COMPUTER LAB 1 – MATLAB BASICS

Learning Objectives
- To practice using MATLAB data analysis tools for manipulation of data.

Where Learning Outcomes Assessed:
- In Pre-lab report, Prac lab report, mid-semester and final exam.

This computer lab involves 3 tasks centred on: 1) Data manipulation, 2) loading csv files and data comparison, and 3) loading practical lab csv files, smoothing noisy data, creating data structures and saving data. **(You will need to demonstrate these skills in the pre-lab submission and other assessment pieces).**

Task 1: MATLAB has a number of script programming and plotting tools. In the first task you will familiarise yourself with creation of signals and plot vectors to help you with the practical lab that occurs in middle of semester. Please consider two sinusoidal waveforms, with frequency 0.1 radian / time instant (zero phase) and 0.5 radian / time instant (zero phase). Hint: at time instant k, y1(k)=sin(0.1*k).

(a) Please develop MATLAB commands that produce a vectors **y1** and **y2** that represent 100 time instants of each of the signals. Plot both signals on the same graph (but the second waveform should be plotted with a red line).

(b) Using MATLAB, determine the time instant, k, that each waveform signal crosses from above zero to below zero (after k=0). Then separately shift to the left the elements each vector so that this time instant becomes the first element in the vector (discarding the elements before this time instant). Then admen or add to each vector new elements equal to 0 so that vectors remain 100 elements long. Create a second figure, and plot both of the vectors on the same figure (but the second waveform should be plotted in red).

Task 2: In this 2nd task you will load in some data from a CSV file (representing a noisy measured waveform), and then compare this data with the two sinusoidal waveforms described in the first task on the basis of RMS error (smaller RMS means closer). There is a MATLAB function called *rms(.).* Create an error vector that is the difference between various vectors.

(a) Load the data from the CSV file *sinedata.csv* as the vector yn and then plot. Hint: csvread.

(b) Compare the vector yn with above waveforms on the basis of RMS value. Which vector is closer? What does that mean. **Hint:** recall that you concatenated these vectors with 0, and you shouldn't use these parts of the vectors in the comparison (maybe use only first 25 elements).

PTO… Task 3 and hints on next page

Task 3: In this 3rd task, you will load an example of servo motor position data that was captured in last year's practical lab.

(a) Open the csv file openloop_1.csv in excel to inspect the structure of the file, and use this information to load the csv data into MATLAB and plot the two represented signals (one signal is the response, the other is a time vector). You may need to type "help csvread" in MATLAB to work out how to skip the initial unnecessary information present in the csv file.

(b) Use MATLAB's *smooth(.)* (or *smoothdata(.),* depending on the version of MATLAB you are using) function to smooth that response data and plot both the raw and smoothed response data on the same figure. You may need to type "help smooth" to learn the syntax of MATLAB's smooth command. NOTE: in latest versions of MATLAB the *smooth(.)* command has changed name to *smoothdata(.).*

(c) Use MATLAB structures to store both the raw and smoothed vector in one variable. For example, perhaps data.raw and data.smoothed. Then save this structure variable (and only this variable) into a matlab data file with the filename "smoothdata.mat". You may need to type "help save" to learn the syntax of matlab's save command.

Congratulations you have just learned the basics of create data vectors, manipulating data, using RMS to understand noisy data, loading data from csv files, smoothing data, saving data, and using data structures in matlab (the above steps will be required when working with data captured in the practical lab.)

MATLAB **HINTS:** You might find it helpful to use the following MATLAB commands: ***help, clc, clear, for, end, find (y1>0), hold on, hold off, plot(y1), figure(2), kv=1:1:100,*** **y1(25:end),** ***ones,*** **csvread, smooth, save and rms.** You should use script files and structures

See all the MATLAB hints pdf.