Single Photon Experiments 2023

Tai Xiang

May 4, 2023

Contents

All code and 3-D printing files can be found in the github resposity at https://github.com/tyxiang0530/qrand.

1 Single Photon Detection Set-up

The basic set-up involves a helium-neon laser in the IR band driven into a BBO crystal. Photons that enter the BBO crystal from the laser undergo spontaneous parametric down-conversion and are then detected within two photon detectors, as shown in figure 1.

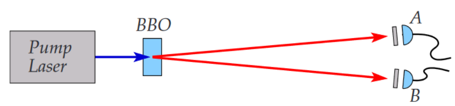


Figure 1: Basic set-up for the SPCM experiment.

Additional beam-splitters can be added between the BBO and the detectors to yield a set-up such as the one shown in figure 2.

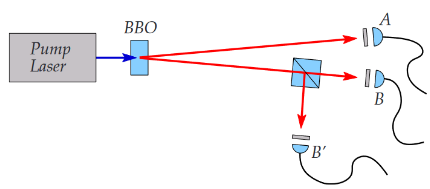


Figure 2: Beam-splitter set-up for the SPCM experiment.

This set-up will be the primary set-up used for the experiments and upgrades listed in the later sections.

2 Photon Statistics and Randomness

One of the extensions I explored with the single photon counting module was to apply it to the task of random number generation. Exploiting the quantum intepretation of photons traveling within a beam-splitter and the Poissonian nature of light, a high bitrate random number generator can be developed from the current CCM experimental set-up. The random numbers generated from this set-up can then be used to map out password strings or generate security keys.

With the beam-splitter source of randomness, the set-up generates random bits by observing what detector measures a photon. In the set-up shown in figure 2, a 50–50 beam-splitter is placed between the detector B and B'. In the quantum mechnical interpretation of light, the 50–50 beam-splitter causes an incident photon to have a 50–50 chance of arriving within detector B or B'. Assigning a bit of 0 to B and a bit of 1 to B', we then have a 50–50 chance of getting a bit of 1 or 0. By inserting an additional detector A' that is analogous to B and B', and then adding a 50–50 beam-splitter between A and A', another random bit can be generated.

Furthermore, since photons are emitted from the laser in a Poissonian nature, we can split off a Poissonian distribution fitted to the count of arriving photons, assign buckets to each, and draw random bits in accordance to what bucket a time-step of photon counts falls into. This illustrated in figure 3.

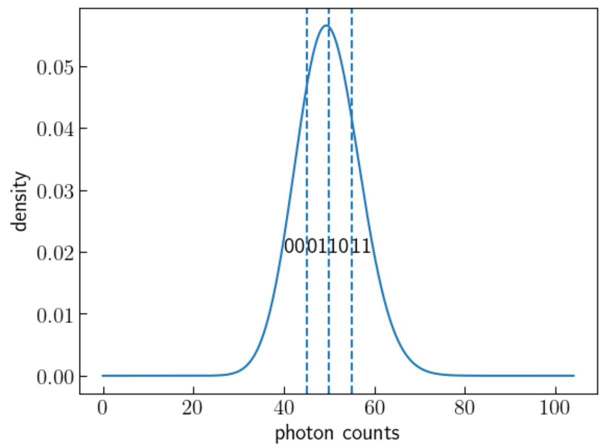


Figure 3: Poissonian distribution segmented into regions of equal area. A bit combination is then assigned to each region.

For example, taking an experiment fitted to the Poissonian displayed in figure 3, if there are 40 incident photons, then we get an output bitstring of 01, if there are 45, we output 01, and so on. A Poissonian can be drawn for each detector in the experiment. All software pertaining to the various Poissonian operations mentioned in this report can be found in the python file PoissonianOps.py.

In an experiment with two beam-splitters and four detectors utilizing each of the ports available on the SPCM-AQ4C module, a total bitrate of 10 bits per integration time-step can be achieved. In the generation of a password or security key, there are a total of 10 numbers, 40 symbols, 26 lower-case letters, and 26 upper-case letters to choose from, yielding a total of 102 possible selections. A total of 7 bits can reach every single combination of this, with $2^7 - 102 = 28$ additional mappings that are not used and can be overflowed to the next time-step. Each experimental time-step generates 10 bits, so we use 7 to generate a character and overflow the additional 3 bits as well as the additional mappings to the next time-step for efficiency.

We validate the random bits we generate with the NIST statistical test suite for random number generates. The NIST statistical test suite contains a series of tests utilized to determine the randomness of input number sequences. The results of the tests are shown in table 2, where a score of 8/10 is considered passing.

Statistical Test	**Proportion**
Frequency	10/10
Block Frequency	10/10
Cumulative Sums	10/10
Longest Run	10/10
Longest Run (Rank)	10/10
Longest Run (FFT)	10/10
NonOverlapping Template	10/10
NonOverlapping Template	9/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	9/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	8/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	9/10
NonOverlapping Template	9/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	9/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
NonOverlapping Template	10/10
Random Excursions Variant	10/10

The random numbers generated with our set-up pass the NIST statistical tests. However, although the set-up performs extremely well on these tasks, it should be noted that the passing of statistical tests is not a mathematically rigorous definition of "randomness". The set-up merely performs well on human-defined metrics of randomness.

To ensure that our experimental set-up is running accordingly and that there is no systematic error that is causing non-Poissonian measurements, I have developed a python file Main.py that has a method gen_report that automatically generates a report that fits a set of Poissonians to each randomness source we are relying on and investigates the fit. The report generated also enables the user to analyze the CDF of each channel, as well as a set of basic statistics for the given experimental run such as the count rate and the total counts over time.

3 Scattering Reduction

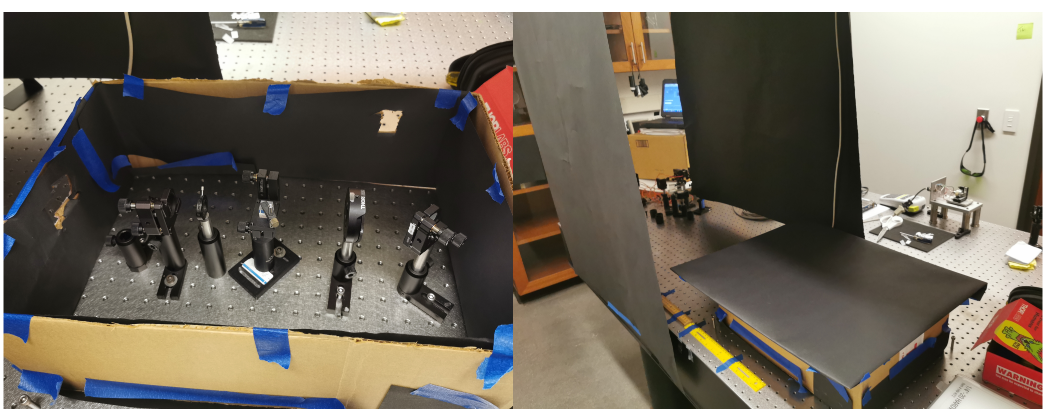
An additional improvement I have made to this experimental set-up is the development of multiple schemes to reduce scattering from background noise sources. This section will cover a set of optimizations and developments on this front.

3.1 Black Paper and Boxes

A very basic optimization that I performed to reduce scattering was to place long sheets of black paper along the general beam-path of the laser. This was done to ensure scattered light from sources external to the experiment, such as light from the laptop screen and light from the controls of the coincidence counting module, do not interfere with the experiment and generate accidental coincidences.

Additionally, I also added a box that fits the optical components that are along the beam-path prior to the BBO crystal. I cut a small hole that enables the beam to travel into the box and another small hole that enables the beam to exit. Any scattering that arises from the components inside the box is thus contained within it and does not leak out into the room and contaminate the photons arriving at the detectors.

The additional pieces of paper along the beam-path as well as the box enclosure as shown in figure 4.

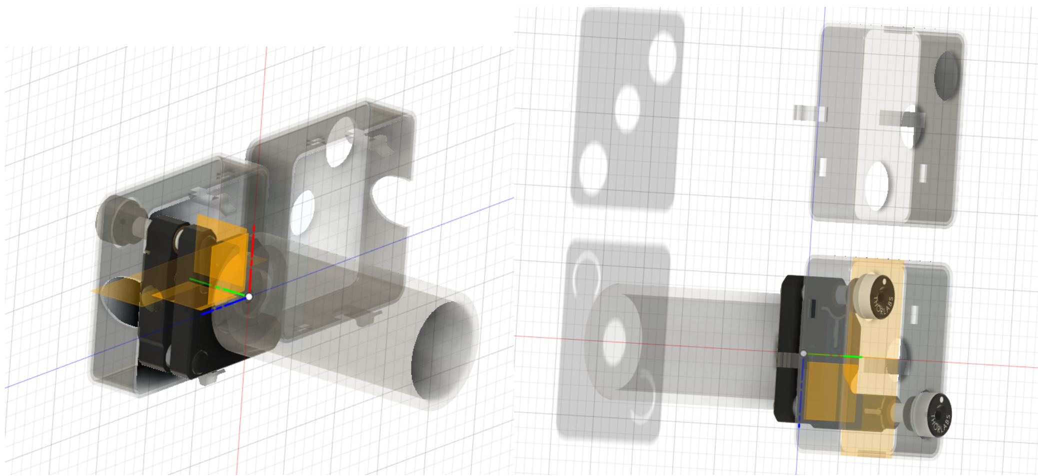


(a) (b)

Figure 4: Figure 14a shows the box that encloses the optical elements and figure 4b shows the additional black paper placed around the experiment.

3.2 Enclosures

Another scattering reduction optimization I have made is to create a 3-D printable enclosure for the detectors that reduces the chance of entry for noisey scattered photons. The two-piece enclosure fits and snaps over the detectors and adds a cylindrical nose to the front. The long cylindrical nose reduces the chance of photons that do not arise from the beam-path from hitting the detector. The CAD designs are shown in figure 5.



(a) (b)

Figure 5: Figure 5a displays the side view of the enclosure. Figure 5b shows a back view. The two-piece enclosure snaps together through two joints and can easily be taken apart. The cylindrical nose must be glued on to the front of the enclosure.

Physically, the enclosure is shown in figure 6.

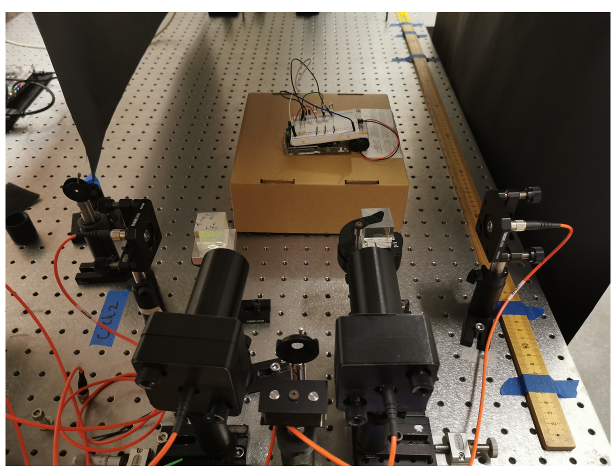
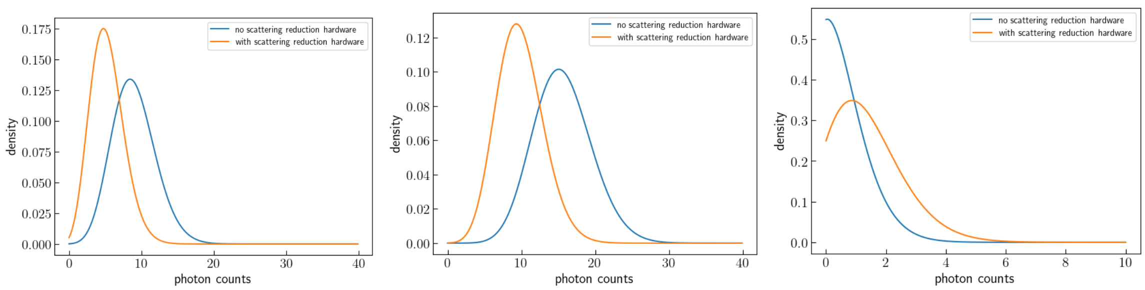


Figure 6: Enclosure inserted into experiment.

The cylindrical nose-tube is also designed to fit a filter, such as the FR RG780. When the enclosure was first designed, the current filter was used in the experiment. When these optimizations are applied to the basic experimental set-up shown in figure 1, we observe the results shown in figure 7.



(a) (b) (c)

Figure 7: Figure 7a shows the difference between the Poissonian distribution of photon counts per integration time-step in detector A with and without scattering reduction hardware. Figure 7b shows the difference between the Poissonian distribution of photon counts per integration time-step in detector B with and without scattering reduction hardware. Figure 7c shows the difference between the Poissonian distribution of photon counts per integration time-step in A-B detector coincidences with and without scattering reduction hardware.

We observe that the scattering reduction steps that have been taken cut down on the Poissonian mean of photon arrivals in channel A and B without a major reduction in A–B coincidences. This suggests that the optimizations are cutting out background nose but are not affecting the coincidences generated from down-converted photons (the signal).

3.3 Filters

Filters that only enable light in the IR range to pass through are effective in reducing noise, as our laser operates in the IR range. At the beginning of the experiment, we utilized the filters of type *abcd*. An easy optimization we can make to the experiment is to improve the quality of the filter. We test the FR RG780 in place of the *abcd*. The FR RG780 is a longpass filter with a filter profile that is shown in figure 8.

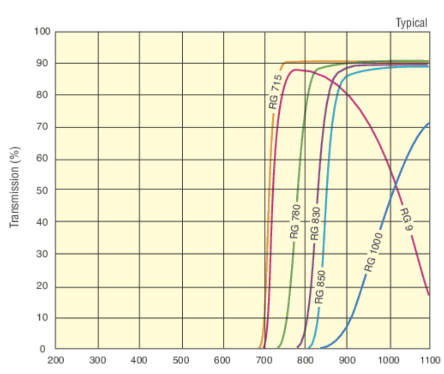
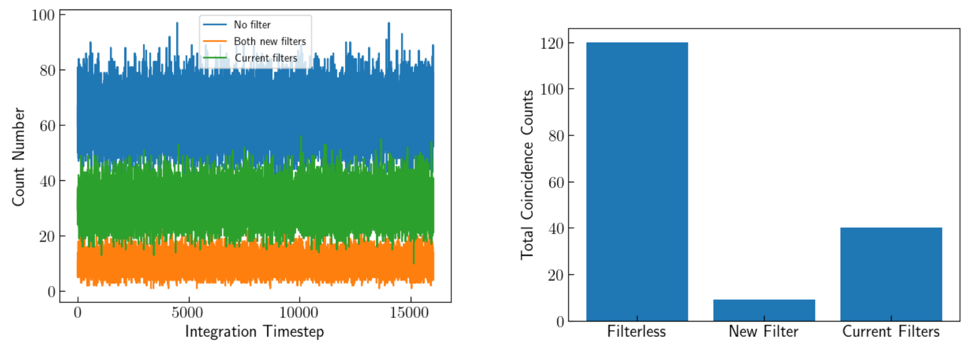


Figure 8: Filter profile of RG780 shown in green.

We compare the results of the experiment with no filter, with the old filters, and with the new RG780 filters. We perform tests with the purple laser, a maglite torch, and an IR LED array.

3.4 Laser

For the laser, when analyzing initial counts over time, we observe the results shown in figure 9.



(a) (b)

Figure 9: Figure 9a displays the count numbers measured over each integration timestep for a set-up with no filters, old filters in both the nose tube and following the output optical fiber, and the RG780 filter in both the nose tube and following the output optical fiber. Figure 9b shows the coincidences measured in each configuration.

We observe that the variance in counts go down in accordance to 1 over the shot noise, and we see that the RG780 filter outperforms the current filter in regards to attenuation of the measured count numbers. The coincidence counts also decrease.

Another useful metric we can examine is the α value for each configuration. α is the anticorrelation parameter. For two detectors and a channel for coincidence measurement, we have

$$\alpha_{2d} = \frac{R_c}{R_{acc}^{(2r)}}. \tag{1}$$

Here, R_{acc} denotes the rate of accidental coincidences and is defined to be

$$R_{acc} = 2\tau R_1 R_2 \tag{2}$$

where τ is the coincidence window and R_1 and R_2 denote the count rates for channel 1 and channel 2. R_c denotes the coincidence rate. For correlated sources, such as photon pairs arriving from SPDM, we expect $\alpha > 1$. For an uncorrelated source, such as white light from a torch, we expect $\alpha = 1$.

For each of the filter configurations, we can calculate some α value, finding the results shown in table 3.4.

Configuration	α Value
No Filters	0.645
Current Filters	0.670
RG780	0.764

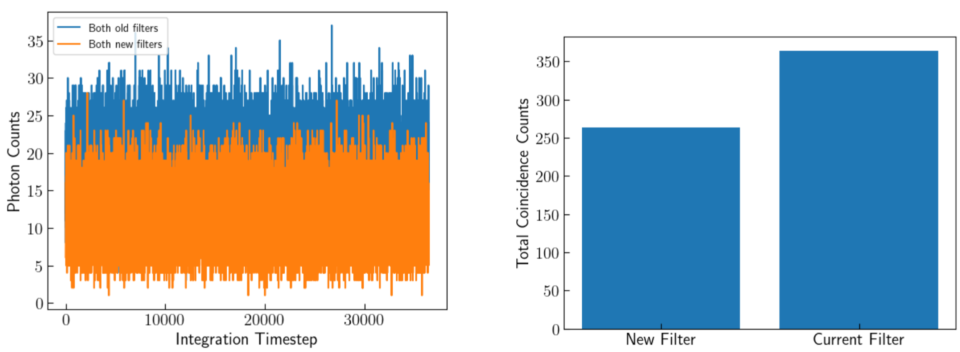
Table 1: Table of α values for various filter configurations with a laser photon source.

Curiously, for the laser source, we do not observe an α that is greater than 1, which is what we would expect for a correlated photon source. This defect will be addressed in section 4. However, with the new RG780 filter, we achieve a higher α value, indicating that the accidental coincidences from the experiment are decreases and we observe stronger correlation. This suggests that the RG780 filter outperforms our current filters.

3.5 Torch

We perform the same set of tests described above with a maglite torch. The torch serves as an uncorrelated photon source, and sends light outwards in a conal shape. We additionally test this configuration to understand how well the new filter performs in the presence of a source that primarily emits photons at a wavelength that is attenuated by the filter.

We once again examine the counts over time and the coincidences measured, though in this test we omit the baseline filter-less case as the counts coming in are too high and excessively compress the axis of our plots. We observe the results shown in figure 10.



(a) (b)

Figure 10: Figure 10a displays the count numbers measured over each integration timestep for a set-up with old filters in both the nose tube and following the output optical fiber and the RG780 filter in both the nose tube and following the output optical fiber. Figure 10b shows the coincidences measured in each configuration.

Once again, we witness an attenuation in counts and in coincidences. For the case of the torch, we observe the α values shown in table 3.5.

Configuration	α Value
Current Filters	0.638
RG780	0.677

Table 2: Table of α values for various filter configurations with a torch photon source.

Interestingly, we once again observe results that contradict theory in that our α value is not close to 1. However, we do once again observe that the RG780 filters outperform the current filters in reducing the number of accidental coincidences and yielding a higher α value.

3.6 IR LED Array

We also examine the performance of our filters when an IR LED array is placed in front of our detectors. This varies from the torch in that the wavelength of photons emitted by the IR LED is a range the filter does not attenuate but the photon source once again only produces uncorrelated photons. We set-up our IR LED array in accordance with the schematic shown in figure 11.

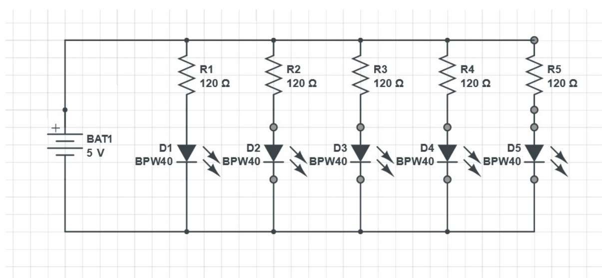
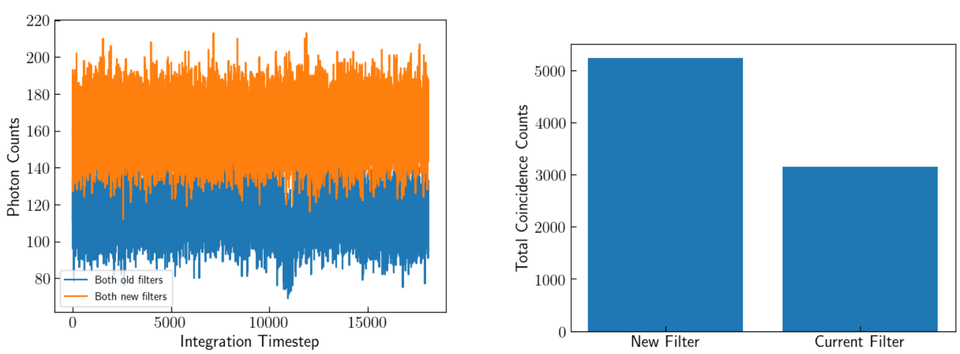


Figure 11: Schematic of IR LED array

We perform the same measurements as those in the above sections and once again omit the filterless detector due to high counts. We observe the results shown in figure 12.



(a) (b)

Figure 12: Figure 12a displays the count numbers measured with the IR LED array set-up over each integration timestep for a set-up with old filters in both the nose tube and following the output optical fiber and the RG780 filter in both the nose tube and following the output optical fiber. Figure 12b shows the coincidences measured in each configuration.

Interestingly, we observe that for the IR LED array, we observe a higher quantity of counts with the RG780 than with the old filters. This suggests that the RG780 filter attenuates the actual signal less, and has a steeper slope at the cutoff 780nm frequency.

When we examine the α values for this set-up, we observe the values shown in table 3.6.

Configuration	α **Value**
Current Filters	0.66
RG780	0.63

Table 3: Table of α values for various filter configurations with a torch photon source.

Here, we observe that the RG780 filter yields a lower α value than that of the current filter, suggesting that it measures more accidental coincidences than the current filters. Though this is a negative, the strong cutoff and the high transmission of photons within the signal bandwidth demonstrates that the RG780 is a superior filter than the current filter.

4 Defects in Current Coincidence Counting Module (CCM)

In the previous section, we observe a set of measured α values that do not align with theoretical results, namely the measurement of uncorrelated light yields αs on the order of 10^{-1} and the measurement of correlated sources does not yield $\alpha > 1$.

We believe that this result is due to defects in the current coincidence counting module. To further investigate this result, we analyze the pulse-shaper and the channel-by-channel variance of the counter to better understand why this error may arise.

4.1 Coincidence Windows and the Pulse Shaper

We first analyze the pulse shaper in the CCM. The pulse shaper alters the waveform of a photon detection event, decreasing the pulsewidths (equivalent to the parameter τ) to those shown in table 4.1.

Pulsewidth Setting	**Pulsewidth**
Short	10 ± 2.5ns
Medium	14 ± 2.7ns
Long	18 ± 2.8ns
Unaltered	25ns

Table 4: Table of α values for various filter configurations with a torch photon source.

Coincidences are measured by examing waveform overlaps between two detectors, and thus a decrease in pulsewidth may lead to a decrease in accidental coincidences, as two waveforms must arrive within a smaller increment of time for overlap, and subsequently coincidence to occur.

We investigate the α values yielded for different pulsewidths.

Pulsewidth Setting	α
Short	0.653
Medium	0.755
Long	0.748
Unaltered	0.930

Table 5: Table of α values for various filter configurations with a torch photon source.

We find these results to be quite odd. A scaling of α with the pulsewidth can be expected, as an increase in pulsewidth leads to an increase in coincidences, as pulses from the two detectors are now more likely to overlap. However, scaling at which this occurs, especially where medium and long equal one another , is rather strange and unexpected. This leads us to believe that there may something wrong with our pulse-shaping circuit.

We first validate the pulse-shaper settings by measuring output waveforms from the CCM on an oscilloscope. The pulsewidths are sound, and align well with the values we have in table 4.1.

4.2 Channel Variance

We continue to investigate the current CCM and analyze if there is any variance when the detectors are put into differing input channels. If some variance occurs, this would suggest that there is some inconsistency between the various inputs of the CCM. We put detectors A and B into channel 1 and 2, channel 2 and 3, and channel 3 and 4, with each taking measurements from the same experimental system. We measure α for each, observing the results shown in table 4.2.

Pulsewidth Setting	α
1, 2 input	0.830
2, 3 input	0.651
3, 4 input	0.685

Table 6: Table of α values for different channel inputs.

This drastic variance in α value suggests that there is something wrong with our CCM. Each experimental run takes measurement from the same system, and the CCM ought to process inputs from different channels in the same manner. However, we see that in the results shown in table 4.2, this is not the case.

5 Altera DE-2 FPGA CCM

We seek to develop an alternative coincidence counting unit with an Altera DE-2 FPGA CCM. This is based upon the information provided from Mark Beck and their DE-2 coincidence unit.

The Altera DE-2 is a design and education board with a wide variety of uses. We primarily leverage it for the Cyclone FPGA in the system and use it to count coincidences. Similar to the current CCM, the Altera FPGA can take in a set of ttl signals, check for overlap between the signals, and count coincidences. Like the current CCM, the Altera FPGA is easily programmable and has a high enough bit count to measure every possible combination of coincidences for a four input experiment (A, A', B, B').

5.1 Breakout Board

To convert the output TTL signals from the SPCM AQ4C, we need a breakout board to convert the output signals into 3.3V logic signals, which the Altera FPGA uses. To do this, we utilize the schematic for our voltage divider shown in figure 13.

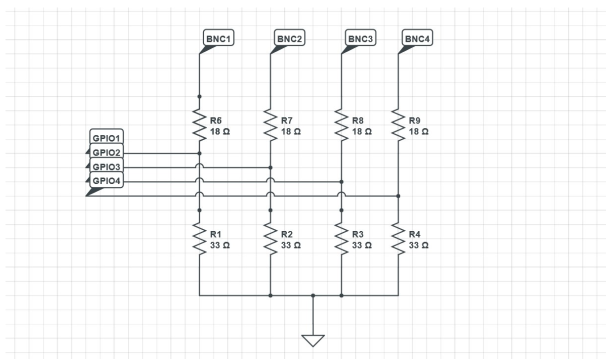
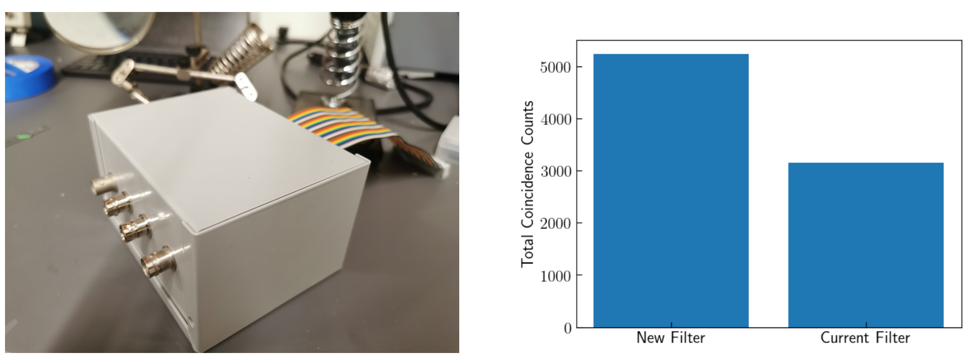


Figure 13: Breakout board schematic.

Signals enter into the circuit through four individual BNC inputs for each channel and exit exit through a ribbon cable and then into the DE-2 GPIO inputs. The full breakout board is shown in figure 14.



(a) (b)

Figure 14: Figure 14a displays the enclosure for the breakout board circuit. Figure 14b displays the implemented breakout board circuit

.

It should be noted that there are only 4 GPIO pins that are used on the FPGA, and thus outputs from the voltage divider must be connected to the correct pins on the ribbon cable to ensure that the four GPIO pins are correctly addressed. Here, we use a raspberry pi ribbon cable, and thus each of the inputs must be sent to the raspberry pi ribbon cable inputs in accordance to table 5.1.

Channel	Pi Input	FPGA Input
1	1	GPIO13
2	2	GPIO15
3	3	GPIO17
4	4	GPIO19

Table 7: Table of channel inputs to ribbon cable and FPGA inputs

When this circuit is implemented, we observe that the signals from the SPCM AQ4C have been adequately attenuated to the correct levels, as shown in figure 15.

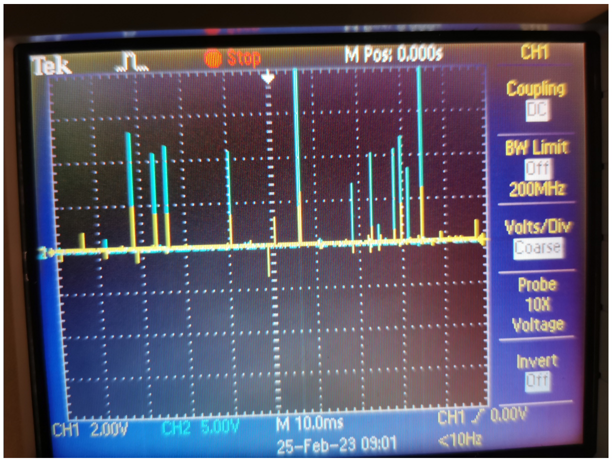


Figure 15: Attenuation of SPCM AQ4C output signals to 3.3V logic.

5.2 Software

The software used to drive the Altera-based CCM (ACCM) is taken from Mark Beck's website. The advantage of the ACCM over the old CCM is that it can be modified for use with a wide variety of other experiments, such as testing Bell's inequality, local realism, single photon polarization states, and more. Some of the experiments require some additional upgrades in equipment, but the core software can all be executed on the ACCM.

We validate the ACCM by performing a basic experiment with the maglite torch. We hook the ACCM up with the coincidence counting library in labview. Through the front panel, we observe that we are getting counts, and we also observe that we have count fluctuations that are consistent with expectations when the torch is moved around. We take data for three different positions of the maglite torch and observe some concerning results shown in table 5.2.

Trial Number	α
1	36.43
2	62.70
3	42.10

Table 8: Table of α values for different trials with the ACCM.

These results are highly inconsistent with the outputs we expect for the torch source and suggest an excessive coincidence count. There are further issues in the ACCM that need to be fixed.

5.3 Possible Problems

Some additional refinements and improvements can be made to the ACCM setup to possibly resolve some of the problems that we have observed. One major improvement would be to design the voltage divider circuit in a CAD program such as Altium and create a PCB. The current circuit involves a somewhat unreliable and unrobust perfboard configuration that likely will not hold up to student use in the future and can be slightly unreliable. Issues in the breakout board can cause noisy extra counts or kill signal and is detrimental to the overall experiment.

An additional area that can be investigated further with the ACCM is the labview software provided with the hardware. I have not been able to dig into the software too much, and have not found out a way to run tests with varied coincidence windows or with some of the other bells and whistles provided with the software. Perhaps the fine-tuning of certain settings could resolve the current issues with α measurements.

6 Extensions

Aside from the possible fixes to the ACCM setup, another major extension to the single photon experiment would be to try and utilize the full capabilities of the Altera board and implement some of the additional labs on Mark Beck's page. This could lead to the development of a full lab course that covers much more experiments in quantum mechanics and has a greater applied bend. The lab setups are all well-documented on Mark's website and should be fairly straightforward to set up, though they will require some additional equipment.

As for the random number generation component of this project, it would be Interesting for someone to take the random number generation capabilities of the current set-up and extend it to a remote web-app that enables users to remotely run experiments and generate passwords. This would likely involve the addition of some servos and controllers that allow a user to remotely toggle on and off the experiment, as well as taking the outputs of the labview program and directly sending them into automated processing that directly turns the string of random numbers into a password that is then streamed to the user.

Another set of fun extensions to the random number generation is to investigate how various attack schemes can alter the experiment. For instance, if the room were to be heated by an attacker, causing upwards laser drift and an increase in the Poissonian mean, then the output "random" bits would be biased towards a certain direction. One could explore various methods of attacking this experiment and also various methods of prevention, such as periodic re-calibration and validation of the set-up.