R•I•T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Sciences & Technology

ISTE-782 Assignment 2 - Data Scavenger Hunt

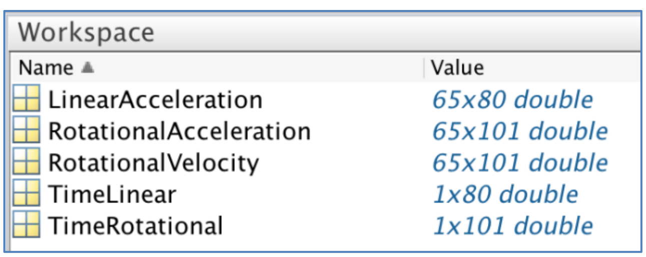
The purpose of this assignment is to give you the opportunity to manipulate, display, and compute features from raw data. Over the course of this assignment, you will be guided through a process of manually finding interesting phenomena in a data set. There is no specific process that will work for all data sets, but this specific process should give you a feel for how you would initially analyze your own data set that you'll find for your course project.

Part 1 – Opening Data in Matlab

The data set you'll be using is on myCourses in the Content area under Assignment 2 and is called waveforms.mat. Save the file to your computer. Open Matlab and navigate to the directory where you saved the data set. Either double click on the waveforms.mat file in the directory listing or type the command below to load it programmatically.

```
load('waveforms.mat');
```

In your workspace, you will see five matrices, as shown below. Notice the dimensions of each matrix. How many instances of data do we have in the first three matrices? How do you know?



The matrices are defined as follows:

LinearAcceleration: This matrix contains 65 instances of 80 linear acceleration readings taken by an accelerometer. The times at which each of these measurements occurred may be found in the TimeLinear matrix. Note that those two matrices have the same number of columns. The unit of each accelerometer reading is in g's (9.81 m/s^2).

RotationalVelocity: This matrix contains 65 instances of 101 rotational velocity measurements taken by a gyroscope. The times at which each of these measurements occurred may be found in the TimeRotational matrix. Note that those two matrices have the same number of columns. The unit of each gyroscope reading is in radians per second (rad/s).

RotationalAcceleration: This matrix contains 65 instances of 101 rotational acceleration measurements that are derived from the rotational velocity measurements taken by the gyroscope. Rotational acceleration for our purposes is simply the difference between consecutive rotational velocity measurements and represents the rate of change in rotational velocity from reading to reading. Like the rotational velocity matrix, the measurement times come from the TimeRotational matrix. The unit for these measurements is in radians per second squared (rad/s^2).

TimeLinear: This matrix has a single row of 80 times at which the accelerometer took its measurements, beginning at 0 seconds, and ranging to approximately 0.050 seconds, or 50 milliseconds (ms).

TimeRotational: This matrix has a single row of 101 times at which the gyroscope took its measurements, beginning at 0 ms, and ranging to approximately 50 ms.

Now that you know what each of these matrices represent, let’s look at the data!

Part 2 - Data Manipulation and Visualization

First of all, the two “time” matrices are not very pleasant to look at. Notice the “min” and “max” values for these two matrices in the Workspace. The times are in seconds, but when we talk about the time scales at which the accelerometer and gyroscope measurements were taken, we really want this data to be in milliseconds. How can we convert from seconds to milliseconds? Use the lines of code below, filling in the proper conversion constant so that the arrays are in units of milliseconds instead of seconds.

```
TimeLinear = TimeLinear * <conversion constant>;
TimeRotational = TimeRotational * <conversion constant>;
```

Now let’s see what the linear acceleration, rotational velocity, and rotational acceleration waveforms look like. Before we visualize the data, let’s note the “min”, “max”, and “mean” values for each of these matrices in the Workspace. These “summary statistics” about the data are the min, max, and mean for the entire 2D matrices. If we want to see the min, max, or mean for a single waveform instance, use the following code example for the matrix and summary statistic of interest.

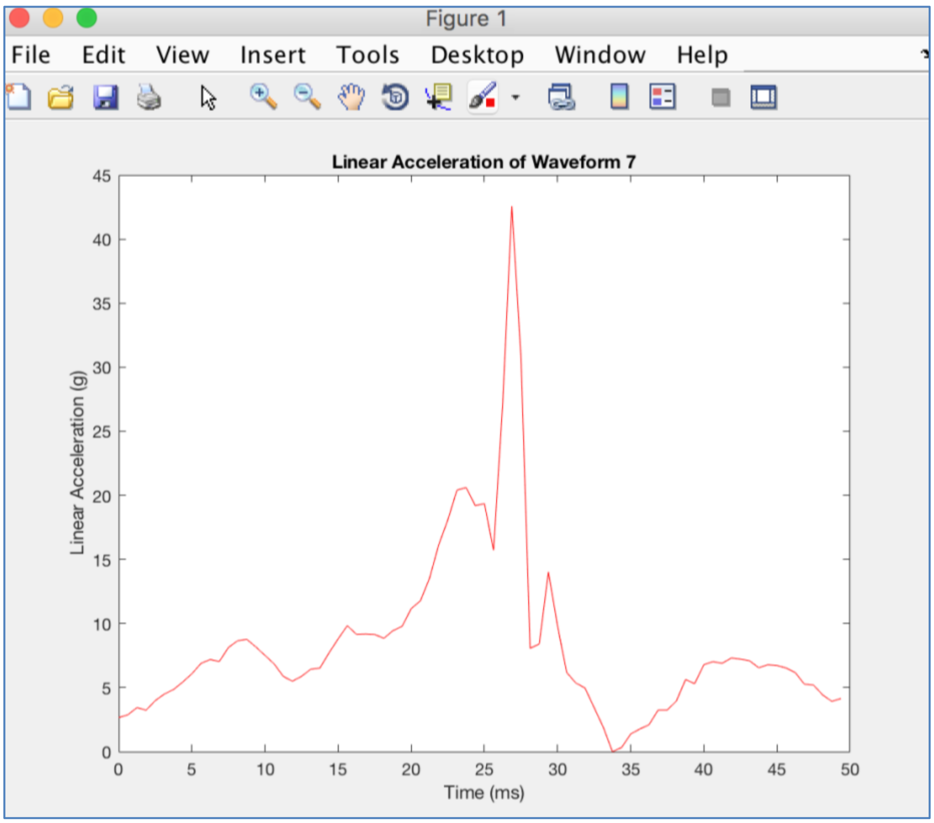
```
mean( LinearAcceleration( 3, : ) )

ans =

    8.1696
```

This example computes the mean of all the 80 accelerometer readings from the 3rd instance of a linear acceleration waveform that was measured by the accelerometer. Notice that the “:” operator is used in place of providing the range of indices (1:80) of the row of data. If Matlab does not output the same answer, something is incorrect and you should seek assistance from the instructor.

What does this waveform data look like?! Let’s plot one of the waveforms and see. Enter this code line by line in the Command Window and see how the plot you create evolves as the commands are entered.

```
plot( TimeLinear, LinearAcceleration( 7, : ), '-r' );
title( 'Linear Acceleration of Waveform 7' );
xlabel( 'Time (ms)' );
ylabel( 'Linear Acceleration (g)' );
```

This set of commands plots the 7th linear acceleration waveform instance using a red line and adds a proper title and axes labels. It's always important to know what the units of measure are.



Your plot should look something like this.

To save the figure to a file, use this command and fill in the path where you want it to go.

```
print(gcf, fullfile( '<your path>', [ 'Linear Acceleration
Waveform ' num2str( 7 ) '.png' ] ), '-dpng', '-opengl');
```

Notice that the second argument to the “fullfile” function is the filename. The bracket notation allows you to concatenate strings and numeric values together. The num2str function is converting the number 7 to a string. Your file should be called “Linear Acceleration Waveform 7.png” when it’s saved. Close the figure once you’ve confirmed that it saved properly.

Using a for loop and the subplot command, create an image file for every data instance that contains all three waveforms (Linear Acceleration, Rotational Velocity, and Rotational Acceleration) for that instance in a single figure. The code should look something like below. Since the code is rather lengthy, create a new .m script and run the code from that script. The script should begin with a load command to ensure that the data set is open. Your script will also need to scale the time matrices from seconds to milliseconds, as we did above.

One final note: if you want to properly compare the waveforms between data instances, you should use the same scale on the y-axis for each individual waveform plot. To do this, you need to first find the maximum value for Linear Acceleration, Rotational Velocity, and Rotational Acceleration in the entire data set, and then use the ylim command in each individual plot. For example, if the maximum Linear Acceleration in the entire data set is 80g, then the y-axis on every Linear Acceleration plot you create should go up to 80g. It is recommended that your script also calculates these maximum values and use them in the ylim command for each plot.

```
for i = <loop bounds>
    figure; % create a new figure
    subplot( 3, 1, 1 );
    hold on;
    <plot command for linear acceleration waveform>;

    title( 'Linear Acceleration of Instance <number>' );
    xlabel( 'Time (ms)' );
    ylabel( 'Linear Acceleration (g)' );

    subplot( 3, 1, 2 );
    hold on;
    <plot command for rotational velocity waveform>;

    title( 'Rotational Velocity of Instance <number>' );
    xlabel( 'Time (ms)' );
    ylabel( 'Rotational Velocity (rad/sec)' );

    subplot( 3, 1, 3 );
    hold on;
    <plot command for rotational acceleration waveform>;

    title( 'Rotational Acceleration of Instance <number>' );
    xlabel( 'Time (ms)' );
    ylabel( 'Rotational Acceleration (rad/sec^2)' );

    print(gcf, fullfile( '<your path>', [ 'Instance '
num2str( i ) '.png' ] ), '-dpng', '-opengl');

    close all; % close any open figures
end % i loop
```

Fill in the loop bounds, output directory path, and the proper plot commands. Once you've created the files, take a look at all of the 65 images. Do any of the sets of waveforms stand out over the others? Compare the respective individual waveforms from each data instance as you go through the images one at a time.

Part 3 – Creating, Analyzing, and Selecting Features

Out of the 65 data instances, 5 of them are statistically “different” than the other 60 instances. In this part of the exercise, we will be creating features that will help us complete a data scavenger hunt. We’ll be generating several features and only some of them will turn out to be useful in finding the data instances that are different than the rest.

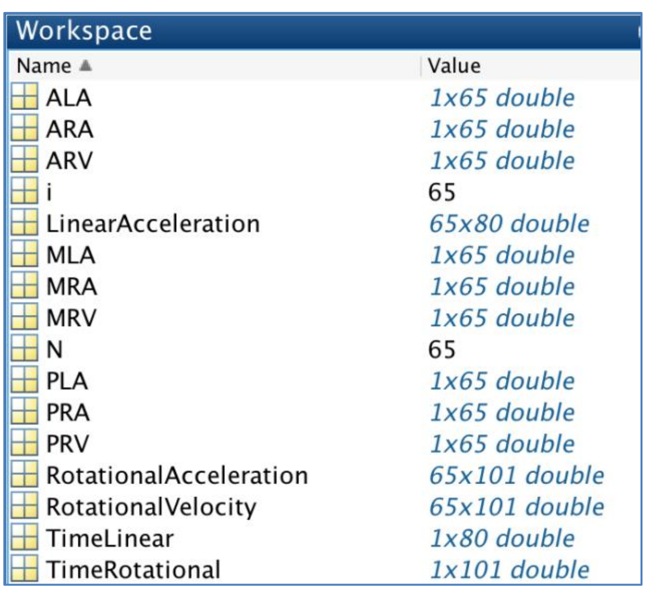
Create a new .m script that starts off by loading the waveform data set. Next, write three separate loops that iterate across the three respective waveform matrices and compute the min, max, and mean of each individual waveform. Save these features to the following matrix names. *Hint: to write the cleanest possible code, you should pre-allocate these matrices using the “zeros” command and fill these matrices in during their respective loops.*

MLA: minimum linear acceleration
ALA: average (mean) linear acceleration
PLA: peak (max) linear acceleration

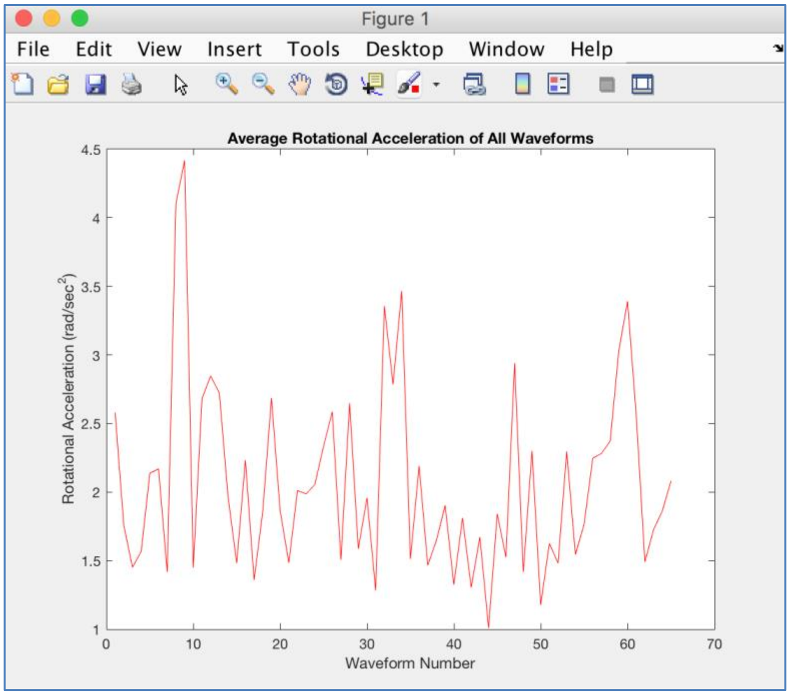
MRV: minimum rotational velocity
ARV: average (mean) rotational velocity
PRV: peak (max) rotational velocity

MRA: minimum rotational acceleration
ARA: average (mean) rotational acceleration
PRA: peak (max) rotational acceleration

After generating the features, your Workspace should look something like this.

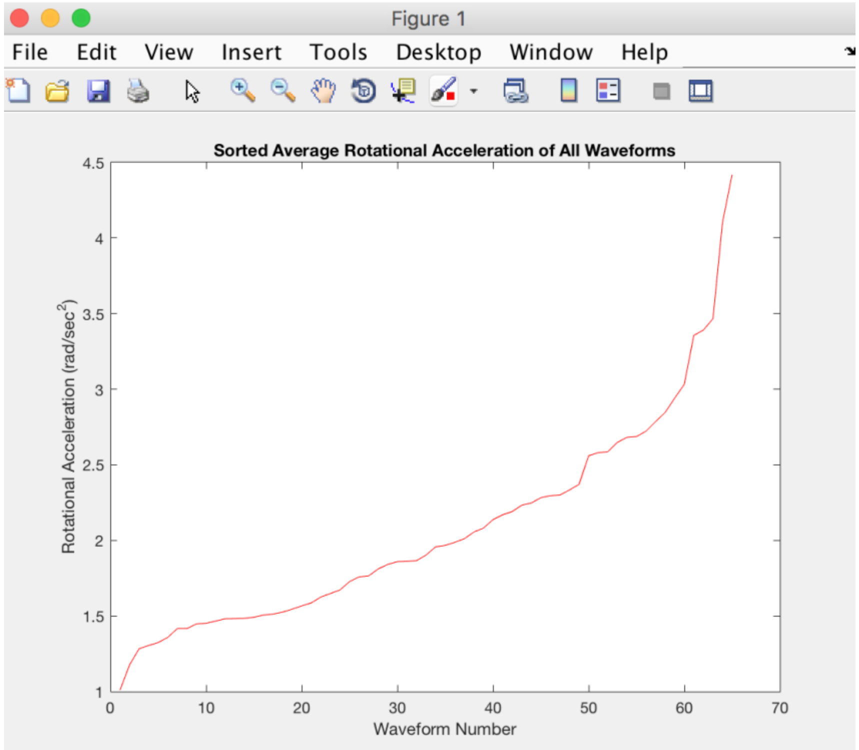


Take a look at the summary statistics for each of these features. Specifically, note the range of values (minimum to maximum) of each feature. Plot each of the features on their own plots and visually compare them. The plotting you do is up to you.



Above is an example of a feature plot, which in this case is the average rotational acceleration (ARA). Note the range of values goes from ~1 rad/sec^2 to ~4.5 rad/sec^2. How does this range compare with other features? Perhaps it's easier to look at the data if it's sorted.

Try using the sort command and then plotting the feature data. Note that when you sort data, you need to keep track of which index each instance was prior to sorting. Use Matlab's help to see how you can keep track of the original indices that the data came from. You'll need to know this when you report which data instances are different than the rest.



For ARA, sorting the data does not seem to be particularly helpful. The range of the data for this feature is very small (~3.5 units) and is probably not going to be of use in determining which instances are different than others. Do this same analysis for all of the features, keeping in mind the range of each feature's data and if any of the data instances stand out from the rest.

After visualizing all of the features, only a subset of them will appear to be helpful. In the table on the next page, note whether or not each feature should be selected for further analysis. Using the selected features, note the five data instances that you found to be different than the others and the features and their respective values that you used. Indicate the data instances in the table below by their original waveform indices and only include values for features that allowed you to differentiate these instances from the others. If a feature you selected was not actually helpful, it should not be mentioned in the table.

R•I•T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Sciences & Technology

Data Scavenger Hunt Results

Feature	Selected (Yes/No)
MLA	No
ALA	No
PLA	Yes
MRV	No
ARV	Yes
PRV	No
MRA	No
ARA	No
PRA	Yes

Data Instance Index	Feature PLA	Feature ARV	Feature PRA
8	110.9405		27.9306
9	131.7017	20.2883	44.5424
11	130.7306		25.1077
19	127.1141		30.2862
20	151.7148	23.3975	23.0736

Assignment

Title: Data Scavenger Hunt
Due: By the specified due date for your section in the Assignment 2 dropbox on myCourses.

Deliverables:
A write-up of your response to the instruction questions in MS Word format (no PDFs). Be sure to include your name within your submission. Use the following naming convention for your file before posting:
[your_last_name]_[first_ initial]_assignment_2.doc (or docx). For example: Golen_E_assignment_2.doc.

Instructions:

Write a one page[1] essay where you describe why you chose three of the original nine features and which two of those features were the most useful in determining the five data instances that were "different". You may optionally use a plot of the features to help with your discussion. You must show a plot of one of the "different" data instances and a plot of a "normal" data instance and compare them to each other visually. Specifically, you should describe the particular features that make them distinct from one another.

[1] 1 page is defined as 12-point font, 1.5 lines spacing, 1" margins.