Table of Data:

Abstract:

Making use of the photoelectric effect through a photomultiplier tube (PMT) allows the investigation of photon statistics. In this experiment, two statistical models were compared to data taken from a PMT device with an attached counter. In addition, general methods of error reduction and their implementation in observation were explored. As might be expected, an increased number of samples per observational trial generally reduced the error in the measurement of the rate of photons striking the detector. It was also found that the mean value of the photon counts and the variance of the value have a linear relationship, a hallmark of Poisson statistics. As the length of the time samples increased, the Poisson distribution for the probability of counts ceases to be the superior model, giving way to the smooth Gaussian distribution.

Introduction:

This experiment was an exploration of the statistical behavior of photons. The setup was relatively simple – a PMT attached to a device the registers a count when struck by a pulse. The PMT makes use of the photoelectric effect, bouncing a single photon from the LED on dynodes. This process results in the photoelectric emission of multiple electrons from a single photon. When these electrons reach the detector, they register as a pulse, and a photon is recorded by the counter.

Using this photon counting device and some python programs, the data was shifted into probabilities – information about the number of times a set count number would appear during a sample of arbitrary length. These probabilities were compared to two probability distributions to find one that would accurately model the photon behavior. This modeling was then tested at extremes - for very high and low numbers of samples – and over multiple trials.

Observations & Data:

The primary acquisition device used in this experiment was the PMT and its attached USB-4301 counting device. The setup had only rudimentary controls – an overall power switch, a switch to turn the lamp on and off and an unlabeled brightness dial. An associated python script called PMT.py was provided to gather data from the photon counter. This program contained a function called photoncount that took a number of time samples (nsamp), and the length of each sample (tsamp). After processing information for a length of time equal to nsamp multiplied by tsamp, the function returned an array of numbers representing the number of photons counted during the time specified for each sample.

Observations were undertaken over the course of two weeks when the entire lab team could be present. Only one group member recorded data (see Table ~\ref{datatable}), but each individual analyzed the resulting output in its graphical form. Together, it was decided whether the data sets collected were in keeping with predictions or if there were explainable discrepancies.

Since data with relatively large gaps in between each session, efforts were made to ensure that conditions were similar. However, due to the fact that the brightness dial on the PMT was unmarked, it was difficult to record completely consistent data. Ideally the situation would have been such that only the parameters fed to PMT.py would have changed between each step.

Often, the PMT itself would experience issues. Even when the light was off, some count activity was expected, but for long stretches, the PMT would simply output an array of zeroes. When running a loop that gathered many data sets at once, the start of this behavior was not always obvious and had the potential to ruin the set. Another error with the machine or its partner software caused it to report counts in excess of several thousand, regardless of the length of the sample time. This was even more challenging to identify, as only a few counts were misreported, though the fix in this case was simply to remove outlying points.

As mentioned above, the counter reports photons even when the light is off in the PMT. There are a few reasons for this: first that the PMT is not perfectly insulated, and so some stray photons might have made it in from the outside. In addition, the PMT is at room temperature. For everyday materials, 273K is not hot enough to give an electron the energetic kick it needs to escape its bonds (E = frac{3}{2}kT where k is the Boltzmann constant and T is temperature). However, by virtue of the number of electrons and thermodynamic probabilities, some will escape and cause a false report of a count.

Data Reduction & Methods

Data was analyzed using the Python language, largely with original functions written for the task. Plotting the data at every step was done with the pyplot module, part of matplotlib.

The first part of the analysis done for this lab was mostly concerned with plotting data and developing intuition for expected outcomes. Plotting raw data was clearly uninformative (see Figure~\ref{fig:section4}), so a function was written to convert the array of counts into a histogram. It was simply a matter of finding how many times a given number of counts was recorded and plotting that information in a histogram. This proved to be much more relevant than the original plot of the number of counts versus time.

After using the histogram function to plot count frequency for varying parameters, the next step was to make some statistical predictions. This required writing two new functions – one for a Poission distribution, and another for a Gaussian distribution. In addition, it was necessary to calculate the mean and standard deviation of the data sets. The actual procedure and formulae are outlined below (see Section~\ref{sec:calculations}, but the functions needed were not so complex as to be challenging to model. However, due to the limitations of the written factorial function needed for the Poisson distribution, it was necessary to use the scipy.stats Poisson function on data sets with a high number of counts per sample.

Both of these distributions provide output as probability, so a final modification to the raw data was necessary. Rather than plot just the frequency of counts, as in Figure~\ref{fig:histograms}, each such frequency was divided by the total number of counts recorded to provide a probability.

The final matter of concern with regard to the methods used were possible sources of error (as outlined in Section~/ref{sec:observations}). Of particular concern were dark counts, the photon counts re

Calculations & Modelling

The instructions for modeling the behavior of the photon counts were provided in the lab handout, but required some tweaking to achieve the desired results, particularly in the python implementation. One of the most significant challenges was comprehending the behavior of the probability functions and understanding each of the variables.

The first step towards plotting either of the probability distributions was calculating the mean and standard deviation of each data set (see Equations~\ref{eqn:mean},~\ref{eqn:standard deviation}).

With this information, and the histogram modified to display probabilities, it was possible to overplot the Poisson and Gaussian distributions (see Equations~\ref{eqn:poisson},~\ref{eqn:gaussian}).

It is interesting to note that the Poisson distribution (unlike the Gaussian distribution), does not take standard deviation into account. Instead, it bases the standard deviation on the mean ($\sigma$=$\sqrt{\mu}$). This relationship is later demonstrated in the photon data in Figure~\ref{fig:section7}. The other relevant feature of the Poisson distribution is its discrete nature. Since it depends on the factorial of x, it is only meaningful for integer values. This makes sense for a photon count distribution, where the number of counts per time sample is necessarily discrete in order to be physically sensible.

The Gaussian distribution is more widely applicable than the Poisson distribution in that it is continuous