Investigating the Viability of Using Avalanche Photodiodes for State Detection of BaF in the NL-eEDM Experiment

Tesse Tiemens
S3652327

July 2022

First supervisor: Lorenz Willmann
Second supervisor: Mick Mulder

Abstract

Electron electric dipole moment (EDM) measurements provide for sensitive tests of physics beyond the Standard Model, and the NL-eEDM collaboration aims to be such a measurement. This experiment, which uses Barium Monofluoride molecules to measure the EDM, requires the detection of small amounts of infrared photons as its main signal for the measurement. In this thesis we propose a new method of detecting these photons using avalanche photodiodes, which has the potential to overcome some of the limits of the current method using photomultiplier tubes. The performance of one such diode was tested in terms of sensitivity, voltage response, and noise. It was found that the current setup did not reach the shot noise limit, and so would not be an improvement to the experiment. However, improvements to the measurement electronics are possible and a few of these were explored and recommended for further investigations.

university of groningen

Contents

1 Introduction **2**
1.1 The NL-eEDM experiment . . . 2

2 Photon detection **3**
2.1 Photo-multiplier tubes (PMTs) . . . 3
2.2 Photodiodes . . . 3
2.3 Avalanche Photodiodes . . . 4
2.4 In conclusion . . . 5

3 The design **6**
3.1 Goals . . . 6
3.2 Signal specifics . . . 6
3.3 APD selection . . . 6
3.4 Electronics . . . 6

4 Noise characteristics **7**
4.1 Noise in the APD . . . 7
4.1.1 Dark current shot noise . . . 7
4.1.2 Signal shot noise . . . 8
4.2 Noise in the amplification circuit . . . 9
4.2.1 Thermal noise . . . 9
4.2.2 Op-amp i_n and e_nC . . . 10
4.2.3 Other sources of noise . . . 11
4.3 In Conclusion . . . 11

5 Testing **11**
5.1 Optics . . . 11
5.2 Calibration . . . 12
5.3 Data collection and processing . . . 13
5.4 Error analysis . . . 14

6 Results and Discussion **14**
6.1 Signal intensity . . . 14
6.2 Reverse voltage . . . 17

7 Outlook **19**
7.1 Tests on the current setup . . . 19
7.2 Changes to the electronics . . . 19

8 Conclusion **20**

A Processing code **21**

1 Introduction

This thesis is connected to the NL-eEDM collaboration, which aims to test the electric dipole moment (EDM) of the electron. As the maximum allowed value of this dipole moment is orders of magnitude smaller in the Standard Model than in several of its extensions, and a nonzero value of the EDM would violate the parity (P) and time-reversal (T) symmetries, measurements of this EDM make for good tests of new physics [1], and even dark matter candidates [2]. A number of modern experiments make use of an amplification of this effect created within atoms, as shown by P.G.H. Sandars in [3], and polar molecules [4]. The EDM measurement can then be performed by making use of the structure of the Hamiltonian:

$$\mathcal{H} = (\vec{\mu} \cdot \vec{B} - \vec{d_e} \cdot \vec{E}_{\mathit{eff}})I \tag{1}$$

Where $\vec{\mu}$ is the magnetic dipole moment, $\vec{B}$ the magnetic field, $\vec{d_e}$ the electric dipole moment, I the spin of the system, and $\vec{E}_{\mathit{eff}}$, the effective electric field felt by the electron. This effective electric field is not the same as the applied electric field, but is instead amplified by the structure of the molecules by a factor in the order of $\sim 10^2$, the specific value depending on the molecule used. This amplificaiton allows for more sensitive measurements of the EDM at lower field strengths, and is one of the main benefits of using polar molecules. As we cannot know the magnetic field well enough, we measure the energy eigenstates of this Hamiltonian with both parallel and anti-parallel E and B fields, so that the $\vec{\mu} \cdot \vec{B}$ terms cancel out and we are left with the term including the electron EDM, causing an energy difference $\delta = 2\vec{d_e} \cdot \vec{E}_{\mathit{eff}}$.

The current lower bound on the electron EDM was set to $1.1 \cdot 10^{-29} e{\cdot}\text{cm}$ by [5] using ThO, but more relevant are the $1.0 \cdot 10^{-27} e{\cdot}\text{cm}$ set using YbF in [6], and the $1.3 \cdot 10^{-28} e{\cdot}\text{cm}$ using HfF$^+$ in [7], as the molecules used are more similar to the BaF used in the NL-eEDM experiment.

1.1 The NL-eEDM experiment

The main contribution NL-eEDM, located at the University of Groningen, brings to the field is the usage of barium monofluoride (BaF), and a great amount of experience using a traveling wave Stark decelerator to slow down stontium monofluoride (SrF) [8], a molecule very similar to BaF. Aiming to reach an upper limit of $5 \cdot 10^{-30} e{\cdot}\text{cm}$, the experimental setup uses the slow pulses of BaF molecules from the Stark decelerator and, as described in [9], employs the ground state of these molecules, which splits into the $F = 1$ and $F = 0$ hyperfine levels. Using an all optical method, the molecules are transferred from the $F = 0$ state to the $F = 1$ state, ending up in the $|F, m_F\rangle = \frac{1}{\sqrt{2}}(|1, +1\rangle - |1, -1\rangle)$ superposition between the $m_f = 1$ and $m_f = -1$ magnetic sub-levels. This superposition will evolve according to the Hamiltonian in Equation 1, and the different energy levels between the m_F states creates a phase-difference ϕ [6], causing an oscillation between the initial 'bright' state, and the $|F, m_F\rangle = \frac{1}{\sqrt{2}}(|1, +1\rangle + |1, -1\rangle)$ 'dark' state, which the optical setup does not interact with. The angle ϕ can then be measured by projecting the superposition back to the energy eigenstates $F = 0$ and $F = 1$, so that the population distribution between the states now encodes information about ϕ and thus a possible EDM.

Since the energy of the different hyperfine levels is different, the populations of molecules in either F state can be measured through laser-induced fluorescence (LIF) to the $A^2\Pi_{1/2}$ excited state using a 860 nm near infra-red (NIR) laser [10]. This will, depending on the exact laser wavelength, make the molecules in either hyperfine state emit NIR photons when transitioning back to the ground state, which we can detect. As this detection of NIR photons is ultimately the main way of probing the EDM in the experiment, efficient detection of these photons is of utmost importance. In fact, following [9], the statistic uncertainty σ_d in the measurement of the EDM is given by:

$$\sigma_d = \frac{\hbar}{e} \frac{1}{2|P|E_{\mathit{eff}}\tau\sqrt{\dot{N}T}} \tag{2}$$

Where e is the electron charge, $|P|$ the polarization of the molecule, E_{eff} the effective electric field acting on the electrons, τ the interaction time, increased by slowing the molecules, T the total measurement time, and $\dot{N}$ the molecule detection rate. From this importance on molecule detection, our research question arises: **Is there a better method of detecting the laser-induced fluorescence photons in the NL-eEDM experiment?**

2 Photon detection

In order to answer this question, we need to start by discussing the various options available for photon detection: photo-multiplier tubes, photodiodes, and avalanche photodiodes.

2.1 Photo-multiplier tubes (PMTs)

As of the writing of this thesis, the detection of the LIF photons in the NL-eEDM setup is done using photomultiplier tubes (PMTs), which use the photo-electric effect to knock an electron out of a thin strip of material, which is then attracted to a high-voltage dynode, releasing more electrons as it hits it. This happens a number of times, resulting in a typical gain of $10^6 e/\gamma$ [11]. This level of gain makes it possible to have measurement setups that are limited primarily by noise generated from background counts and fluctuations in the Poisson-like processes taking place within the PMT, rather than electrical noise in the readout circuitry, allowing for very accurate low-light measurements. These devices are not without their limitations however, some of which have implications for the NL-eEDM experiment.

Firstly, as PMTs rely on the photoelectric effect in a photo cathode to 'knock' electrons free and into the multiplication stages, there is a lower limit on the energy an incoming photon can have to trigger a pulse, and for most models 860nm photons lie below this threshold. There are PMTs that are sensitive to these wavelengths, but with a quantum efficiency (QE), the percentage of incoming photons that actually get detected, of $\sim 10\%$ instead of the more typical $\sim 25-30\%$, due to the need for special photo-cathodes. As the statistical uncertainty σ_d of the experiment is inversely proportional to the square root of the amount of molecules detected (see equation 2), such a low QE is undesirable.

Secondly, as the luminescence from the molecules is non-directional and diffuse, it is beneficial to cover as large a solid angle as possible with photo-detectors, ideally by placing multiple detectors side-by-side. The physical size of PMTs and their attached amplification equipment makes this difficult however, and so a significant portion of the luminescence is left unused.

Lastly, PMTs, due to the nature of the multiplication stages, are limited in the photon rates that can accurately be measured. Specifically, the typical 'dead time', the time needed between two counted photons, is about 10ns. This poses an issue, as the final goal of the NL-eEDM experiment is to have more molecules pass by the sensor than would be countable by a PMT, and so, once again, part of the signal goes to waste, increasing statistical and systematic uncertainty. During a recent test run of the experiment, one of the PMTs used already reached this limit at times, and so did not provide an accurate measurement of the number of photons.

2.2 Photodiodes

Photodiodes are semiconductor devices, in which the photoelectric effect is used to excite an electron from the valence band into the conduction band. Unlike in the PMT, after this excitation the electrons do not need to leave the material, allowing for lower energy photons to be detected, and thus a higher quantum efficiency, especially for wavelengths in the near infrared. Photodiodes can be run in two modes: photovoltaic and photoconductive mode.

In photovoltaic mode, no external voltage is applied to the diode, and the diode creates its own current and voltage, akin to a solar cell. In this mode however, the internal capacitance of the diode is quite high (limiting the bandwidth of the system), and, more importantly, if the diode is hooked up to some kind of load resistance, the photo-current through the resistor can create a voltage close to the forward voltage of the diode, stopping any more current from being generated.

In photoconductive mode a small reverse bias voltage is added, overcoming the issue with the voltage generated by any load resistors, and lowering the internal capacitance by increasing the voltage drop over the semiconductor junction. On the other hand, this does add an extra source of noise from a small leakage current, or "dark current", created by the bias voltage applied [12].

2.3 Avalanche Photodiodes

Avalanche Photodiodes, first invented in 1952 by Jun-ichi Nishizawa [13], are a kind of photodiode that uses large (typically 100-300V) reverse voltages in order to create an internal current gain. Whereas in traditional photodiodes each photon excites a single electron-hole pair through the photoelectric effect, an avalanche photodiode functions more akin to a proportional chamber, using a Townsend avalanche to excite additional electrons to reach a typical internal gain of $\sim 100e/\gamma$. This internal gain allows APDs to detect lower light levels than 'traditional' photodiodes, without sacrificing bandwidth in order to suppress thermal noise [14]. Similarly to regular photodiodes, the quantum efficiency of APDs is

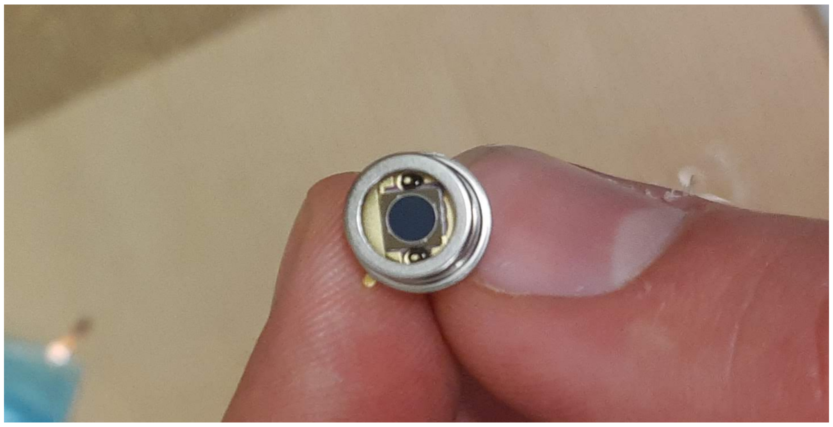


Figure 1: An avalanche photodiode. The inner black circle is the photosensitive area.

significantly better than that of PMTs, reaching as high as 70%, even at 860nm. [14]. Next to that, the high bias voltage allows for a very wide dynamic range, which is useful for the experiment, as the signal strength can vary strongly, depending on the performance of other parts of the experiment.

One more benefit, which is more specific to the NL-eEDM experiment, is that the small size of photodiode packages (see Fig 1) allows them to be mounted inside the vacuum chamber, much closer to the source. This would remove the need for a lot of the optics required to get the signal to the PMTs, which needed to be outside the vacuum, as well as potentially allow for a larger solid angle to be sampled.

However, the benefits that come with using APDs do not come without their own challenges. Firstly, the gain of an APD is still much smaller than that of a PMT, requiring significantly more attention to be paid to the readout electronics. Secondly, the higher reverse bias voltage creates a significant amount of dark current, some of which is also amplified by the avalanche mechanism, creating more noise (see Section 4). Lastly, as can be seen in Figure 2, the gain of an APD has a strong dependence on both voltage and temperature, potentially requiring active temperature stabilization or compensation to avoid adding an extra source of uncertainty to the measurement.

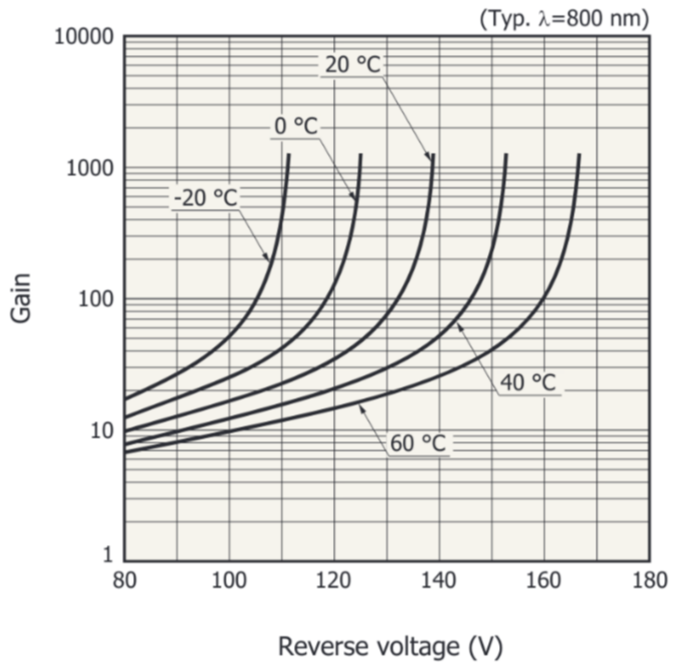


Figure 2: Gain curves of the avalanche photodiode used, taken from [15]. Note the strong temperature dependence on both temperature and voltage. Next to that, these curves are given for a large range of APD types, and so together with manufacturing variances, might not fully reflect the specific photodiode in use, creating the need for testing.

2.4 In conclusion

Type	Gain	Size	Dynamic range	QE at > 800nm	Dark current	Temp. dependence
PMT	$\sim 10^6$	Large	Low	$10 - 20\%$	Low	Med.
PD	1	Small	High	$60 - 80\%$	Low	Low
APD	~ 100	Small	High	$60 - 80\%$	High	High

Table 1: An overview of the detector types discussed in this section. A number of columns are left intentionally vague as the specific values vary rather strongly from setup to setup, or are simply hard to compare.

An overview of the matter discussed in this section can be found in Table 1. As photodiodes do not have a lot of issues that PMTs create, but regular photodiodes lack the gain to be able to detect the low light levels in the experiment, this investigation will mainly focus on avalanche photodiodes and their viability of using them in the experiment.

3 The design

3.1 Goals

The goal of the detector is, as mentioned above, to measure the amount of molecules in a given state. This means that we are interested in the *integral* of the incoming signal, rather than the photon rate at any given time, and thus the noise in the signal is the standard of these integrals for a given amount of incoming photons. As the PMTs allow us to reach the shot noise limit on this variance, the point where the limiting factor is no longer the internal background noise of the system but rather the noise inherent in any quantized signal, the goal for the APD-based detector is also to reach this limit.

3.2 Signal specifics

With the current PMT setup, the amount of photons detected is approximately $300 - 500$ in pulses of approximately $60\mu s$. With the APDs increased QE, but decreased detection size (3mm vs 5mm diameter, see next subsection), we expect the photon counts detected by the APD to be approximately 3 times that, so $\sim 1.5 * 10^3$, but the end goal of the experiment is to have signals that are 10-100 times larger than that.

3.3 APD selection

The APD chosen for this project was the Hamamatsu Si APD S2384, as it grants a good balance between gain, size, and terminal capacitance, while being maximally sensitive in the 800nm band [15], having a quantum efficiency between 70 and 80% at the desired wavelengths. The exact unit used was rated for a gain M of $60e/\gamma$ at a reverse voltage of 147V, with a breakdown voltage of around 160V. However, as can be seen from Figure 2, it has rather strong temperature dependence. Hamamatsu does also manufacture APDs with a lower temperature dependence, but those require significantly higher reverse bias voltage, which would in turn create higher electric fields, which could harm the experiment, and are generally less easy to handle.

3.4 Electronics

The APD was placed in a lenstube and attached to two coaxial connectors: a LEMO connector for bias voltage, and an SMA connector for the outgoing signal. The bias voltage was provided by a Gossen Konstanter 14K160R0.3 power supply, monitored by a Fluke model 175 multimeter. Signal amplification was handled by a Thorlabs AMP120 trans-impedance amplifier with a gain of 10^5V/A and a Standford Research SR560 pre-amplifier as a second stage amplifier and filter. The output was connected to a Rigol DS1074 Oscilloscope.

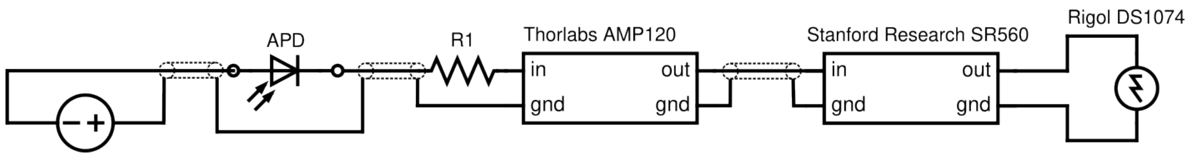


Figure 3: Overview of the measurement electronics. From left to right: the power supply for the reverse bias voltage, hooked up to the APD via a LEMO connector, which is then hooked up to the Thorlabs AMP120 100kV/A trans-impedance amplifier through a 100kΩ resistor $R1$ to suppress noise. The output of the AMP120 is connected to a Stanford Research SR560 pre-amplifier with a built-in band-pass filter, the output of which is hooked up to a Rigol DS1074 oscilloscope to read out the signal.

A resistor $R1$ of 100kΩ was added in series with the AMP120 to suppress thermal noise in the system (see section 4). An overview of the setup can be found in Figure 3. For all measurements, the

SR560 was set to a bandpass mode between 10Hz and 30kHz with a 6dB/octave roll-off on either side, while the gain was varied to match the incoming signal strength. The lowpass set at 30kHz functions to filter out high-frequency noise, while allowing for rise times of under 15μ**s**, which is fast enough for our 60μs pulse.

4 Noise characteristics

As the goal of the design is to reach a certain low level of noise, the following section will discuss all the likely sources of noise, and estimate their contributions. However, first we must discuss what noise is, and how we will describe it.

In short, noise is a random fluctuation in an electrical signal, be it a current or a voltage. It is often described as a root-mean-square (rms) voltage or current, which corresponds to the variance in a constant signal. Besides this total rms level, as the total intensity of noise seen is dependent on the total bandwidth of the system, noise values and formulae are also often given as a noise *density*, which has units of V$\sqrt{\text{Hz}}$. This means that the total noise increases with the square root of the bandwidth, unless it is explicitly dependent on frequency, in which case the noise density should be integrated over the entire bandwidth to find the total noise.

For our our signal however, we add a third figure to this, as we are interested in the noise in our signal, the *integral* noise. As our signal is a count of electrons, our integral noise, $N_{n\text{el}}$, will also be a number of electrons. In order to convert to this, we can treat it similarly to averaging a DC signal, and change the bandwidth to $B = 1/2T$, with T the time of integration. For the rms sources that go with $\sqrt{B}$, this comes down, after integration and conversion to a count of electrons using the electron charge q, to an integral noise as given in Equation 3.

$$N_{n\text{el}} = \frac{1}{q}\sqrt{\frac{T}{60kHz}}I_{n\text{rms}} \tag{3}$$

In order to keep with convention and remain comparable with literature, this section will calculate mainly rms values, but at the end there will be an overview where these will be converted to integral noise values, as well as expected signal-to-noise ratio (SNR) values.

4.1 Noise in the APD

4.1.1 Dark current shot noise

According to Hamamatsu in [14], APDs have two internal sources of noise. The first source of noise is tied to the dark current through the APD; the leakage current that flows without a signal present. This dark current consists of two parts: the non-amplified surface current I_{ds}, and the body current, amplified by the avalanche processes in the APD, I_{dg}. These can not be measured directly, but only inferred from a single dark current measurement $I_d = I_{ds} + MI_{dg}$, and are not given separately by the Hamamatsu data-sheets. As can be seen in Figure 4, for most of the reverse voltage range I_{ds} dominates, but as the reverse voltage increases, the gain increases (nonlinearly), and so I_{dg} starts to dominate.

As these are of course quantized, they carry with them shot noise. However, due to to the construction of the APD, the shot noise is also amplified by the avalanche process, and so needs to be multiplied by the gain M, plus an extra noise figure x, added to the exponent of the gain[1], giving us an rms noise current as given in Equation 4.

[1]This is a simplification, ideally we would multiply by an excess noise factor F, dependent on M, but this works relatively well, and since we don't have exact values for the dark currents the uncertainty in those values dominates the small error created by this approximation,

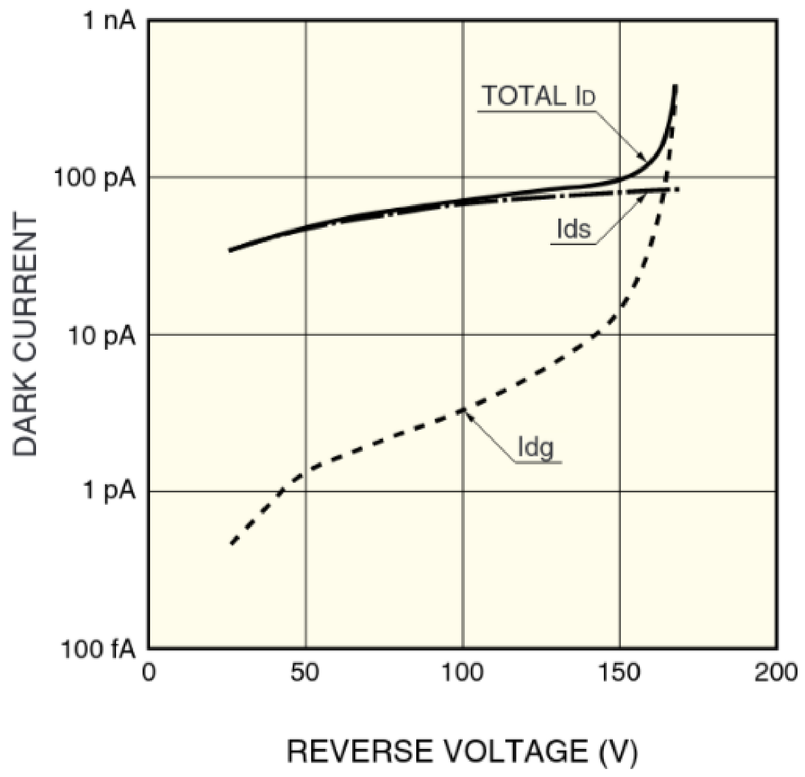


Figure 4: Si APD dark current behaviour. Not representative of the actual APD used. Taken from [14].

$$I_{nd} = \sqrt{2q\,B\,(I_{dg}\,M^{2+x} + I_{ds})} \tag{4}$$

Where q is the electron charge and M is the APD gain. As we do not know the exact values for I_{ds} and I_{dg}, it is hard to accurately predict a figure for this noise, but by fitting to the dark current curves given by the Hamamatsu data-sheet [15], as well as the given excess noise figure $x = 0.3$ we find that in the area where I_{ds} dominates we expect the rms noise current to be well below $10^{-11}A$, while at higher gains it can climb to above $10^{-10}A$.

4.1.2 Signal shot noise

Our second source of noise, the signal shot noise, similarly needs to be multiplied by M^{1+x}. This results in an rms current noise of:

$$I_{ns} = \sqrt{2q\,B\,I_L\,M^{2+x}} \tag{5}$$

Where I_L is the *unamplified* photocurrent, equal to the rate of detected photons multiplied by the charge of the electron; for a pulse of 1500 photons in $180\mu s$ it is $1.3 \cdot 10^{-12}A^2$. Filling in our other figures ($B = 30$kHz, $M = 60$, $x = 0.3$), we get $I_{ns} = 1.25 \cdot 10^{-11}$A. Note that the excess noise factor makes it such that with increasing gain the shot noise grows with $M^{1.15}$, where the signal grows with M, causing the signal-to-noise ratio (SNR) to have a distinct maximum at a certain gain, as can be seen in figure 5, which also gives an overview of all the different noise sources and their expected rms current values.

[2]Note that this is a simplification, as our pulses are gaussian and so will not have a constant current.

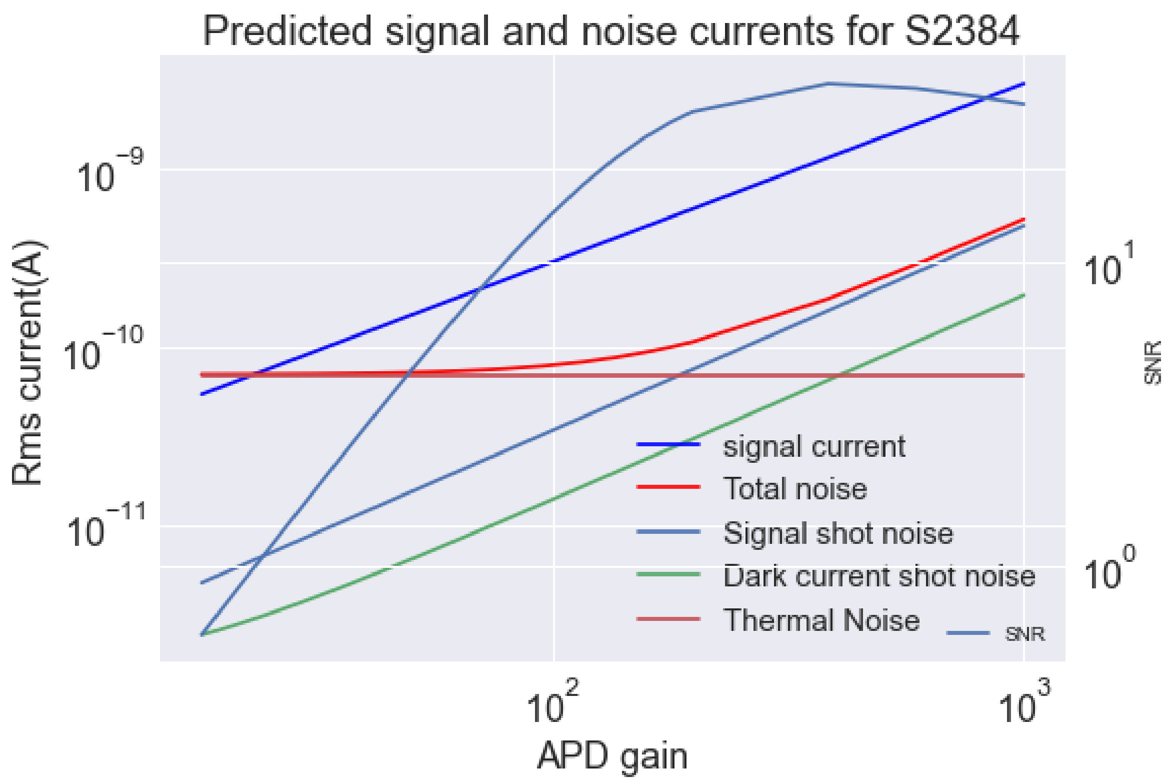


Figure 5: Calculated signal and noise figures for various APD gains for pulses of 5000 photons. Note that 10^3 is an unrealistic gain for this APD; in the experiment the highest used gain was around 175. Any higher would cause a significant increase in dark current and possibly degrade the device.

4.2 Noise in the amplification circuit

4.2.1 Thermal noise

Any circuit that uses resistors in the signal path will suffer from thermal noise, also known as Johnson noise. This means that ideally we would not have any resistors near the APD side of the amplifier, but the Thorlabs AMP120 has a 100Ω load resistor before its first stage amplifier, likely for impedance matching or circuit protection purposes. With a voltage-signal, 100Ω would not create too much of an issue. However, since our signal is in the form of a current, the thermal noise goes with the inverse of the load resistance R_L of the circuit:

$$I_{nt} = \sqrt{\frac{4k_bTB}{R_L}} \tag{6}$$

where k_b is Boltzman's contstant, T is the temperature, and B is the bandwidth of the system. This means a small resistance like 100Ω creates quite a large thermal noise, hence why we added the 100kΩ resistor, $R1$ to the setup.

Filling in this resistance and our 30kHz bandwidth, we find that the rms noise current (at $293K$) is $6.9 \cdot 10^{-11}A$, assuming a 1% error resistor. Further increasing load resistance would lower this noise, but would also decrease the bandwith of the system, as the terminal capacitance of the APD acts together with the resistor to form a low-pass filter with a cutoff frequency given by:

$$f_c = \frac{1}{2\pi C_t R_L} \tag{7}$$

Where C_t is the capacitance of the APD. In our case the S2384 is rated at $40pF$, giving us a cutoff frequency of approximately $40kHz$ with the $100k\Omega$ resistor. This is enough to not interfere with our signal, even allowing for extra capacitance added by wiring, which would be added to C_t.

There are ways of designing the measurement electronics to have larger resistances without having this bandwidth issue, some of which are discussed in greater detail in Section 7.2.

4.2.2 Op-amp i_n and e_nC

After the 100Ω resistor discussed in the previous subsection, the AMP120 uses in its first amplification stage an OPA301 op-amp, which has input current and voltage noise density values of $i_n = 1.5fA/\sqrt{Hz}$ and $e_n = 3nV/\sqrt{Hz}$ respectively [16]. For the current noise, with our bandwidth, we find an rms current noise of $I_n = 0.2\text{pA}$, which is negligible against all other sources of noise.

The voltage noise couples with the terminal capacitance of the APD and the capacitance of the wires to create what Horowitz and Hill call e_nC noise [12]. They describe it as translating into current noise as $i_{ne} = 2\pi e_n f C_{in}$, where f is the frequency and C_{in} is the input capacitance. This approach however is a simplification that only works at higher frequencies and is incompatible with the high load resistance we have added. Instead, we must use the approach given in [17], where we model the amplifier circuit as shown in Figure 6.

From this we can find the transfer functions, and from there calculate the noise current i_{ne} generated by e_n to be:

$$i_{ne} = e_n\left(\frac{j\,2\pi f\,C_{in}}{1 + j\,2\pi f\,R_L C_{in}} + \frac{1}{Z_f}\right) \tag{8}$$

Assuming Z_f, the feedback impedance, to be a $100k\Omega$ resistor in parallel with a 10pF capacitor, and C_{in} to be 100pF to account for wire capacitance, we use scipy to integrate this with respect to $\sqrt{f}$ to find an rms noise current of $6.9 \cdot 10^{-12}\text{A}$.[3] Being about an order of magnitude lower than the thermal noise, this too is negligible.

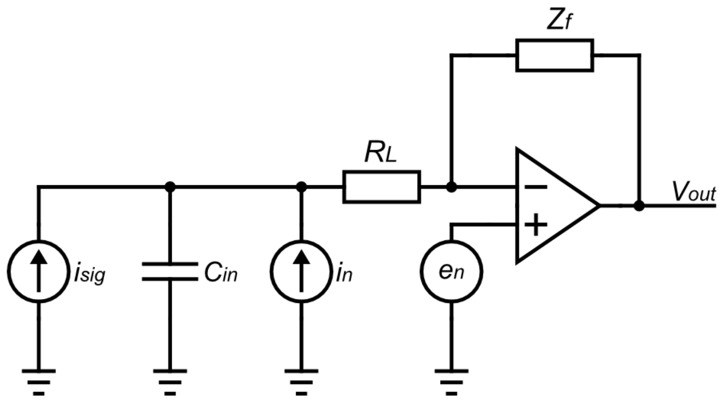


Figure 6: Noise model of the amplifier, used to find the contributions of the op-amp i_n and e_nC. The approach used was taken from [17], but adapted to our specific situation.

[3]we integrate w.r.t. $\sqrt{f}$ as e_n is expressed in $\text{V}/\sqrt{\text{Hz}}$

4.2.3 Other sources of noise

Other sources of noise include the $100k\Omega$ feedback resistor, noise from the later stages of the AMP120 and its power supply, noise generated by the SR560, pick up noise in the wires, and input noise on the oscilloscope, all of which are unknown.

4.3 In Conclusion

An overview of the noise sources discussed in this section can be found in Table 2. The integral values in this table were calculated using Equation 3, except for the amplifier e_nC, as it does not go with $\sqrt{B}$, and so had to be re-integrated. From this table, it can be seen that the resistor thermal noise is strongly dominating, with the signal shot noise in a distant second place. However, as was discussed in section 4.1.1, at higher gain levels we would expect dark current shot noise to dominate.

Noise source	rms (A)	Integral (electron count)	SNR
Dark current shot noise	$< 10^{-11}$	$< 3 \cdot 10^3$	> 30
Signal shot noise	$1.25 \cdot 10^{-11}$	$4.3 \cdot 10^3$	21
Thermal	$6.9 \cdot 10^{-11}$	$2.4 \cdot 10^4$	3.7
Amplifier e_nC	$6.9 \cdot 10^{-12}$	$2.5 \cdot 10^3$	37
Amplifier i_n	$0.2 \cdot 10^{-12}$	68	$1.3 \cdot 10^3$
Total	$7.0 \cdot 10^{-11}$	$2.4 \cdot 10^4$	3.7

Table 2: Overview of the noise figures for a pulse of 1500 detected (so ignoring the quantum efficiency of the APD) photons in 180μs, and a gain of 60, yielding $9 \cdot 10^4$ electrons. The signal-to-noise ratio (SNR) here is defined as SNR $= \frac{\text{integral signal}}{\text{integral noise}}$.

5 Testing

In order to figure out if the APD and the connected circuit actually behave like we expect, they should be tested in a real-world setting. Specifically, we are interested in the noise behaviour at different signal levels, as well as the gain and noise behaviour for different reverse voltage levels on the APD. The following section will outline how this was done, and how the data was processed.

5.1 Optics

The APD was tested in a controlled setting in a setup as shown in Figure 7, using an 860nm laser light and a chopper wheel to create pulses of $80\mu s$ FWHM at a rate of 550Hz. The intensity of the light reaching the APD could be controlled using the various neutral density filters (NDFs) set in the NDF wheel (consisting of 2 wheel in series, so that a combination of two NDFs could be made), together with additional filters mounted in the lens tubes. From these filters, the optical attenuation available ranged between $10^{2.85}$ and $10^{8.7}$. Fluctuations in the laser were accounted for by directing part of the beam to hit a regular silicon photodiode (Thorlabs SM05PD28) running in photovoltaic mode, which was read out over a 1kΩ resistor to ground.

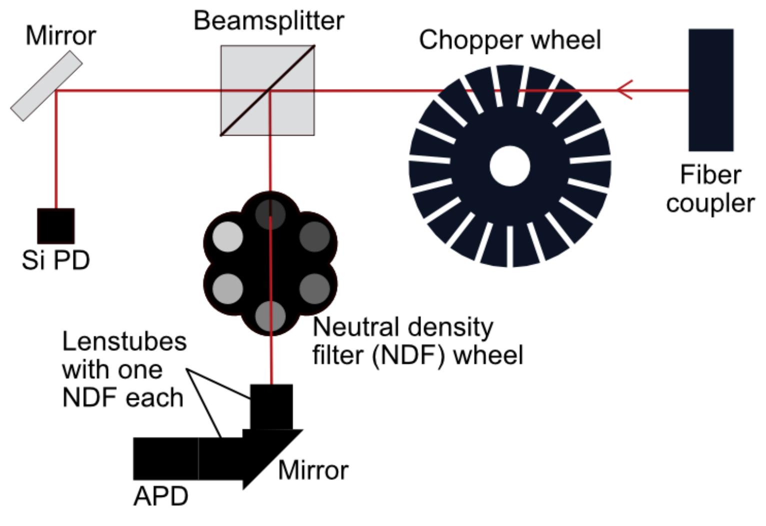

Mirror
Beamsplitter
Chopper wheel
Fiber coupler
Si PD
Lenstubes with one NDF each
Neutral density filter (NDF) wheel
Mirror
APD

Figure 7: A schematic representation of the optics setup used for testing the APD. The laser beam coming from the fiber coupler, was cut into pulses using a chopper wheel, after which it was split 2-way using a beamsplitter. One beam was sent to a regular photodiode via a mirror for alignment, while the other beam traveled through a number of neutral density filters as well as a mirror for alignment before reaching the avalanche photodiode.

5.2 Calibration

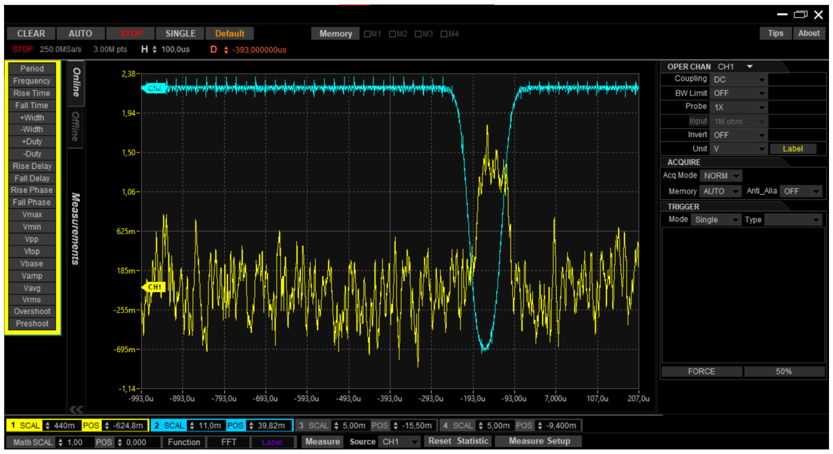
All the parts of the setup were re-calibrated with the help of a Coherent Lasermate-Q laser power meter. With the chopper wheel stationary in an open position to create a steady beam, the NDFs in the NDF wheels were measured to have attenuation strengths $10^{0.9}, 10^{1.4}, 10^{2}$, and $10^{2.5}$ for the first wheel, and $10^{0.5}$ for the only one used from the second wheel. The NDFs in the lenstubes were measured to both have an attenuation of $10^{2.85}$. All of these measurements had an uncertainty of $\pm 10^{0.05}$. The splitting ratio of the beamsplitter was found to be $54 \pm 3\%$ by measuring the signal on both sides of the splitter. The SiPD was measured to have a responsivity of 0.24 ± 0.01A/W by measuring the incoming laser power and the voltage over the resistor with the oscilloscope in DC coupling mode. An overview of all these values can also be found in Table 3.

Measurement	value
First wheel NDF strengths	$10^{0.9}, 10^{1.4}, 10^{2}$, and $10^{2.5}$
Second wheel NDF strength	$10^{0.5}$
Lens tube NDF strength	$10^{2.85}$
Beamsplitter splitting ratio	$54 \pm 3\%$
SiPD responsivity	0.24 ± 0.01A/W

Table 3: Calibration values for the various optical elements. All NDF attenuation strength values have an uncertainty of $\pm 10^{0.05}$.

5.3 Data collection and processing

Data was gathered by connecting the oscilloscope to a computer via USB, and using the Rigol software to save traces of the APD and SiPD signals. Either due to the software or the link with the oscilloscope, this was limited to 1200 data points per trace, yielding a temporal resolution of $1\mu s$. The data was saved as .csv files, and grouped in *sets*, each set aiming to measure the effect of changing a single parameter given other parameters (except amplifier gain) constant. These sets contained a number of *series*, each containing 20 traces, describing one particular configuration of the setup. All this data was then loaded into a set of custom python scripts for analysis. For an example of a signal trace, see Figure 8.



STOP 250.0MSa/s 3.00M pts H 100,0us D -393,000000us

OPER CHAN CH1
Coupling DC
BW Limit OFF
Probe 1X
Input 1M ohm
Invert OFF
Unit V
ACQUIRE
Acq Mode NORM
Memory AUTO
Anti_Alia OFF
TRIGGER
Mode Single

1 SCAL 440m POS -624,8m
2 SCAL 11,0m POS 39,82m
3 SCAL 5,00m POS -15,50m
4 SCAL 5,00m POS -9,400m
Math SCAL 1,00 POS 0,000
Source CH1

Figure 8: A trace as generated by the setup. The blue trace is from the regular photodiode, while the yellow one is from the APD. As this trace is from a test measurement, the noise in the blue trace is caused by a noisy laserdiode. During the actual measurements this was not present.

In order to find the integral of the traces, the peak of the SiPD signal was found using Scipy, telling us when in time the pulse arrived. This information was then used to get rid of any DC offsets in either signal by removing the area of the peak, averaging over the remaining data, and shifting the signal by said average. The signals were then integrated using trapezoid integration, all the signals for the series were averaged to find the signal value, and the standard deviation was taken to find the noise value, as the noise we are interested in is the variation in the integral of the signal. Finally, the signals were divided by the amplification used to create the trace, in order to be able to compare signals between amplification levels.

For these averages, variation in laser power was taken into account by comparing the integral of the SiPD signal for the trace with the average of all the SiPD traces for the set and scaling the APD signal integral by the according amount to get the adjusted electron count. The background rms noise was found by cutting out the peaks of the APD signals, and taking the root mean square of the remainder of the signal. This was then, where necessary, converted to an integral noise using Equation 3. Signal electron counts were found using Equation 9, in which q is the electron charge in Coulombs, A is the total amplification, and F_s is the scaling factor used to adjust for fluctuations in laser power. For the plots showing electron per photon counts, this scaling factor was left out.

$$N_{\text{el}} = \frac{1}{q\,A\,F_s} \int_{t_1}^{t_2} s(t)dt \tag{9}$$

5.4 Error analysis

On the photon count side, the sources of error taken into account were the uncertainty in the sensitivity of the regular photo-diode, the uncertainty in the load resistance of that photodiode, and the uncertainty in the attenuation factors of the neutral density filter, all together leading to a systematic uncertainty of 23%. For statistical errors, mainly the error in the offset removal was considered.

This too was a source of statistical error on the electron count side, together with the uncertainty in the photon count involved in removing the fluctuations in laser power. The error in the noise is given by the 95% confidence interval on the standard deviation created by the fact that we only took 20 traces per series. One notable source of error missing from this analysis is the error in amplification, which was assumed to be negligible.

6 Results and Discussion

6.1 Signal intensity

The first measurement set was done to determine the behaviour of the APD and measurement electronics at various signal intensities, by varying the combination of NDFs in the beam path towards the APD. The pulse-length used for integration - found by taking the width of the SiPD pulse at 2.5% of its total height - was approximately $180\mu s$, with variations around that number due to size differences in the slits of the chopper wheel. The results of this set can be found in figures 9, 10, and 11. As expected, the signal behaves linearly with the incoming photon counts, while the noise flattens out towards $2 \cdot 10^4$ electrons - corresponding to an rms current of approximately $6 \cdot 10^{-11}$A, with the background rms current sitting at $6.1 \cdot 10^{-11}$A. Reasons for this being higher than expected could be resistance in the wires, or stray capacitance of the connection leads lowering the effective bandwidth of the system. On the high signal side, the noise grows nearly linearly with the incoming signal, instead of following the signal shot noise as expected (plotted in red in Figure 9).

At first it was thought this was due to variations in the laser intensity, however, since the same rise showed up in the background rms noise, and the laser intensity variations were accounted for by the SiPD, a more likely explanation is that the source of this noise is between the output stage of the SR560 and the input of the oscilloscope, and grows linearly with the signal due to the fact that the amplification was decreased linearly with the signal, and the final measurements are divided by the amplification to get the results in the graphs. This theory is further supported by the fact that the inverse of the amplification indeed grows at a very similar rate to the rise in the noise, as shown in Figure 11. However, as can also bee seen from that figure, the noise in the signal departs from the thermal noise before the background noise does, and stays greater than it after this. A possible explanation for this could be that the noise source is in some way dependent on the signal passing through, which could be the case for a noise source in the output stage of the SR560.

Another thing worthy of noting is that while the gain is relatively stable throughout the intensity range, as can be seen in Figure 10, it is closer to 35 than the 60 it should be at the used voltage. One reason for this could be that the diode was damaged during shipping or soldering to the connectors, lowering the gain. Another could be that the actual quantum efficiency of the APD is lower than the 75% used to create this graph, as this was read off of the Hamamatsu datasheet [15], but might not fully apply to the diode used. Combined with the uncertainty in the amount of photons coming in, this could account for the discrepancy seen.

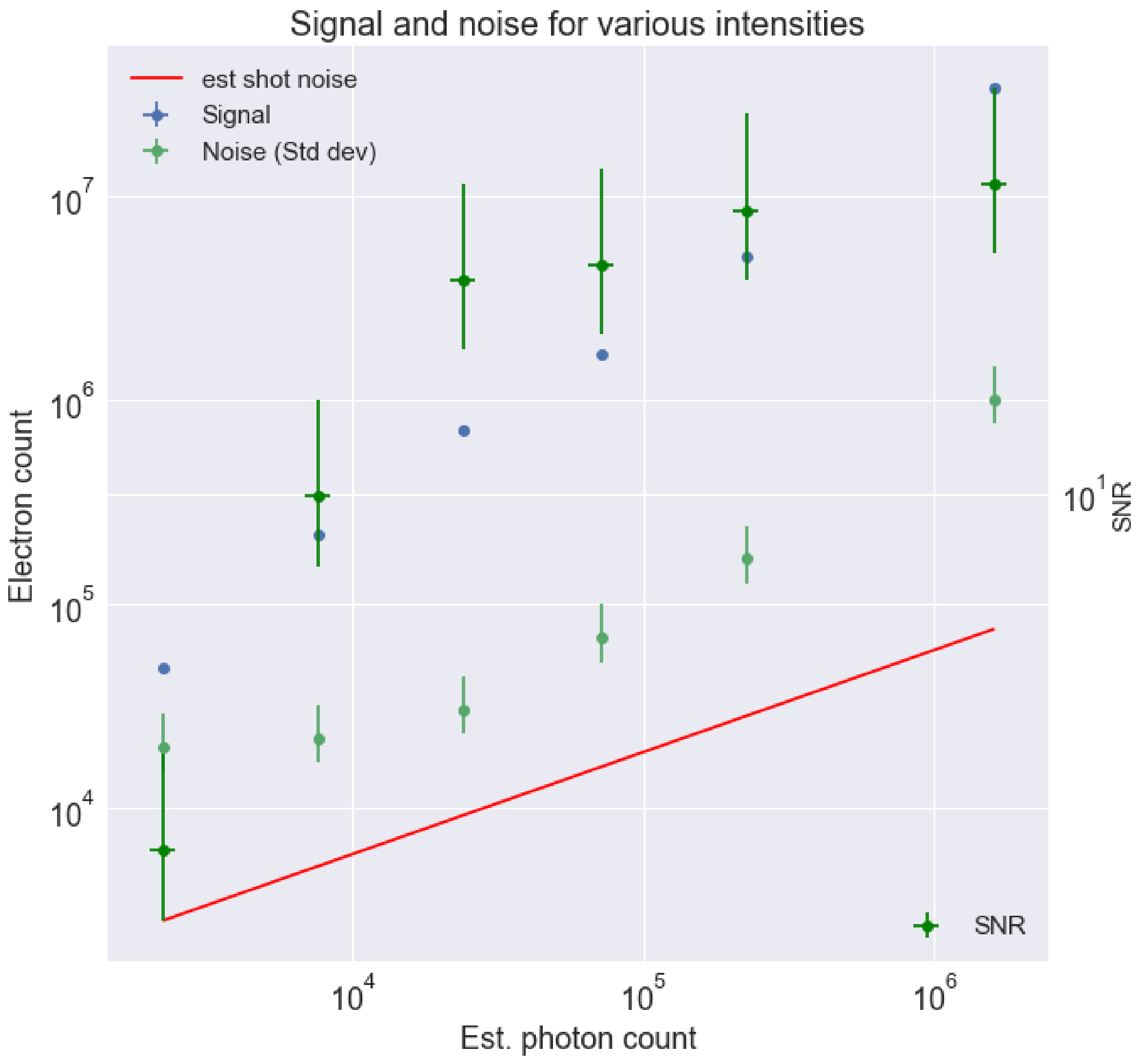


Figure 9: Signal and noise for various intensities. The shot noise prediction was calculated using Equation 5 with a gain of 35, while the electron counts were calculated using Equation 9. The photon intensities have a systematic uncertainty of 23%.

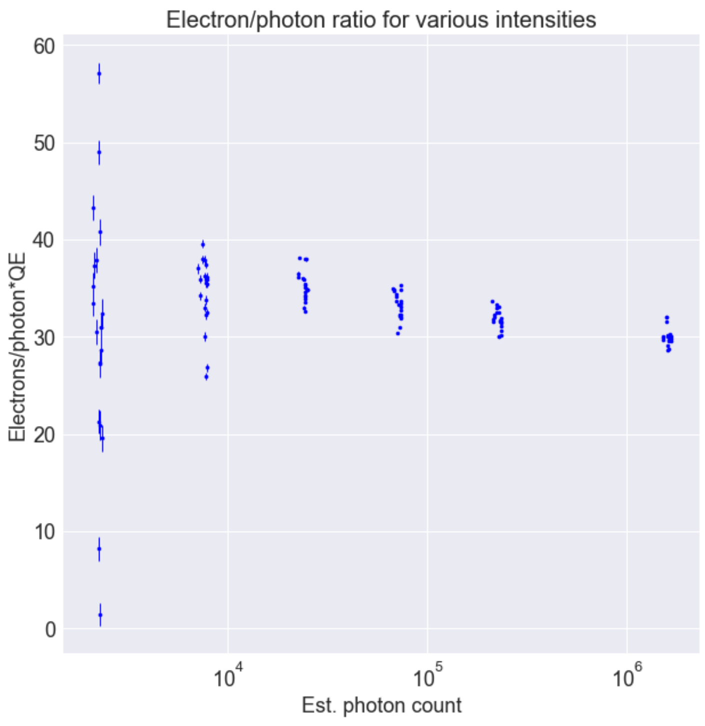


Figure 10: Calculated electron/photon ratios for each distinct trace. A greater spread indicates a smaller SNR. Note that the photon counts used to calculate the ratios include the QE of the APD. Also note that the 23% systematic uncertainty in photon counts should be taken into account not only on the x axis of this graph, but also on the electron/photon ratio.

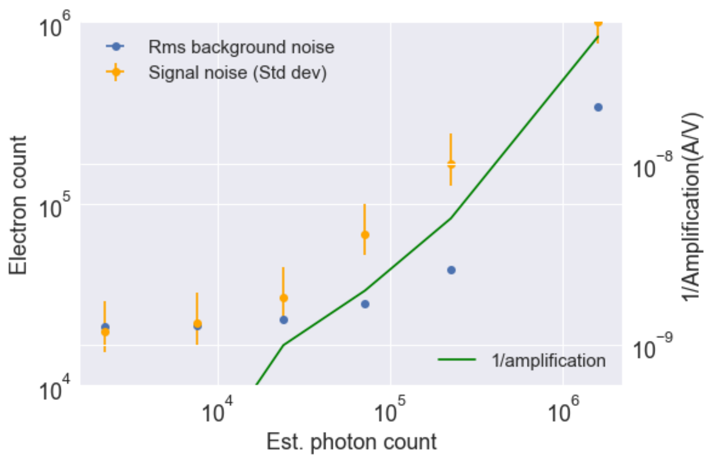


Figure 11: Comparison between the noise calculated from the rms noise in the background signal, the actual noise in the signal, and the inverse of the amplification. Note that while the inverse of the amplification was scaled to the same ratio as the noise graphs, its vertical offset is arbitrary. The photon intensities have a systematic uncertainty of 23%.

6.2 Reverse voltage

In order to characterise the behaviour of the APD over a range of different reverse voltages, and determine if it was possible to reach the shot noise limit at low signal strengths, two sets of measurements were done; one at a signal strength of $2 \cdot 10^3$ photons, and one at $9 \cdot 10^3$ photons. Both runs tested voltages between $145V$ and $155V$, and both ran at high enough amplifications to prevent the final-stage noise identified in Section 6.1 from dominating. As can be seen from the logarithmic plots in Figures 13 and 14, the gain grew strongly with voltage. The noise grew with voltage too, and, as can be seen in Figure 12, it followed the behaviour as expected from a combination of thermal noise and dark current shot noise with dark current figures of $I_{ds} = 1 \cdot 10^{-10} A$ and $I_{dg} = 5 \cdot 10^{-11}$. A possible explanation for the large fluctuations in the noise is that dark current could possibly be more strongly varied than the other noise sources, needing greater sample sizes to get an accurate reading of the standard deviation.

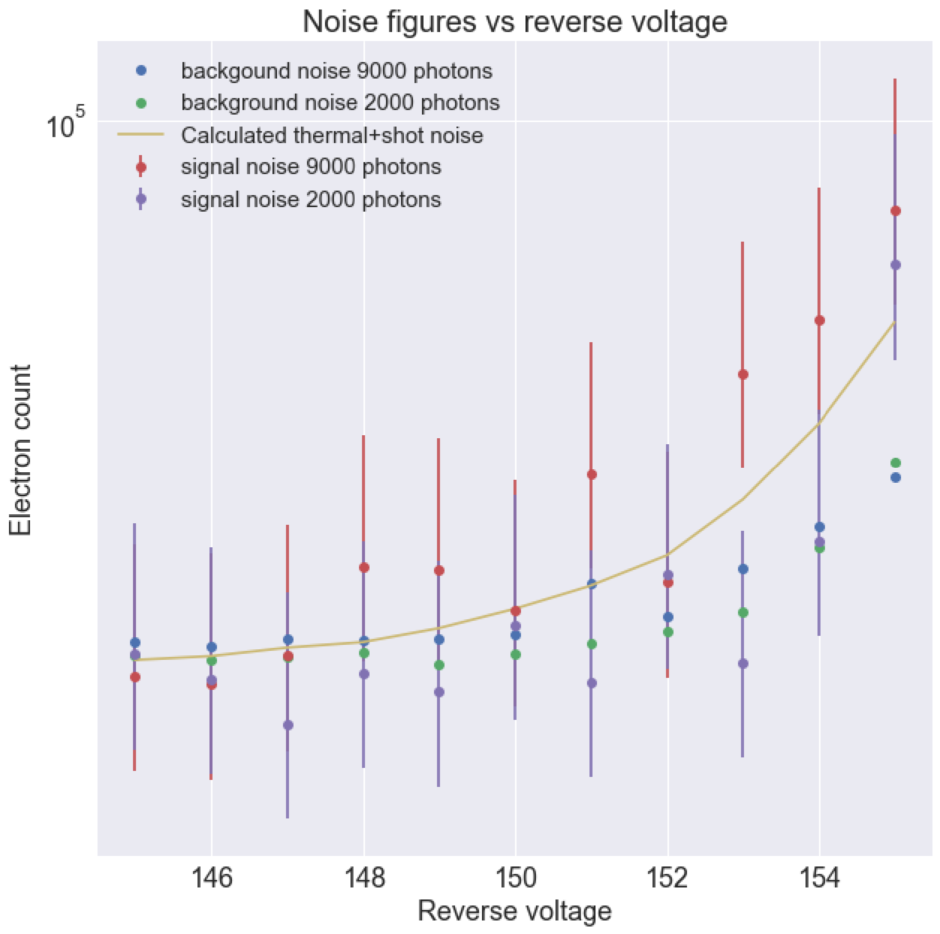


Figure 12: Background rms and signal noise curves for the two reverse voltage measurement sets. The expected noise was calculated by combining dark current shot noise calculated using Equation 4 with the thermal noise of $6.1 \cdot 10{-}11A$ as found in Section 6.1. As can be seen, the noise figures for both the signal and background noise follow the calculated trend rather well. However, where the 2 background traces agree quite well with each other and with the $2 \cdot 10^3$ electron signal, the noise for the $9 \cdot 10^3$ electron signal is significantly higher, possibly due to signal-dependent noise in the final amplifier stage.

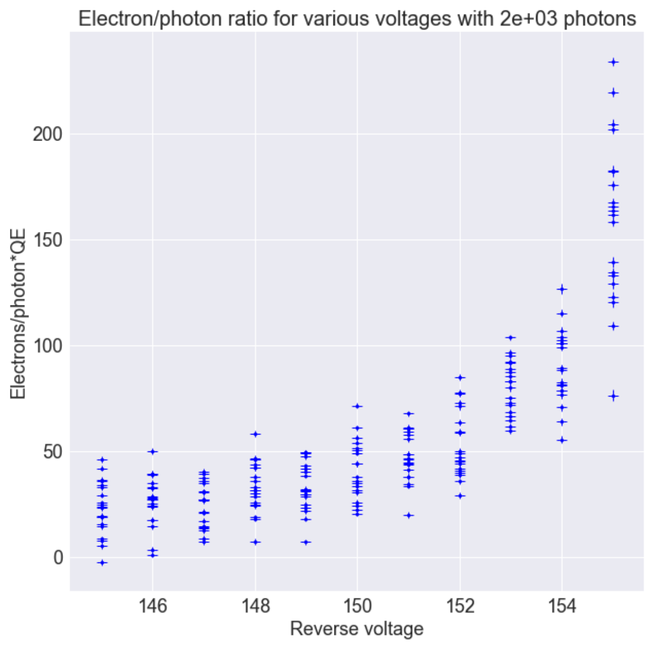


Figure 13: Calculated electron/photon ratios for each distinct trace at an incoming signal of $2 \cdot 10^3$ incoming photons. Notice on this graph, and the graph below, the strong dependence of the gain on the reverse voltage, as well as the increase in noise at higher reverse voltage.

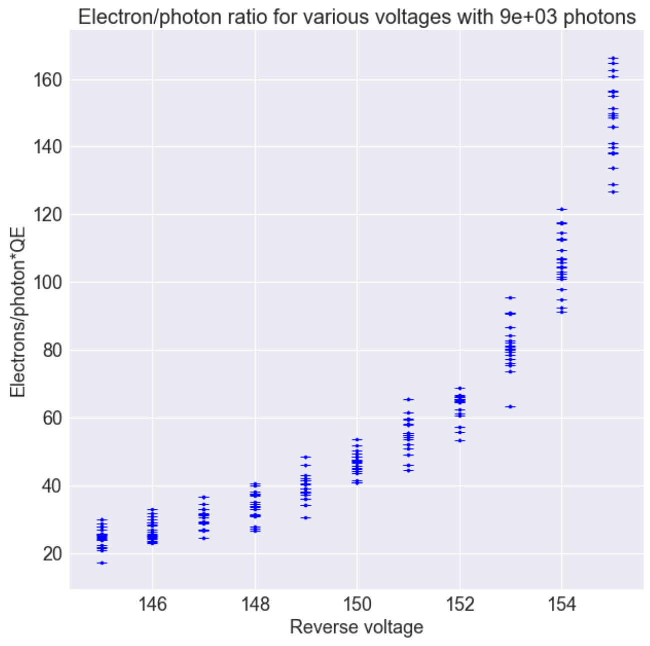


Figure 14: Calculated electron/photon ratios for each distinct trace at an incoming signal of $9 \cdot 10^3$ incoming photons. A greater spread indicates a smaller SNR. Note that the photon counts used to calculate the ratios include the QE of the APD. Also note that the 23% systematic uncertainty in photon counts should also be taken into account on the electron/photon ratio.

7 Outlook

7.1 Tests on the current setup

A number of further tests can be performed with the current setup. Firstly, it would be interesting to see if it is possible to reach the shot noise limit at higher signal strengths with a different second stage amplifier to eliminate the 1/amplification noise. Secondly, as the gain the specific APD used has quite a large temperature dependence, it might be interesting to see if it is possible to achieve higher gains with lower dark current shot noise at lower temperatures. Another important test undertake is to determine the temporal stability of the gain by leaving the setup running for an elongated period of time while regularly taking measurements, as it will have to run for a long period of time in the NL-eEDM experiment as well.

7.2 Changes to the electronics

One thing that could be considered to reduce thermal noise is to use multiple, smaller APDs placed side-by-side. The smaller Hamamatsu APDs have larger gains and, just as importantly, lower terminal capacitance, allowing for larger load resistances without losing the required bandwidth, reducing thermal noise. Connecting them in parallel should also allow the signal to benefit from the fact that while the signals can sum, the noises will sum as a sum-of-squares, effectively scaling with the square root of the number of APDs. Doing the calculations for the range of low bias-voltage APDs offered by Hamamatsu, we find that adding together 4 of the smaller S3884 photodiodes should yield the same level of thermal noise with an expected gain of 100 instead of 60, but the real-world behaviour needs to be measured. Summing before the trans-impedance stage might be possible by using an individual bootstrap or cascode as suggested in [17] to negate the summing capacitances of the diodes and overcome the e_nC.

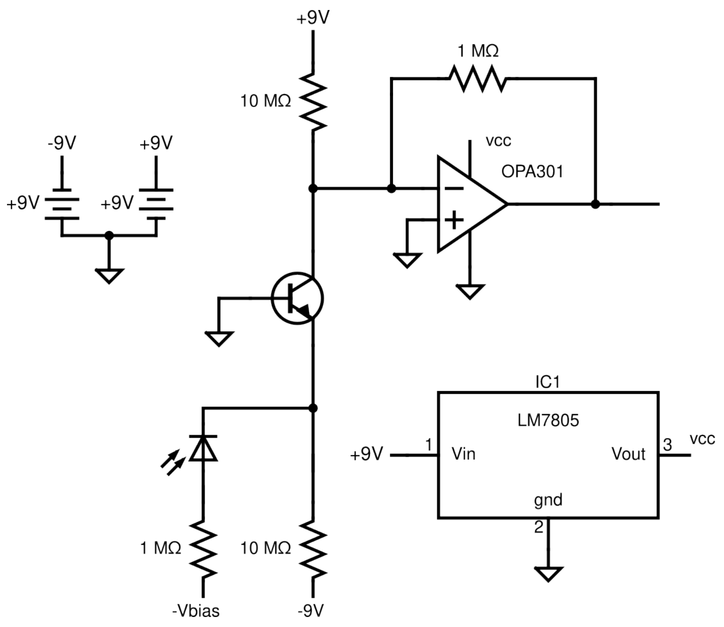


Figure 15: A possible cascode setup. Using 9V batteries provides a very clean power supply, creating a 1μA bias current through the transistor, giving it an internal resistance of $r_c = 25\text{k}\Omega$. The $10\text{M}\Omega$ resistors should have a factor 10 less thermal noise than we see now, allowing us to potentially reach the shot noise limit sooner. The supply voltage for the OPA301 is created by an LM7805 linear voltage regulator, which smoothly steps down the 9V to the 5V needed by the op-amp.

This cascode setup generally has significant benefits, as it can be used in a circuit as shown in figure 15 to reduce noise significantly. In this setup, we keep the resistance of the transistor low by supplying a very clean $\sim 1\mu$A bias current, setting the resistance 'felt' by the APD to about 25kΩ, leaving the bandwidth high. On the other hand, as the transistor does not have thermal noise, and the resistors are far larger, we can expect the thermal noise to be a factor 10 smaller than in the current setup. Besides, as the capacitance of the transistor can be far lower than that of the photodiode (3pF vs 40pF), we can reduce our e_nC noise, and we can increase the gain resistor as that too is limited by coupling to the photodiode capacitance to reduce bandwidth.

Creating our own circuit like this would also allow us to have the amplificiation circuit and the photodiode on one PCB, reducing pickup noise in wires, as well as allowing us to directly attach our own filters and second stage amplifiers, designed for low noise operation with our specific signal.

8 Conclusion

The viability of using avalanche photodiodes for the state-detection in the NL-eEDM experiment was investigated. A successful measurement method was devised, and measurements were taken to determine the behaviour of the setup under varying reverse voltages and signal intensities. As in all of these runs other sources of noise dominated, the shot noise limit was not reached, and so the current setup is likely not viable. In order to improve this, a number of improvements to the setup were suggested, as well as some more tests.

References

[1] Maxim Pospelov and Adam Ritz. "Electric dipole moments as probes of new physics". In: *Annals of Physics* 318.1 SPEC. ISS. (2005), pp. 119–169. ISSN: 00034916. DOI: 10.1016/J.AOP.2005.04.002.

[2] Tanya S. Roussy et al. "Experimental constraint on axion-like particle coupling over seven orders of magnitude in mass". In: *Physical Review Letters* 126.17 (June 2020). DOI: 10.1103/PhysRevLett.126.171301. URL: http://arxiv.org/abs/2006.15787%20http://dx.doi.org/10.1103/PhysRevLett.126.171301.

[3] P. G. H. Sandars. "The Electric Dipole Moment of an Atom". In: *J.Chem. Phys* 14 (1963), p. 939.

[4] P. G. H. Sandars. "The Search for Violation of P or T Invariance in Atoms or Molecules". In: *Atomic Physics 4* (1975), pp. 71–92. DOI: 10.1007/978-1-4684-2964-0{_}6.

[5] "Improved limit on the electric dipole moment of the electron AcMe collaboration*". In: (2018). DOI: 10.1038/s41586-018-0599-8. URL: https://doi.org/10.1038/s41586-018-0599-8.

[6] J. J. Hudson et al. "Improved measurement of the shape of the electron". In: *Nature* 473.7348 (May 2011), pp. 493–496. ISSN: 00280836. DOI: 10.1038/NATURE10104.

[7] William B. Cairncross et al. "A precision measurement of the electron's electric dipole moment using trapped molecular ions". In: *Physical Review Letters* 119.15 (Apr. 2017). DOI: 10.1103/PhysRevLett.119.153001. URL: http://arxiv.org/abs/1704.07928%20http://dx.doi.org/10.1103/PhysRevLett.119.153001.

[8] Parul Aggarwal et al. "Deceleration and trapping of SrF molecules". In: *Physical Review Letters* 127.17 (Mar. 2021). DOI: 10.1103/PhysRevLett.127.173201. URL: http://arxiv.org/abs/2103.07968%20http://dx.doi.org/10.1103/PhysRevLett.127.173201.

[9] Parul Aggarwal et al. "Measuring the electric dipole moment of the electron in BaF". In: *European Physical Journal D* 72.11 (Nov. 2018). ISSN: 14346079. DOI: 10.1140/EPJD/E2018-90192-9.

[10] P. Aggarwal et al. "Lifetime measurements of the A (2)Pi(1/2) and A (2)Pi(3/2) states in BaF". In: *Physical Review A* 100.5 (Nov. 2019), p. 052503. ISSN: 2469-9926. DOI: 10.1103/PHYSREVA.100.052503. URL: https://research.rug.nl/en/publications/lifetime-measurements-of-the-a-2pi12-and-a-2pi32-states-in-baf.

[11] Glenn G. Knoll. "Radiation Detection and Measurement (4th Edition)". In: *John Wiley & Sons, Inc.* (2010), pp. 205–356.

[12] Paul Horowitz and Winfield Hill. *Art of Electronics, The.* 3rd ed. Camb.U.P., 2015. ISBN: 0521809266. URL: https://books.google.com/books/about/The_Art_of_Electronics.html?id=LAiWPwAACAAJ.

[13] Edward M.D. Fisher. "Principles and Early Historical Development of Silicon Avalanche and Geiger-Mode Photodiodes". In: *Photon Counting - Fundamentals and Applications* (Mar. 2018). DOI: 10.5772/INTECHOPEN.72148.

[14] Hamamatsu. *Characteristics and use of Si APD (Avalanche Photodiode).*

[15] Hamamatsu. *Si APD S12023 series, low bias operation, for 800 nm band.* URL: www.hamamatsu.com.

[16] Texas Instruments. *Low-Noise, High-Speed,16-Bit Accurate, CMOS Operational Amplifier datasheet (Rev. D).*

[17] Philip C.D. Hobbs. "Photodiode Front Ends: The Real Story". In: *Optics and Photonics News, Vol. 12, Issue 4, pp. 44-47* 12.4 (Apr. 2001), pp. 44–47. ISSN: 1541-3721. DOI: 10.1364/OPN.12.4.000044. URL: https://opg.optica.org/abstract.cfm?uri=opn-12-4-44%20https://opg.optica.org/opn/abstract.cfm?uri=opn-12-4-44.

A Processing code

The code used to process the measured signals was originally made to run in a Jupyter Notebook, and so will be easiest to read in multiple sections. The first section sets up all the necessary code to get all the data from the traces, as well as prepare some things for the later sections.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal
import scipy.integrate as integrate
import scipy.stats as stats

plt.style.use('seaborn')
plt.rcParams.update({'font.size': 18,
  'xtick.labelsize' : 18,
  'ytick.labelsize' : 18})

set up some useful functions for errorprop
def errormult(A,B,errA,errB):
  out = np.multiply(A,B) #use numpy so we can hand it arrays
  out_err = np.multiply(out,np.sqrt(np.divide(errA,A)**2+np.divide(errB,B)**2))
  return out,out_err

def errordiv(A,B,errA,errB):
  out = np.divide(A,B) #use numpy so we can hand it arrays
  out_err = np.multiply(out,np.sqrt(np.divide(errA,A)**2+np.divide(errB,B)**2))
  return out,out_err

def erroradd(A,B,errA,errB):
  out = A+B
  out_err = np.sqrt(errA**2+errB**2)
  return out,out_err

```

```python
def erroravg(data, errs):
  avg,sow = np.average(data,weights=np.divide(1,errs**2),returned=True)
  avg_err = sow**(-1/2) # use the sum of weights to calculate the propagated error
    since this will likely be higher
  return avg, avg_err

#get a bunch of constants in
sipd_power = 0.24 #A/W, approx, measured against lasermate
sipd_err = 0.01
timestep = 1*10**(-6) #each point is one us

ratio, ratio_err = errordiv(300,260,10,10) #ratio between the beam going to the APD
    and SiPD

electronNo = 6.24 * 10**18 #electrons per coulomb

#get the energy per photon
wavelength = 850 * 10**(-9)
wavelerr = 1 * 10**(-9)
hbar = 6.626176 * 10**(-34)
c = 299792458
photon_energy, energy_err = errordiv(hbar*c,wavelength,0,wavelerr)
photon_energy

att_err = 10**(-0.05) #attenuation uncertainty due to rounding to the nearest 10**0.1

def peakfinder(trace):
  peak_data = scipy.signal.find_peaks(trace,height=0.05,width=50,distance=20,
    rel_height=0.975)
  ips = np.array([peak_data[1]["left_ips"][0],peak_data[1]["right_ips"][0]]) #
    intercepts (at rel_height)
  return ips.astype(int)

#class for control traces from the SiPD
class ctl_trace:
    def __init__(self,trace):
    self.trace = trace #set up the trace
    self.ips = peakfinder(self.trace) #find the peaks, since we will need them anyway
    self.offset_err = np.nan

  def recenter(self):
    #we want to find the general offset in the data so we can make sure there's
    #no contribution from that in the integral
    nopeak = np.delete(self.trace, range(self.ips[0]-30,self.ips[1]+30)) #get all of
    the trace without the peak
    offset = np.average(nopeak) #since the noise should be equal on both sides the
    average should give us an estimate
    self.offset_err = stats.sem(nopeak) #get the error in this estimate, since it'll
    self.trace -= offset #set it back

  def integrate(self,resistance,resistor_err):
    #we integrate to find the total amount of incident photons
    peak_slice = self.trace[self.ips[0]:self.ips[1]]
    integral = integrate.simpson(peak_slice,dx=timestep) #use simpson integration to
    find the integral in Vs
    integral_amps,erramps = errordiv(integral,resistance,0,resistor_err) #in As or
    Coulombs
    integral_joules,errjoules = errordiv(integral_amps,sipd_power,erramps,sipd_err) #
    almost there
    self.photons,self.errphotons_syst = errordiv(integral_joules,photon_energy,
    errjoules,energy_err)
    self.errphotons_stat = (self.ips[1]-self.ips[0])*timestep*self.offset_err # the
    sipd power one is systematic, this one is statistical
    return self.photons, self.errphotons_syst,self.errphotons_stat

```

```python
#class for signal traces from the APD
class sig_trace:
  def __init__(self,trace,ips):
    self.trace = trace #save the data
    self.ips = ips
    self.integral = 0 #I want this set so I can check if it's been changed or not
    self.electrons = np.nan
    self.rms = np.nan
    self.offset_err = np.nan
    self.integ_err = np.nan
    self.electron_err = np.nan

  def recenter(self):
    #we want to find the general offset in the data so we can make sure there's
    #no contribution from that in the integral
    nopeak = np.delete(self.trace, range(self.ips[0],self.ips[1])) #get all of the trace without the peak
    offset = np.average(nopeak) #since the noise should be equal on both sides the average should give us an estimate
    self.offset_err = stats.sem(nopeak)
    self.trace -= offset #set it back
    #print(offset)

  def integrate(self):
    #we integrate to find the total amount of incident photons
    peak_slice = self.trace[self.ips[0]:self.ips[1]]
    self.integral = integrate.trapezoid(peak_slice,dx=timestep) #trapezoid so we don't filter out the noise
    self.integ_err = self.offset_err * timestep * (self.ips[1]-self.ips[0])
    return self.integral, self.integ_err

  def background(self,amplification):
    totAmp = amplification*10**5 #adding the AMP120
    nopeak = np.delete(self.trace, range(self.ips[0]-30,self.ips[1]+30)) #get all of the trace without the peak
    rms = np.sqrt(np.mean(nopeak**2)) #find voltage rms
    self.rms = rms / totAmp #turn back into amps
    return self.rms

  def toElectrons(self,amplification):
    #turn the signal + the amplification from second amplifier into an approximate electron count
    totAmp = amplification*10**5 #adding the AMP120

    #make sure we actually have an integrated signal
    if self.integral == 0:
    integ, integerr = self.integrate()
    else:
    integ = self.integral
    integerr = self.integ_err

    #divide by totAmp to find amount of coulombs, then mult by electronNo to get actual counts
    self.electrons = integ * electronNo / totAmp
    self.electron_err = integerr/integ * self.electrons #treating electronNo as errorless bc it practically is

    return self.electrons, self.electron_err



def process_signals(series_list,runsperlist=20):
  lists = len(series_list)
  totruns = lists*runsperlist

```

```python
#first get the average incoming light on the SiPD
#this way, we can adjust for fluctuations in light, since the mirror should be splitting with
#the same ratio all the time
siPD_cuts = np.empty(totruns)
siPD_cuts_err_syst = np.empty(totruns)
siPD_cuts_err_stat = np.empty(totruns)

#we do the same trick we did for getting the control values
for i in range(lists):
  series = series_list[i,0]

  #get the resistance from the details file
  details = np.loadtxt(series.format("details"),delimiter=',',skiprows=1,dtype=object)

  #exec cannot edit local variables, so if called in a local scope you need to put it in a dict
  #we call locals() to get the current locals dict, then extract the right values
  #putting 'resistance' in the exec tho will cause problems when tryna extract the value
  exec("resistance_l ="+details[2,1],globals(),locals())
  exec("resistor_err_l ="+details[2,2],globals(),locals())
  resistance = locals()["resistance_l"]
  resistor_err = locals()["resistor_err_l"]

  for run in range(1,runsperlist+1):
    # load the data
    new_data=np.loadtxt(series.format(run),delimiter=',',skiprows=2)
    #invert it since it's negative, then chuck it into the class
    data = -1*new_data[:,2]
    trace = ctl_trace(data)
    #get rid of the offset
    trace.recenter()
    #get the integral
    siPD_cuts[i*runsperlist+run-1],siPD_cuts_err_syst[i*runsperlist+run-1], siPD_cuts_err_stat[i*runsperlist+run-1] = trace.integrate(resistance,resistor_err)

#do averaging, first weighted average, then the sem. I still need to figure out how to combine error and
cut_power,cut_err = erroravg(siPD_cuts,siPD_cuts_err_stat)
cut_err = 0 # we treat this as arbitrary, as it is only used for scaling up and down, and its error will be trumped by inc_phot_avg error

#set up global arrays
inc_photons = np.empty(totruns)
inc_photons_err_sys = np.empty(totruns)
inc_photons_err_stat = np.empty(totruns)
electrons = np.empty((totruns,2))
voltages = np.empty((totruns,2))
inc_photons_avg = np.empty((lists,2)) #value, error
signals_avg = np.empty((lists,2)) #avg, std error
noise_avg = np.empty(lists)
noise95 = np.empty((lists,2))
voltages_short = np.empty((lists,2))
rms_avg = np.empty(lists)
ampls = np.empty(lists)

for j in range(lists):
  #get the filename and attenuation
  series = series_list[j,0]
  att = 10**(-1*(float(series_list[j,1])))

  #get the details from the details file
```

```python
    details = np.loadtxt(series.format("details"),delimiter=',',skiprows=1,dtype=object)

    #see above for explanation what the fuck is going on
    exec("ampl_l ="+details[1,1],globals(),locals())
    exec("resistance_l ="+details[2,1],globals(),locals())
    exec("resistor_err_l ="+details[2,2],globals(),locals())
    exec("voltage_l = "+details[0,1],globals(),locals())
    exec("voltage_err_l = "+details[0,2],globals(),locals())

    ampl = locals()["ampl_l"]
    resistance = locals()["resistance_l"]
    resistor_err = locals()["resistor_err_l"]
    voltage = locals()["voltage_l"]
    voltage_err = locals()["voltage_err_l"]


    #set up some useful local arrays
    #run_photons = np.empty(runsperlist)
    series_sigs = np.empty(runsperlist)
    series_sigs_errs = np.empty(runsperlist)
    series_rms = np.empty(runsperlist)

    for run in range(runsperlist):
      #get the full counter
      fc = j*runsperlist + run

      # load the data
      new_data=np.loadtxt(series.format(run+1),delimiter=',',skiprows=2,)

      #invert ctl data since it's negative, then chuck it into the classes
      ctl = ctl_trace(-1*new_data[:,2])
      ips = ctl.ips

      #print(ips[1]-ips[0])

      sig = sig_trace(new_data[:,1],ctl.ips)

      #0-adjust both
      ctl.recenter()
      sig.recenter()

      #find the background signal rms
      series_rms[run] = sig.background(ampl)
      #print(rms)

      #reset ips so we do integrate over the full signal
      #sig.ips = [0,1200]

      #set the integrals going
      phot, photerr_sys,photerr_stat = ctl.integrate(resistance,resistor_err)
      signal,sigerr = sig.toElectrons(ampl)

      #get photon and electron count and signal scale factor
      att_err_local = np.sqrt(3)*((att/att_err)-att) #we want to find the local attenuation error (as its multiplicative), we use the top side as it's bigger
      photfac, photfacerr = errormult(att, ratio, att_err_local, ratio_err)
      inc_photons[fc],inc_photons_err_sys[fc] = errormult(phot,photfac,photerr_sys,photfacerr) #get the systematic errors
      inc_photons_err_stat[fc] = photerr_stat/phot * inc_photons[fc] #since the integral is the only source of statistic error we can use this

      electrons[fc] = sig.toElectrons(ampl)
      voltages[fc] = [voltage,voltage_err]
      #we ignore the systematic error as it is cancelled out by cut_power (note cut_err set to 0)
      scalefac, scalefacerr = errordiv(cut_power, phot, cut_err, photerr_stat)
      #print(scalefac)
      #rescale signal and save for averaging
      series_sigs[run],series_sigs_errs[run] = errormult(signal, scalefac,sigerr, scalefacerr)


    #calculate averages
    inc_photons_avg[j,0],inc_phot_avg_err_stat = erroravg(inc_photons[runsperlist*j:runsperlist*(j+1)],inc_photons_err_stat[runsperlist*j:runsperlist*(j+1)])
    inc_phot_avg_err_sys = np.average(inc_photons_err_sys[runsperlist*j:runsperlist*(j+1)]) #get the systematic error as average of the systematic errors
    inc_photons_avg[j,1] = inc_phot_avg_err_stat #combine both for plotting

    signals_avg[j] = erroravg(series_sigs,series_sigs_errs)
    signals_avg[j,0] = np.average(series_sigs) #the weighted avereage was giving too much priority to a very low value so this is the temp solutio
    noise_avg[j] = np.std(series_sigs)
    noise95[j] = [0.2395*noise_avg[j],0.4606*noise_avg[j]] #confidence intervals
    voltages_short[j] = [voltage,voltage_err]
    rms_avg[j] = np.average(series_rms)
    ampls[j] = ampl

  return ampls, signals_avg,noise_avg,noise95.transpose(),inc_photons_avg,voltages_short,electrons,inc_photons,inc_photons_err_sys,inc_photons_err_stat,voltages,rms_avg

```

The second section deals with making the graphs for Figures 10, 9, and 11:

```
series_list = np.array([["Measurments_2/1045att_{}.csv",4.85],
                ["Measurments_2/1050att_{}.csv",5.7],
                ["Measurments_2/1055att_{}.csv",6.2],
                ["Measurments_2/1059att_{}.csv",6.6],
                ["Measurments_2/1064att_{}.csv",7.1],
                ["Measurments_2/1070att_{}.csv",7.7]
                ])

ampls, signals_avg,noise_avg,noise95,inc_photons_avg,voltages_short,electrons,inc_photons,inc_photons_err_sys,inc_photons_err_stat,voltages,rms_avg = process_signals(series_list)

SNR,SNerR = errordiv(np.stack([signals_avg[:,0],signals_avg[:,0]]),np.stack([noise_avg,noise_avg]),np.stack([signals_avg[:,1],signals_avg[:,1]]),noise95)


SNR = SNR[0,:].transpose()

#calculate expected shot noise figures
pulselength = 180*10**-6
e_charge = 1/electronNo
gain = 35
bandwidth = 30000

IL = e_charge*inc_photons_avg[:,0]/pulselength
Inrms = np.sqrt(2*e_charge*bandwidth*IL*(gain)**2.3)
Inint = Inrms* np.sqrt(pulselength/(2*bandwidth)) * electronNo

plt.figure(figsize=(10,10))
plt.title('Signal and noise for various intensities',size=20)
ax = plt.gca()

ax.errorbar(inc_photons_avg[:,0],signals_avg[:,0],xerr=inc_photons_avg[:,1],yerr = signals_avg[:,1],marker='o',ls='',label='Signal')
ax.errorbar(inc_photons_avg[:,0],noise_avg[:],xerr=inc_photons_avg[:,1],yerr = noise95
    ,marker='o',ls='', label='Noise (Std dev)')
ax.plot(inc_photons_avg[:,0],Iint,label='est shot noise',c='r')
ax.set_yscale('log')
ax.set_ylabel('Electron count',size=18)
ax.set_xlabel('Est. photon count',size=18)
ax.legend(prop={'size': 15})

ax2=ax.twinx()
ax2.set_yscale('log')
ax2.errorbar(inc_photons_avg[:,0],SNR[:],xerr=0.1*inc_photons_avg[:,0],yerr=SNerR[:],
    marker='o',ls='',label='SNR',c='g')
ax2.legend(loc=4,prop={'size': 15})
ax2.set_ylabel('SNR',size=15)

plt.xscale('log')
ax2.set_xlabel('Est. photon count',size=18)
plt.setp(ax2.get_xticklabels(),visible=True,color='k')
plt.show()


inc_photons_err = np.sqrt(inc_photons_err_stat**2)
print("systematic photon error is {:.2f} %".format(100*np.average(inc_photons_err_sys/
    inc_photons)))
electronQE,electron_QErr = errordiv(electrons[:,0],0.75*inc_photons[:],electrons
    [:,1],0.75*inc_photons_err)

plt.figure(figsize=(10,10))
plt.title('Electron/photon ratio for various intensities',size=20)
plt.errorbar(inc_photons[:],electronQE,xerr=inc_photons_err,yerr=electron_QErr,fmt='b.
    ',elinewidth=1)
plt.xscale('log')
plt.xlabel('Est. photon count',size=18)
plt.ylabel('Electrons/photon*QE',size=18)
plt.show()

rms_background_int = rms_avg * np.sqrt(pulselength/(2*bandwidth))*electronNo
plt.errorbar(inc_photons_avg[:,0],rms_background_int,xerr=inc_photons_avg[:,1],marker=
    'o',ls='',label='Rms background noise')
plt.errorbar(inc_photons_avg[:,0],noise_avg[:],xerr=inc_photons_avg[:,1],yerr=noise95,
    marker='o',ls='',label='Signal noise (Std dev)',c='orange')

plt.xscale('log')
plt.yscale('log')
ax = plt.gca()
ax.legend(prop={'size': 15})
ax.set_ylabel('Electron count',size=18)
ax.set_xlabel('Est. photon count',size=18)
ax.set_ylim(10**4,10**6)

ax2 = ax.twinx()
ax2.plot(inc_photons_avg[:,0],1/(ampls*10**5),label='1/amplification',c='g')
ax2.set_yscale('log')
ax2.set_ylabel('1/Amplification(A/V)',size=18)
ax2.set_xlabel('Est. photon count',size=18)
ax2.legend(loc=4,prop={'size': 15})
ax2.set_ylim(6*10**-10,6*10**-8)

plt.show()
```

The last two sections create the graphs for Figures 4, 13, and 14, as well as a few graphs that were omitted in this report.

```
series_volts = np.array([["Measurements_3/145rev_{}.csv",7.7],
                  ["Measurements_3/146rev_{}.csv",7.7],
                  ["Measurements_3/147rev_{}.csv",7.7],
```

```python
                   ["Measurements_3/148rev_{}.csv",7.7],
                   ["Measurements_3/149rev_{}.csv",7.7],
                   ["Measurements_3/150rev_{}.csv",7.7],
                   ["Measurements_3/151rev_{}.csv",7.7],
                   ["Measurements_3/152rev_{}.csv",7.7],
                   ["Measurements_3/153rev_{}.csv",7.7],
                   ["Measurements_3/154rev_{}.csv",7.7],
                   ["Measurements_3/155rev_{}.csv",7.7],
                   ])

ampls, signals_avg,noise_avg,noise95,inc_photons_avg,voltages_short,electrons,
    inc_photons,inc_photons_err_sys,inc_photons_err_stat,voltages,rms_avg =
    process_signals(series_volts)

noiseold = noise_avg
noise95old = noise95

SNR,SNerR = errordiv(np.stack([signals_avg[:,0],signals_avg[:,0]]),np.stack([noise_avg
    ,noise_avg]),np.stack([signals_avg[:,1],signals_avg[:,1]]),noise95)
SNR = SNR[0,:].transpose()

plt.figure(figsize=(10,10))
plt.title('Signal and noise for various voltages with {:.0e} photons'.format(
    inc_photons_avg[-1,0]),size=20)
ax = plt.gca()

ax.errorbar(voltages_short[:,0],signals_avg[:,0],xerr=voltages_short[:,1],yerr =
    signals_avg[:,1],marker='o',ls='',label='Signal')
ax.errorbar(voltages_short[:,0],noise_avg[:],xerr=voltages_short[:,1],yerr = noise95,
    marker='o',ls='',label='Noise (Std dev)')
ax.set_yscale('log')
ax.set_ylabel('Electron count',size=18)


ax2=ax.twinx()
ax2.set_yscale('log')
ax2.errorbar(voltages_short[:,0],SNR[:],xerr=voltages_short[:,1],yerr=SNerR[:],marker=
    'o',ls='',label='SNR',c='g')
ax2.legend(prop={'size': 15})
ax.legend(prop={'size': 15})
ax2.set_ylabel('SNR',size=18)

ax2.set_xlabel('Est. photon count',size=15)
plt.show()

inc_photons_err = np.sqrt(inc_photons_err_stat**2)
print("systematic photon error is {:.2f} %".format(100*np.average(inc_photons_err_sys/
    inc_photons)))
electronQE,electron_QErr = errordiv(electrons[:,0],0.75*inc_photons[:],electrons
    [:,1],0.75*inc_photons_err)

plt.figure(figsize=(10,10))
plt.title('Electron/photon ratio for various voltages with {:.0e} photons'.format(
    inc_photons_avg[-1,0]),size=20)
plt.errorbar(voltages[:,0],electronQE,xerr=voltages[:,1],yerr=electron_QErr,fmt='b.',
    elinewidth=1)
#plt.xscale('log')
plt.xlabel('Reverse voltage',size=18)
plt.ylabel('Electrons/photon*QE',size=18)
plt.show()

#I wanna use the rms stuff later
rms_old = np.copy(rms_avg)
```

```python
series_volts = np.array([["Measurements_4/145rev_{}.csv",7.1],
                   ["Measurements_4/146rev_{}.csv",7.1],
                   ["Measurements_4/147rev_{}.csv",7.1],
                   ["Measurements_4/148rev_{}.csv",7.1],
                   ["Measurements_4/149rev_{}.csv",7.1],
                   ["Measurements_4/150rev_{}.csv",7.1],
                   ["Measurements_4/151rev_{}.csv",7.1],
                   ["Measurements_4/152rev_{}.csv",7.1],
                   ["Measurements_4/153rev_{}.csv",7.1],
                   ["Measurements_4/154rev_{}.csv",7.1],
                   ["Measurements_4/155rev_{}.csv",7.1],
                   ])

ampls, signals_avg,noise_avg,noise95,inc_photons_avg,voltages_short,electrons,
    inc_photons,inc_photons_err_sys,inc_photons_err_stat,voltages,rms_avg =
    process_signals(series_volts)

SNR,SNerR = errordiv(np.stack([signals_avg[:,0],signals_avg[:,0]]),np.stack([noise_avg
    ,noise_avg]),np.stack([signals_avg[:,1],signals_avg[:,1]]),noise95)
SNR = SNR[0,:].transpose()

plt.figure(figsize=(10,10))
plt.title('Signal and noise for various voltages with {:.0e} photons'.format(
    inc_photons_avg[-1,0]),size=20)
ax = plt.gca()

ax.errorbar(voltages_short[:,0],signals_avg[:,0],xerr=voltages_short[:,1],yerr =
    signals_avg[:,1],label='Signal')
ax.errorbar(voltages_short[:,0],noise_avg[:],xerr=voltages_short[:,1],yerr=noise95,
    label='Noise (Std dev)')
ax.set_yscale('log')
ax.set_ylabel('Electron count',size=18)
ax.legend(prop={'size': 15})

ax2=ax.twinx()
ax2.set_yscale('log')
ax2.errorbar(voltages_short[:,0],SNR[:],xerr=voltages_short[:,1],yerr=SNerR[:],label='
    SNR',c='g')
ax2.legend(prop={'size': 15})
ax2.set_ylabel('SNR',size=18)

ax2.set_xlabel('Est. photon count',size=18)
plt.show()


inc_photons_err = np.sqrt(inc_photons_err_stat**2)
print("systematic photon error is {:.2f} %".format(100*np.average(inc_photons_err_sys/
    inc_photons)))
electronQE,electron_QErr = errordiv(electrons[:,0],0.75*inc_photons[:],electrons
    [:,1],0.75*inc_photons_err)

plt.figure(figsize=(10,10))
plt.title('Electron/photon ratio for various voltages with {:.0e} photons'.format(
    inc_photons_avg[-1,0]),size=20)
plt.errorbar(voltages[:,0],electronQE,xerr=voltages[:,1],yerr=electron_QErr,fmt='b.',
    elinewidth=1)
plt.xlabel('Reverse voltage',size=18)
plt.ylabel('Electrons/photon*QE',size=18)
plt.show()

gains = signals_avg[:,0]/inc_photons_avg[:,0]
x = 0.3
Ids = 10**(-10)
Idg = 5*10**(-11)

dark_shot_sq = 2*e_charge*bandwidth*(Idg*gains**(2+x)+Ids)
```

```python
dark_shot = np.sqrt(dark_shot_sq)
dark_shot_int = dark_shot * np.sqrt(pulselength/(2*bandwidth))*electronNo

thermal_noise = 2*10**4

est_noise = np.sqrt(thermal_noise**2+dark_shot_int**2)

rms_background_int = rms_avg * np.sqrt(pulselength/(2*bandwidth))*electronNo
rms_old_int = rms_old * np.sqrt(pulselength/(2*bandwidth))*electronNo

#plot the RMS noise
plt.figure(figsize=(10,10))
plt.plot(voltages_short[:,0],rms_background_int,marker='o',ls='',label='backgound
    noise 9000 photons')
plt.plot(voltages_short[:,0],rms_old_int,marker='o',ls='',label='background noise 2000
     photons')
plt.errorbar(voltages_short[:,0],noise_avg,yerr=noise95,marker='o',ls='',label='signal
     noise 9000 photons')
plt.errorbar(voltages_short[:,0],noiseold,yerr=noise95old,marker='o',ls='',label='
    signal noise 2000 photons')
plt.plot(voltages_short[:,0],est_noise,label='Calculated thermal+shot noise')
plt.xlabel('Reverse voltage',size=18)
plt.ylabel('Electron count',size=18)
plt.title('Noise figures vs reverse voltage',size=20)
plt.yscale('log')
plt.legend(loc=2,prop={'size': 15})
ax = plt.gca()

plt.show()
```

Of course, if anyone wishes to access the raw data referred to in this code, we will happily comply with any requests sent to t.e.tiemens@student.rug.nl.