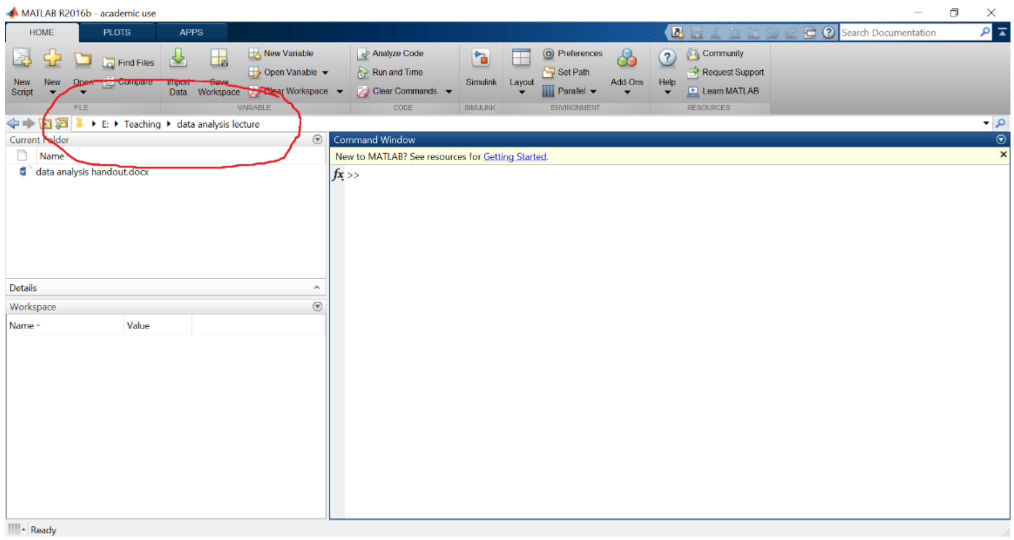
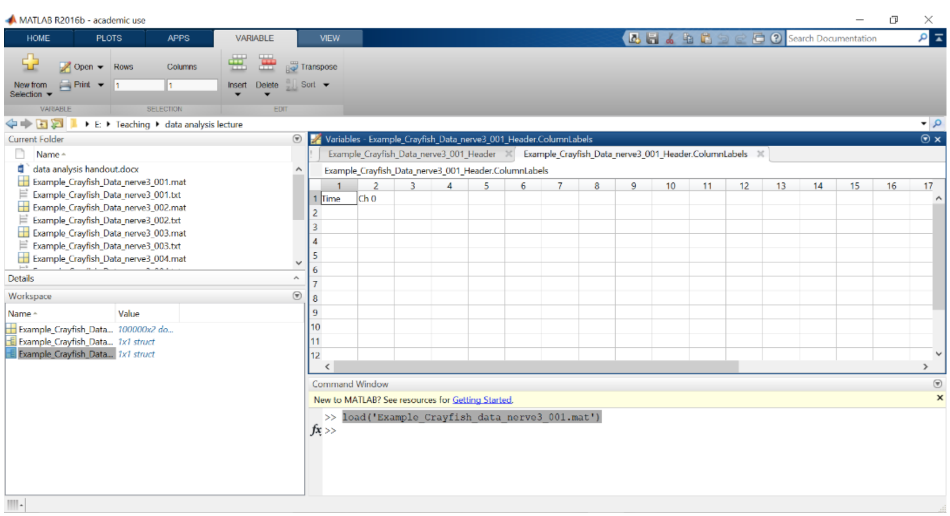
Data Analysis in MATLAB

Lab 1: The speed limit of the nervous system (comparative conduction velocity)

Importing Data into MATLAB



- Change your **Current Folder** to the folder where your data is located.
- Import data by double clicking it in the **Current Folder** window or by using the *load* function in the **Command Window**.
    - load('your_file_name.mat')
    - Example: load('Example_Crayfish_data_nerve3_001.mat')
- You can double-click these variables in the **Workspace** to view their contents in the **Variables** window:



- In the Header.ColumnLabels structure, you can see what the different columns of your data are. Here, the first column is the time and the second column is the electrode.

Plotting in MATLAB

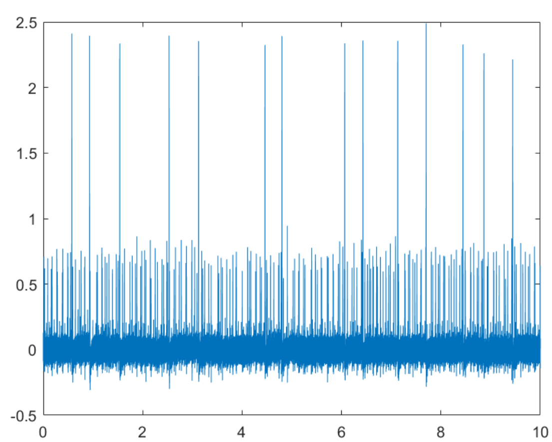
Variables - Example_Crayfish_Data_nerve3_001

Example_Crayfish_Data_nerve3_001

100000x2 double

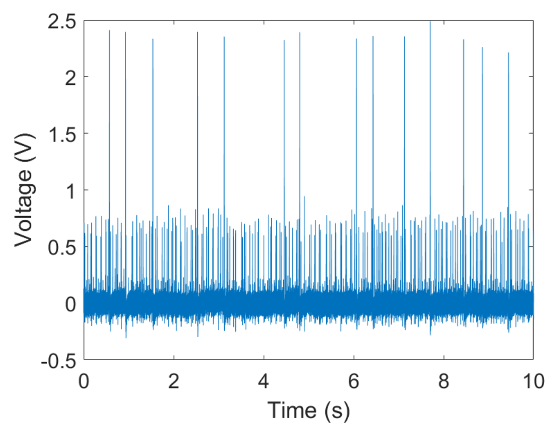
	1	2	3	4	5
1	-10	0.0626			
2	-9.9999	0.0649			
3	-9.9998	-0.0857			
4	-9.9997	0.0550			
5	-9.9996	-0.0436			
6	-9.9995	-0.0223			
7	-9.9994	0.1293			
8	-9.9993	-0.0410			
9	-9.9992	0.0580			
10	-9.9991	0.0162			
11	-9.9990	-0.0176			
12	-9.9989	0.0751			

- Use the imported data to create a time vector *t* and a voltage vector in your workspace.
    - Example: vector = Example_Crayfish_data_nerve3_001(:,2);
- NOTE: Your time vector starts at -10 s, not 0 s. This is because when you manually triggered your DAQ, you kept the 10 seconds before the trigger.
- You can plot two vectors against each other using the MATLAB *plot* function:
    - Example: plot(t,x)

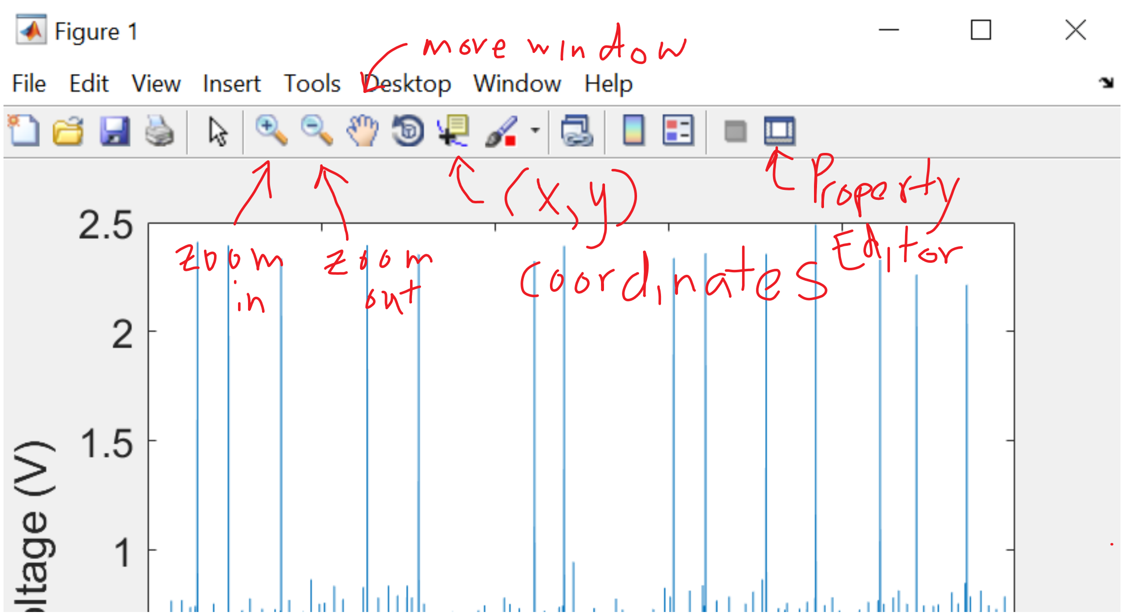


- Here are some functions that will help you create nice MATLAB plots:
    - *xlabel* labels the x-axis.
        - Example: xlabel('Time (s)')
    - *ylabel* labels the y-axis.
    - Set your font size using the following command:

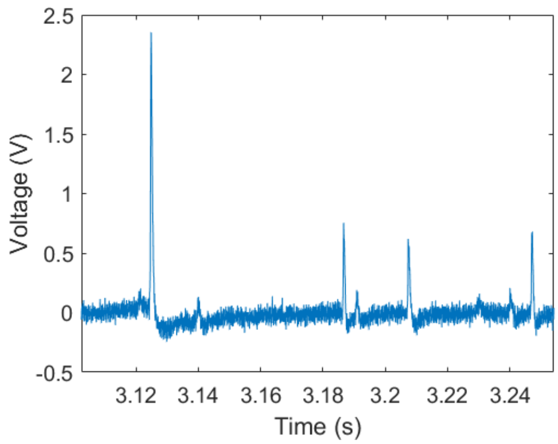
        - set(gca,'FontSize',#)
        - Example: set(gca,'FontSize',15)
    - *hold on* will enable you to plot multiple voltage traces in the same window.
        - Example: hold on; plot(t,x2)
    - *legend* creates a legend for multiple traces.
        - Example: legend('Electrode 1','Electrode 2')
    - You can also use the **Property Editor** button (the last button on the toolbar at the top of the **Figure** window) to make these changes and more!
- After using these functions, this is what your plot can look like:



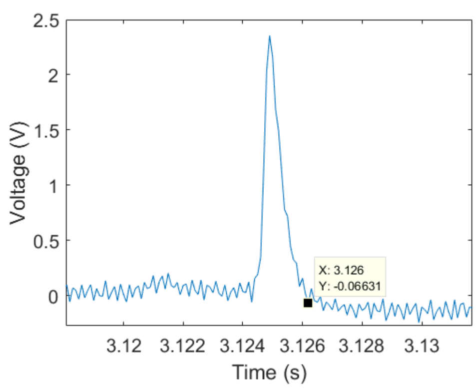
- You can also use the toolbar in the **Figure** window to manipulate your plot.



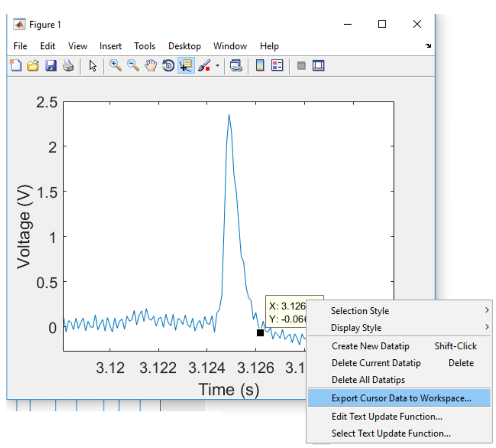
- You can zoom in, zoom out, move the plot so different portions of the data are visible, and click on the trace to find the (x,y) coordinates of different data points (i.e. the peaks!).
- Zoom in on one of the spikes by selecting the zoom in tool and dragging a box around the area of interest.



- How would you describe the shape of an action potential? What distinguishes it from noise or motion artifacts?
- Zooming in even further can let you measure the width of your action potential using the (x,y) coordinates (Data Cursor) tool.

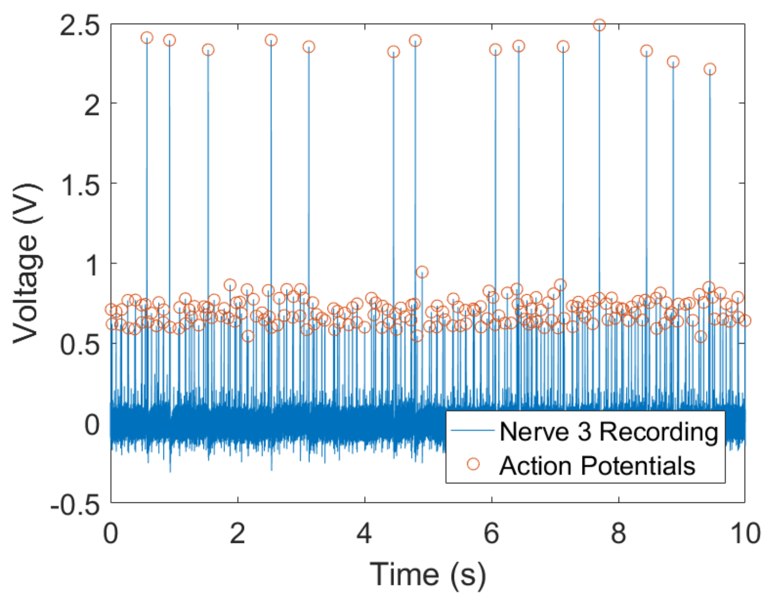


- To get more decimal places for the (x,y) coordinates, you can click on the box and export the cursor data to your workspace.

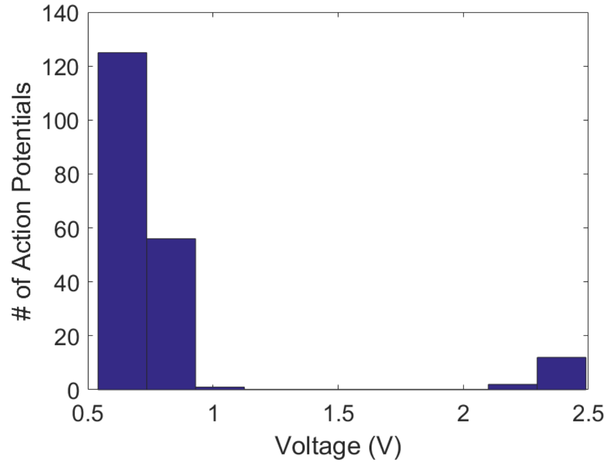


Peak Detection

- MATLAB has a built-in peak detection function, *findpeaks*, though you may also use manual analysis methods or write your own peak detection function.
- Consult the help file on *findpeaks*: https://www.mathworks.com/help/signal/ref/findpeaks.html
- To find the amplitude of the action potentials in your signal, you can use *findpeaks* or another method:
  - Example: [amp,loc] = findpeaks(data,Fs,'MinPeakHeight',0.5,'MinPeakDistance',0.005)
  - Fs is the sampling frequency of your data. The recordings you took should have Fs = 10000 Hz.
  - You can set a threshold for the peak height using MinPeakHeight, and a refractory period between detections using MinPeakDistance. Here, I set a threshold of 0.5 V and a refractory period of 5 ms. These parameters will vary from recording to recording.
  - The variable *amp* will give you the amplitude of the peaks, and the variable *loc* will give you the time they occur.
- I used *hold on* and the plot function to overlay my detected amplitudes on the voltage trace. To plot single data points instead of a line, modify your *plot* command:
  - Example: plot(t,x,'o')



- To make a histogram of your action potentials, use the matlab *hist* command.
  - Example: hist(amp)



- Based on this histogram, how many neurons do you think are present in this recording?

Simple Statistics in MATLAB

- *mean* – calculates the mean of a data set
  - mean(data)
  - If the data input is a matrix, the output of this function will be the mean for each column of the matrix.
- *std* – calculates the standard deviation of a data set
  - std(data)
  - If the data input is a matrix, the output of this function will be the standard deviation for each column of the matrix.
- HINT: You can calculate the baseline of your electrode recordings using the mean. Since peaks are very rare events, taking the mean of your data can give you a good estimate of the minimum voltage of your action potentials.

Try these analyses on your own data and create plots for your transition assignment!