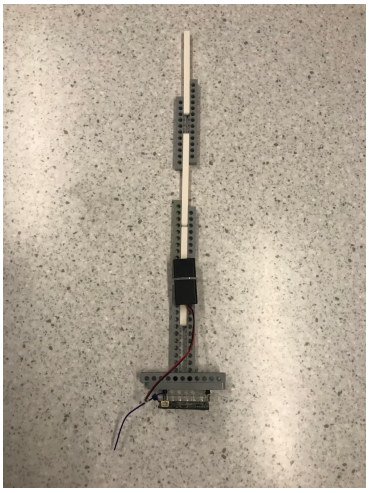
Andres Hernandez
Cheonan Kougba
Project 1
ES2

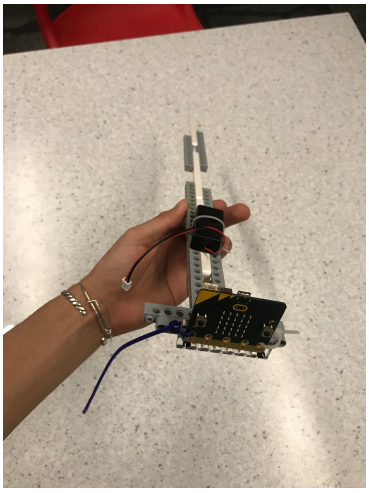
Pendulum Project Report - ES2

Introduction

In this project, the physical properties of a pendulum were examined by using micro:bit accelerometer data to extract significant values from the motion of a pendulum. This included acceleration in x y and z dimensions and angular position of the pendulum in degrees. In order to do this, multiple scripts were written in order to perform each respective step of the process. Firstly, a script was written to allow the micro:bit to take in data, writing this data to a text file which could later be analyzed. The second script was used to parse this data file and extract information from it. The Numpy module was also used in this script to produce plots of time vs angular position and time vs angular acceleration, both of which were filtered using the Scipy median filter function. This was done to better analyze the extrema of the plots so that the period of the pendulum could be extracted from the data. This process was repeated for 2 lengths of pendulum.

The setup for the micro:bit pendulum is pictured below:



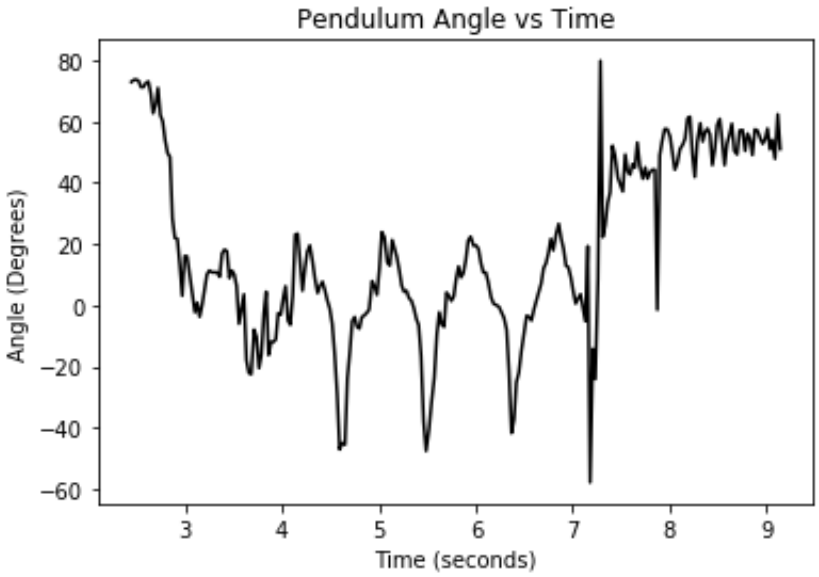


The micro:bit was attached perpendicular to the pendulum to adapt to the code written, which took the angle of the micro:bit as 0 when it was perpendicular to the pendulum, which was the desired value when measuring angles in this case. The micro:bit was attached using zip ties, and the pendulum was rotated using a straight gray lego piece (Picture 3), which allowed for smooth swinging with minimal friction. This allowed us to take a better average of the pendulum's period and obtain a better numerical estimate.

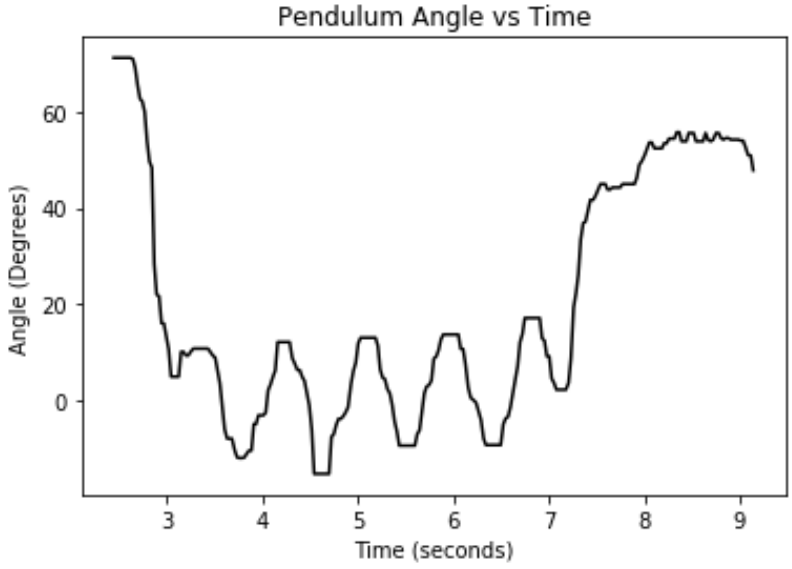
Results

Pendulum 1
Rod length: 0.25 meters
Avg Determined period: 0.88 seconds
Theoretical period: 1.00 seconds
Real vs Theoretical deviation: 0.12 seconds

Unfiltered Raw Data:



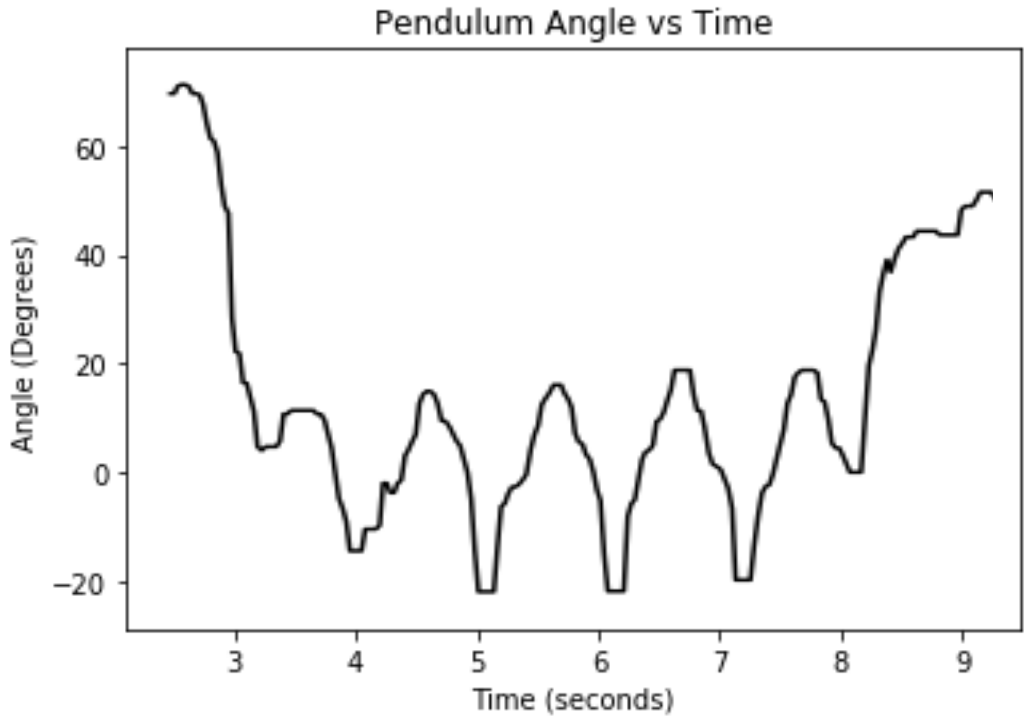
Median Filtered Data:



Pendulum 2

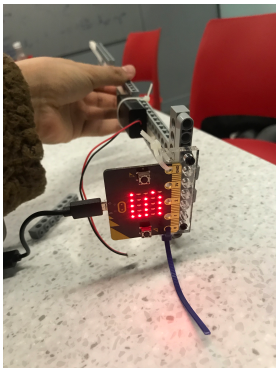
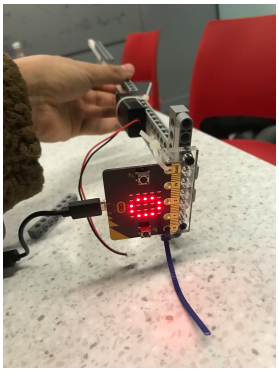
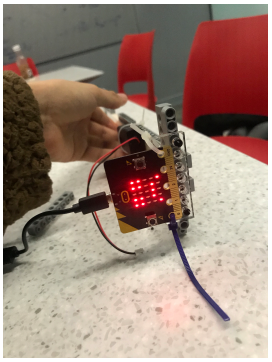
Rod Length: 0.55 meters
Avg Determined Period: 1.23 seconds
Theoretical Period: 1.49 seconds
Real vs theoretical deviation: 0.26 seconds.

Median Filtered Data:



After two trials, the micro:bit began showing “error 50” and displaying a frown face when the mu editor accelerometer code was flashed to it. Under micro:bit error codes at https://support.microbit.org/support/solutions/articles/19000016969-micro-bit-error-codes, it can be seen that the error has to do with the accelerometer component of the hardware.

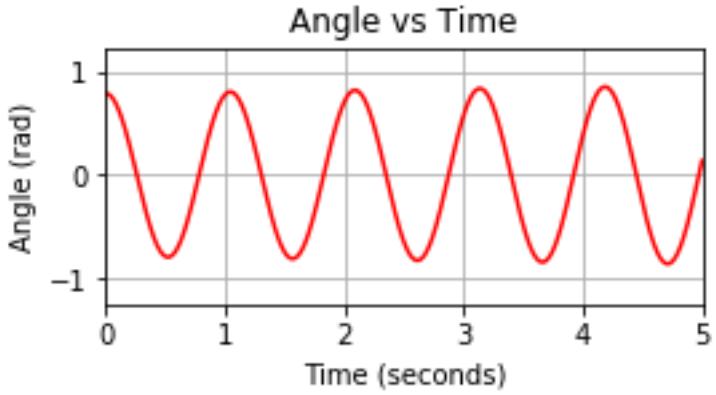
50	An error occurred with the micro:bit's accelerometer component.

Since only one micro:bit was at our disposal, this error prevented us from acquiring more data for different lengths, something that was out of our control. Fortunately, we were still able to extract meaningful data at 2 different pendulum lengths.

Simulated Data

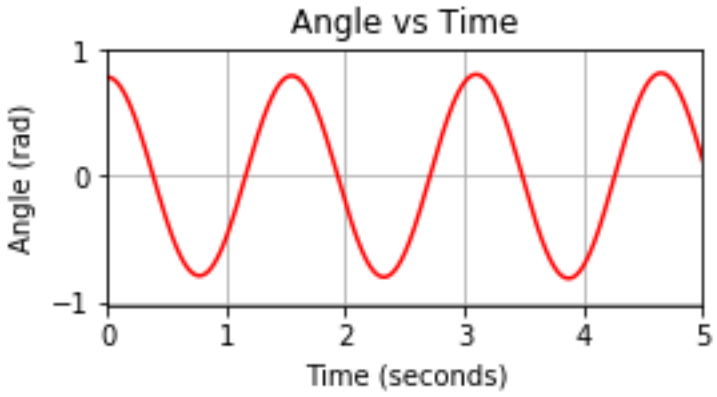
Pendulum 1
Rod length: 0.25 meters
Avg Determined period: 0.88 seconds
Theoretical period: 1.046 seconds
Real vs Theoretical deviation: 0.125 seconds




```
The period for the length 0.25 meters is 1.0460523026151307
seconds
```

Pendulum 2:

Rod Length: 0.55 meters
Avg Determined Period: 1.23 seconds
Theoretical Period: 1.55 seconds
Real vs theoretical deviation: 0.32 seconds




```
The period for the length 0.55 meters is 1.5500775038751937
seconds
```

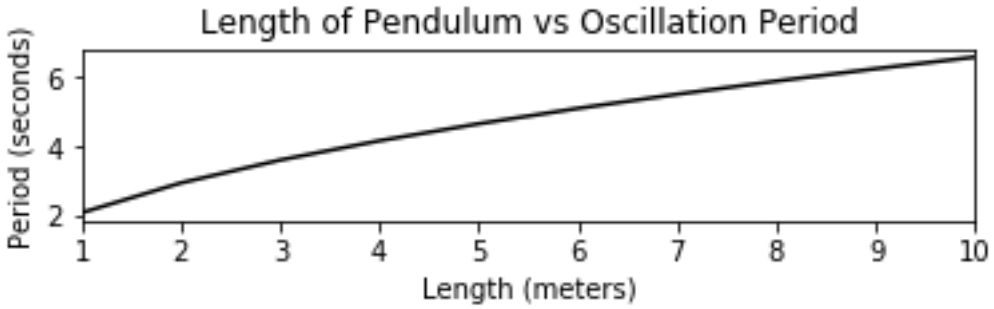
Discussion

The period of the pendulums were calculated using the spicy module find peaks. This module distinguishes peaks from the rest of the data to a specified prominence; that is, only returning the peaks that are a certain minimum distance away from the rest of the data. This reduced the amount of wrongly found peaks returned. This returned an array of indices. Since the data has a 1:1 ratio, finding the value of time for these same indices returned the time in seconds at which the peaks occurred. After this, the average was found by finding the separation in seconds between successive peaks and dividing this by the total amount of peaks.

Real world vs simulated comparison:

As can be seen from the real world data and the simulated data for respective pendulum lengths, micro:bit accelerometer data and parsing was relatively successful in determining the period of the pendulums. To calculate a form of uncertainty, the absolute difference between the micro:bit value and the theoretical values was calculated. In both examples, the difference turned out to be 11.9% and 20.6% of the theoretical period value, respectively. This is a significant error, and can be attributed almost entirely to the noise/error in accelerometer values from the micro:bit. Other possible errors could be physical errors in the setup of the pendulum. Lego pieces are not the most physically reliable objects to base accurate measurements around, and unbalanced forces caused by pieces out of place or pieces moving during a pendulum swing could have minimally contributed to errors in data collection.

Another valuable piece of information extracted from the project was the relationship between the length of the pendulum and the period of oscillation. The simulation script was used in order to plot a graph of length vs period:



As can be seen, the length is proportional to the square root of the period. This plot makes sense as the theoretical equation for period is:

$$T = 2\pi\sqrt{\frac{L}{g}}$$

This suggests that the script is computing periods correctly, or at least with the correct relationship between length and period. Since our micro:bit restricted us from taking more than 2 sets of data, we were not able to produce a plot of length vs period for real world data. However, based on the analysis of the error percentage of the micro:bit, it would be expected that this plot would 'noisily' resemble the simulation plot above.

Conclusion

In conclusion, this project was not only successful in illustrating how to use a micro:bit as a data reader, but it also helped us gain knowledge in various different areas of python and computation in general. For example, data parsing and creation of simulations that model real real world data, and incorporating mathematical models into programs like using accelerometer values to compute pendulum angle. It was also valuable to work with arrays, as they prove more useful for certain mathematical operations that lists fail to provide. Modules like Scipy were also useful in showing how data can be analyzed using existing functions built into python documentation, and will prove useful in the future when analyzing more complex physical systems involving motion. The project was also a useful first connection between coding and real world engineering problems, and showed the usefulness of being acquainted with computational methods.