Single Photon Experiments 2023

Tai Xiang

May 2, 2023

Contents

0.1 Summary

0.2 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.

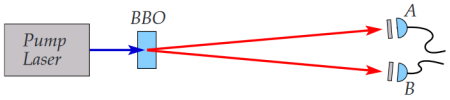


Figure 1: Basic set-up for the SPCM experiment.

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.

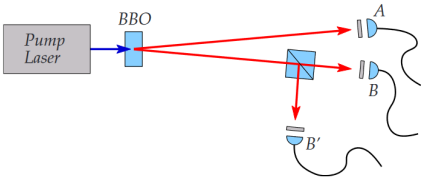


Figure 2: Beam-splitter set-up for the SPCM experiment.

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

0.3 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.

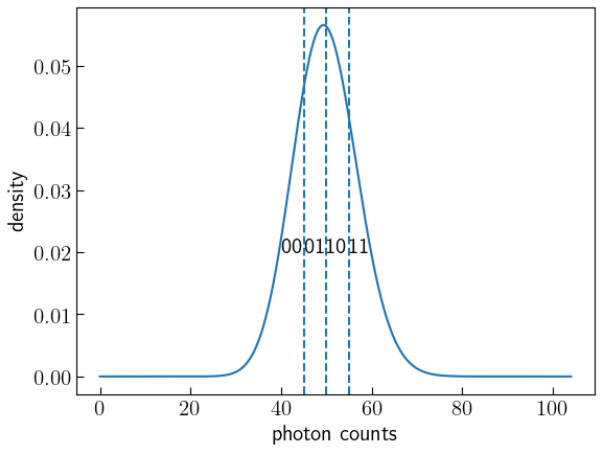


Figure 3: Poissonian distribution segmented into regions of equal area. A bit combination is then assigned to each region.

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.

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 0.3, 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.

0.4 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.

0.4.1 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.

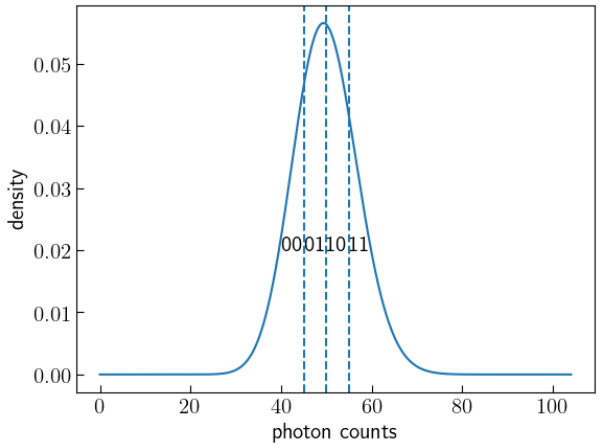
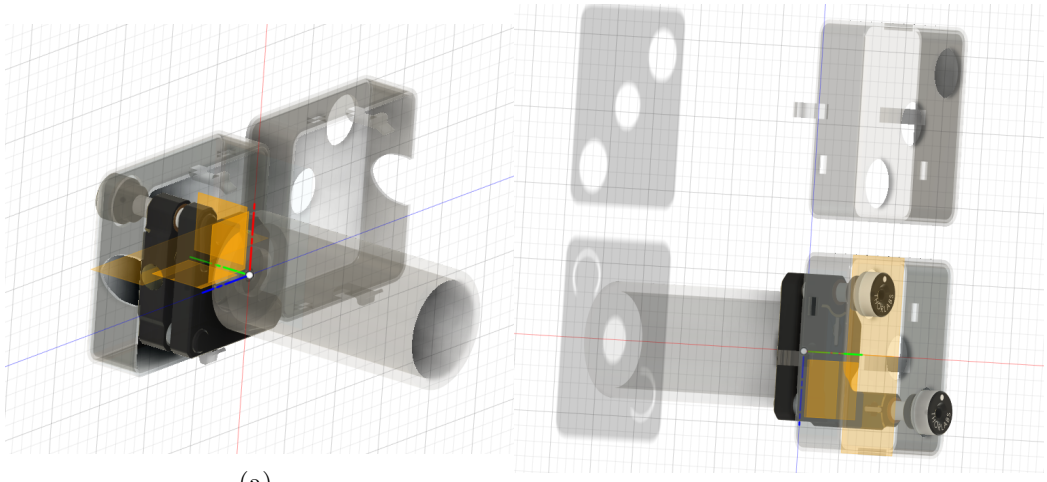


Figure 4: Poissonian distribution segmented into regions of equal area. A bit combination is then assigned to each region.

0.4.2 Enclosures

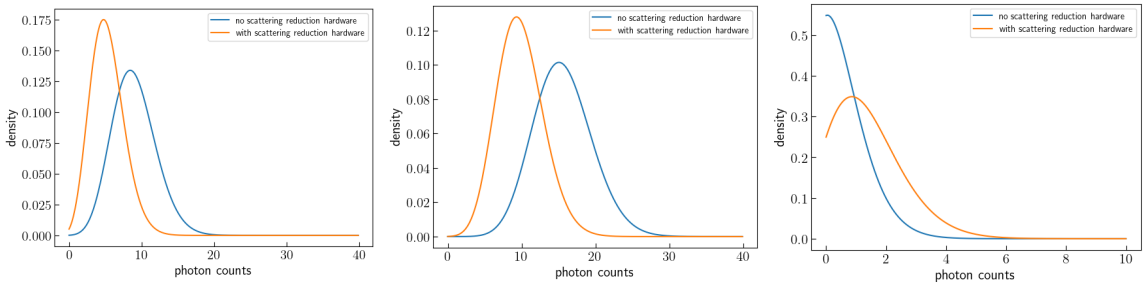
Another scattering reduction optimization I have made is to create a 3-D printable enclosure for the detectors that reduces the chance of entry for noisey scattered photons. The two-piece enclosure fits and snaps over the detectors and adds a cylindrical nose to the front. The long cylindrical nose reduces 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.

The cylindrical nose-tube is also designed to fit a filter, such as the FR RG780. When the enclosure was first designed, the ___ 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 6.



(a) (b) (c)

Figure 6: Figure 6a shows the difference between the Poissonian distribution of photon counts per integration time-step in detector A with and without scattering reduction hardware. Figure 6b shows the difference between the Poissonian distribution of photon counts per integration time-step in detector B with and without scattering reduction hardware. Figure 6c 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).

0.4.3 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 7.

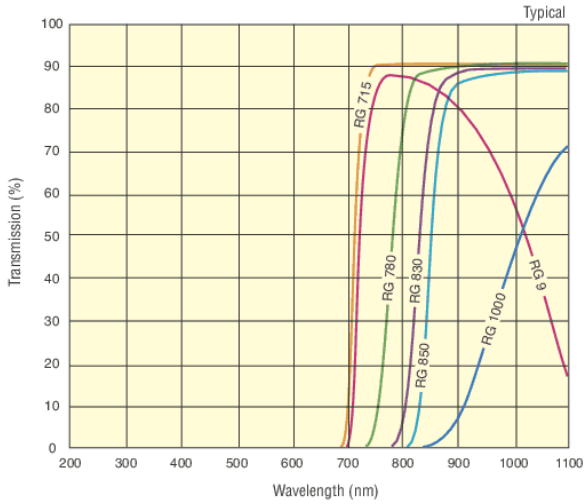
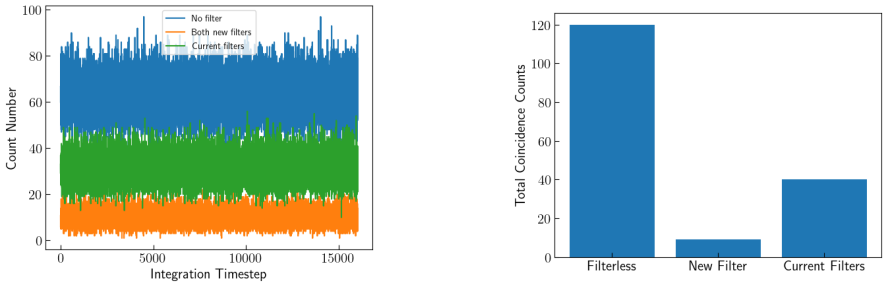


Figure 7: Filter profile of RG780 shown in green.

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.

For the laser, when analyzing initial counts over time, we observe the results shown in figure 8.



(a) (b)

Figure 8: Figure 8a 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 8b 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. A useful metric we can examine is

0.5 Defects in Current CCM and the FPGA CCM