If there were no resistive components, how would the response change?

"Give a man a fish and you feed him for a day; teach a man to fish and you feed him for a lifetime"