SE 165-265 Homework #2
Assigned April 10th, 2024; Due April 17th, 2024 (11:59 pm Pacific Time)

Assignment Goal:

1. Perform basic statistical analysis with Matlab.

2. Reinforce material presented in the 4th lecture

Background Reading: *Structural Health Monitoring: A Machine Learning Perspective, Appendix A Signal Processing for SHM. Chapter 6: Introduction to Probability and Statistics*

Introduction
We will continue to use experimental data from the 4-story structure. You can refer to Homework #1 to recall how the experiment was performed.

Remember, for all the tasks that requires **subplots**, when creating the figure, use:

```
figure('Renderer', 'painters', 'Position', [10 10 1200 900])
```

This will make the figure larger, to make sure all subplots are visible, and the titles are not overlaying each other.

Task 1: Download the file **data3SS2009.mat** from the SE 165-265 CANVAS Data File folder, 4-story structure data. **Load** this file in Matlab. Notice that this is the same data set from HW1. This file contains measurements from all 5 sensors.

Loading this file will give you a 3-D matrix called **dataset** ($\mathbf{8192 \times 5 \times 850}$), where 8,192 corresponds to the data points, 5 corresponds to channels 1 to 5, and 850 is the number of measurements. In this assignment, **we will only work with data from channel 5** (data from the accelerometer on the top floor).

Save the data from channel 5 in a new variable called **testingData**.

For all the following tasks, we will work with data from the following states:

1. State 1 (columns 1 to 50), which we consider to be the baseline.
2. State 5 (columns 201 to 250), a case that represents some source of variability, but no damage.
3. State 10 (columns 451 to 500), a case with a low level of damage (hard to distinguish from undamaged cases).
4. State 12 (columns 551 to 600), a case with an intermediate level of damage.
5. State 14 (columns 651 to 700), a case with the highest level of damage.

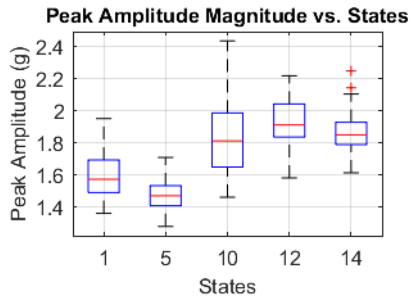
A. We will calculate some basic statistics for each of the 5 states, **using all 50 measurements from each state**. I've added an a few additional slides on Canvas (Files > Data & Codes > Statistics Functions) that have the Matlab commands for these statistics. The file is called Basic Statistics Functions. You can also refer to Table 7.1 on the textbook. The basic statistics are:

- peak amplitude magnitude
- mean
- mean square
- root mean square
- standard deviation
- skewness
- kurtosis.

➢ **A.1.** Create a **boxplot** of these quantities as a function of state. Use the appropriate units for each quantity. You can either have the 7 boxplots in separate figures or in a single figure using subplot/tiledlayout.

➢ **A.2. Answer the question:** Are there any observable trends in these statistics with increasing amounts of damage? Write your answer in your code, using %.

Hint: Results from the first statistics requested.



B. This task is to get familiar with the central limit theorem. We will work with measurements from **State 1** only $(8192\ points \times 50\ measurements)$.

To generate more samples, for each of the 50 measurements, break down the 8,192-points time series into eight 1,024-points time series. Your new matrix should be (1024×400).

Using **histfit**, generate a histogram from **the average value (mean) of each column** of this new matrix. The **histfit** command does both parametric and non-parametric density estimation. It generates the histogram (non-parametric estimate) and by default overlays a normal distribution on the histogram. The normal distribution is based on the mean and standard deviation of the data (parametric estimate).

C. In this task, we will work with the first record of each state (**columns 1, 201, 451, 551, and 651** only). We will use the histogram and kernel density estimator in Matlab to estimate the density functions for the five acceleration signals.

You will generate 5 plots, one per state. You can present them in a (3x2) subplot or in separate figures. The results from steps 1-3 below should all be done for these 5 plots:

1. For each time history (**first** record of each state) use the **histfit** command with the default number of bins (square root of number of data points, approx. 90). Use the default normal density function associated with this command (by default, it will be a red line).

   - Add the appropriate labels.
   - Add titles to the plots, informing **which state is being plotted, the number of bins being used, and the bin width**. The bin width will be different for each set of data (see 2.c.i. below on calculating the bin width).

2. Use the command **ksdensity** with the default parameters to obtain a kernel density estimate of density function.

   Overlay this estimate on the figures generated in step 1.

   a. Use green line and a width of 2 for this plot.
   b. Note that this density estimate should be similar to the normal distribution generated in step 1, but not exactly the same.
   c. Note that you **will have to scale the ksdensity estimate** to make it match the estimates obtained in step 1.

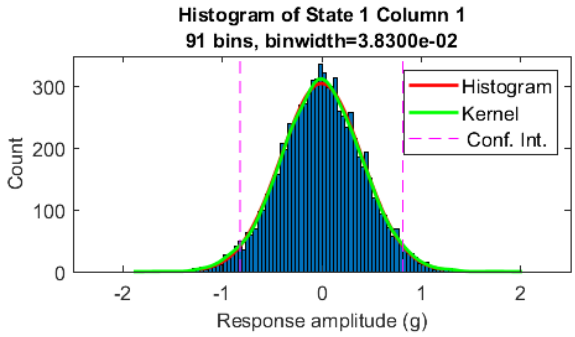
   The scale factor is 1/(N*h) where "N" is the number of data points (8192) and "h" is the bin width generated by the **histfit** command. You will have to divide the **ksdensity** amplitudes by this scaling factor.

   i. An easy and low-tech way to determine the bin width is to plot the histogram and use the data curser in the figure window.
   ii. Place the data curser in successive bins of the histogram and it will give you the coordinates of the center point of the respective bins.
   iii. Subtract the smaller x-coordinate from the larger and this will be the bin width. This value will be different for each data set.
   iv. Hint: A more efficient way to determine the bin width is by exploring the output H(1) from H=histogram(…).

3. Add two vertical lines to the figures generated in step 1 that represent the 95% confidence intervals for the parametric distribution estimate. Remember that for a normal distribution the 95% confidence intervals correspond to +/- 2 standard deviations from the mean. Use the mean and standard deviations from Part A to generate these lines.

4. Use a legend to inform what each line represents.
5. For all plots, fix the x-axis at [-2.5 2.5] and the y-axis at [0 350].

Hint: results from state 1. All of your 5 plots should contain all these elements.



D. Use the Matlab command **normplot** to overlay the normal probability plots for the five time histories (again, first record of each state - columns 1, 201, 451, 551, and 651 from testingData), using different color plots for each state. Add a legend to this plot so I can tell which plot corresponds to the different signals.

E. We will perform a principal component analysis (PCA) on the data from most of the sensors on the column. The idea is to produce a single time-history that is a linear combination of all 50 measurements for each state. Therefore, for each of the 5 states we will use the 8192-points time series of all 50 measurements (columns 1 to 50, 201 to 250, etc.).

For each state, you will do the following:

a. Save the data from all 50 measurements into a new matrix $[d]_{8192x50}$. The subscripts indicate the dimension of the matrices used in this example. Note, you can call it anything you want. I'm going to use [d] in the problem description.

b. Subtract the mean from each measurement and produce a new data matrix $[dm]_{8192x50}$.

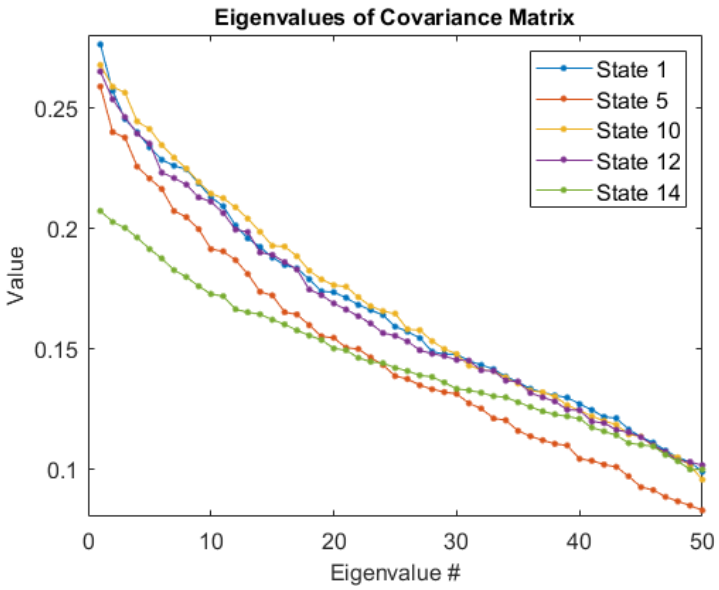
c. Form the covariance matrix of [dm]. You can use the Matlab command **cov** to do this. The covariance matrix will be of dimensions 50x50.

d. Calculate the eigenvalues and the eigenvectors of the covariance matrix using the **eig** function.

e. Sort the **eigenvalues** in order of highest to lowest. You can use the "**sort**" function with the '**descending**' parameter to do this.

f. Sort the **eigenvectors** in a corresponding manner (using the indexes from the eigenvalues' sorting).

g. In the same figure, overlay a plot of the sorted **eigenvalues** for each of the five different states in descending order. Add a legend and the appropriate title and labels.



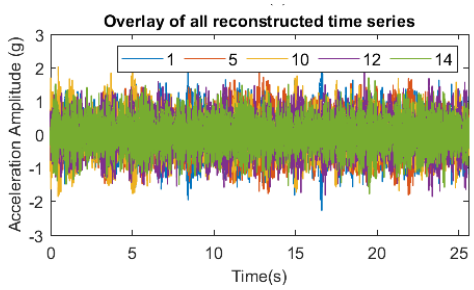
h. Now form new time histories, $[tn]_{8192x1}$, that are linear combinations of the 50 signals we are analyzing from each state. To do so, use the eigenvector corresponding to the largest eigenvalue, $[v1]_{50x1}$, and form the product $[d]_{8192x50}*[v1]_{50x1}=[tn]_{8192x1}$.

i. You also need to create a time vector. You know that the sampling frequency (Fs) is 320Hz, the time step is 1/Fs, and the total number of data points is 8192.

j. Plot the time vector × the new time series. Create one plot per time series (either 5 figures or a 3x2 subplot). Add the appropriate title and labels. For all plots, fix the y-axis at [-3 3].

k. In another plot, overlay the plot of these five new time histories corresponding to the different states. Add legends, the appropriate title, and labels.

Hint: results from overlay (from k.).



l. Finally, in another figure, overlay a plot of the auto (power) spectrum of these five new time histories. Use a Hanning window with 8 averages and no overlap in this calculation (similar to HW 1). Plot on log scale for the Y-axis (**semilogy**). Add legends, the appropriate title, and labels.

Check list for submission:

For all plots

- Titles that indicate what you are plotting.
- Labels with the appropriate units.
- Legends when plotting more than one case.
- For spectral quantities, only plot the positive portion of the spectrum up to the Nyquist Frequency.

You will lose points if you forget the titles, labels, and legends, but they should be "easy points" if you always get this done, even if your plot content is incorrect.

Files to upload

- .m file **"FirstName_LastName_HW2.m**
- .pdf file **"FirstName_LastName_HW2.pdf**

Please insert comments in your m-file so I can follow what you are doing. Please tell me what your variable naming scheme is in these comments.