SE 165/265 Homework 1

Assigned April 3rd, 2024; Due April 15th, 2024 (11:59 pm Pacific Time)

Before starting this assignment, read the **HW Submission Guidelines** posted on Canvas carefully (Files -> Additional Instructions).These instructions apply to **all the coding assignments** you will work on this quarter.

Please execute all tasks below in a single .m file.

Assignment Goals:

1. Start familiarizing the students with reading in experimental data files and plotting them.
2. Perform basic signal processing with Matlab.
3. Reinforce material presented in the 3rd and 4th lectures.

Introduction: Background information on the Dataset used in this problem.

The dataset for this problem comes from the 4-story structure pictured below. You can read about tests that were performed to generate these data in report entitled "Structural Health Monitoring Algorithm Comparison Using Standard Data Sets" that is also posted in this folder (Files → Data & Codes → 4-story structure data) and in Chapter 5 of the course reference book.

A random excitation was applied at the base floor with an electrodynamic shaker. A **load cell (Channel 1)** was attached at the end of a stinger to measure the input force from the shaker to the structure. Four **accelerometers (Channels 2-5)** were attached at the center line of each floor on the opposite side from the excitation source to measure the system's response, as shown in Figure 1.

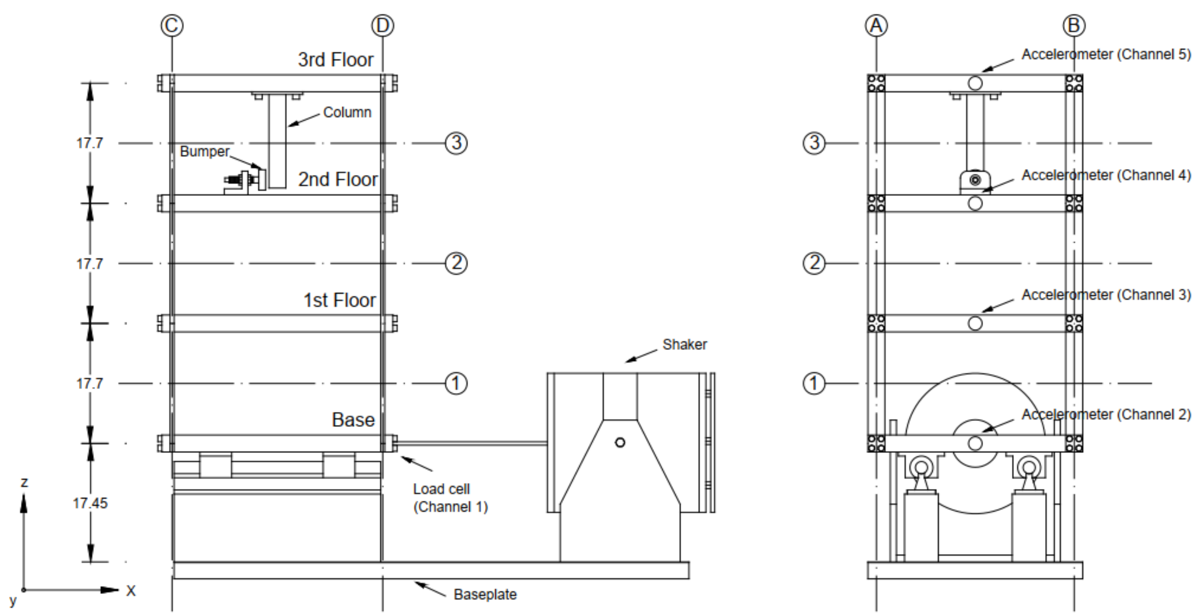
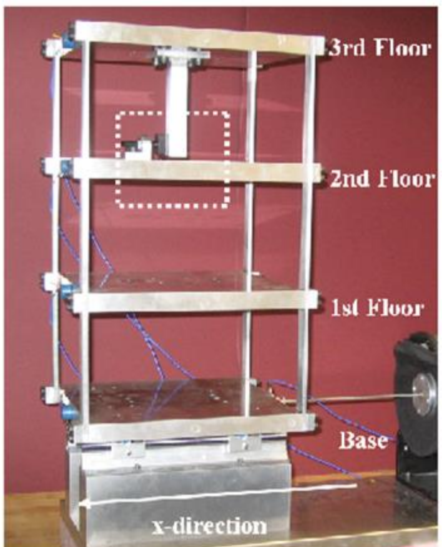


Figure 1- Basic dimensions of the four-story building structure (all dimensions are in cm).

Acceleration measurements provided are from all the 5 sensors (1 load cell and 4 accelerometers). To consider the variability of the data, 50 measurements were taken at 17 different structural states for a total of 850 measurements for each sensor (i.e., column 1-50 contain data from State #1, column 51-100, State #2, etc.). The 17 states are described in Figure 2.

The analog sensor signals were discretized with 8,192 data points sampled at 3.125 ms intervals corresponding to a sampling frequency of 320 Hz. These sampling parameters yield time histories of 25.6 s in duration. You will need this information to create your time vector.

A band-limited random excitation in the range of 20-150 Hz was used to excite the structure. This excitation signal was chosen to avoid the rigid body modes of the structure that are present below 20 Hz.



Structure pictured above. Damage and simulated operating conditions described in table to the right. Bottom-left diagram shows mass added to base floor. Bottom-right diagram shows bumper system for simulating damage.

Label	State Condition	Description
State #1	Undamaged	Baseline condition
State #2	Undamaged	Mass = 1.2 kg at the base
State #3	Undamaged	Mass = 1.2 kg on the 1^{st} floor
State #4	Undamaged	87.5% stiffness reduction in column 1BD
State #5	Undamaged	87.5% stiffness reduction in column 1AD and 1BD
State #6	Undamaged	87.5% stiffness reduction in column 2BD
State #7	Undamaged	87.5% stiffness reduction in column 2AD and 2BD
State #8	Undamaged	87.5% stiffness reduction in column 3BD
State #9	Undamaged	87.5% stiffness reduction in column 3AD and 3BD
State #10	Damaged	Gap = 0.20 mm
State #11	Damaged	Gap = 0.15 mm
State #12	Damaged	Gap = 0.13 mm
State #13	Damaged	Gap = 0.10 mm
State #14	Damaged	Gap = 0.05 mm
State #15	Damaged	Gap = 0.20 mm and mass = 1.2 kg at the base
State #16	Damaged	Gap = 0.20 mm and mass = 1.2 kg on the 1^{st} floor
State #17	Damaged	Gap = 0.10 mm and mass = 1.2 kg on the 1^{st} floor

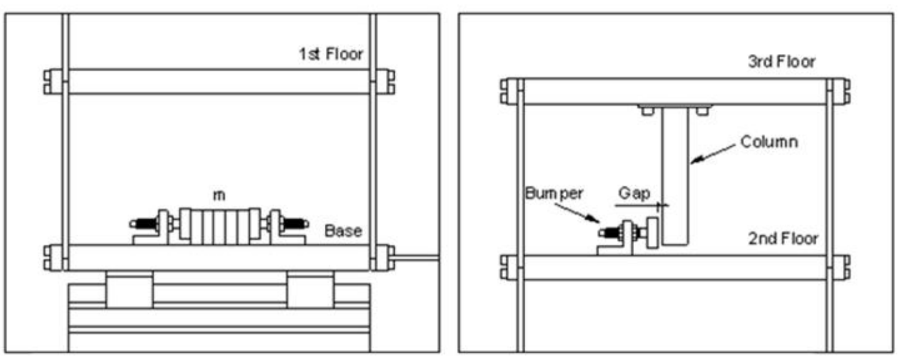


Figure 2- Experimental setup.

Task 1: Plotting the time series:

Download the file **data3SS2009.mat** from the SE 165-265 CANVAS → Files → Data & Codes → 4-story structure data.

Load this file in Matlab. This is how your code should look like so far:

```
% Your Full Name
% SE 165/265 (Depending on the class you are enrolled)
% Homework #1
clc; clear; close all

%% Task 1
load data3SS2009.mat %data from 4-story structure
```

Loading this file will give you a 3-D matrix called **dataset** ($\mathbf{8192} \times \mathbf{5} \times \mathbf{850}$), where:

- 8,192 corresponds to the data points.
- 5 corresponds to channels 1 to 5.
- 850 is the number of measurements.

In this assignment, we will work with data from channel 1 (input time history from the load cell) and channel 5 (data from the accelerometer on the top floor).

Save the data from channel 1 in a variable called **inputData** ($\mathbf{8192} \times \mathbf{850}$). Save the data from channel 5 in a variable called **testingData** ($\mathbf{8192} \times \mathbf{850}$). You may delete the **dataset** variable if you wish, we will not need it anymore.

In the subtasks 1.A-F below you will be generating acceleration-time history plots. These plots should have **time** on horizontal axis and **acceleration** amplitude on the vertical axis. Acceleration amplitudes are in g's and you can plot the amplitude in g (obs: g is the gravitational acceleration, where $1g=9.8m/s^2$).

For these plots, use the same vertical axis scale to facilitate comparison of the different signal amplitudes. Note the **min** and **max** commands in Matlab will be helpful in identifying the required vertical axis range.

For tasks 1. A-D, you will use subplots. When calling 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.

1.A. Generate the time vector – you can do this by using the sampling information given with the experiment's information.

Using **subplot** or **tiledlayout,** create a 3x3 grid**,** and plot the acceleration vs. time for the first data record (the 1st measurement) corresponding to each undamaged state (states#1 -#9). Those 9 records correspond to columns 1, 51,101, 151, …, 401 in the **testingData** matrix.

Add a title to each plot indicating which state and column it represents. Remember to label all axes and to use the same vertical scale for all plots.

1. B. Repeat task **1.A**, but instead of plotting the entire signals, only plot the range 100-300 points. Limit the x-axis from time(100) to time(300).

1. C. Using **subplot** or **tiledlayout,** create a 3x3 grid, and generate a second acceleration vs time plot for the first data record corresponding to each of the damaged states (states 10#-#17). Those 8 records correspond to columns 451, 501, 551, …, 801 in the **testingData** matrix. As you will have only 8 plots, the last subplot slot will be blank.

Add a title to each plot indicating which state and column it represents. Remember to label all axes and to use the same vertical scale for all plots.

1. D. Repeat task **1.C**, but instead of plotting the entire signals, only plot the range 100-300 points. Limit the x-axis from time(100) to time(300).

1. E. Plot the acceleration vs time for the first data record corresponding to state #1 (Column 1 in the **testingdata** matrix) and overlay a plot of the first record for state #15 (Column 701). This plot should be an overlay with each time history plotted in a different color. **Include a legend indicating the states.**

1. F. Repeat task **1.E**, but instead of plotting the entire signals, only plot the range 100-300 points. Limit the x-axis from time(100) to time(300).

Once again, do not forget the title, axes labels, and a legend indicating the undamaged and the damaged states.

Task 2. Signal Processing

For all spectral quantifies, only plot the portion of the spectrum up to the Nyquist Frequency. The sampling frequency was given at the Introduction (320 Hz).

For all the following tasks, we will analyze data from the **first data record** of the following states:

- State 1, which we consider to be the baseline.
- State 5, a case that represents some source of variability, but no damage.
- State 10, the case with a low level of damage (hard to distinguish from undamaged cases).
- State 12, a case with an intermediate level of damage.
- State 14 a case with the highest level of damage.

Therefore, we will only use columns 1, 201, 451, 551, and 651.

For **tasks 2 A-E**, we will use the **testingData.** For **tasks 2 F-H**, we will also need the **inputData.**

Once again, for all the tasks that requires subplots, when calling the figure, use:

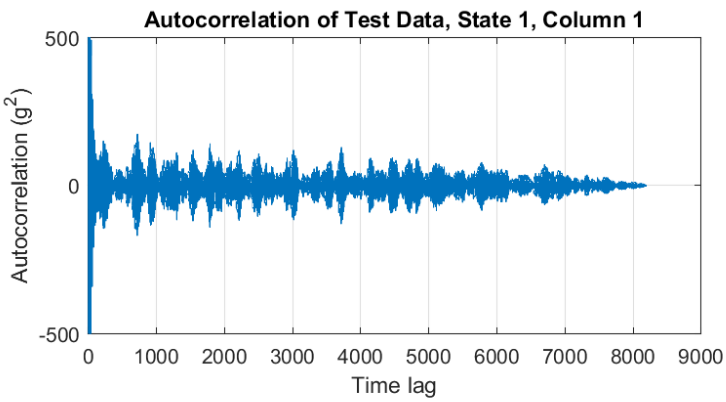
```
figure('Renderer', 'painters', 'Position', [10 10 1200 900])
```

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

2.A. Using **subplot** or **tiledlayout,** create a 3x2 grid**,** and plot the autocorrelation function vs. time lag for each of the five time series (States 1, 5, 10, 12, 14, corresponding to columns 1, 201, 451, 551, and 651 of **testingData**). There will be a blank subplot as we only have 5 plots and 6 slots, that is fine.

Notice that you can plot the autocorrelation vs indices starting at data point 8,192 and going to 16,383 in the vector returned by the **xcorr** function. Set the vertical axis limits [-500 500]. MATLAB command: **xcorr**.

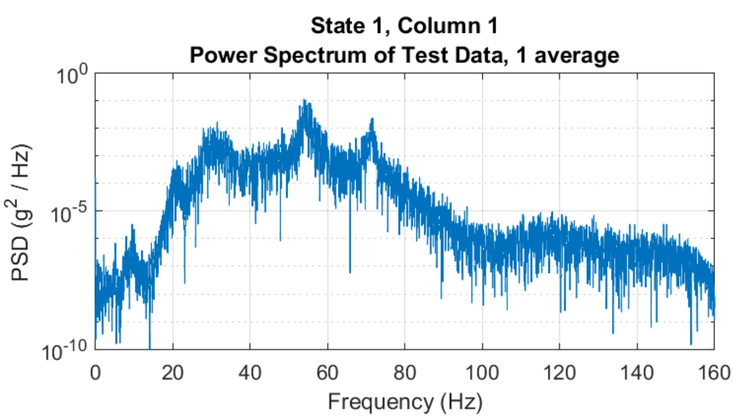
Hint: result from subplot(3,2,1)



2.B. Create a 3x2 grid and plot the power (auto) spectral density function for each signal with a Hanning window applied to the data. This plot should be based on a <u>single average</u> of the entire 8,192 pts signal. Do not use overlap or detrending. Matlab command: **pwelch**.

The horizontal axis should be in Hz. Place appropriate label on the vertical axis. Vertical axis should be a log scale (do not take the log of the psd, use the **semilogy** plotting command). For comparison purposes <u>use the same axis limits for all five signals.</u>

Hint: result from subplot(3,2,1)



2.B.1. Next, create another 3x2 grid, and plot an averaged power spectrum for each signal by dividing the time series into 8 blocks of 1,024 points. Apply Hanning window to each of the blocks. Do not use overlap or detrending. Again, plot with a log vertical scale and use the same limits on the vertical scale.

Obs: You do not need to window the signal and create the averages manually. This can all be done by setting the appropriate options on the **pwelch** command in Matlab.

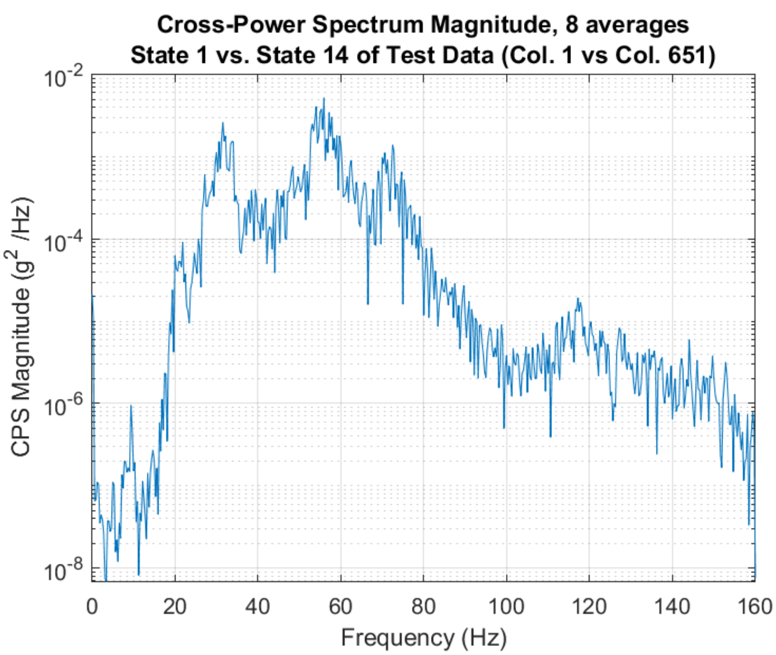
2.C. Calculate the cross-correlation function between State 1 and each of the other states (States 5, 10, 12, and 14). Create a 2x2 grid and plot these four cross-correlation functions vs. time lag.

Note you can plot the cross-correlation vs. indices starting at data pt 8192 and going to 16,383 in the vector returned by the **xcorr** function. Set the vertical axis limits [-200 200]. MATLAB: **xcorr(x,y),** where x refers to state 1.

2.D. Create a 2x2 grid and plot the cross-power spectrum magnitude of each of the other states (States 5, 10, 12, and 14) relative to the undamaged signal (State 1). Apply a Hanning window and use 8 averages. Do not use overlap or detrending. MATLAB: **cpsd(state 1,state Y, …)**.

2.D.1. In a separate figure, plot all 4 CPS overlaid and use a legend to indicate each case.

Hint: result from 2.D. subplot(2,2,2)



2.E. Repeat Tasks 2.B, B.1, D, and D.1, but this time low-pass filter the signal at 80 Hz using a 6th order Butterworth filter.

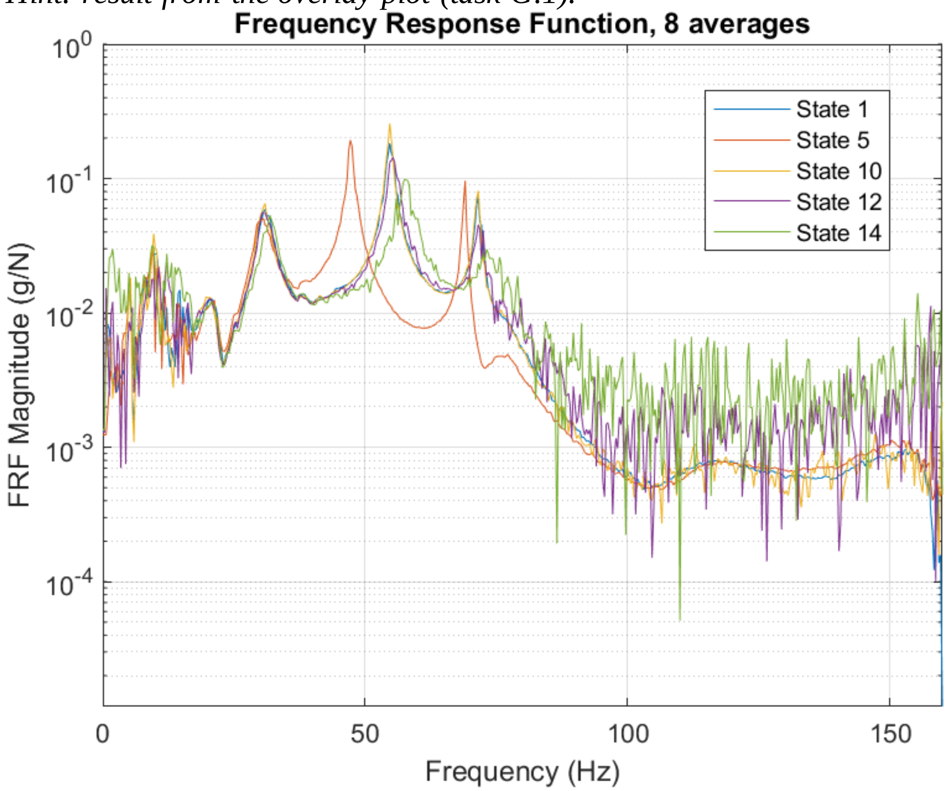
2.F. The input time history from the shaker was saved as **inputData**. The unit for the input data is N. For all five states (columns 1, 201, 451, 551, and 651) calculate the power spectrum for each input time history by breaking the signal in 8 averages. Apply a Hanning window to each block and no overlap. Overlay the plots of the five input PSDs in a single figure. MATLAB: **pwelch**.

2.G. Calculate the Frequency Response Function for each of the 5 signals from **testingData** (the original, unfiltered data) using the corresponding input from **inputData**. Break the time history into 8 windows. Apply a Hanning window to each block and form an averaged FRF. Do not use overlap or detrending. In a 3x2 grid, plot the FRF magnitude of each of the 5 states. **Use a log vertical axis for this plot.**

MATLAB: **tfestimate()**.

2.G.1. In another figure, plot an overlay of all five FRF magnitude plots. Use a log vertical axis for this plot. Use a legend to indicate each case.

Hint: result from the overlay plot (task G.1).



2.H. Calculate the Coherence function for each of the 5 signals from **testingData** (the original, unfiltered data) using the corresponding input from **inputData**. Use a Hanning window, with 8 averages and no overlap. Plot each case in a 3x2 grid**. Use a linear vertical scale.**

2.H.1. In another figure, overlay the coherence function of the five states. Use a linear vertical scale. Use a legend to indicate each case.

MATLAB: **mscohere()**.

Hint: the coherence value (y-axis) should vary between 0 and 1.

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_HW1.m**
- .pdf file **"FirstName_LastName_HW1.pdf**

NOTES

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.

"Matlab help" is a great resource for understanding the functions. Just type help function_name to know more about a certain function and how to use it.