Developing a Solid State Rain Gauge

Alishan Hassan
University of Virginia
85 Engineer's Way
Charlottesville, VA 22903
703-344-5196
ah2xt@virginia.edu

ABSTRACT

In this paper, we are going to take a look at the progress made towards the development of a wireless, solid-state, rain gauge, or disdrometer. The goal of this project is to create a small, power efficient, and inexpensive rain sensor that could be distributed by the thousands throughout a city, to enable real-time rainfall monitoring. This project investigates suitable designs for both the microcontroller electronics and the sensor itself. The platform used in this project is the Particle Photon, a low-power, cloud connected microcontroller with built-in Wi-Fi and an upgrade path to the Particle Electron, a 3G enabled model. The sensor is solid state, designed using an inexpensive piezo element. The impact of the drops creates a stress on the piezo element, which produces an electrical signal that relates to the size of the drop. By collecting raw data of these raindrops, a rainfall rate could be calculated.

CCS Concepts

Applied computing → Physical sciences and engineering → Earth and atmospheric sciences

Hardware

Keywords

Rain gauge; disdrometer; piezoelectric; electret; Particle Photon; solid state; internet of things

1. INTRODUCTION

Coastal urban areas are undergoing significant stormwater management issues. A rising sea level, subsidence, as well as frequent extreme weather is exacerbating these issues. The perfect case study for this is Virginia Beach, a city that is facing some intense water management challenges. Virginia Beach has eight pumping stations to transfer water stored in Best Management Practices to larger water bodies with additional storage capacity to prevent flooding. When the forecast predicts significant amounts of rainfall, a pumping station's intake can be lowered to increase capacity for runoff. To aid this infrastructure, we are looking to develop low cost, wireless, rain sensors that can provide real-time data to better manage water resources, in hopes to mitigate nuisance flooding. Ultimately, we would want to be able to have real-time data points for rainfall, tide levels, and soil moisture. One of the issues we'll need to address is the balance of maintenance, portability, and reliability. There is a need to create a sensor that is unhindered by mechanical breakdowns, movement, and size constraints. There is also a need to make sure the sensor is able to capture data quickly and perform reliable computations that rival existing weather stations. In our case, this means capturing every drop of water and converting it into a real-time rainfall rate across hundreds of nodes throughout a city.

Attempts at building similar rain sensors aren't new. There are portable weather stations, such as models from Vaisala, which provide a number of tools to obtain weather data, including a rain sensor. Tipping buckets sensors are often stored in various locations to obtain local rainfall data, with some sort of a connected computer keeping track of the data. Wireless solutions exist in the form of GSM based internet of things systems, which use GPRS to transmit real-time data. Attempts to create solid state sensors have also been undertaken at, among other places, at other universities.

Our goal is to use a Wi-Fi equipped Particle Photon (and eventually its 3G equipped brother, the Particle Electron) as a cloud connected platform to read data off a solid state sensor of our own design. This solid state sensor will effectively listen to the falling raindrops and, based on the strength of the signal, determine the size of the raindrop calculate the rainfall rate. Calibration is critical towards the success of this implementation. Therefore, a lot of data in multiple conditions needs to be collected. We are running experiments that simulate rainfall by using a watering wand attached to a 10 foot long PVC pipe. This allows us to hold the wand at a height that permits the droplets of water to reach terminal velocity. The sensor is set up connected to the Particle Photon, which is connected to a computer which logs the data. We let the droplets fall on the sensor and collect the data, using either Excel or NumPy to analyze the data.

Thus far, the data collected has a number of issues. While it is easy to see, visually, the impacts of the drops, converting the drops into a rainfall rate is proving to be troublesome. Current computations are showing virtually no trend in the errors, which vary from .5 inches/hour to 4.5 inches/hour. It will be necessary to improve the functions as well as collect much more data.

2. Background and Related Work

Weather Stations

Implementations of distributed rain sensing systems have been implemented before. These tend to be in the form of typical weather stations which, in addition to measuring rainfall, also measure temperature, pressure, wind speed, among other weather attributes. These include systems like the Vaisala Automatic Weather System, which actually has its own solid-state rain sensor built in. It has a software interface that allows you to remotely collect data and, functionality-wise, would be suitable for our real world use cases. However, where it falls short is price and portability. The Vaisala system is relatively large, particularly when our use case requires deploying hundreds of nodes throughout an urban area. More importantly, there is the cost. The Vaisala rain sensor, alone, costs in excess of $2000. For our implementation, it is necessary to bring the cost down as much as possible.

Sensors

There are a number of different sensors that are used to measure rainfall. A classic system developed in the early 20th century is effectively a funnel that channels rain into a graduated cylinder.

The cylinder overflows into an outer container and allows a measurement to be manually taken. Many weather stations rely on a tipping bucket sensor. The principle is rather simple – a small bucket fills up with water as it rains. When it's full, it flips over allowing another bucket to start collecting water. Each flip triggers a reed switch which sends an electrical signal signifying that a flip occurred. This removes the need to constantly check a reading on a graduated cylinder. The downside of a tipping bucket is that it often underestimates rainfall during downpours. Weight precipitation gauges are also used, which simply consist of a bin that is weighed to determine rainfall. This tends to be more accurate than a tipping bucket, but also requires much more maintenance.

Other sensors that have been used include different types of solid state sensors. In theory, their benefit is much lower maintenance and potentially lower cost than tipping buckets or weight precipitation gauges, due to the lack of mechanical systems. Some solid state sensors include optical rain gauges and acoustic disdrometers. The former uses a laser for measurement and the latter listens to raindrop impacts to determine their size and sum up the rainfall. Using an acoustic setup is notable, because not only does it have the benefits that come with being solid state, it's also relatively inexpensive. Furthermore, other projects have made some decent strides in implementing acoustic rain gauges in the field.

Disdrometer

Acoustic disdrometers have been investigated in earlier studies. The Vaisala actually uses one for its rain measurements, and there have studies to implement a spherical disdrometer that can be used on buoys. The principle of measuring hydrometeors using a disdrometer is sound. A master's thesis at Delft University of Technology investigates developing a low cost disdrometer that can be used in urban environments. The project used a piezo element with a PVC cap for the droplets to land on and used an audio recorder to log the data. While calibration needs work, the project shows that such a setup is an effective method to measure rainfall. This idea will be used in the development of our rain sensor.

Communication

Internet of Things devices tend to communicate with the web using a GSM connection. There were a number of ways to implement such a thing. Some, relatively simple, weather devices have relied on using SMS to transmit updates at a preprogrammed rate. They can receive basic commands in the form of a text message. This limits versatility, so a full GPRS/EDGE or even HSPA data connection is used. For this project, an interesting microcontroller was discovered: the Particle Electron, which features functionality that is comparable to Arduinos but also has 3G connectivity and minimal power consumption.

3. Approach

The overall idea is to use a Particle Photon microcontroller (eventually to be upgraded to a Particle Electron) as the brains of the unit. The sensor is connected via an analog input on the board, after undergoing amplification, and the Photon reads the data (with a 5ms delay) and writes it to serial. A USB connection connects the Photon to a PC, which logs the data and performs computations (eventually computations will need to be done no the device to minimize data usage). The sensor is a piezo-element with a cap super-glued to the top, and held up by a PVC pipe. The setup also includes space for a tipping bucket sensor that provides a ground truth.

Sensor

In this experiment, we used a simple 20mm diameter piezo element built by Murata Electronics. These cost around $1.06/unit, making them the cheapest option that was investigated. This bare diaphragm is connected directly to the microcontroller with a cap installed on top. Electret microphones were also purchased and installed in a test jig, but have not yet undergone testing.

Particle Photon

The core of setup is the Particle Photon, a Wi-Fi capable, cloud connected, microcontroller with analog and digital I/O, an ARM Cortex M3 CPU, 1 MB flash, 128 KB RAM, and a programming capabilities similar to an Arduino. For the purposes of reading an analog signal off of our sensor, transmitting over serial or through cloud, and using minimal power, the Photon proved to be a very capable platform. Average current consumption, with Wi-Fi on, is around 80 mA, while deep sleep mode brings the consumption down to a minimal 80 uA. Initially we used an Arduino paired with a SIM900 2G board, and the Photon proved to be considerably more power efficient. The Particle Electron will be used in the deployed sensors because it adds 2G or 3G functionality through T-Mobile at a rate of $2.99/month for 1 MB of data.

Instrumentation Amplifier

The setup in this experiment includes an instrumentation amplifier, the Analog Devices AD623. Particularly with lighter rainfall, an amplifier is necessary for drop detection, and even with heavier rainfall it provides more discrete values that can be used to calculate the size of a drop. During experimentation, the amplifier showed a number of issues using a single voltage supply, namely that the drop in voltage to the zero point is delayed. A dual supply setup did not exhibit this issue, thus further investigation is needed. Operation amplifiers did not exhibit this behavior but had higher noise and DC offsets. Proper functionality of the AD623 is ideal for the final design.

Digital Potentiometer

The magnitude of the amplification of the AD623 is controlled by a resistor. A digital potentiometer, like the MCP41100 used in this project, offers the benefit of having the ability to be controlled by the Photon. Using the Serial Peripheral Interface (SPI) Bus, the Photon can have direct amplification control, with 256 levels, which can be useful for increasing sensitivity during light rainfall or minimizing noise during downpours.

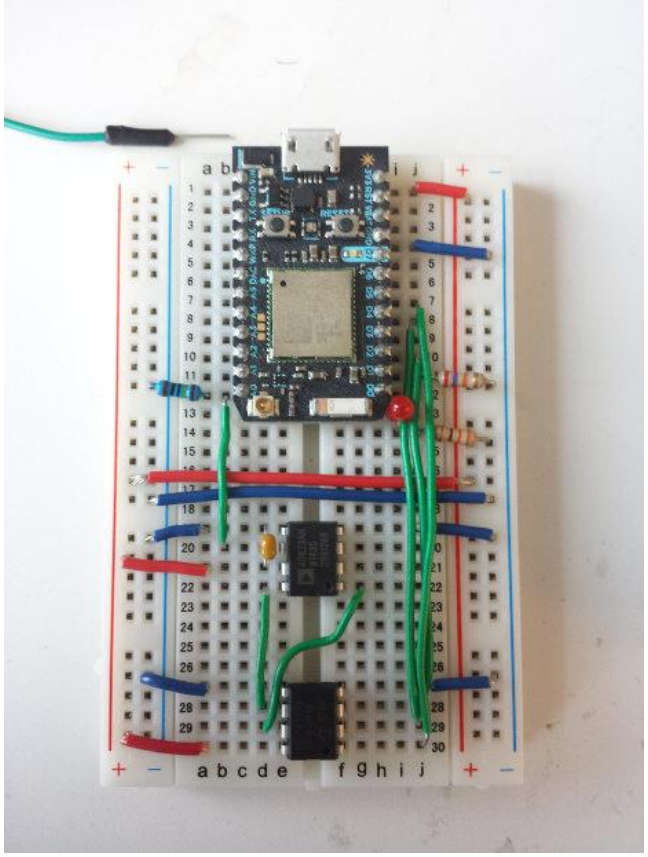

Figure 1: Electronics Configuration

Caps

The piezo element is covered by a cap. In practice, choosing the right, size, shape, and material for the cap is very important. The cap's acoustic properties need to maximize the signal and minimize the outside noise. The shape of the cap also needs to prevent rain drops from accumulating on top and not running off. Furthermore, the radius has to minimized so that the magnitude of the signal is consistent across the surface (typically, the signal is stronger when the raindrop makes an impact closer to the center of the piezo element).

Figure 2: Machined Caps

The only caps used thus far are a stainless steel flat cap that was cut out, using aviation snips, from a sheet of metal and an aluminum foil cap cut out from a drip pan. The stainless steel cap easily oxidizes, does not deform much upon impact, and allows raindrops to accumulate on the top, but was used for much of the data collection, primarily due to manufacturing delays of other caps. The aluminum cap proved to be lacking in durability and easily tore with basic handling. Other caps have been made, including three machined caps that were made on a lathe out of aluminum, nylon, and acrylic. Other caps were made out of two different types of PVC, sanded smoothly with a rounded surface. The machined caps have the benefit of being thin and versatile, but are custom made parts that would be costly and time-consuming to manufacturer. The PVC caps use off the shelf PVC caps, sanded smooth, and are therefore cheaper and more readily available. All of these caps are ready to be tested.

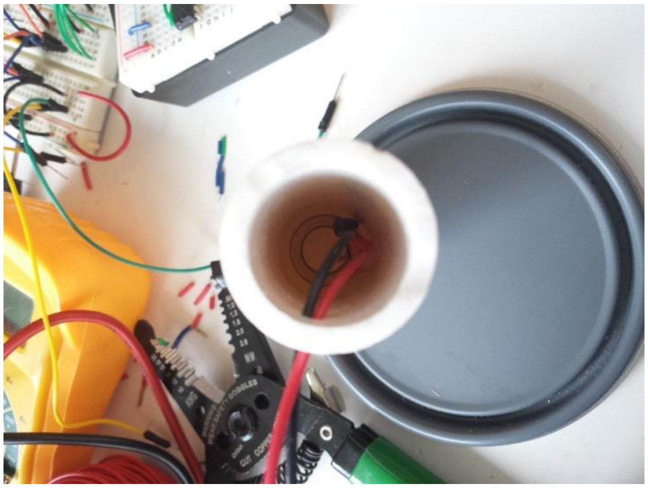

Figure 3: Piezo Element Installed in PVC Pipe

Construction

Assuming the electronics are built as shown in Figure 1, construction of different sensors is relatively quick and simple. 1" wide PVC pipes are cut to a reasonable height. It is imperative to wear a mask while cutting PVC, so avoid lung issues. A small hole is drilled near the bottom side. This is to permit wires to enter the pipe. Now, take an arm span length of red wire and black wire to extend the length of the piezo element wires. Solder them together and apply heat shrink. Next, take the appropriate cap and super glue a piezo element on the bottom as close to the center as possible. Next, fish the wires through the hole that was just drilled and keep pulling until the piezo element sits plat on the top of the tube. Use super glue to attach the element/cap to the PVC tube. Finally, either duct tape or hot glue the tube to the base, and connect the wires to the Photon (either the input on the amplifier or directly to the analog input, depending on what's being tested at the time).

The electronics breadboard is stored inside of a project box which has three holes, one for USB to go the computer, and two for sensor inputs (the piezo element and a tipping bucket for this case).

Tipping Bucket

A tipping bucket sensor is also connected to the Photon. Because it's simply a reed switch, it takes an input voltage and connects to one digital input on the Photon. The purpose of the tipping bucket is to serve as a ground truth. As the rain falls, the Photon is logging both the data from the piezo element as well as the points when the bucket flips.

4. Experimental Setup

The basic experimental setup has the piezo sensor and the tipping bucket connected to the electronics box, which is connected via two USB extensions to a laptop that is protected from getting wet. All power is provided by the laptop. A Python program on the laptop logs the data and writes it to a text file. The data comes in a formatted string which includes the timestamp, elapsed time, piezo element values, when the bucket is flipping, and the cumulative bucket flips. This data can easily be exported into Excel or another Python program which uses NumPy for data analysis.

Figure 4: Experimental Setup

Sources of Rain

It became somewhat difficult to actually provide a reliable method to get varying levels of rainfall. The obvious method is to simply go out during rainfall events and collect data. The more instances of this, the more data that can be collected for calibration purposes.

Figure 5: Testing During Actual Rainfall

This wasn't entirely predictable, so other methods were investigated. The most important thing is to make sure that the drops reach terminal velocity when the make an impact on the sensor. This generally means having the drops fall from a high enough point that terminal velocity is achieved.

One attempted was to use an oscillating sprinkler. If placed on a table, the drops did fall from high enough to reach terminal velocity. There were, however, issues with aiming and making sure the same amount of rain reached both the piezo sensor and the tipping bucket. A more localized approach involved using a shower. The issue, of course, here was that the shower was pressurized and not high enough, which would complicate the calibration process.

Ultimately, a "rain machine" was made using a watering wand attached to a 10 foot PVC pipe using zip ties. The experimenter would stand on a ledge just outside Rice Hall that was about four feet high. Combined with the height of the experimenter, the 10 foot pipe, and the roughly 2-3 feet of length from the wand, and the drops could easily hit terminal velocity. The water simply came from a spigot, and the hose was clamped onto the pipe for structural stability. The end result was a consistent, localized, rainfall that could somewhat be controlled by the water pressure.

Figure 6: "Rain Machine" in Action

The wand, using a standard hose connector, also enabled the use of other nozzles. A simple garden nozzle was purchased to allow a different shower stream as well as a flat and mist setting to introduce more rainfall types. Ultimately, though, the rainfall rate was very high, at points exceeding the world rainfall rate record, so some realism was sacrificed. Ultimately, real rainfall data is ideal for the purposes of calibration.

At the end of the day, the goal is to collect as much data as possible at varying levels of rainfall intensity. This allows us to run a data analysis and determine the error in the rainfall rate at varying levels of intensity.

Calibration

There is question as to whether or not the tipping bucket is completely reliable, especially at these higher rainfall rates. An analog rain gauge and a graduated cylinder are included to confirm that the tipping bucket is, indeed, a reliable ground truth. In addition, some pipettes are also included for bench testing, to make sure the cap being used is reasonable and that the electronics are detecting drops.

5. Results

Tremendous levels of data were collected over the course of this experiment. The raw data files with corresponding Excel plots are included in the Github repository. The data analyzed in this paper is was a relatively consistent bit of rainfall using the "rain machine" with no amplification to isolate issues and a flat stainless steel cap. The figures given in this section can be reproduced for other datasets by simply changing the data file used in the analysis.py program. Note, earlier data analysis was done using Excel, thereby requiring the data files to be run through Excel for some cases.

Figure 7 shows the cumulative flips of the tipping bucket. This shows a relatively consistent rainfall rate. Figure 8 shows the piezo element raw values over the same period of time.

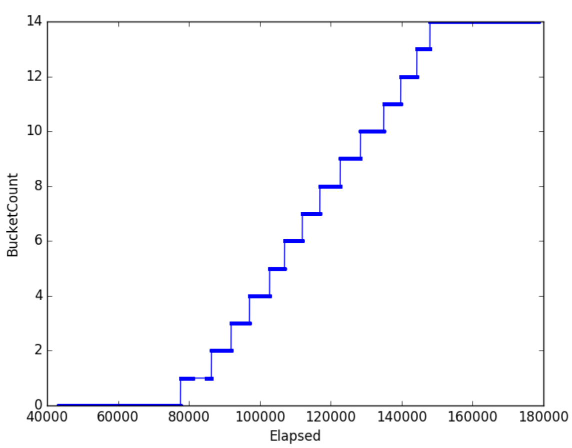

Figure 7: Cumulative Tipping Bucket Flips

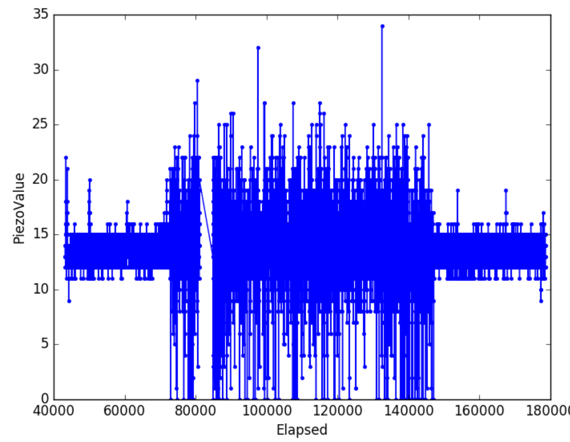

Figure 8: Graph of the Piezo Element Readings Over Time

The idea, here, is that each flip represents .01" of rainfall. So, what the data analysis does is that it notes the start point of each flip, and uses the piezo element data between those two time points to represent .01" of rainfall. The same function is applied to those sets of data and the quality of the function is assessed by an error plot.

Using Excel, the first attempt at data analysis involved simply summing all of the piezo values and applying a linear or quadratic function until the average error is minimized. The error plot is given in Figure 9.

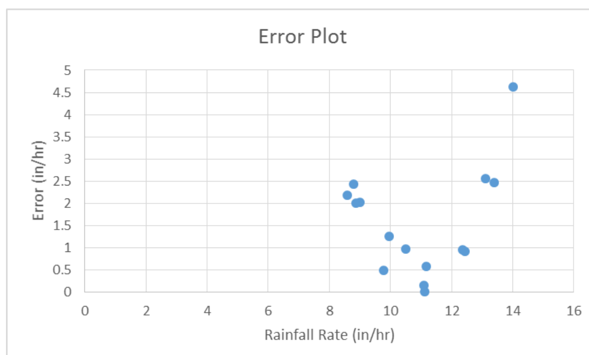

Figure 9: Error Plot with Linear Summation

The fundamental problem with this approach is that it assumes that every point collected is some sort of hydrometeor that should amount to the total rainfall. This is, of course, not true, because even when there are no drops falling, the piezo element is reading in the range of the mid-teens. Therefore, the next iteration involved pruning values below a certain point, since it was known that these weren't raindrops. The error plot is given in Figure 10, and, as it can be seen, the error is much greater. This is likely due to the fact that each bucket flip happened in roughly the same amount of time, so the pruned values normalized the computed rainfall. Because the rainfall is constant, this wasn't caught in the first data analysis iteration. Any variation in rainfall rates would make this very obvious.

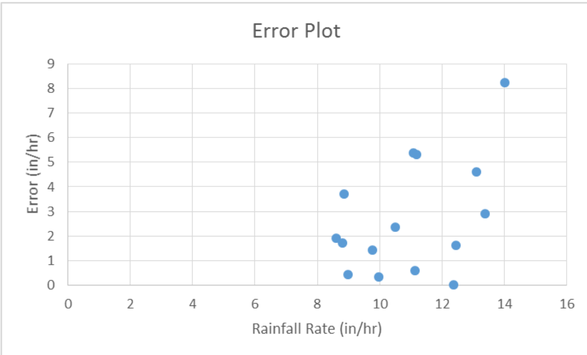

Figure 10: Error Plot When Values Were Pruned

When examining the plot of a single raindrop, it's clear that there's a peak value that signifies the greatest stress on the piezo element. Directly afterwards, there's an echo which represents the piezo element returning to its zero state. The Photon isn't reading negative values, but if it could, then we would see some obvious dips into the negatives after a drop makes an impact. This can be verified with a multimeter or oscilloscope. Therefore, there needs to be a way to look at the data and pinpoint exactly where a raindrop made an impact on the sensor. This is relatively easy to do visually, but to do so in software requires tools more powerful than Excel. While Matlab was shortly used in the experiment, the remaining analysis was done with a Python program using NumPy and SciPy. The most important usage was peak detection, which allowed us to throw out data points that were obviously not drops and try to get more discrete representations of the rain drops.

Note, the following data assumes that there is a linear relationship between the strength of the signal and the size of the drop, although bench testing suggests this isn't true. The analysis was run using the assumption of a quadratic relationship as well, but the results were not as good. The peak values are summed and a least squares analysis is used to determine a function that fits the points that represent the actual rainfall. The blue points on Figure 11 represent the actual rainfall as verified by the tipping bucket, graphed against the sum of the peaks. The lines represent the functions determined from the least squares analyses.

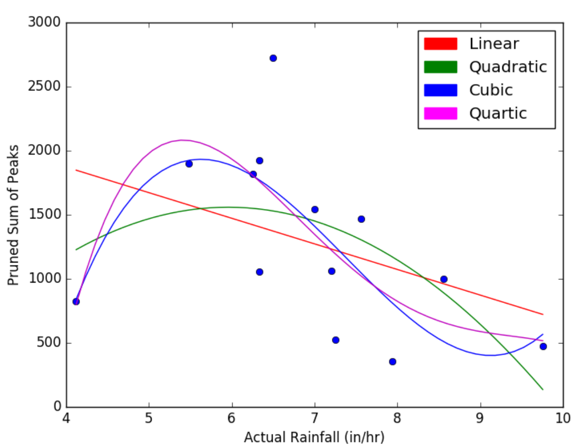

Figure 11: Attempt to Find Function of Best Fit

Unfortunately, while the cubic and quartic functions start to resemble the desired function, sine if the points are so far off the mark (ie. represent such a large error) that no conclusion can be made. In theory, the error should have some correlation to the rainfall rate itself.

Just for the sake of completeness, a least squares analysis of degree 10 was also performed. This is shown in Figure 12. Unfortunately, while many of the points are nearly hit and the error is reduced, the function is not at all reliable for extrapolating rainfall rates given other data.

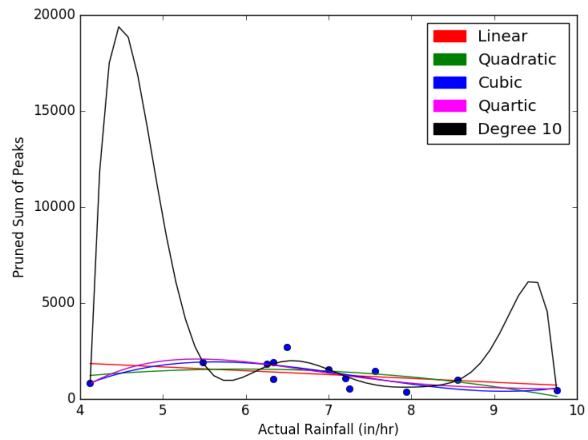

Figure 12: Introduce Degree 10 Function

Much tweaking was done to the program, but ultimately the results are lacking. Certainly, bench testing with slow droplets of water suggest that the principle is sound. However, in the field, especially with rainfall this heavy, there are a number of potential problems, such as lighter drops not being detected, multiple drops hitting at the same time resulting in what appears to be a very large drop, or simply a polling rate that doesn't catch all of the drops. Ideally, more data needs to be collected, and some calibration should be done in the lab using single droplets. If the calibration curve can be tuned well, then the results of this experiment can be promising.

6. REFERENCES

[1] Winder P. M., Paulson, K. S. Direct and indirect measurement of rain drop size distributions using an acoustic water tank disdrometer. *IOP Publishing Measurement Science and Technology* (April. 2013). DOI=10.1088/0957-0233/24/6/065801

[2] de Jong, Stijn. 2010. Low Cost Disdrometer – Improved design and testing in an urban environment. *Delft University of Technology*