

# Lab 2: Q Factor

Robert Purcaru

October 27, 2020

## Contents

<b>1</b>	<b>Auhtor's Note</b>	<b>1</b>
1.1	General	1
1.2	Note to Nathan	2
<b>2</b>	<b>Experimental Design</b>	<b>2</b>
<b>3</b>	<b>Data</b>	<b>6</b>
<b>4</b>	<b>Sources of Error</b>	<b>10</b>
4.1	Measurement Errors	10
4.2	Analysis Errors	10
<b>5</b>	<b>Q Factor</b>	<b>11</b>
5.1	Counting Cycles	11
5.2	Determining $\tau$	11
<b>6</b>	<b>Relationship Between Period and Amplitude</b>	<b>12</b>
<b>7</b>	<b>Appendix</b>	<b>13</b>
7.1	Calculations	13
7.2	Programs Used	13

## 1 Auhtor's Note

### 1.1 General

This report features information pertaining to both labs 2 and 3. Modifications and additions to existing sections are shown as **red text** embedded into the existing section, while new sections have the section header written in red while the body is black. Note that small or stylistic changes - like corrections in grammar or minor clarifications - are left in black. New Figures are shown with red captions.

Figures 9 and 10 show the graphs of quadratic and cubic power series. The captions are very similar just because the graphs depict very similar things, I included both to illustrate that continuing down the power series doesn't bring us closer to a non-zero C. A justification is included in section 6

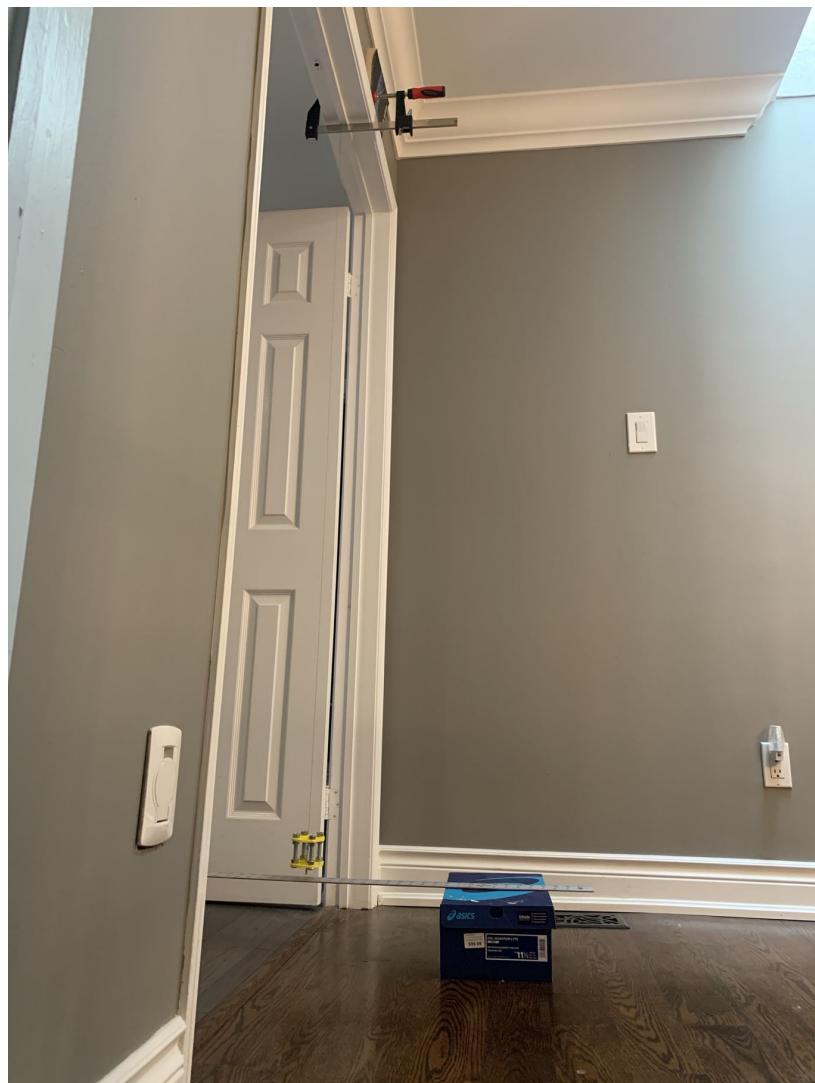
## 1.2 Note to Nathan

In my initial pendulum design, it proved impossible to capture a 90 degree range of motion because the camera was looking down at the pendulum (this was the only arrangement that worked as a result of the narrow space I have to work with). In an attempt to rectify this, I tried to manually find the horizontal distance the pendulum travelled using a ruler fixed against the backdrop and a camera that I moved as the amplitude damped, recording the position every 5 oscillations. However, since I only have one ruler long enough to both measure the distance and be visible on camera, the affects of parallax became increasingly significant as the distance between the peak of the swing and the ruler increased. Since the ruler is about 6cm wide, and the vertical displacement of the mass is more than 25 times greater than that, the measurement uncertainty became more significant than the statistical uncertainty reported by the regression, making the results statistically zero. These results were excluded because the calculations to convert measured horizontal displacement needed to be tuned in a way that I could not justify with integrity to get the expected result. My results will therefore only include data obtained in the original trial.

If this were to be attempted again, the measurement of the angle could be performed by placing the ruler near the pivot and measuring the points where the string can be seen overlapping with the ruler on the top and bottom of the ruler, followed by the use of trigonometry (given that the distance between the top and bottom of the ruler is a known quantity). This could potentially the measurement uncertainty down enough to allow us to consider our result statistically non-zero.

## 2 Experimental Design

The pendulum used in this experiment was constructed by fixing an adjustable G clamp to the top of a doorframe and suspending the mass from the throat of the clamp using a length of fishing line tied in fixed loop (bowline) at one end and attached to the weight at the other (Figure 1). This ensured a large range along which the length of the string could be adjusted and a wide field of view for the camera to capture. To prevent damage to the doorframe and allow for an increased clamping force, a pair of books were used between the clamp and the door frame. The clamp was tightened as much as was possible by hand to minimize any unwanted movement in the setup. The string length can be varied by using a shorter or longer piece of fishing line attached to the mass and the clamp in the same way.



**Figure 1:** Experimental setup used, excluding tripod and recording device. Left half of setup is obscured by wall.

The mass used was composed of four 3/8-inch x 2-1/2-inch steel carriage bolts attached to a pair of 3D-Printed plates, one of which received an additional 3/16-inch x 1-1/2-inch steel eyebolt used to attach the mass to the end of the string. This design was chosen because it allowed for the location the string is mounted to to be varied continuously along the axis of the eyebolt, meaning it can be adjusted so that the string is closer to the center of mass for any number of carriage bolts. The mass can be adjusted by either removing or adding carriage bolts in pairs of two, giving four different configurations of the mass (Figure 2). Four carriage bolts were used for this experiment, despite the capacity for eight, because only four were available at the time the experiment was conducted. If more variation in the mass is required, larger plates can be printed to accept more carriage bolts or additional weights can be fixed to the part of the eyebolt the extends beyond the lower plate. The fishing line used was chosen because of its high tensile strength to weight ratio.



**Figure 2:** All combinations of bolts used to vary mass available with plates that were used for this experiment. The mass with four carriage bolts (second from the right) was used in this experiment.

Before the mass was attached to the string, the position of the eyebolt was adjusted until the mass could be balanced side ways on the side of a pencil. However, once the mass was attached to the string, this configuration proved very unstable; the mass would rock about the point the string was attached to violently and irregularly. To address this, the eyebolt was raised by about  $6 \pm 0.5$  threads on the eyebolt, corresponding to  $0.50 \pm 0.02$  inches, or  $1.27 \pm 0.05$ cm. The string was then attached before its length was measured several minutes later to allow the pendulum to stabilize and to account for any strain in the string lengthening it. The length of the string was measured using a soft tape measure and found to be  $184.5 \pm 0.1$ cm between the top of the clamp and the nearest end of the interior of the loop of the eyebolt. However, since the eyebolt was raised by  $1.27 \pm 0.05$ cm, the distance between the center of mass and the pivot in the neutral position can be calculated as  $184.5 \pm 0.1 + \frac{1.27 \pm 0.05}{2} = 185.1 \pm 0.1$ cm, assuming the change in position of the center of mass is negligible from moving the eyebolt.

The masses of the mass and string were measured using a kitchen scale, giving  $269.8 \pm 0.1$  grams for the mass and  $0.2 \pm 0.1$  grams for the mass of the string. Since the mass of the string is three orders of magnitude smaller than the mass of the mass, we can consider it negligible for the purposes of this experiment. It was also found that the grooves cut into the throat of the G clamp were sufficient to prevent excessive lateral movement of the string while the experiment was being conducted (Figure 3).



**Figure 3:** View of string hanging between teeth on clamp.

Finally, a measuring stick was placed underneath the pendulum and aligned with the eyebolt so that the 0 centerline marker lined up with the neutral axis of the pendulum (Figure 4). The video was taken from a top down view at a resolution of 1080x1920 pixels, where the pendulum swings along the axis of the 1920 pixels.



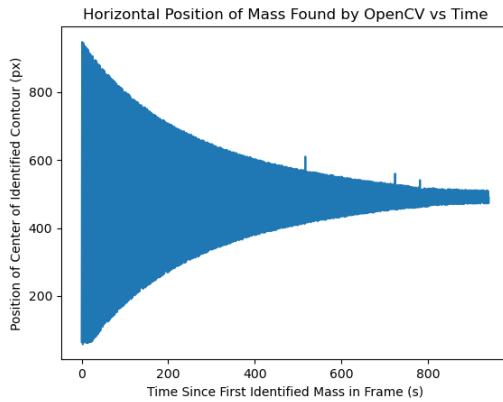
**Figure 4:** Side view of mass zeroed against ruler.

### 3 Data

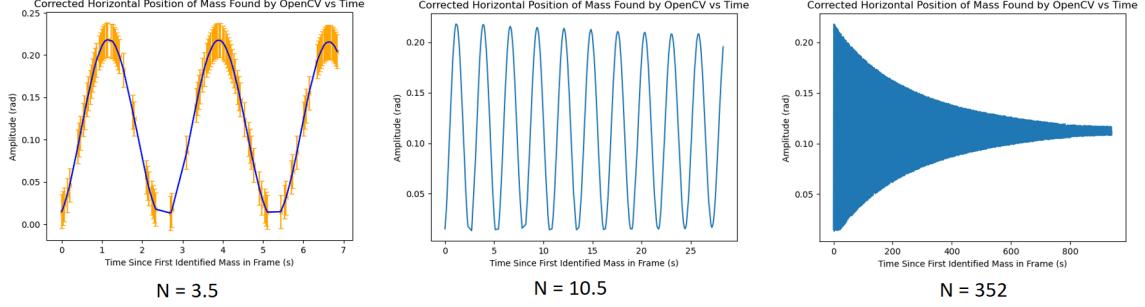
To collect data, a video was recorded by mounting an iPhone X to a tripod and placing it close to the string approximately, 1.3 meters above the mass, giving a top down view of the pendulum. This video was then analyzed in python using the OpenCV library [1]. The program used looks at every frame in the video for the specified color, approximates the shape formed by the color to a circle, and reports the pixel position of the center of the circle. (Figure 5).



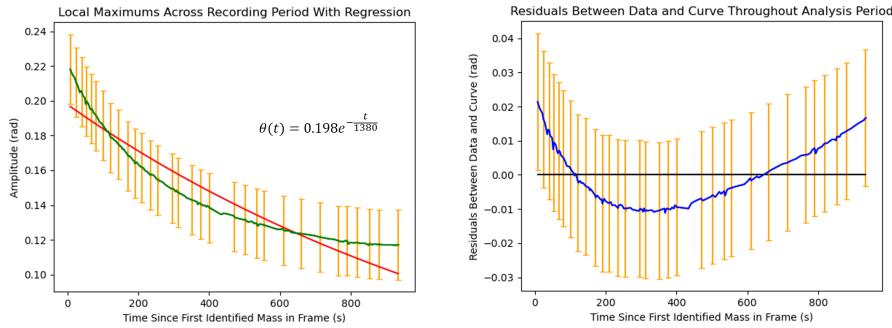
**Figure 5:** Screenshot of thresh (left) and contour (right) views provided by OpenCV during video. The thresh view shows which areas of the frame are within the specified colour threshold determined in advance by identifying the RGB values of the yellow mass. The contour shows the circular approximation made by OpenCV in green and the center of the circle in blue. The position in pixels from the top of the frame is recorded every time the mass can be identified.



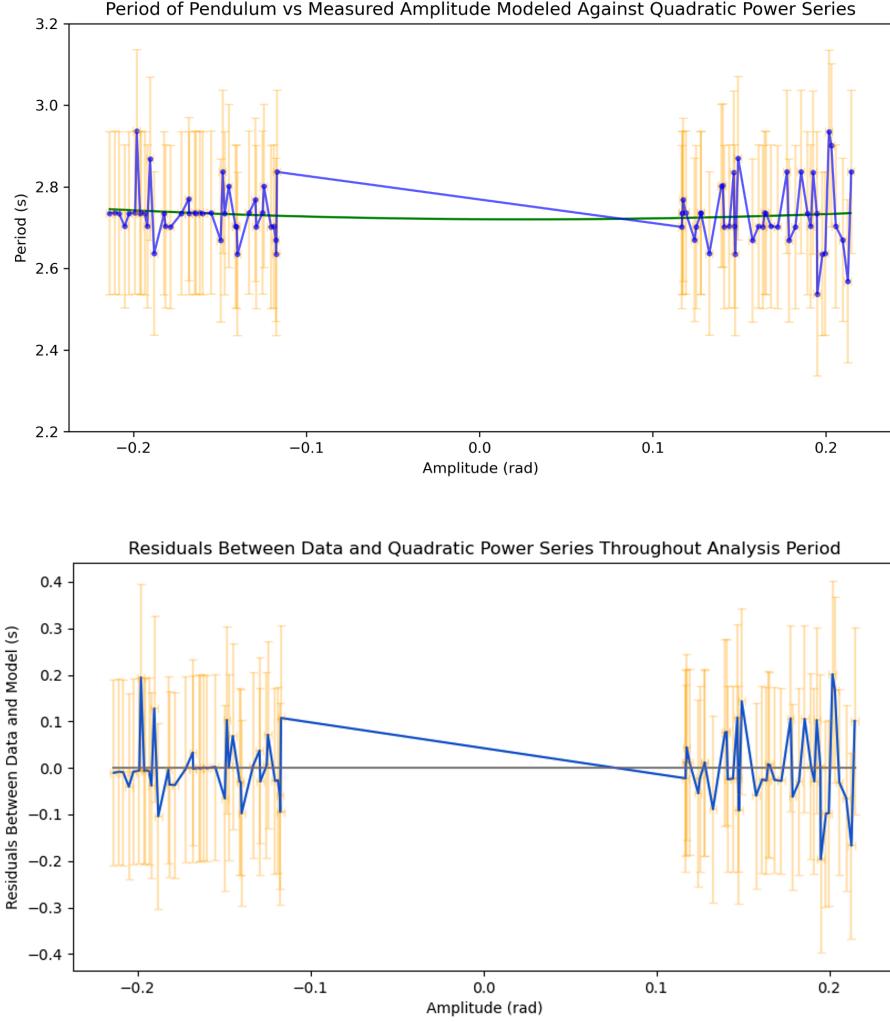
**Figure 6:** Raw data as mapped by OpenCV at specified frame. Video was recorded at 29.97 frames per second. Three distinct outliers can be seen around 520 seconds, 730 seconds, and 790 seconds. These occur because a foot entered the frame at those times, causing a tracking error in OpenCV. Error bars are excluded from this graph because data from OpenCV is interpreted with uncertainties in Figure 7; this graph only depicts information developed by OpenCV.



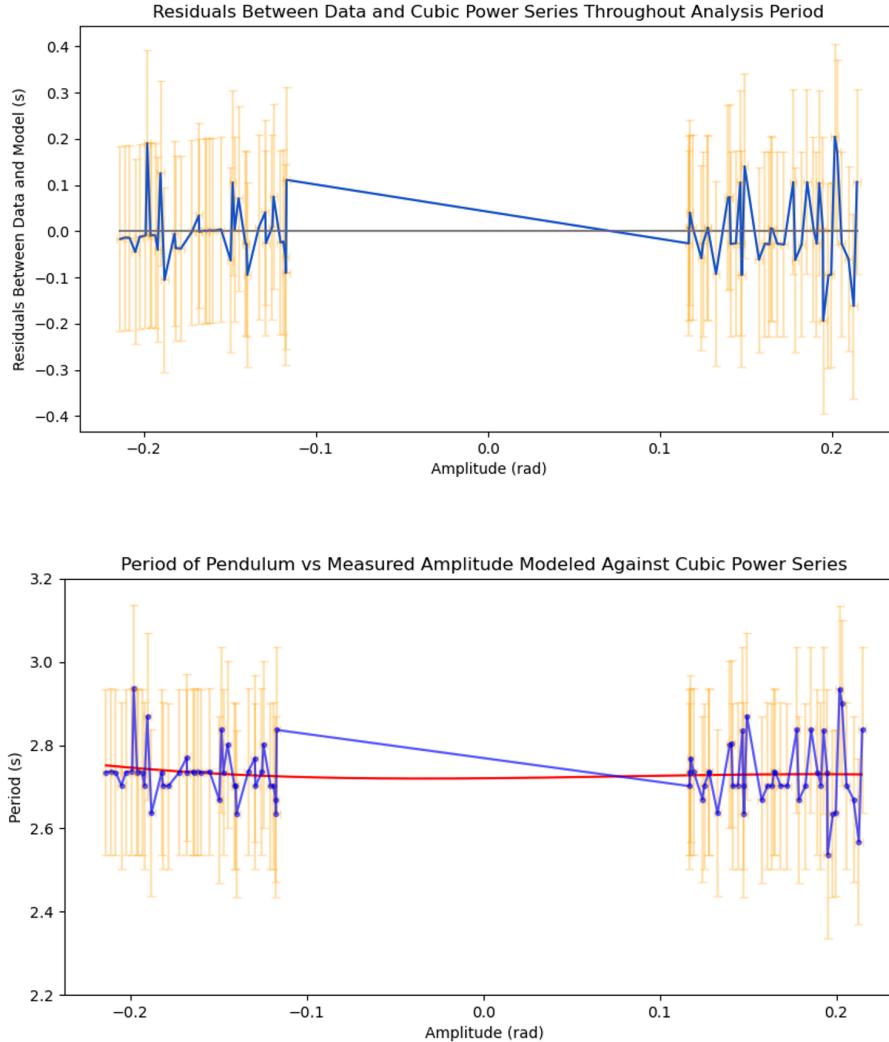
**Figure 7:** Data collected at specified frame, correcting for outliers and tracking errors after approximately  $N$  cycles. The graph featuring 3.5 cycles (left) depicts vertical error bars marking an uncertainty of  $\pm 0.02$  radians (see 4.2) and horizontal error bars marking  $\pm 0.02$  seconds (half of a frame). Graphs for 10.5 and 352 cycles (center and right) are shown to better illustrate the decay of the amplitude of the pendulum. It is worth noting that during the first few cycles the mass goes out of frame at its maximum amplitude on one side. This can be seen in the apparent sharp edges in the early troughs of the graph depicting 10.5 cycle and the 'blunt' edge of the bottom left side of the graph depicting 352 cycles.



**Figure 8:** Graph depicting relative maximums at specified times as determined by Open CV (green) with overlaid regression (red) in the form  $\theta(t) = \theta_0 e^{-t/\tau}$ , where  $a = 0.198$  and  $\tau = 1380$  (left). The graph depicting the residuals (blue) between the data and the regression (right) illustrates some significant discrepancy between the model and the measurement (a black line at 0 rad discrepancy is used to help illustrate this). This is further discussed in 4.2. Both graphs feature error bars (orange) every 5 data points, illustrating a measurement error of 0.02 radians in the amplitude. Note that there are some large gaps between error bars due to problems determining a local maximum for certain times as a result of measurement errors.



**Figure 9:** Graph depicting period of pendulum as a function of amplitude, shown in blue, modeled against regression of quadratic power series in the form  $T = T_o + B\theta_0 + C\theta_0^2$  [2], shown in green, (top). Uncertainties, shown in orange, associated with the period (vertical axis) are the measurement uncertainty of 0.2 seconds and the uncertainty in the amplitude (horizontal axis) is the measurement error of 0.02 rad (see 4.1 for explanation of both). The line of best fit has parameters  $T_o = 2.72$ ,  $B = -0.02$ , and  $C = 0.4$ . The values were found using the optimize library in SciPy [3]; a discussion of the uncertainties associated with these values can be found in section 6. The 'jump' in the graph between about -0.11 rad and 0.11 rad is a result of the recording being stopped before the pendulum reached rest. Otherwise, the data used was recorded in the same manner as described in section 2. The second graph (bottom) shows the residuals between the curve and the data. It can be seen that all of the data points fall within the uncertainty, meaning we can consider the result to be statistically zero for the amplitudes used when modeled against a quadratic power series. A calibration line at 0 is provided in gray.



**Figure 10:** Graph depicting period of pendulum as a function of amplitude, shown in blue, modeled against regression of cubic power series in the form  $T = T_o + B\theta_o + C\theta_o^2 + D\theta_o^3$  [2], shown in red, (top). Uncertainties, shown in orange, associated with the period (vertical axis) are the measurement uncertainty of 0.2 seconds and the uncertainty in the amplitude (horizontal axis) is the measurement error of 0.02 rad (see 4.1 for explanation of both). The line of best fit has parameters  $T_o = 2.7$ ,  $B = 0.04$ ,  $C = 0.4$ , and  $D = -2$ . The values were found using the optimize library in SciPy [3]; a discussion of the uncertainties associated with these values can be found in section 6. The 'jump' in the graph between about -0.11 rad and 0.11 rad is a result of the recording being stopped before the pendulum reached rest. The second graph (bottom) shows the residuals between the curve and the data. It can be seen that all of the data points fall within the uncertainty, meaning we can consider the result to be statistically zero for the amplitudes used when modeled against a cubic power series. A calibration line at 0 is provided in gray.

## 4 Sources of Error

### 4.1 Measurement Errors

The pendulum and its amplitude was supervised throughout the experiment. The exposed portion of the eyebolt was used as a reference to find the position of the pendulum against the ruler underneath the pendulum. The ruler was zeroed within 0.0625 inches (Figure 4) so that any error resulting from an offset as the ruler was read could be minimized. Still, the measurement error here is significant. Given that the ruler was being read while the weight was moving, an uncertainty of about  $\pm 0.3$  inches seems reasonable, as this gives 4 increments in both direction to account for parallax and reading error at the start of the experiment. As the experiment proceeded, and the pendulum slowed down, it became easier to read the ruler, thereby increasing the confidence of the measurement and decreasing the error to about  $\pm 0.1$  inches, or about 2 increments on the ruler in both directions. This corresponds to 6 pixels, and using Equation 9, we find that the uncertainty in the amplitude is 0.02 radians, or about 1% of the maximum amplitude. This explanation is confirmed with the results obtained in equation 2

### 4.2 Analysis Errors

The OpenCV library doesn't explicitly provide any information for uncertainty or accuracy of the values provided. However, from experience, OpenCV provides the center of a relatively good circle to within about a pixel. Since the circle identified was broken by the bolts (Figure 5), the error is likely around 8 pixels at the start when the pendulum is moving at its fastest, and decreasing as it slows down, resulting from motion blur. Using Equation 9, we obtain a value of  $\pm 0.02$  rad for our uncertainty in the reported amplitude of the pendulum. As the pendulum slows and motion blur become less significant, this value likely decreases, however we will use  $\pm 0.02$  radians as it is the largest source of uncertainty we find. This is a significant amount and should be addressed in future experiments. The measurements can be made more accurate and more frequently if OpenCV can identify the mass more consistently. To do this, a cover that is an unbroken color should be made to obscure the nuts on the top of the mass. This will allow the OpenCV to develop a better approximation of the pendulum, thereby reducing the error. It would also be worth trying a view perpendicular to the axis of rotation ('in front' of the pendulum) as this would largely eliminate measurement errors due to parallax.

For the purposes of measuring the relationship between period and amplitude, we must consider how the period is measured. The data collected by OpenCV often reported the mass as near stationary (the change in position between frames was less than 3px) for up to six frames at a time (0.2 seconds), during which the reported position would vary unpredictably around some average value. This happened most commonly during the later stages of the experiment when the pendulum was moving more slowly. In order to determine the amplitude, I took all of the points where the mass was reported as 'stationary' and averaged the position that was reported along with the corresponding time stamps. The period was then determined by taking the difference between neighboring time stamps. The error in the reported period is therefore  $\pm 0.2$  seconds, corresponding to the greatest number of near stationary frames that were reported. The only way this could be improved is with a higher definition recording device that could give OpenCV more pixels to analyze and therefore better report the position of the pendulum.

## 5 Q Factor

### 5.1 Counting Cycles

By watching the first cycle and later reviewing the video manually, it was found the initial amplitude was  $13.75 \pm 2\%$  inches, as read on the ruler bellow. This corresponds to about  $905 \pm 2\%$  pixels of the video and  $0.22 \pm 2\%$  rad. If we consider the parallax error in reading the ruler negligible, we find that the amplitude corresponding to  $e^{-\tau}$  of 14.0 inches is about  $0.605 \pm 2\%$  inches. This corresponds to about  $41 \pm 2\%$  pixels. Counting in real time, I found that the amplitude reached 0.605 inches after 328 complete oscillations. Reading the raw data, we find that the amplitude reaches 4% of its original amplitude after  $332 \pm 1\%$  cycles, corresponding to 15 minutes and 5 seconds of video time. The uncertainty was adapted from the raw data, where the position was reported by openCV to 10 decimal places on a linear pixel scale of 1 unit to 0.1 pixels. Taking the width of the frame (1920px), we find an uncertainty that rounds to 1%. While these two values are very close and within the uncertainty of the other, the uncertainty of reading the ruler is greater than the uncertainty using OpenCV; the ruler used only has marks every 0.0625 inches, which must be judged to line up with the center of the weight in real time. The data generated by the program on the other hand has an estimated error of 8 pixels, which likely decreases as the pendulum speed decreases. We will therefore consider  $332 \pm 1\%$  as the Q factor determined by counting the oscillations until 4% amplitude is reached.

### 5.2 Determining $\tau$

To affirm our value of Q obtained by counting, we can consider the formula

$$Q = \pi \frac{\tau}{T} \quad (1)$$

where  $T$  is the period of one complete oscillation and  $\tau$  is the time constant of decay, as provided by [2]. To find the period, we can take the average of the differences of neighboring maximums to get 2.73s. To find the error in this measurement, we can take the standard deviation of the difference of consecutive maximums, giving us 0.07s. Therefore, the period of one oscillation is  $2.73 \pm 0.07$  seconds. To find a value for  $\tau$ , we can use the formula

$$\theta(t) = \theta_o e^{-t/\tau} \cos(2\pi \frac{t}{T} + \phi_o) \quad (2)$$

where  $\theta(t)$  is the amplitude in radians as a function of time,  $\theta_o$  is the initial amplitude in radians, and  $\phi_o$  is the phase constant that accounts for the difference between when the analysis starts and the pendulum starts moving, as provided by [2]. If we only consider the maxima of our data,  $\cos(2\pi \frac{t}{T} + \phi_o) = 1$ , so we are left with

$$\theta(t) = \theta_o e^{-t/\tau} \quad (3)$$

The maximums calculated and the line of best fit approximated using Equation 3 are shown in Figure 8. The value for  $\tau$  used in the approximation is  $1380 \pm 3\%$  seconds. This is provided by the square root of the first element in the covariance matrix give by the curve fit function in the SciPy library [3]. However, it is clear from the graph that this approximation is very rough. Keeping that in mind, the value we calculate for using this method is  $1590 \pm 3\%$  (Equation 8 in 7.1). This value may be excessively large as a result as the relatively small original amplitude; about  $0.218 \pm 3\%$  radians (about 12.5 degrees). A small amplitude like this leads to a decreased maximum speed and, assuming air resistance, given by the equation:

$$D = C d \rho \frac{v^2 A}{2} \quad (4)$$

taken form [4], is the largest slowing factor, the pendulum would be impacted less by air resistance at slower speeds ( $v$ ). However, this may also be a result of an analysis error. Future experiments will attempt to reduce the uncertainty resulting from analysing the data using methods described in 4.2.

## 6 Relationship Between Period and Amplitude

Using the SciPy optimize library [3], I performed a regression on my data using both quadratic and cubic power series, seen in figures 9 and 10. The parameters and the respective uncertainties determined for the quadratic power series in the form

$$T = T_o + B\theta_0 + C\theta_o^2 \quad (5)$$

are:  $T_o = 2.72 \pm 0.02$ ,  $B = -0.02 \pm 0.04$ , and  $C = 0.4 \pm 0.8$ . From this, we find that both  $B$  and  $C$  can be described as statistically zero. Since  $B$  is statistically zero, the pendulum is likely not significantly asymmetric at the amplitude range tested. Furthermore, since the value of  $C$  is statistically zero, we can say that for the amplitude range tested, amplitude and period are independent. Therefore, in future labs, this range will be used to measure the relationship between different variables in the setup.

Similarly, when I performed the cubic regression with the function

$$T = T_o + B\theta_0 + C\theta_o^2 + D\theta_o^3 \quad (6)$$

I got parameters  $T_o = 2.72 \pm 0.02$ ,  $B = 0.04 \pm 0.2$ ,  $C = 0.4 \pm 0.8$ , and  $D = -2 \pm 5$ . Again,  $B$  and  $C$  are statistically zero, meaning the conclusion we developed using the quadratic equation is corroborated by the cubic regression.

Both quadratic and cubic regressions produce a value of  $2.72 \pm 0.02$  seconds for the period of the pendulum. However, the measurement uncertainty of 0.2 seconds is greater. Therefore the period of the pendulum in the range of tested amplitudes is  $2.7 \pm 0.2$  seconds, or  $2.7$  seconds  $\pm 7\%$ .

The ranges shown in figures 9 and 10 were determined by the number of oscillations measured. While initially, I planned to include as many oscillations as were performed in my Q factor experiment, it became clear from the data collected during the Q lab that measuring the amplitude of smaller oscillations would be significantly more challenging compared to measuring the amplitude of larger oscillations. This was a result of the amount of time it seemed the pendulum was still for small values of  $\theta$ . A balance seemed to exist around 0.1 radians of amplitude where the time it the mass seemed to be still was less than or equal to about six frames (about 0.02 seconds).

I chose to include the cubic regression to see if continuing down the power series would cause the value of  $C$  to converge, potentially giving a statistically non-zero result after a certain number of iterations. However,  $C$  remained statistically zero through the cubic power series, leading me to conclude that, within the range of amplitudes I tested my pendulum, period is independent of amplitude.

If this experiment were to be carried out again, I would modify my setup as described in section 1.2. I would expect that there would be some value for  $\theta$  where  $C$  is no longer statistically significant. Speaking to my colleagues, I would expect this value to be anywhere in the range of 0.35 rad to 0.90 rad. This number would indicate at what angle I could no longer consider the period and amplitude to be independent, meaning that future experiments would have to use amplitudes less than the determined value.

## 7 Appendix

### 7.1 Calculations

Approximation for Q using  $\tau$  obtained in Figure 8. Uncertainties were obtained from the covariance table that was obtained from the SciPy regression, and is propgated as the largest percent among the two.

$$Q = \pi \frac{2.73 \pm 0.07}{1380 \pm 35} = 1590 \pm 40 \quad (7)$$

$$Q = \pi \frac{2.73 \pm 3\%}{1380 \pm 3\%} = 1590 \pm 3\% \quad (8)$$

Conversion equation used to determine angle from position P, expressed in pixels. The camera was mounted such that the lense was within  $5.0 \pm 0.1\text{cm}$  of the string attatched to the mass. If we consider this distance to be negligible compared to the lenth of the aparatus, we find that the angle can be expressed as the arcsin of the horizontal position of the pendulum divided by the length of string. This also assumes that the difference between the mass and the measureing device is negligible (it was measured to be  $4.8 \pm 0.1\text{cm}$  at the maximum distance the camera could still find it's position.)

$$\theta(P) = \arcsin \frac{\frac{2.54\text{cm}}{\text{in}} \frac{31.875 \pm 0.25\text{in}}{1920\text{px}} P}{185.1\text{cm} \pm 0.1\text{cm}} \quad (9)$$

$$\theta(P) = \arcsin \frac{\frac{2.54\text{cm}}{\text{in}} \frac{31.875 \pm 1\%}{1920\text{px}} P}{185.1 \pm 0.1\%} \quad (10)$$

$$\theta(P) = \arcsin (0.000228 * P) \pm 0.1\% \quad (11)$$

### 7.2 Programs Used

The following is an excerpt of the program that tracks the mass. The program looks for the colors specified in the given frame, approximates the region the color is identified in with a circle and records the center of the identified circle.

```

def find_mass(frame, show_trackbars = False):
    frame = cv2.GaussianBlur(frame, (7, 7), 0)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    hul, huh, sal, sah, val, vah = 23, 40, 126, 255, 127, 255

    if show_trackbars:
        hul=cv2.getTrackbarPos(hl, wnd)
        huh=cv2.getTrackbarPos(hh, wnd)
        sal=cv2.getTrackbarPos(sl, wnd)
        sah=cv2.getTrackbarPos(sh, wnd)
        val=cv2.getTrackbarPos(vl, wnd)
        vah=cv2.getTrackbarPos(vh, wnd)

    HSVLOW = np.array([hul, sal, val])
    HSVHIGH = np.array([huh, sah, vah])
    thresh = cv2.inRange(hsv, HSVLOW, HSVHIGH)
    floodfill = thresh.copy()
    h, w = thresh.shape[:2]

```

```

mask = np.zeros((h+2, w+2), np.uint8)
cv2.floodFill(floodfill, mask, (0,0), 255)
floodfill_inv = cv2.bitwise_not(floodfill)
thresh_filled = thresh | floodfill_inv
edged = cv2.Canny(thresh_filled, 30, 200)
all_contours, hierarchy = cv2.findContours(edged,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

filtered_contours = []

for contour in all_contours:
    area = cv2.contourArea(contour)
    x, y, w, h = cv2.boundingRect(contour)
    if abs(max(w, h)/min(w, h)) < 1.5:
        if area > 100:
            filtered_contours.append(contour)
        break

if len(filtered_contours) < 1:
    return -1, -1, -1, thresh_filled

mass_contour = filtered_contours[0]
center, radius = cv2.minEnclosingCircle(mass_contour)

return int(center[0]), int(center[1]), int(radius),
thresh_filled

```

## References

- [1] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [2] B. Wilson, *PHY180 Lab Project (2020)*.
- [3] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [4] “The drag equation,” May 2015.