

Calculating the Acceleration of Gravity Using an Accelerometer Data Logger

1 Introduction

Occasionally, it's a good idea to check some of the fundamental laws of physics for validity. You never know if those physicists made a mistake or the nature of the universe recently took a turn for the worse. Here, a USB Accelerometer made by Gulf Coast Data Concepts was used to determine the periodicity of a simple gravity pendulum and to calculate acceleration of gravity.

2 Objective

Calculate the acceleration of gravity using a simple gravity pendulum.

3 Test Setup

First, several assumptions simplified the experiment into an easily modeled system. A thin string and a large mass pendulum diminished the affects of friction and air drag. Also, a long pendulum arm and small swing angle allowed the motion to be approximated as a simple harmonic motion. The oscillation period was simplified to the following formula:

$$T = 2\pi\sqrt{\frac{l}{g}} \quad \text{Equation 1}$$

Where l is the length of the pendulum between the pivot point and center of gravity, g is the gravitational acceleration, and T is the period of one complete swing.

Solving for g :

$$g = \frac{l}{\left(\frac{T}{2\pi}\right)^2} \quad \text{Equation 2}$$

For this experiment, a large lead weight with an eyehook served as the pendulum bob. The USB accelerometer was attached to the weight using the ubiquitous, but universally practical, duct tape such that the x-axis pointed inline with the pendulum arm. The weight was connected to the pivot point with 50lb test Berkley "Gorilla Tough" fishing line. This type of fishing line is very strong and does not stretch much when loaded. An eyehook attached to a high ceiling worked as the pivot point. The fishing line was attached to the pivot and the weight such that the weight hung several inches above the ground. The length between the center of the weight and the pivot point was measured to be 99 inches, or 8.25 feet.

Using the Java program “*XLR8R*” provided by Gulf Coast Data Concepts, the USB Accelerometer was configured with the following settings:

```
gain = low  
deadband = 0  
samplesperfile = 10000  
statusindicators = Normal
```



Figure 1: Supplies for Experiment



Figure 2: Experiment Setup

4 Test Procedure

After assembling the pendulum and ensuring the USB Accelerometer was operating (blinking blue LED), the pendulum was put into motion by pulling the weight back several feet while maintaining the string taut and releasing the weight. The pendulum was allowed to swing for several minutes then stopped.

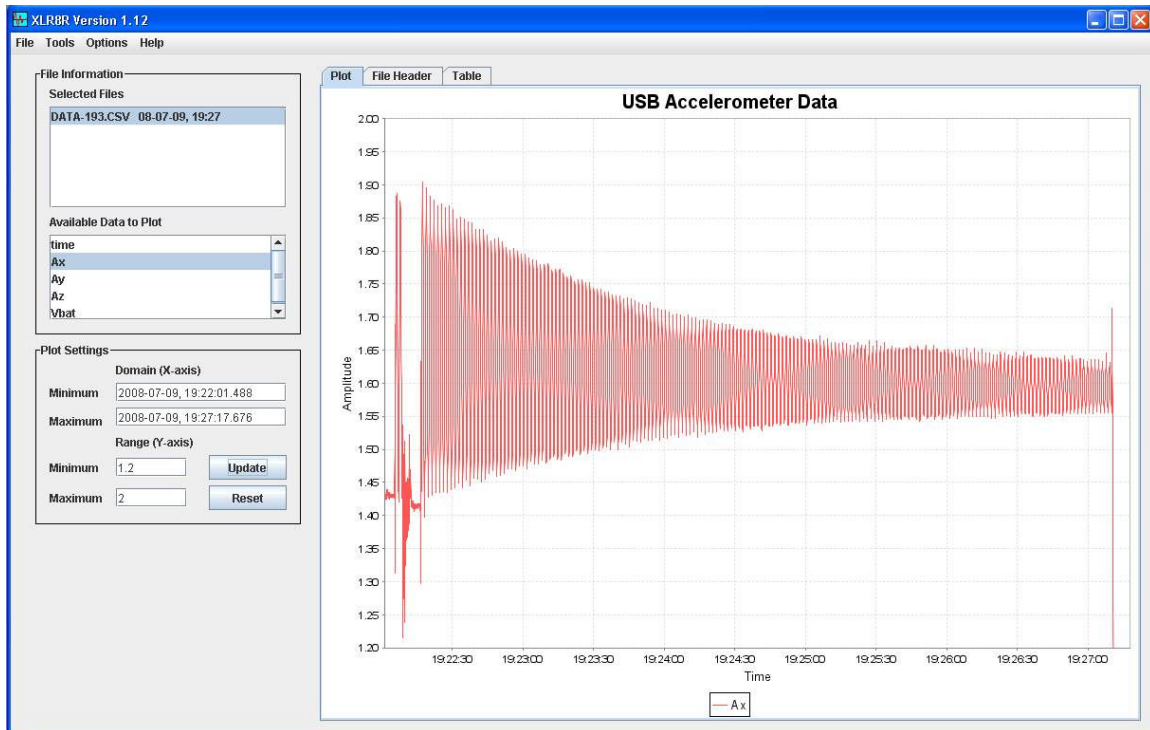


Figure 3: Test Data Presented in XLR8R

The data was reviewed using the *XLR8R* application and a 60-second segment of x-axis data was exported for further analysis. Two methods were used to determine the pendulum's period of oscillation.

First, a Fast Fourier Transform (FFT) converted the time domain data to frequency domain. The result is a plot showing the dominant frequency within the data (this is similar to the spectrum analyzer used by music players). The FFT was performed using "R", which is a free open source mathematics package available at www.r-project.org. The R script used is presented in the Appendix.

The second method for determining the pendulum's period wasn't as fancy but was very effective. Plotting the 60-second segment of data in *XLR8R*, the number of cycles observed was counted and divided by the time frame.

5 Results

The results of the *R* analysis was imported into Microsoft Excel and plotted as a scatter plot. The dominant frequency was determined to be 0.640 hertz. There are two acceleration peaks per cycle since the pendulum swings past center twice. Therefore, the frequency of the pendulum is half that of the FFT result, or 0.320 hertz. The resulting pendulum period was 3.125 seconds.

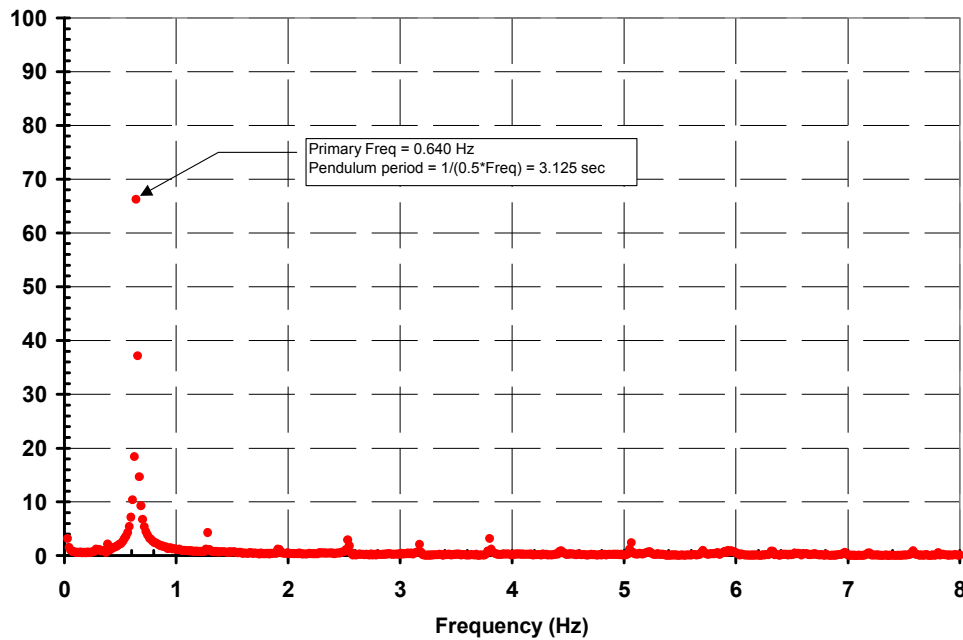


Figure 4: FFT plot

Calculating gravity using the 3.125 second period and Equation 2:

$$g = \frac{8.25 \text{ feet}}{\left(\frac{3.125 \text{ sec}}{2\pi}\right)^2} = 33.35 \text{ ft/sec}^2$$

Counting the cycles observed over a 60-second time frame resulted in a frequency of 0.626 hertz. Again, the pendulum oscillation frequency is half that, or 0.313 hertz. The resulting pendulum period using this method was 3.194 seconds.

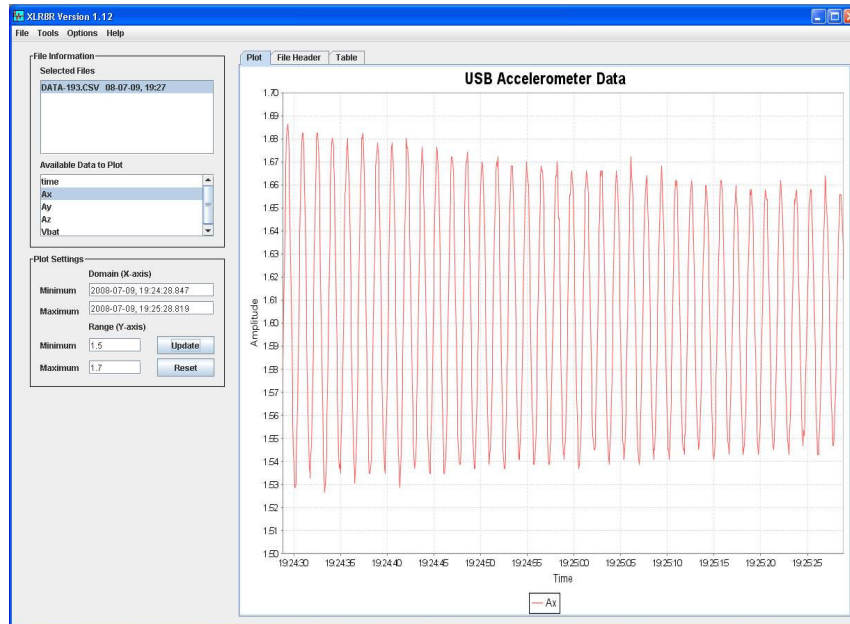


Figure 5: 60-second Segment of Data

Calculating gravity using the 3.194 second period and Equation 2:

$$g = \frac{8.25 \text{ feet}}{\left(\frac{3.194 \text{ seconds}}{2\pi} \right)^2} = 31.93 \text{ ft/sec}^2$$

6 Discussion

Using an FFT to determine the frequency of the pendulum should have resulted in a very accurate assessment of the pendulum's period. However, the gravity constant calculated suggests that FFT provided a period slightly faster than actual. A second FFT was performed using another algorithm separate from *R*, which resulted in a more accurate gravity constant (31.80 ft//sec²). *R* is a widely accepted mathematics program in the scientific research community so these results should not indict the *R* FFT algorithm. More likely, the FFT may not have been properly implemented in this experiment.

The simple and straightforward method of counting cycles from a graph to determine the period was more accurate than the FFT method.

7 Conclusion

Calculating the acceleration of gravity to within 5% of actual using such a crude experiment is impressive. Furthermore, we can conclude that the gravity still works and all is well in the universe (at least in Waveland, MS).

8 Appendix: Script code for R

```
#Author: Gulf Coast Data Concepts
#Date: July 9, 2008
#File Name: "pendulum.r"
#R script for processing pendulum data
#reads data from the file "pendulum.csv"
# - the acceleration data must be in the first column
# - data rate must be 16 hertz
#exports the FFT data to the file "pend_out.txt"
#to run this script, type "source("pendulum.r") at the 'R' command prompt

#read the data from "pendulum.txt"
#comma delimited text, lines starting with ";" are ignored
data<-read.table("pendulum.txt", sep=";", comment.char=";")

#perform FFT on second column of data and store results in data.fft
data.fft<-fft(data[[2]])

#calculate the magnitude of the FFT data and store results in data.mag
data.mag<-Mod(data.fft)

#we want the first half of the FFT, which is 0 to 8 hertz
data.mag.half<-data.mag[1:(length(data.mag)/2)]

#this code builds an array of numbers representing the freq spectrum
x.axis<-1:length(data.mag.half)/(length(data[[1]])/16)

#combine the results into a 2 column array of numbers
#each row will be x.axis versus the FFT magnitude
output<-array(c(x.axis, data.mag.half), dim=c(length(x.axis),2))
#uncomment the following line to have R plot the data
#plot(output, type="l", xlim=c(0,8), ylim=c(0,100), xlab="Frequency",
ylab="Magnitude")
#xlim and ylim define the range and domain limits

#write the results to the output data file
#this file can be imported into Excel and plotted as an XY scatter plot
write.table(output, "pend_out.txt")
#NOTE: zero hertz is the DC offset. This value may be really big compared to the
#rest of the results, depending on the data processed. Ignore the DC offset and
#look for the first peak, which is the primary frequency. Subsequent peaks are
#harmonics of the first peak. The acceleration will peak twice as the pendulum
#swings so the pendulum period is actually half that of the primary frequency
#calculated here.
```