Periodic, Period Doubling, and Chaotic Dripping in a Leaky Faucet

A Senior Project

By

Max Bigras

Advisor, Dr. Glen D. Gillen

Department of Physics, California Polytechnic University SLO

March 17, 2016

Approval Page

Title: Periodic, Period Doubling, and Chaotic Dripping in a Leaky Faucet

Author: Max Bigras

Date Submitted: March 17, 2016

Senior Project Advisor: Dr. Glen D. Gillen

_____

Signature

_____

Date

Contents

List of Tables

List of Figures

1 Introduction

The Leaky Faucet Experiment was first performed by Robert Shaw in 1984 at UC Santa Cruz [1]. Since then it has become an iconic example of a simple system that can exhibit complex behavior. Interesting dripping patterns can be observed such as periodic, period doubling, and chaotic dripping. We built and performed the experiment, visually represent our data in a variety of graphs, and are able to confirm that our apparatus demonstrates periodic, period doubling, and chaotic behavior.

2 Theory

The leaky faucet system is a relatively simple one: a reservoir of water drips from a faucet with some adjustable flow rate (in our case a clamped hose). Water droplets build up at the tip of the faucet (in our case, glass dropper) and eventually break off.

One interesting quantity investigated in this work is to determine is the period between drops, T. In most cases we represent the data using T_n, where T_n is the time between the n^{th} drop and the previous drop. We can then display plots such as: T_n vs. T_{n+1}, a type of Poincare section plot, T_n vs. n, and the number of occurrences N vs T, a histogram plot. Between these three graphs we are able to identify which regions of behavior our apparatus is exhibiting for a given period and are also able to observe transitions made into different regions.

We are all familiar with the rhythmic periodic dripping of a leaking faucet, that is with T_n always equal. Perhaps less know is if the flow rate is very carefully increased the system will move into a period-2 or period doubling region, where $T_n = T_{n+2}$. This type of dripping may sound like two quick drips followed by a pause and then two quick drips again. Increase the flow rate further and the system will theoretically move into a period-4 region, then period-8, continuing double until the period eventually becomes chaotic. In practice we were only able to observe periodic, period-doubling, and chaotic regions.

3 Experiment

3.1 Building the apparatus

A schematic of the apparatus can be seen in Fig. 1. The major experimental components used in this investigation are listed in Tab. 1. Water is pumped from a reservoir up into a Weir cup using a fish pump. The Weir cup keeps the water at a constant level by allowing water to flow back into the reservoir through a tube running up the inside of the cup, this keeps the pressure inside the tube going to the dropper constant. From the Weir cup, water flows through another tube and out a glass dropper. We are able to carefully open and close a clamp around the tubing in order to adjust the frequency of the dripping. When a droplet falls from the dropper it scatters a laser beam normally incident upon a photodiode detector and then falls into another bucket.

The most important quantity for us to measure is the time of each water drop, then later we can calculate the period between drops. When a droplet falls it crosses the beam path between a laser and the photodiode detector, causing a voltage drop to occur. We can then register this voltage drop as an event using an Arduino running custom software.

A close up schematic of the setup of the Arduino, laser, and photodiode detector can be seen in Fig. 2. Getting the laser and photodiode detector pair calibrated required some adjustment. The detector continuously outputs an analog voltage value depending on the intensity of light hitting it; however, it can become saturated if the light is too intense, as is the case when you shoot the HeNe laser directly at the photodiode, so the voltage will not drop appreciably when laser beam is scattered for a split second by the water drop. To bring down this intensity and prevent saturation we used an ND 1.0 filter, decreasing the intensity by a factor of 10. At first we used a BNC coaxial cable to connect the photodiode detector to the Arduino's ground and analog input A0 pin; however, the Arduino was not able to get a meaningful reading from the detector. This was because of an impedance mismatch between the BNC output of the photodetector and the Arduino analog input. By using a 1 MegaOhm impedance oscilloscope probe we were able to match the oscilloscope probe's output impedance to the Arduino's expected input impedance and get a meaningful mV reading from the detector.

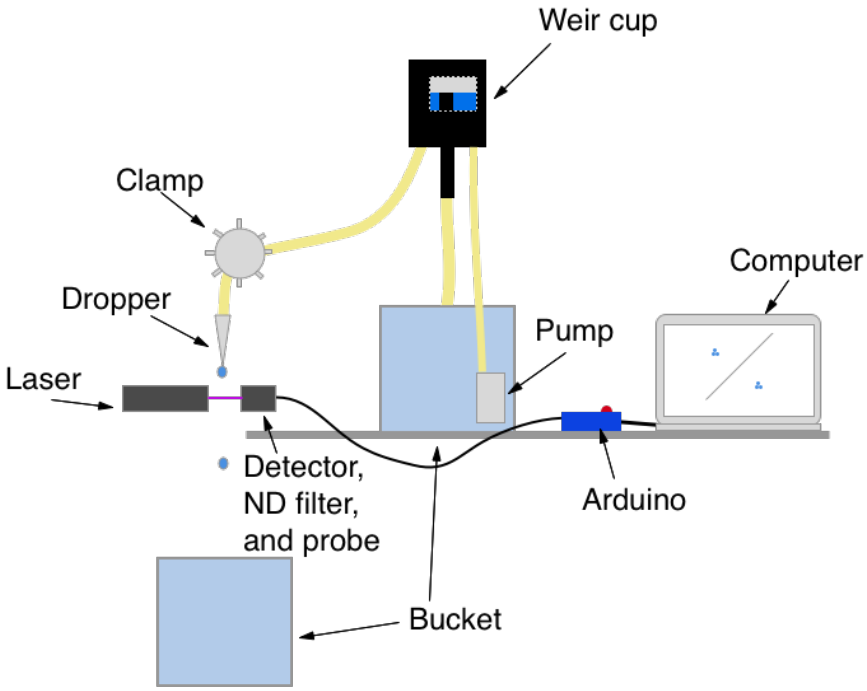

Figure 1: Leaky faucet experimental apparatus

Equipment	**Manufacturer/Model**
Weir cup	Central Scientific Company
HeNe Gas Laser	JDSU 1508P-1 633 nm
Aqua Pump	Rolf C. Hagen Corporation
Microcontroller	Arduino UNO
Photodiode detector	ThorLabs DET210
Oscilloscope Probe	Agilent 10073C
ND Filter	ThorLabs NE10A

Table 1: Equipment List

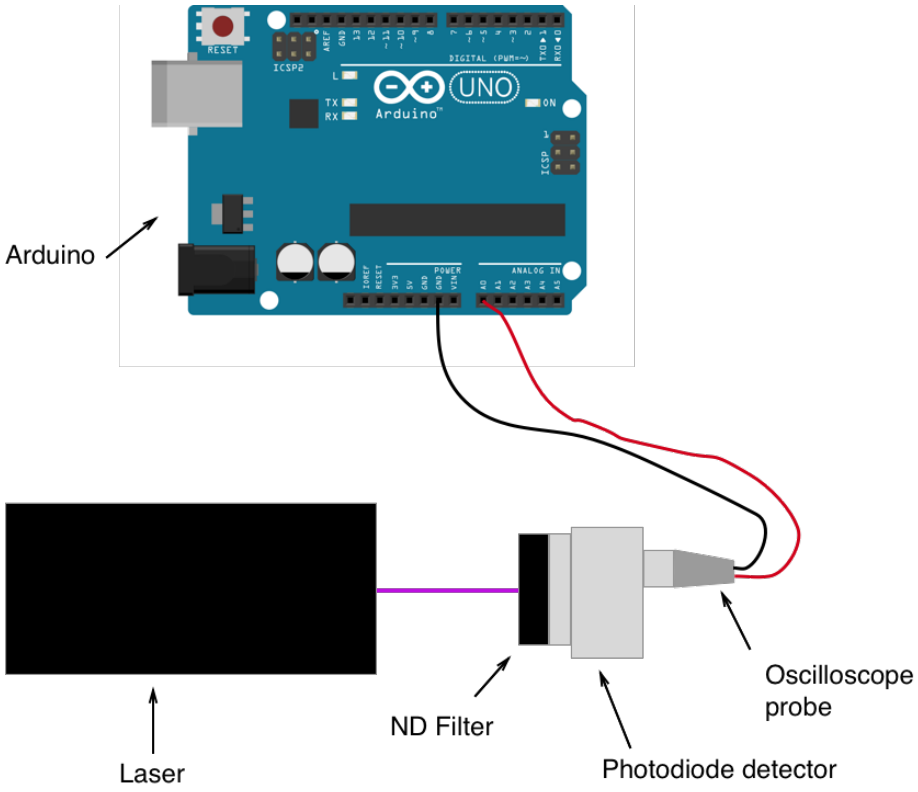

Figure 2: Close up schematic for Arduino, laser, photodiode detector, ND filter, and oscilloscope probe

3.2 Coding

The code for the Arduino can be found in Appendix A. As mentioned above, the most important piece of data for us to record is the time of each water drop so we can later determine the period between drops. The code is straightforward but worth going through carefully.

The `setup()` function on line 5 is run once when the Arduino first turns on. `Serial.begin(9600)` on line 7 is a function that tells the Arduino we would like to communicate with the Serial output using the 9600 band; this is what lets us interface the Arduino with the computer. Inside the `setup()` function we also initialize `time0` to be used later as the initial time to find the period.

The `loop()` function on line 11 is run over and over again as long as the Arduino is on. Lucky for us, the Arduino keeps track of the number of milliseconds that have passed since it was turned on which we access every time the loop is run with the `millis()` function on line 13. The Arduino gives us access to the analog voltage value being read on several pins of our choosing (in this case pin A0) using the `analogRead(pinNumber)` function on line 15. If pin A0 is reading above 500 mV (which it does when the laser is hitting the photodiode detector) then the loop will get skipped, but once the laser beam path is broken the detector will output a lower voltage, which will be read as less than 500 mV by pin A0 and the loop will execute; so it is really on line 15 where we are able to interface between the physical experiment, the Arduino, and the code, which makes line 15 my favorite line.

After stepping into the if-statement on line 15 there is some interesting behavior. Line 16 is where we interface between the Arduino code and Excel. We output the drop number and period to the Serial output; however, we output the data in such a way that it can be imported later into Excel so the data can be processed. Lines 16-17 update the drop number and the initial time to be used when calculating the next period. Finally line 19 addresses the speed at which the Arduino executes an iteration through the `loop()` function. Because the Arduino is able to run through the `loop()` function so fast it is possible that the loop will run twice per a single drop, thus it gets counted twice. Because we were measuring periods on the order of 50 ms, a delay of 10 ms, using the `delay(10)` function, gives a single drop time to make it all the way through the laser while still giving the Arduino time to recover and wait until the next drop, so the double counting problem is solved.

3.3 Running the experiment

Surprisingly, displaying useful processed data from the Arduino in real-time is not as easy and one may think. Theoretically, one should be able to send the raw data shown in the top left of Fig. 3 to another program such as Matlab or Processing, process the raw data to find the period, and then render a plot such as Fig. 4. Then one could easily adjust the clamp on the hose shown in Fig. 1 and find different regions of interest and toggle data capture on or off to be used later. In reality, the Arduino software doesn't play nice with other programs except it's own native Serial Monitor which can be seen in the top left of Fig. 3. While performing the experiment these numbers are often jumping across the screen and it's hard to get a sense of what type of drip behavior is happening until after the data is processed. After repeating the experiment many times and processing the data it became obvious that most of the interesting behavior was happening somewhere around a period of 50 ms, so this was the region we most closely investigated.

To optimize the possibility of a successful trial run we first pulled a paper towel over the bottom bucket, for the drumming noise. Then we started the Arduino software to begin displaying times for each drop. When the software is running we opened the clamp until we were within range of about 50 ms; this was accomplished by preventing the Serial Monitor from auto-scrolling and checking the output to see if it was on the order of 50 ms. Once we were within range we closed the Serial Monitor and re-opened it to begin the trial. The reason why we close and re-opened the Serial Monitor to begin the trial is the counter starters over again except this time exactly in a region of interest which makes processing the data later much easier. Now a trial run is in full swing and it's time to adjust the clamp to increase or decrease the period and hopefully move into different regions of interest. Now, one may think that using an adiabatic approach by slowly opening or closing the clamp is the way to success; however, in reality when this is done often the region is overshot or hidden because the dropper isn't given a chance to stabilize within a specific region. What ends up happening is the data becomes a swoopy blur and no interesting behavior is observed as shown in Fig. 6. The technique we found to be more useful is to instead open or close the clamp as little as possible, but do it quickly and then wait for around 500 drops then make another instantaneous open or close of the clamp. Suffice to say, finding the beautiful display of all three regions in the same trial run, as shown in Fig. 4, took much effort.

3.4 Processing data

One might expect getting measured data out of the Arduino to be easy, especially because the entire community of scientists, hobbyists, students, and anyone performing any type of experiment with the Arduino are likely making some sort of measurement and probably want their data go somewhere besides the Serial Monitor. Actually, finding a way to get data from the Arduino into a usable form like a CSV file in even a pitifully scalable way is non-trivial. We tried several ways including: piping the output to the terminal, piping the output to Processing, piping the output to a file, and crudely copying and pasting the output from the Serial Monitor. Any time another program was being used with the Arduino integrated development environment (IDE) all the programs had to be restarted each time and in a specific order to get reliable results making it impractical for multiple trials. Piping output to a file worked about half the time and only if the entire IDE was shut down in a specific order. Also when trying any of the first three methods mentioned one was not able to open the Serial Monitor first to get an idea of what the current period was and then begin taking data after establishing that we were close to an interesting region. The only reasonable workflow for processing data that allowed us to run many trials and save useable data can be seen in Fig. 3: literally copying and pasting data from the Serial Monitor (using cmd-C because the Serial Monitor doesn't respond to right clicks), and then using the ImportText Wizard in Microsoft Excel. However, as shown on the right in Fig. 3 the Serial Monitor adds it's own newline character to each line so the data ends up skipping every other cell in Excel. To work around this we first copied and pasted the data from the Serial Monitor into Sublime Text, a text editor (keeping in mind to use cmd-A in the Serial Monitor to highlight all the data, otherwise it takes literally 5 minutes of waiting to highlight all the text while the Serial Monitor scrolls), then we copied and pasted the data from Sublime Text into Excel using the ImportText Wizard with Sublime Text automagically (that is, "automatically" but with no explanation for why it works) removed all the mysterious new lines, also shown on the right of Fig. 3. Finally we are able to process our data in Excel as one would normally expect.

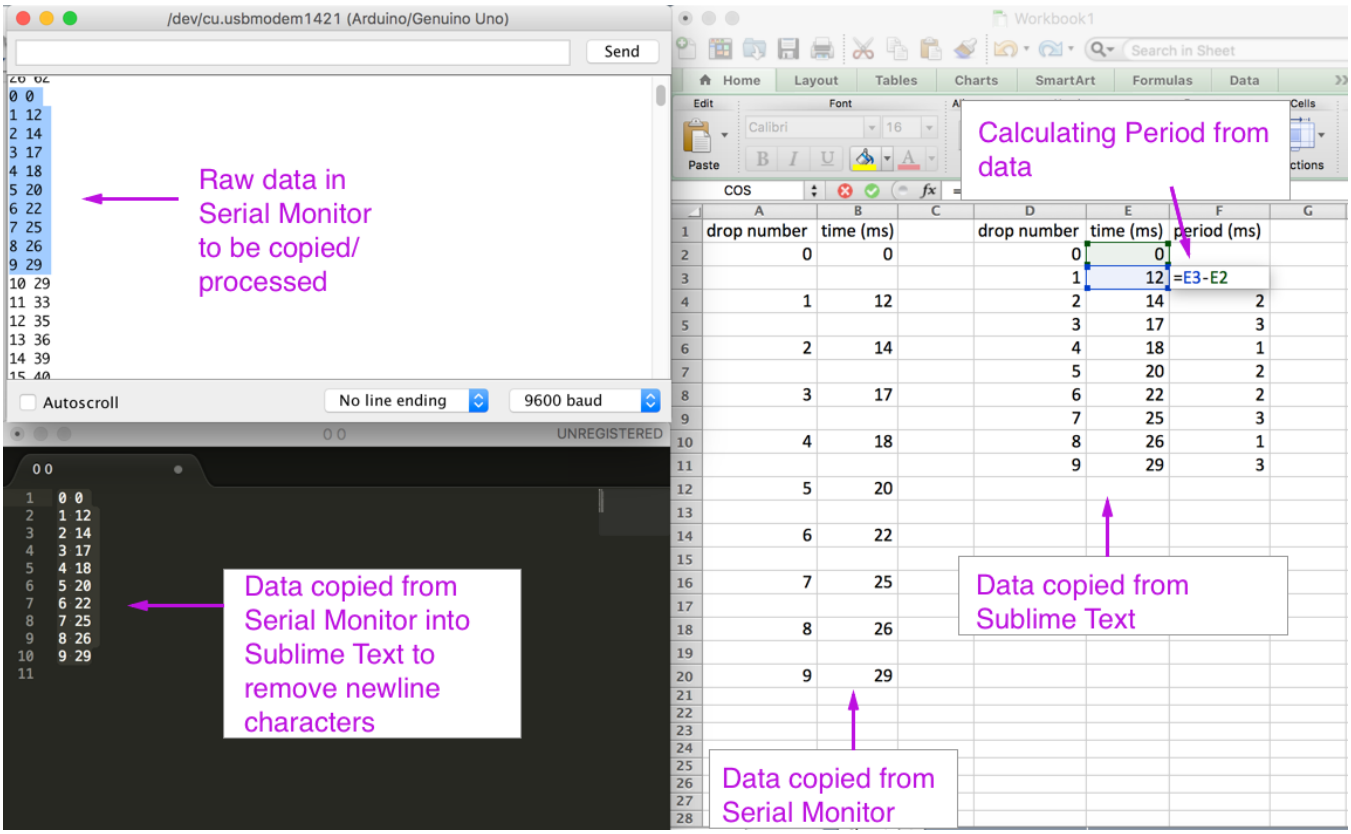

Figure 3: Screen shot of a finished trial and data processing

4 Data, Analysis, and Discussion

4.1 Interpreting the graph

There are three types of graphs that we use to visualize our data. Interpreting each one can take a little getting used to but once understood the information provided by each can be clear.

4.1.1 Period vs. drop number

Figures 4 - 6 are plots of period as a function of droplet number. On these types of graphs each point represents a single period between two water droplets. Figure 4 is an experimental plot of the period-1, period-2, and chaotic regions. Figure 5 is a theoretical plot of the period-1, period-2, and chaotic regions. Figure 6 is an experimental plot showing only periodic regions. Taking a look at Fig. 5 we can understand what is happening more easily because of the resolution. Notice, how for the period-1 region, also called periodic, droplets 21-30, has the same period of 40 ms. However, when looking at droplets 0-9 we see that for each successive droplet there is a toggling between 50 ms and 70 ms, this behavior is characteristic of period-2, or period doubling. Finally, droplets 10-19 seem to be randomly scattered about, never repeating or alternating, this behavior is characteristic of chaotic dripping.

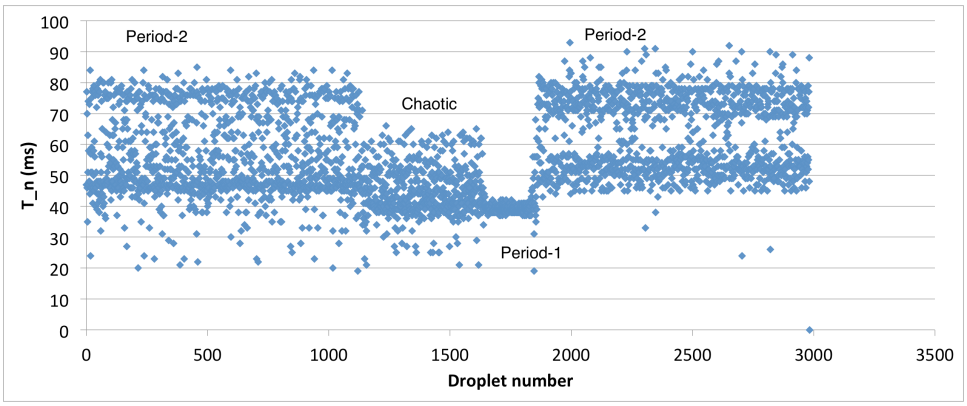

Figure 4: Good data showing Period-1, Period-2, and Chaotic experimental regions

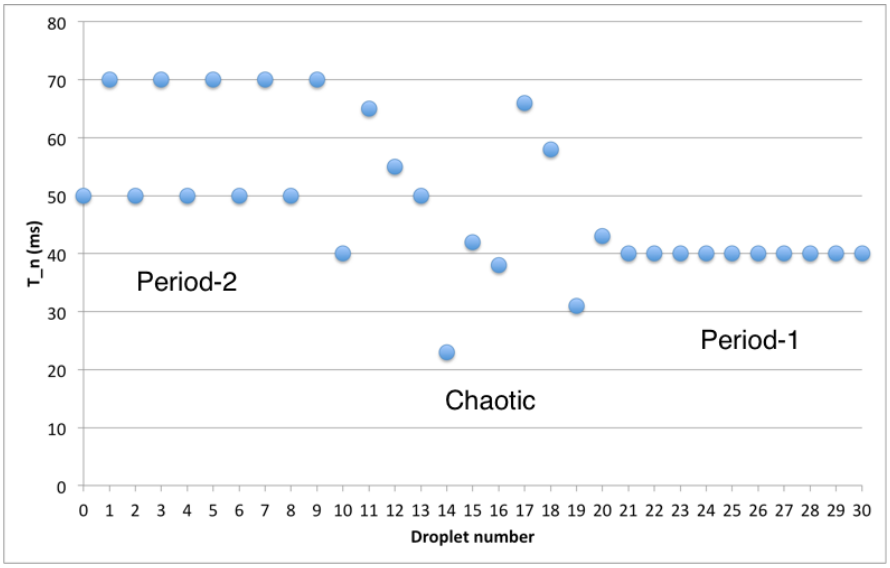

Figure 5: Theoretical Period-1, Period-2, and Chaotic regions with 30 drops

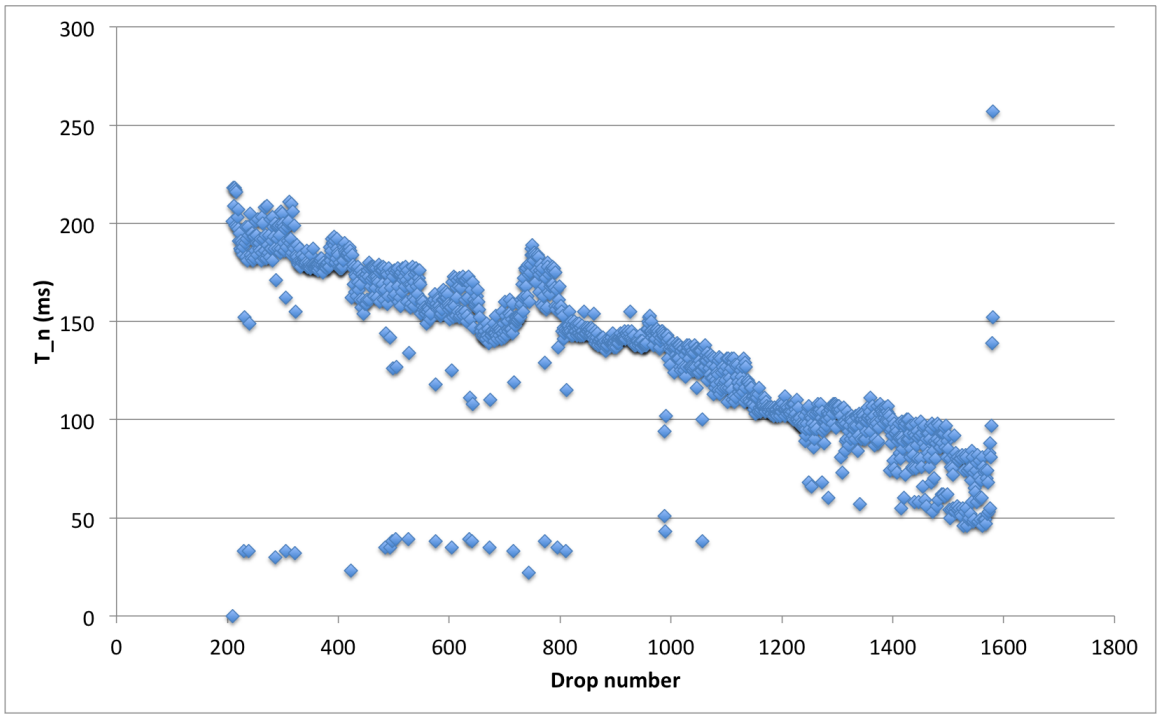

Figure 6: Adjusting clamp adiabatically, instead of in quick direct bursts produces convoluted data

4.1.2 Poincare section graph

Figures 8, 9, 11, 12, 14, and 15 are Poincare section graphs. Poincare section graphs can be used to zoom in on each region of interest. Now, each point on the graph holds two pieces of information, the period of an initial droplet on the x-axis (T_n), and also the period of the next successive droplet on the y-axis (T_{n+1}). The line $T_{n+1} = T_n$ represents regions of periodicity because each successive drop has the same period, therefore each successive droplet will map back to the same point.

Figure 7 shows a visualization of periodic dripping, if we take $n = 1$ then we can see both T_1 and T_2, in red to signify they have the same period; now when we look at the graph on Fig. 8 we can see that $T_1 = 40$ ms but $T_2 = 40$ ms also, and so a point on the graph predictably falls onto the $T_{n+1} = T_n$ line.

We can compare the theoretical and experimental Poincare section graphs to determine which region the system is in. Figures 8 and 9 clearly agree indicating that our system is indeed exhibiting periodic behavior.

Figure 7: Period-1 visualization

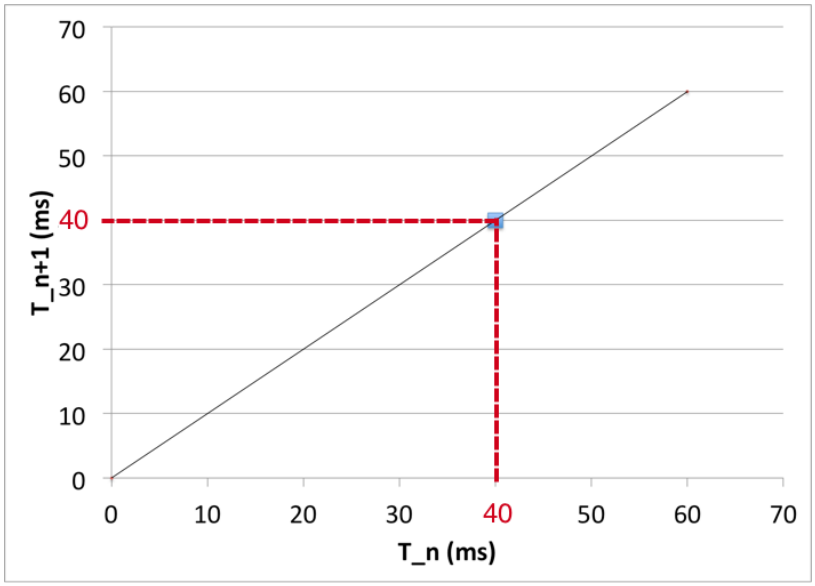

Figure 8: Period-1 theoretical plot

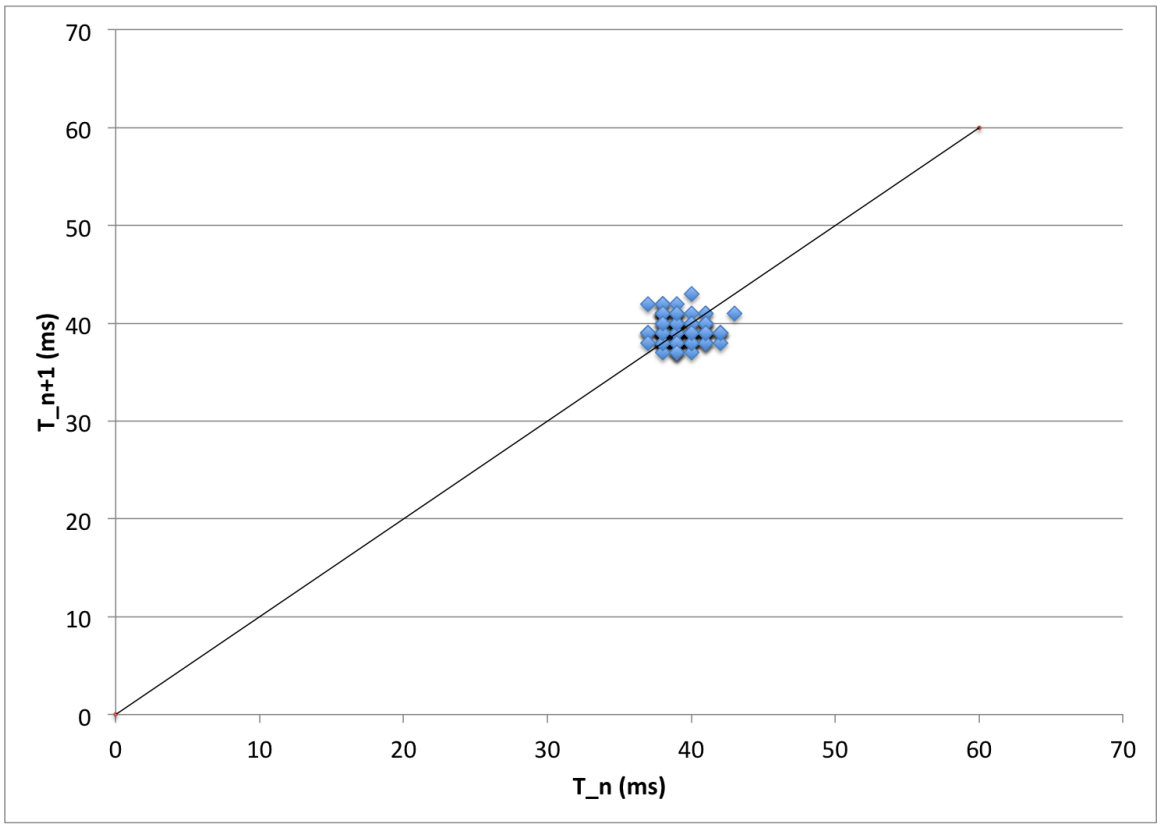

Figure 9: Period-1 experimental plot

Figure 10 shows a visualization of period doubling dripping, if we take $n = 1$ then we can see both T_1 and T_3, in purple to signify they have the same period, while T_2 in green to signify it has a different period; now when we look at the graph on Fig. 11 we can see that $T_1 = 50$ ms but $T_2 = 70$ ms, and so a point on the graph falls above the $T_{n+1} = T_n$ line. Now, if we take $n = 2$, then $T_n = T_2$ in green, and $T_{n+1} = T_3$ in purple. Looking back at the graph on Fig. 11 we can see that $T_2 = 70$ ms but $T_3 = 50$ ms, and so a point on the graph falls below the $T_{n+1} = T_n$ line. Interesting with period doubling the data points will falls symmetrically about the $T_{n+1} = T_n$ line.

Comparing the theoretical and experimental Poincare section graphs on Fig. 11 and Fig. 12 we can see that they clearly agree, indicating that our system is indeed exhibiting period doubling behavior.

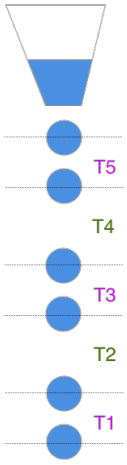

Figure 10: Period-2 visualization

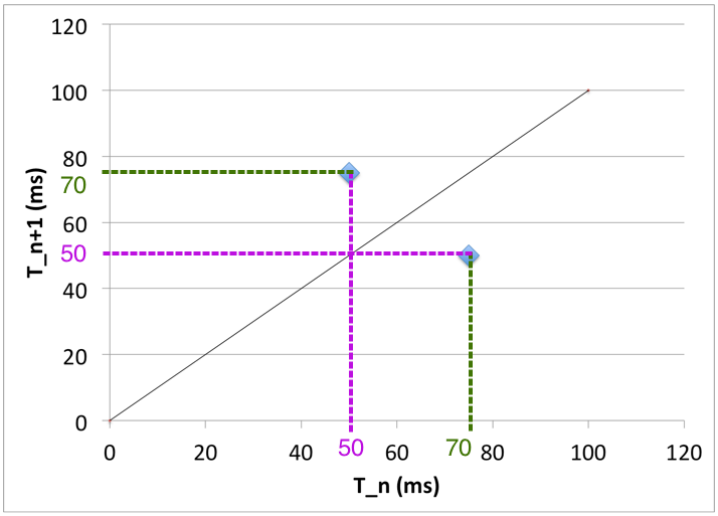

Figure 11: Period-2 theoretical plot

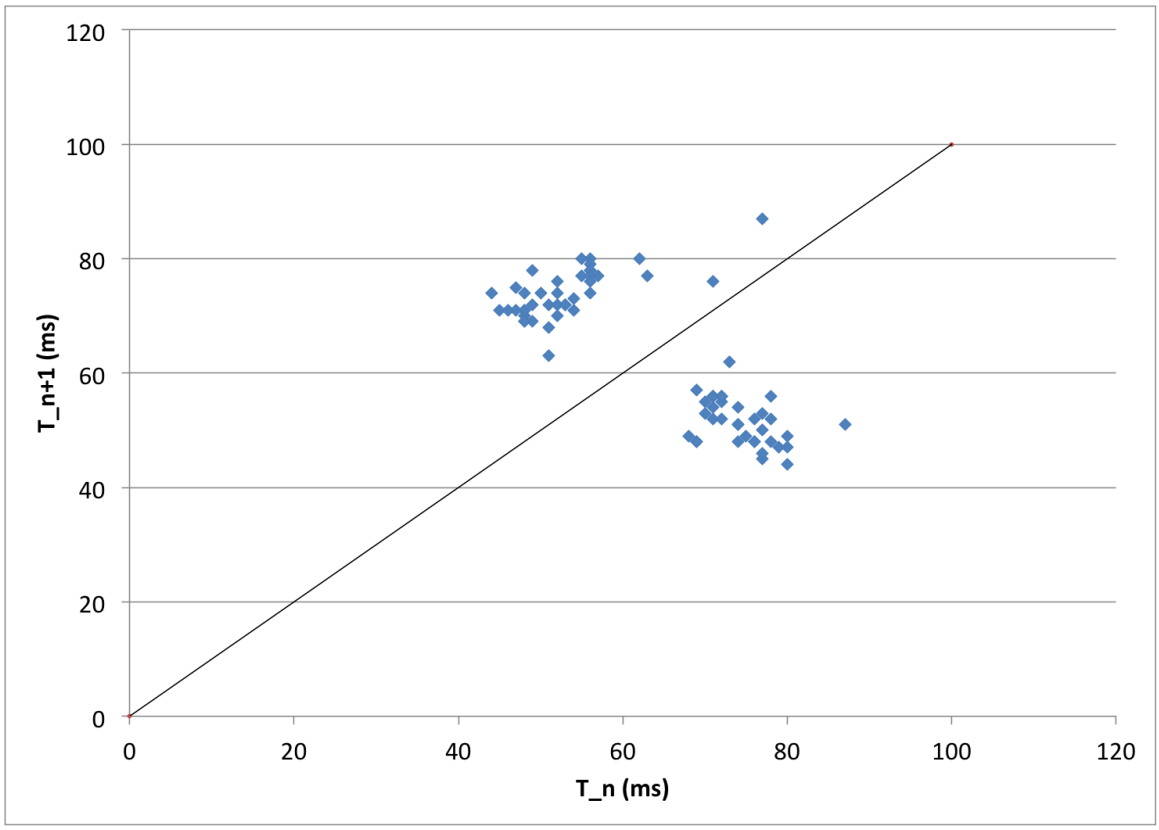

Figure 12: Period-2 experimental plot

Figure 13 shows a visualization of chaotic dripping, if we take $n = 1$ then we can see both T_1, T_2, and T_3 all have different periods; now when we look at the graph on Fig. 14 we can see that there is no predictable pattern and the data points are scattered about the $T_{n+1} = T_n$ line.

Comparing the theoretical and experimental Poincare section graphs on Fig. 14 and Fig. 15 we can see that they clearly agree, indicating that our system is indeed exhibiting chaotic behavior.

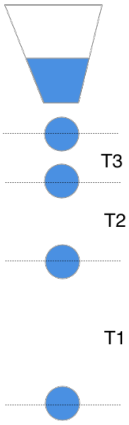

Figure 13: Chaotic visualization

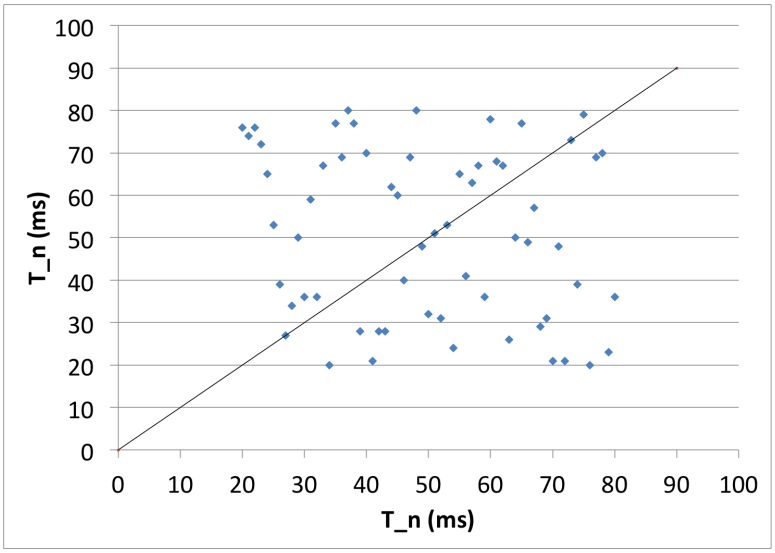

Figure 14: Chaotic theoretical plot

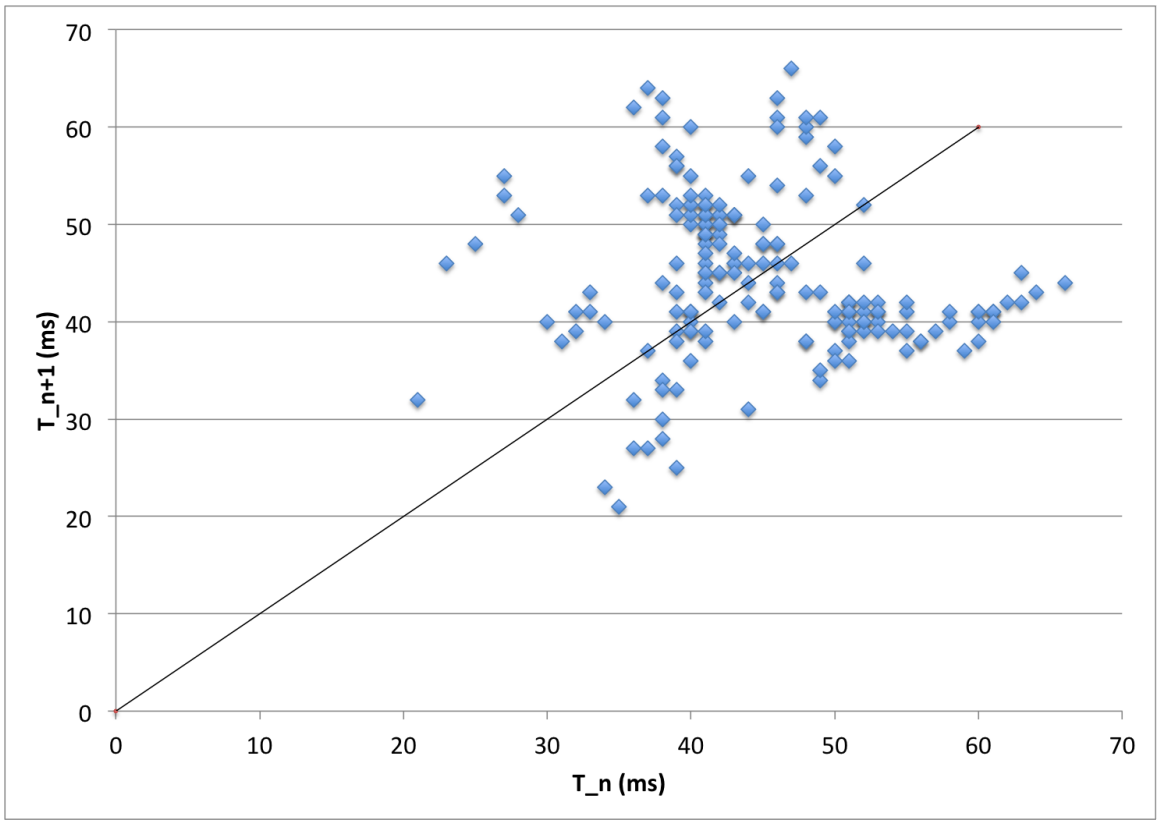

Figure 15: Chaotic experimental plot

4.1.3 Histogram

We also use a Count vs. Period histogram to zoom in on each region of interest. Now, each bar on the graph tells us how often a specific period occurred for a given number of drops.

Fig. 16 shows a theoretical histogram for periodic dripping. As expected we see a spike in the 41 to 42 region, which would be the case if the system were in a periodic situation like the one shown in Fig.7.

We can compare the theoretical and experimental Histogram graphs to determine which region the system is in. Figures 16 and 17 clearly agree indicating that our system is indeed exhibiting periodic behavior.

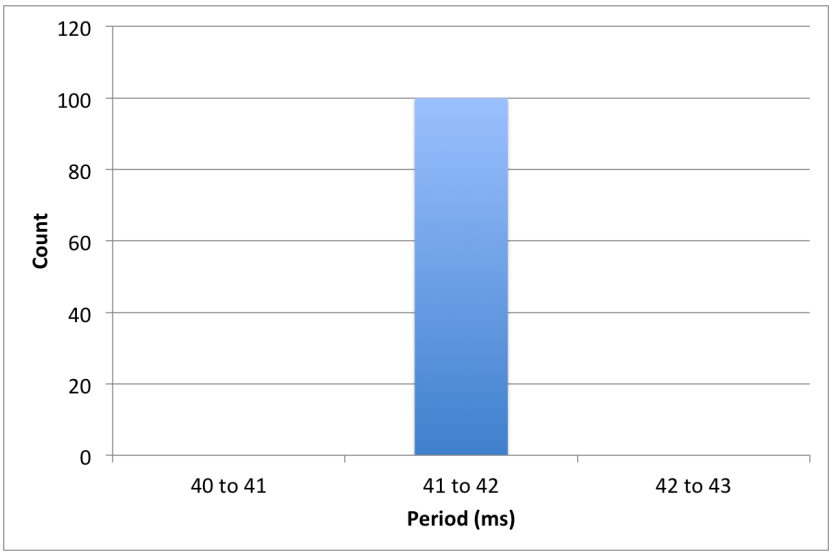

Figure 16: Period-1 theoretical histogram

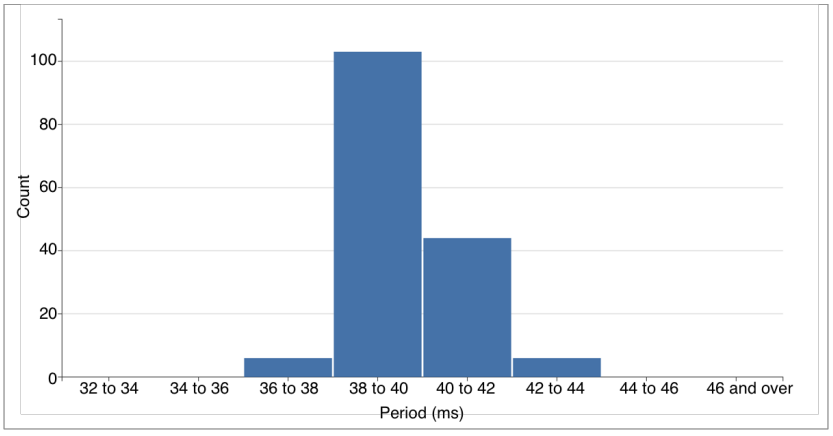

Figure 17: Period-1 experimental histogram

Fig. 18 shows a theoretical histogram for period doubling dripping. We see two spikes in the 50 to 60 and 70 to 80 regions, which would be the case if the system were in a period doubling situation like the one shown in Fig. 10.

Comparing the theoretical and experimental Histogram graphs on Fig. 18 and Fig. 19 we can see that they clearly agree, indicating that our system is indeed exhibiting period doubling behavior.

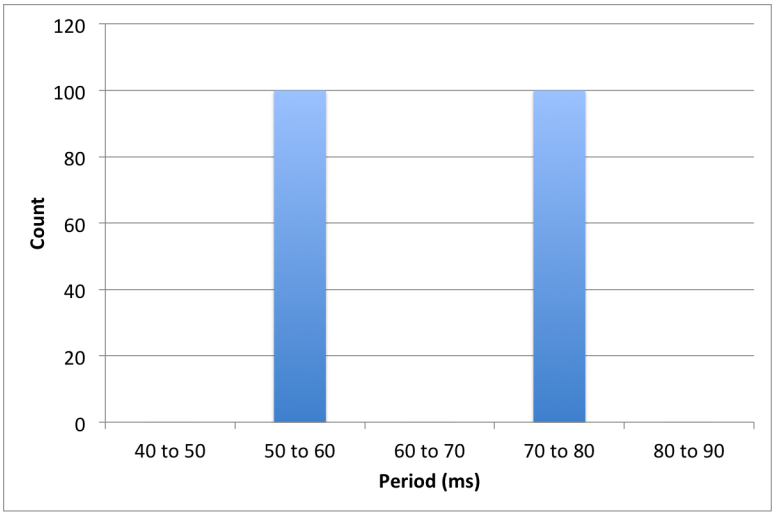

Figure 18: Period-2 theoretical histogram

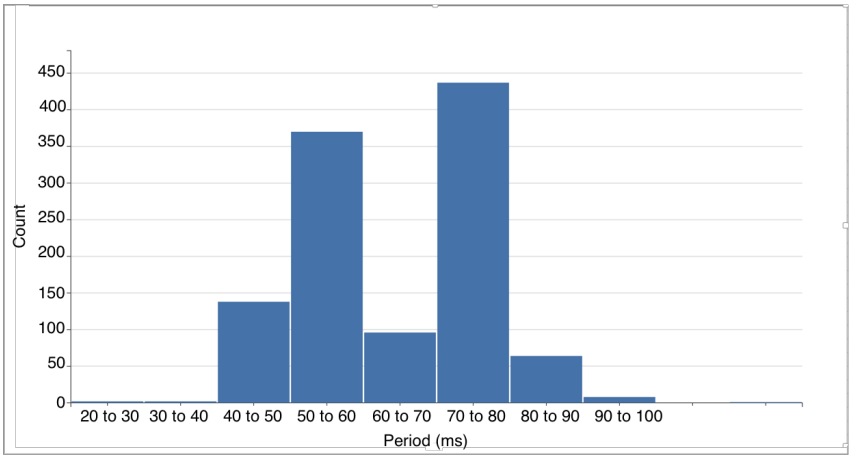

Figure 19: Period-2 experimental histogram

Fig. 20 shows a theoretical histogram for chaotic dripping. We see a wide array of spikes, which would be the case if the system were in a chaotic situation like the one shown in Fig. 13.

Comparing the theoretical and experimental Poincare section graphs on Fig. 20 and Fig. 21 we can see that they clearly agree, indicating that our system is indeed exhibiting chaotic behavior.

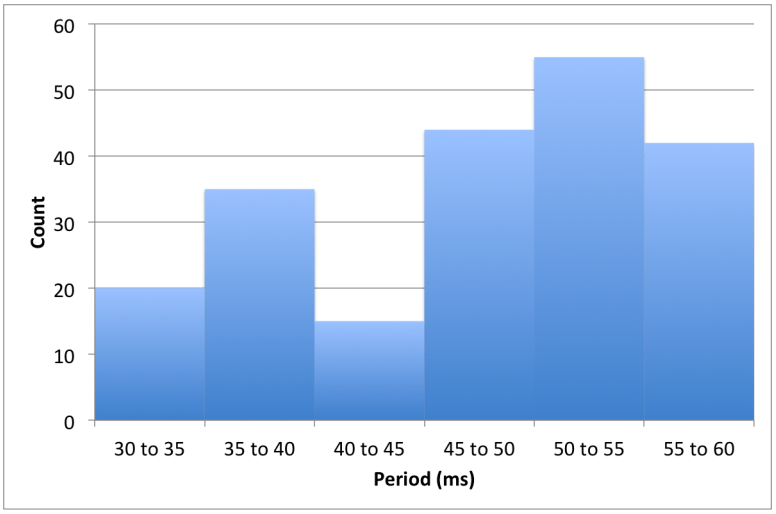

Figure 20: Chaotic theoretical histogram

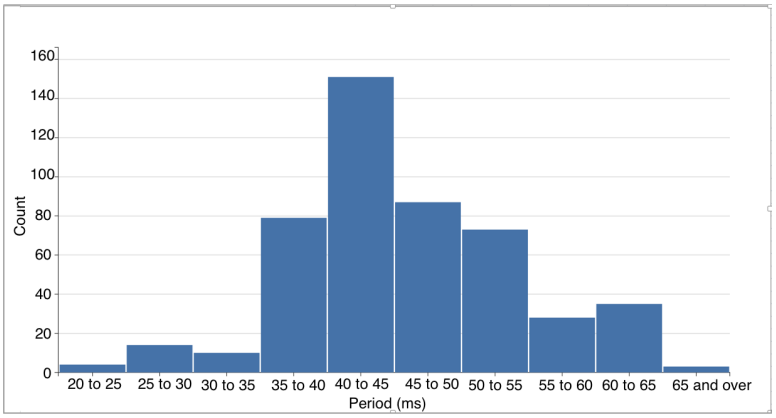

Figure 21: Chaotic experimental histogram

5 Conclusions

We built an apparatus and wrote custom software to measure the time of each water droplet from a dropper. While building and calibrating our apparatus we solved experimental problems such as decreasing the intensity of the laser to prevent the photodiode from saturating, impedance matching the photodiode detector with the Arduino, using correct technique to attenuate the clamp that adjusts the flow rate, and finding a viable way to process data from the Arduino. By adjusting the flow rate into the dropper, determining the period of each drop, and exploring several different types of data plots we confirmed that our dripping faucet system exhibits periodic, period doubling, and chaotic behavior.

A
Arduino Code

```
1   int n = 0;
2   unsigned long time0;
3   unsigned long time1;
4
5   void setup()
6   {
7     Serial.begin(9600);
8     time0 = 0;
9   }
10
11   void loop()
12   {
13     time1 = millis();
14
15     if ( analogRead(A0) < 500 ) {
16         Serial.println(String(n) + " " + String(time1-time0));
17         n = n + 1;
18         time0 = time1;
19         delay(10);
20     }
21   }
```

B
Appendix B

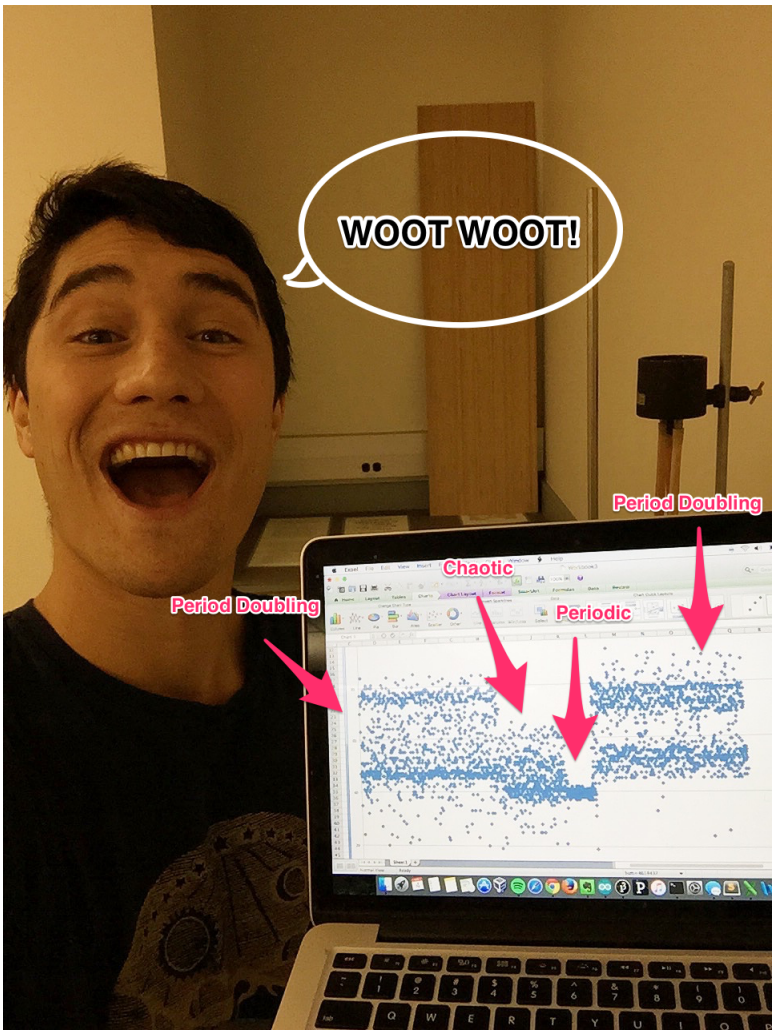

Figure 22: Successful trial run

Infinite thanks to my senior project advisor Dr. Glen Gillen for fun tinkering sessions, reading through long-winded paragraphs, and giving me the equipment and space I needed to work. And another infinite set of thanks to Dr. Matt Moelter for the support and inspiration to see this thing through.

References

[1] R. Shaw, *The dripping faucet as a model chaotic system*, (Aerial Press, Santa Cruz, USA, 1984.)