PHYC30300- Advanced Laboratory I

Electrical Noise

Nathan Power

19311361

Abstract

This experiment used an oscilloscope to generate electrical white noise, which was sampled by a computer equipped with a data acquisition board, and then used for a number of counting statistics experiments. A noise discriminator was used to produce a train of random digital pulses over a certain threshold and to control the rate of the counts.
In the second part of the experiment, the output of the noise generator was passed through a low-pass filter and sampled by the computer. It was found that the autocorrelation function for the sample was a decreasing exponential function with a decay constant equal to the time constant of the low-pass filter.

Introduction

Every real electrical circuit generates electrical noise in addition to a signal that we want to measure. When the noise in your circuit is too large, it can significantly affect your measurements. Therefore, understanding the origin of noise and different ways to reduce it in your measurements is very important.
The noise can be either random noise or coherent interference.[1] The random noise is typically intrinsic, that is it is associated with different physical phenomena occurring within your instruments, e.g. thermal motion of charge carriers in your resistors. As it is intrinsic in nature, we can not get a reading of the random noise. However, due to the random nature of this noise, we can significantly reduce its effect on the result of our measurement. This can be done by the signal averaging. Unlike the random noise, the coherent interference is typically extrinsic to your measuring devices and generally arises from periodic, ”man-made” disturbances, such as current fluctuations in the power line.[1] The interference noise can be significantly reduced by proper shielding and grounding of your electrical circuits, the use of isolating transformers etc.

However, electrical noise can also be used as a source of random events for counting experiments. There are many types of events that occur at a well defined average rate, but for which we cannot predict the outcome of any particular measurement. For example, a light bulb has an average lifetime. [2] But we cannot predict how long a particular light bulb will last. This is a characteristic of a random process. The value of the random variable will vary from trial to trial as the experiment is repeated. For the light bulb the experiment is measuring its lifetime, and the random variable is the number of days it produces light before it burns out.[2] The counts in electrical noise is a random process. We cannot predict exactly when a count will be detected, but only the average number of counts in a certain time interval. If we actually count the number of pulses in several time intervals, we get a different number of pulses each time, but they have a definite mean.

Assume we count the number of times N a given random event occurs in a certain time interval. Theory predicts that if we repeat our experiment k times, we can calculate the average number of counts $M = [\Sigma_1^m N_i]/k$. The actual number of counts will be distributed around this average value. Numbers close to the average will be recorded frequently, numbers very different from the average will be recorded infrequently. [2]

For a large number of rare events we find that the probability of recording a particular number N is given by the Poisson distribution:

$$P(N) \ = M^{N} e^{\frac{(-M)}{N!}} \tag{1}$$

For a large number of frequent events we find that the probability of recording a particular number N is given by the Gaussian distribution [2]:

$$P(N) \ = [\frac{1}{2\pi M}]^{\frac{1}{2}} \ . \ e^{\frac{-(N-M)^{2}}{2M!}} \tag{2}$$

Theory

Poisson vs Gaussian Distributions

The Poisson distribution is used to describe discrete quantitative data in which the population size is large, the probability of an individual event is small. Current examples can be the number of covid cases in a town per day, or the number of admissions to a particular hospital. Here a version of a Poisson distribution is used to quantify superspreading for COVID-19. [4]
The Gaussian distribution is completely described by two parameters μ and σ, where μ represents the population mean and σ the population standard deviation. It is symmetrically distributed around the mean. This population distribution can be estimated by the superimposed smooth `bell-shaped' curve or `Normal' distribution shown.[5]

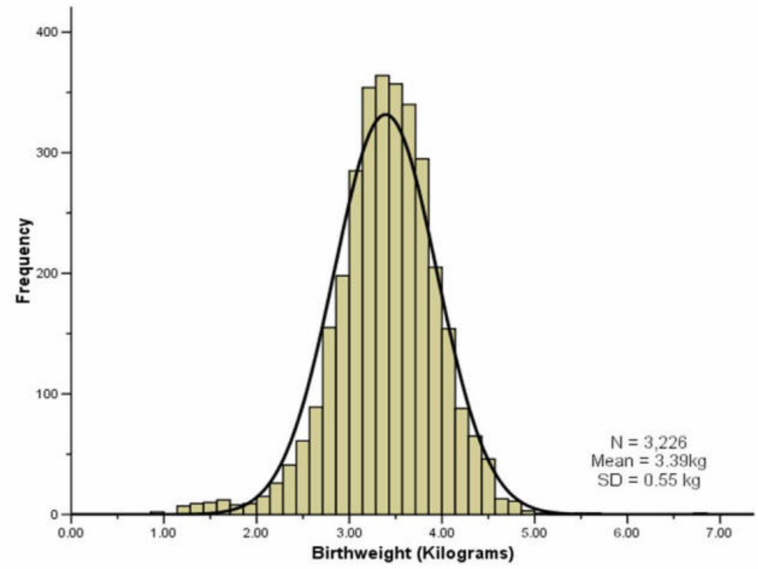


Fig (1) Gaussian distribution showing the frequency of the birthweight of babies [5]

Auto-correlation

Autocorrelation refers to a correlation between a set of time signals with an old version of itself. The two sets of time signals have some time difference between them. Theoretically, white noise has no correlation between the noise voltage at one time and at another as it is a random process.The autocorrelation function is zero everywhere except t=0. This is because the noise voltage changes so rapidly that there is no correlation between the voltage levels over short periods of time. The reason for the rapid changes is that the noise contains high frequency components. But by using the low-pass filter these high frequency components are removed and the noise voltage does not change so rapidly. The filter then produces noise which shows autocorrelation over short periods. It would be found that the autocorrelation function for the sample would be a decreasing exponential function with a decay constant equal to the time constant of the low-pass filter as calculated in [3]

Experimental Procedure

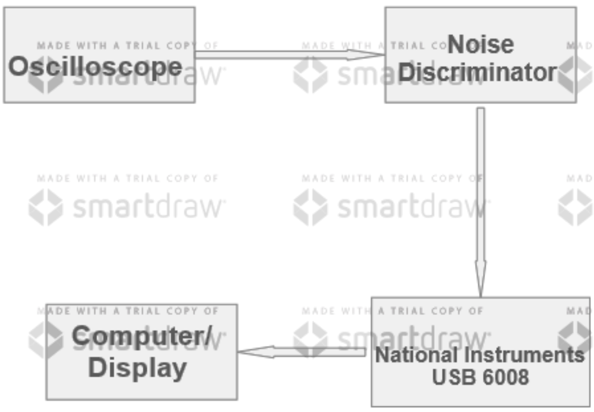


Fig.1 Box Diagram of Circuit

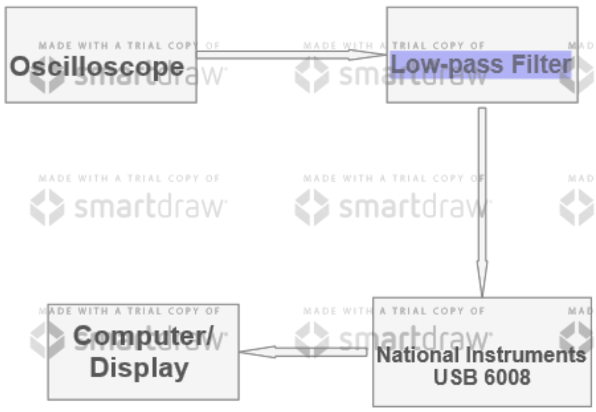


Fig. 2 Experiment Apparatus

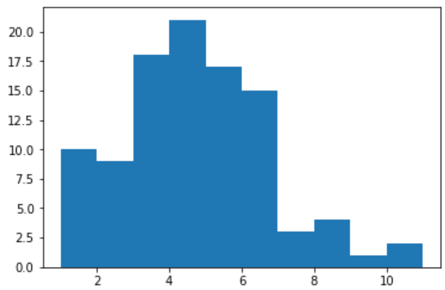
In this experiment, an oscilloscope, noise discriminator and the National Instruments USB 6008 were used. The oscilloscope is used to generate white electrical noise which was then used as a source of random events. The analogue noise was firstly sent through a discriminator to produce a train of random digital pulses. The discriminator produced a digital output pulse when the noise amplitude exceeded the threshold. The discriminator was used to control the count rate of these pulses.

The pulses were then recorded and analysed using the National Instruments USB 6008. Using this, the properties of counting statistics were investigated such as, the Gaussian approximation to the Poisson distribution and the time-interval distribution of events in a Poisson process.

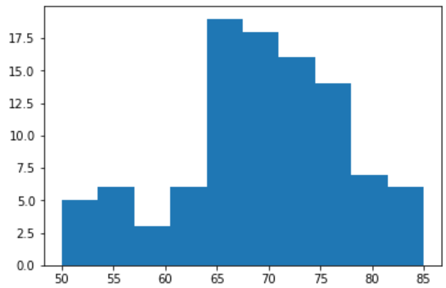
In the second part of the experiment, a low-frequency filter was used to find the autocorrelation function of a sample of generated noise. This is done by passing the noise generated by the oscilloscope through the filter and then recording and analysing the output using the National Instruments USB 6008 model.

Results and Discussion

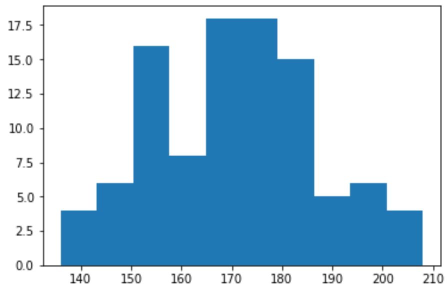
Investigation of properties of counting statistics



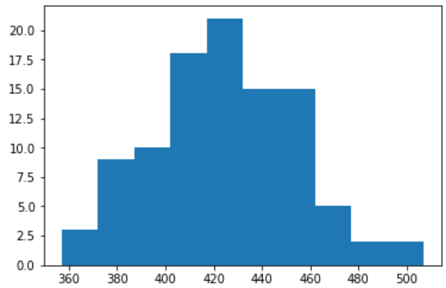
Fig(3) Histogram of Counts with mean = 4.24



Fig(4) Histogram of Counts with mean = 69.12

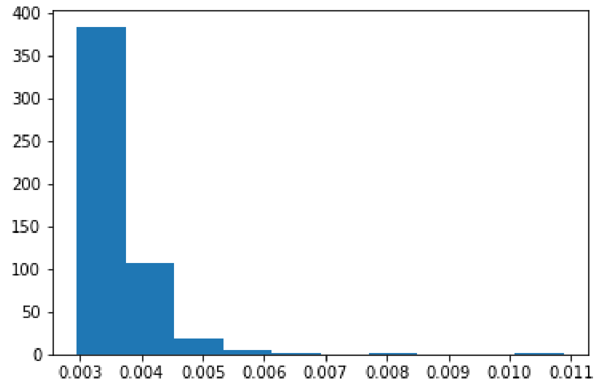


Fig(5) Histogram of Counts with mean = 170.67



Fig(6) Histogram of Counts with mean = 424.5

Firstly, the Poisson distribution was investigated, specifically the Gaussian approximation to the Poisson distribution. Starting from a low count rate and building up, it can be seen that as the mean count rate increases the distribution becomes more symmetrical and Gaussian.



Fig(7) Histogram showing the time intervals between consecutive counts

Next, the time interval between consecutive counts was recorded and analysed.
The distribution of the time intervals is a negative exponential, which is a property of Poisson distribution models. This distribution has a memoryless property, which means it “forgets” what has come before it. In other words, if you continue to wait, the length of time you wait neither increases nor decreases the probability of an event happening. Any time may be marked down as time zero.[6]

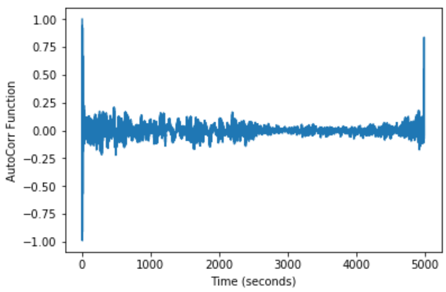
Next, we investigated the weighted mean and the errors of the mean of a sample of noise data. Finding the weighted mean of a sample involves multiplying each data point in a set by a value which is determined by some characteristic of whatever contributed to the data point. Therefore, a larger set of data can be divided into subsets and using the weighted mean formula and the means and errors of each subset, the weighted mean will agree with the mean of the set as a whole. The errors on each set was the square root of the number of counts within the set.

Mean of Total Set = 146.78 +- 10.0	
Mean of Set 1	151.5 +- 3.16
Mean of Set 2	142.75 +- 3.46
Mean of Set 3	146.73 +- 3.87
Mean of Set 4	151.33 +- 3.46
Mean of Set 5	147.75 +- 5.75
Mean of Set 6	142.10 +- 4.24
Weighted Mean = 146.78 +- 4.34	

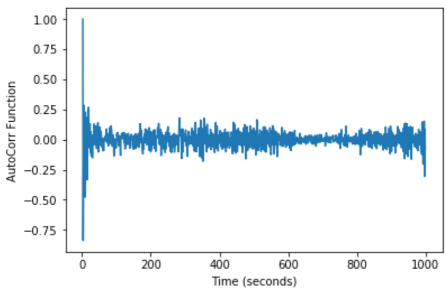
Table(1) Table showing the mean of a set agrees weighted mean of its subsets

Autocorrelation

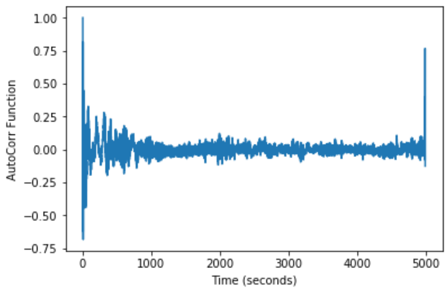
Autocorrelation refers to a correlation between a set of time signals with an old version of itself. The two sets of time signals have some time difference between them. By taking data with and without using the low-pass filter, we can see without the filter that noise is random and there was correlation from the noise voltage at one time and at a later time. The autocorrelation function is zero everywhere except t=0 as the noise voltage changes so rapidly due to high frequency components. But as the low-pass filter removes these high frequency components which shows autocorrelation over short time frequencies. [3]



Fig(8) Autocorrelation function of the noise with no filter



Fig(9) Autocorrelation function of the noise with filter of RC = 2.7 ms



Fig(10) Autocorrelation function of the noise with filter of RC = 4.7 ms

The data recorded in this section was extremely noisy. This makes the graphs difficult to read and not as accurate. This could have been made more clear if more data points were taken. Therefore, it was found that the autocorrelation function for the sample was a decreasing exponential function with a decay constant equal to the time constant, RC, of the low-pass filter.

Conclusion

From our results, it can be clearly seen that our data analysis agrees with the expected results from counting statistics experiments. Theoretically, the difference between Poisson distribution and Gaussian should be greatest when there are few counts, and become smaller as the number of counts increases. This was consistent with our findings. The distribution of the time intervals in our data was found to be a negative exponential, which also agrees with the properties of Poisson distribution models. It was also found that the autocorrelation function for the sample was a decreasing exponential function with a decay constant equal to the time constant of the low-pass filter. This was consistent with the findings found by *J. Loren Passmore Brandon C. Collings,* and *Peter J. Collings*.[3] However, the data recorded in this section was extremely noisy. This could have been made more clear if more data points were taken.

References

[1]Okinawa Institute of Science and Technology Graduate University, *Noise in Electrical Circuits: Origin of Noise Handou*t, Date Last Accessed; 30/11/2021, https://groups.oist.jp/sites/default/files/imce/u155/Handout_3_noise.pdf

[2]The University of Tennessee, *Lab 7: Counting Statistics*, Date Last Accessed; 30/11/2021, http://electron6.phys.utk.edu/phys250/Laboratories/counting_statistics.htm

[3]Passmore, J.L., Collings, B.C. and Collings, P.J., 1995. Autocorrelation of electrical noise: an undergraduate experiment. *American Journal of Physics*, *63*(7), pp.592-595.

[4] Kremer, C., Torneri, A., Boesmans, S., Meuwissen, H., Verdonschot, S., Driessche, K.V., Althaus, C.L., Faes, C. and Hens, N., 2021. Quantifying superspreading for COVID-19 using Poisson mixture distributions. *Scientific reports*, *11*(1), pp.1-11.

[5] MJ Campbell 2016, S Shantikumar 2016, *HealthKnowledge, Standard Statistical Distributions (e.g. Normal, Poisson, Binomial) and their uses*, Date Last Accessed; 30/11/2021, https://www.healthknowledge.org.uk/public-health-textbook/research-methods/1b-statistical-methods/statistical-distributions

[6]Stephanie Glen, *Exponential Distribution / Negative Exponential: Definition, Examples* From StatisticsHowTo.com: Elementary Statistics for the rest of us! https://www.statisticshowto.com/exponential-distribution/

In [4]:
```python
from pydaqmx_helper.counter import Counter

import time
import numpy as np
import matplotlib.pyplot as plt
```

In [5]:
```python
mycounter = Counter()
```

Function used to Count the Pulses

In [6]:
```python
def fun_1(t,N):
    i=0
    output=np.array([])
    while(i<N):
        mycounter.start()
        time.sleep(t)
        val=mycounter.stop()
        mycounter.stop()
        output=np.append(output,val)
        i+=1
    print(output)
    print(len(output))
    return output
```

Function used to record the time interval between consecutive Counts

In [3]:
```python
def TimeInterval(N):
    i=0
    output=np.array([])
    while(i<N):
        t1=time.perf_counter()
        mycounter.start()
        val=mycounter.stop()
        mycounter.stop()
        if val == 1:
            t2=time.perf_counter()
            output=np.append(output,t2-t1)
        i+=1
    print(output)
    return output
```

Weighted Mean

In [141…
```python
vals5=fun_1(0.1,100)
split=np.split(vals5,[10,22,37,49,82])
```

```
[153. 133. 162. 156. 140. 146. 156. 169. 140. 160. 149. 144. 137. 148.
 143. 136. 145. 139. 143. 154. 139. 136. 155. 133. 126. 164. 133. 152.
 139. 128. 154. 144. 139. 157. 159. 156. 162. 156. 175. 157. 149. 144.
 134. 146. 170. 136. 164. 157. 128. 145. 145. 114. 154. 142. 164. 170.
 147. 128. 142. 146. 129. 155. 153. 139. 141. 160. 136. 155. 126. 180.
 155. 166. 150. 136. 144. 159. 129. 154. 167. 155. 152. 138. 145. 144.
 158. 135. 127. 133. 168. 151. 159. 118. 137. 147. 131. 140. 131. 117.
 154. 162.]
100
```

In [154…
```python
print("Mean of Total Set = ", np.mean(vals5),"+-", np.sqrt(len(vals5)))
print("Mean of Set 1 = ", np.mean(split[0]),"+-", np.sqrt(len(split[0])))
print("Mean of Set 2 = ", np.mean(split[1]),"+-", np.sqrt(len(split[1])))
print("Mean of Set 3 = ", np.mean(split[2]),"+-", np.sqrt(len(split[2])))
print("Mean of Set 4 = ", np.mean(split[3]),"+-", np.sqrt(len(split[3])))
print("Mean of Set 5 = ", np.mean(split[4]),"+-", np.sqrt(len(split[4])))
print("Mean of Set 6 = ", np.mean(split[5]),"+-", np.sqrt(len(split[5])))

w_x=(np.mean(split[0])*len(split[0]))+(np.mean(split[1])*len(split[1]))+(np.mean(split[2])*len(split[2]))+(np.mea
weighted_mean= w_x/len(vals5)
w_x_err=(np.sqrt(len(split[0]))*len(split[0]))+(np.sqrt(len(split[1]))*len(split[1]))+(np.sqrt(len(split[2]))*ler
print(w_x_err/100)
```

```
print("Weighted Mean  = ", weighted_mean,"+-", w_x_err/100)
```

```
Mean of Total Set =  146.78 +- 10.0
Mean of Set 1 =  151.5 +- 3.1622776601683795
Mean of Set 2 =  142.75 +- 3.4641016151377544
Mean of Set 3 =  146.73333333333332 +- 3.872983346207417
Mean of Set 4 =  151.33333333333334 +- 3.4641016151377544
Mean of Set 5 =  147.75757575757575 +- 5.744562646538029
Mean of Set 6 =  142.05555555555554 +- 4.242640687119285
4.387940652620032
Weighted Mean  =  146.78 +- 4.387940652620032
```

AutoCorrelation Function

In [ ]:

```
from pydaqmx_helper.adc import ADC
myADC= ADC()
myADC.addChannels([3])
val=myADC.sampleVoltages(100,200)
val1=val[3]
```

In [ ]:

```
def autocorr1(x,lags):
    corr=[1. if l==0 else np.corrcoef(x[l:],x[:-l])[0][1] for l in lags]
    return np.array(corr)
```

In [ ]:

```
lags=np.arange(1,101)
r=autocorr1(val1,lags)
plt.xlabel("Time (seconds)")
plt.ylabel("AutoCorr Function")
plt.plot(np.flip(r));
```