Noise: A Study in Random Uncertainty

1 Learning Objectives

At the conclusion of this activity you should be able to:

- Calculate the mean, standard deviation, and standard error (or standard deviation of the mean) for a sample.
- Explain the difference between the standard deviation and the standard deviation of the mean (SDOM) of a sample.
- Experimentally determine the accuracy and precision of each axis of an accelerometer.

2 Background

2.1 Pre-lab Reading

The pre-lab quiz for this experiment is mostly based on readings from John R. Taylor's "An Introduction to Error Analysis". **Please read Chapters 4.1 through 4.5.**

Key concepts for this lab: *e.g.* average, standard deviation, and standard deviation of the mean (SDOM) are introduced and explained in the reading.

2.2 Useful Laboratory Statistics

2.2.1 Average

The average[1] of N measurements is given by:

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \ldots + x_N}{N} = \frac{\sum x_i}{N} \tag{2.1}$$

and describes our best estimate of the quantity being measured.

In Python, the average of an array * can be found using the `mean()` method. For example:

```
myAverage = myList.mean()
```

*The data types for arrays can occasionally be tricky. Columns selected from Pandas DataFrames are of type Pandas Series. The example shown here works on Pandas Series. If you run into data-type trouble, you can convert a standard Python list into an array using the numpy function: `asarray()`.

2.2.2 Standard Deviation

The standard deviation is used to describe the spread observed in a sample of N measurements. Mathematically, the standard deviation σ_x is given by:

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum\left(x_i - \bar{x}\right)^2}. \tag{2.2}$$

The standard deviation not only describes the spread of an entire sample but it can also be used to estimate the uncertainty associated with an *individual* measurement. That is:

$$\delta_{x_i} = \sigma_x. \tag{2.3}$$

In Python, the standard deviation of an array can be found using the `std()` method. For example:

```
myStdDev = myList.std()
```

2.2.3 Standard Error or Standard Deviation of the Mean

In most cases, as more measurements are made, we expect that the precision of the best estimate for the value of interest should improve. This trend is typically not observed in the standard deviation alone. Instead a quantity called the *standard error* or *standard deviation of the mean* is used.

The standard error[2] describes the uncertainty associated with our best estimate for a given quantity x. Our best estimate is often the average, $\bar{x}$. The standard error of the mean, $\sigma_{\bar{x}}$, is given by:

$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{N}} \tag{2.4}$$

where σ_x is the standard deviation of the sample and N is the total number of measurements.

A Pythonized version of Equation 2.4 can be calculated using the `std()` and `size` methods and `numpy.sqrt()` function. For example:

```
myStdErr = myList.std()/numpy.sqrt(myList.size)
```

The important feature of the standard error of the mean is that as more data are taken (*i.e.* N increases), the standard error of the mean, $\sigma_{\bar{x}}$, decreases. Large samples lead to a higher degree of confidence in the final measurement. Note that Equation 2.4 only applies if the variable being measured is truly random. For example, measuring the length of nail with a ruler an infinite number of times will not yield a measurement with zero uncertainty.

When stating a result, that is the result of an average, it is often appropriate to quote the standard error of the mean as the uncertainty. That is, the final result will be presented as:

$$\bar{x} \pm \sigma_{\bar{x}}. \tag{2.5}$$

2.3 Accuracy and Precision

Accuracy and *precision*[4] are two words that are often used to mean similar things in common conversation. However, in the physics lab, they have separate and distinct meanings.

Accuracy is generally used to describe how close a given measurement is to an accepted or true value. In many "real world" experiments the "true" value is not known making it difficult to quantify how accurate a given measurement may be. Accurate measurements have small *systematic* uncertainties. Thus, the goal of most scientific experiments is to identify, quantify, and control systematic uncertainties.

Precision is generally used to describe the repeatability of a measurement. Precise measurements have small *random* uncertainties. Precision also may be referred to as the *resolution* of a measurement or an instrument.

The ideal scenario is a measurement that is both accurate and precise. It is however possible for a measurement to be both precise and inaccurate or, conversely imprecise and accurate.

2.4 HISTOGRAMS[3]

A histogram is a type of chart that is useful for representing the frequency of measurements graphically. Histograms are often used to illustrate how data are distributed.

Python makes it very simple to plot a histogram. For a given series, you can make a histogram using the `matplotlib` function, `hist()`. For example:

```
import matplotlib.pyplot as plt
plt.hist(myDataSeries)
```

See the Python Help section on Blackboard for more detailed examples on how to visualize data using histograms:
Bb ▸ `PythonTutorials` ▸ `4 Plotting` ▸ `6 Histograms.ipynb`

2.5 `for` LOOPS IN PYTHON

`for` loops give you the ability to easily perform repeated operations in your code.

A simple example is shown in Figure 2.1. More detailed examples are given in the Python Help section on Blackboard. Look under:
Bb ▸ `PythonTutorials` ▸ `2 Python Basics` ▸ `4 Loops.ipynb`

```
In [1]: OddNumbers=[]              # define an empty list named OddNumbers
        for i in range(0,10):      # initialize the for loop.  i will be set to 0,1,2,..9
            OddNumbers.append(2*i+1) # append values to our empty list

        print(OddNumbers)          # the loop is done.  print the values in the list

        [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

Figure 2.1: An example `for` loop.

2.6 ACCELEROMETERS

Modern accelerometers are packaged in small integrated circuit packages called Microelectromechanical Systems (MEMS). Accelerometers are used in nearly all smartphones. For example, the accelerometer is used to tell your phone whether to display an image in portrait or landscape mode depending on the orientation of the phone.

Accelerometer MEMS contain small springs attached to small "proof masses" to measure accelerations. Generally, the accelerometer can measure the acceleration in three axes simultaneously. Figures 2.2 and 2.3 show microscope pictures of an accelerometer.

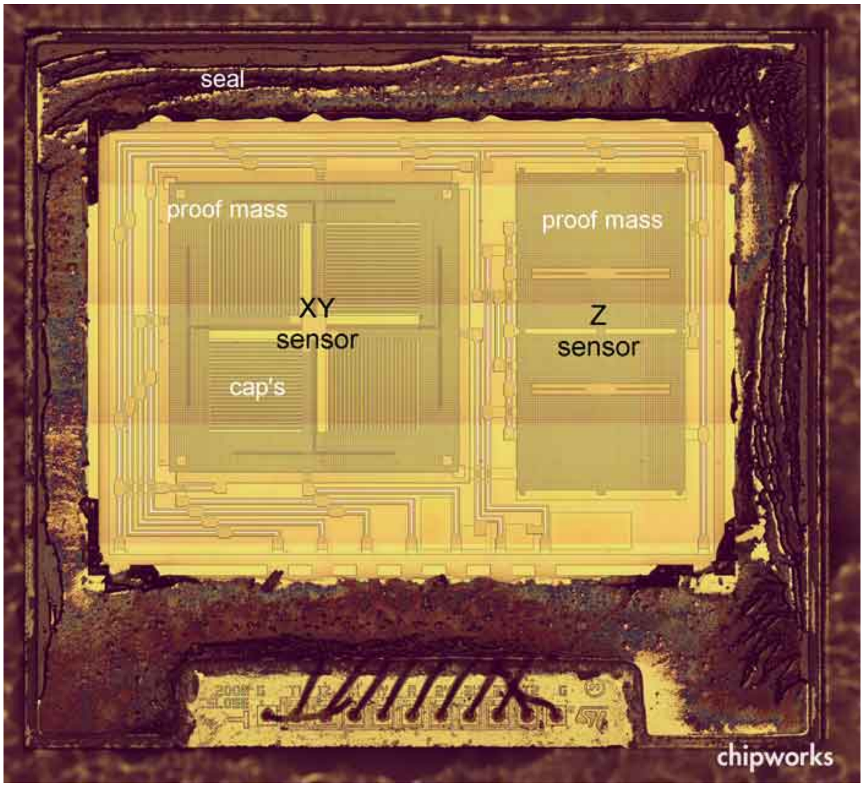

Figure 2.2: A microscope picture of an accelerometer MEMS chip. For this picture, the chips packaging has been removed. The motion of the proof mass is measured using capacitors (labeled "cap's" in the image). The x and y accelerations are measured using one system and the z direction is measured with another. Image Credit: https://memsblog.wordpress.com/2011/01/05/chipworks-2/

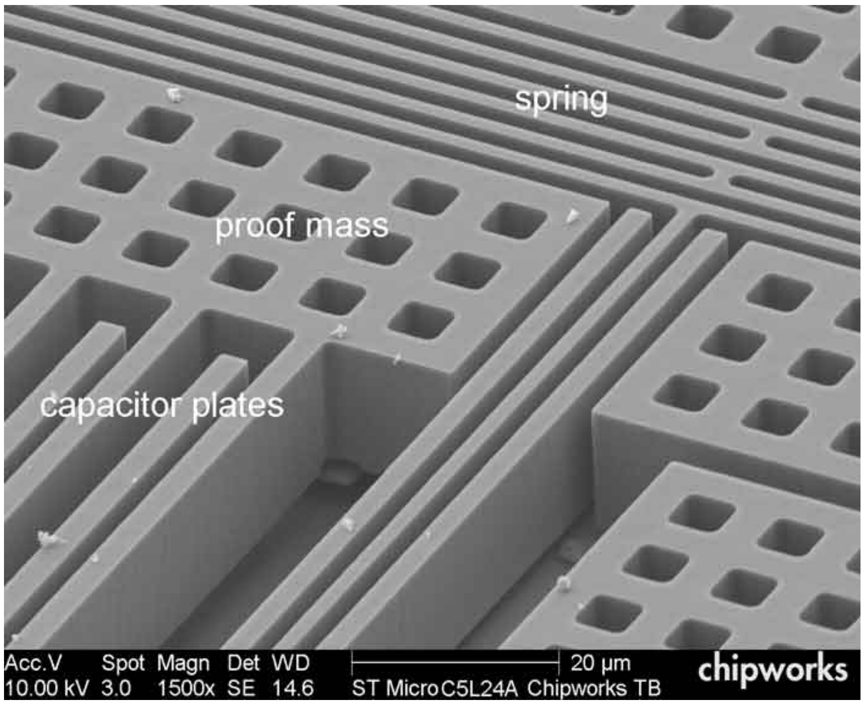

Figure 2.3: A close-up microscope picture of the etched spring and proof mass in an accelerometer MEMS. Motion of the proof mass – under acceleration – changes the spacing between the capacitor plates which can be measured electrically. Image Credit: https://memsblog.wordpress.com/2011/01/05/chipworks-2/

3 PROCEDURE

Your goal in this lab is to characterize the accuracy and precision of the three-axis accelerometer MEMS in your mobile phone.

3.1 SOFTWARE YOU WILL NEED

To make this measurement, you will need to install the "Physics Toolbox Sensor Suite" app on your mobile phone.

Please see the "Physics Toolbox User Guide" for links to the app and detailed information on installing and using the Physics Toolbox app for making measurements and exporting the data.

3.2 TAKING DATA

Take "g-Force Meter" measurements with your phone in three, perpendicular orientations (i.e. flat on a table, standing vertically on its short end, and vertically on its long end). Take at least 10 seconds of data in each orientation.

3.3 SPECIAL IOS INSTRUCTIONS

The data file created by the Physics Toolbox on an iOS device requires some special attention before it can be analyzed. If you took your data using an iOS device, please see the special instructions for reading your data:
`Bb` ▸ `PythonTutorials` ▸ `Physics Toolbox` ▸ `iOSDatestringPhysicsToolboxCSV.ipynb`.

3.4 ANALYSIS SUGGESTIONS

- Plot the acceleration measurements for each axis of the accelerometer as a function of time.
- Create a histogram of the acceleration measurements for each axis.
- Calculate the mean, standard deviation, and standard deviation of the mean for each axis.
- Use a plot to determine the resolution of the accelerometer (smallest step size between two measurements).
- Slice your data into 20 slices. For example, if you have a sample of 10,000 measurements, the first slice will be measurements 1 through 500, the second slice will be measurements 1 through 1000, etc. Note that each slice will be progressively larger until you are looking at all of the data. Calculate the mean acceleration, standard deviation, and standard deviation of the mean for each slice.
- Using the calculated values above, generate a plot of the mean acceleration (with standard deviation error bars) vs. the number of measurements (slice).
- Using the calculated values above, generate a plot of the mean acceleration (with standard deviation of the mean error bars) vs. the number of measurements.

4 LAB NOTEBOOK

Your submission will be evaluated using the following rubric:

LAB NOTEBOOK PRACTICES

- Lab Notebook Mechanics (6 points)
  - Relevant information *e.g.*: your name, your lab partner's name, date, *etc.* is present.
  - The notebook is organized and easy to read. Markdown cells are used for narrative text. Code cells are clearly organized and commented.
  - The ZIP file of the notebook is healthy and runs correctly.
- Data Analysis & Plots (6 points)
  - The notebook tells a scientific story; it is an accurate record of the work that you did.
  - The notebook should show evidence of trial and error. Keep a good record of your work – recording mistakes is useful.
  - Record rough data and plots that you used to verify that the analysis was on the right track. Final versions of plots should be well formatted and meet the plotting guidelines for the course.
  - Use models to identify trends that your data exhibit or other apparent relationships between your independent and dependent variables.
- Results and Comparison (6 points)
  - Clearly state the final result(s) of your experiment. Remember to quote your result with units and appropriate significant digits.
  - Final result plots are well formatted and meet the standards described in the Figure Formatting reference.
  - A useful comparison is made to a known/expected value or another similar result.
  - Choose the best available tools for your comparison (*e.g.* plots, pictures, discrepancy, significance of discrepancy, etc).
- Uncertainty and Error Propagation (6 points)
  - Identify the dominant source(s) of error in your experiment.
  - Support your conclusions with appropriate error estimates and error propagation calculations.
- Physical Interpretation (6 points)
  - Throughout the notebook, interpret the data, rough plots, and final results in terms of the underlying physics.
  - What are you able to conclude from your data? Clearly explain how you arrived at your conclusions from your experimental observations.
  - Reflect on how your experiments connects with the physics concepts you are studying.

ACKNOWLEDGMENTS

Martiń Camilo Monteiro Trabal from the University ORT Uruguay kindly shared his ideas about statistical measurements with smartphone sensors. This lab is based largely on his initial idea. See `https://fisicamartin.blogspot.com/2019/02/medidas-estadisticas-con-sensores-de.html`.

REFERENCES

[1] Taylor, J. R. (1997) *An Introduction to Error Analysis.* 2nd ed., Sausalito, CA: University Science Books. Chapter 4.2 (pp. 97-101).

[2] Ibid. Chapter 4.4 (pp. 102-106).

[3] Ibid. Chapter 5.1 (pp. 122-126).

[4] Ibid. Chapter 4.1 (pp.94-97).