Precision Characterization for Prototype Electronics for read out of the 3.2 Gigapixels camera for LSST

By: Othmane Rifki

Advisors: Gordon Richards, Mitch Newcomer, Rick Van Berg

Senior Thesis

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Physics

Contents

Abstract

The Large Synoptic Survey Telescope (LSST) is an 8.4m ground based telescope that will be used to produce a 6-band wide-field deep astronomical survey of the southern sky. The survey covers a wide range of astronomical topics such as using the deformation of the images of distant galaxies to map the distribution of dark matter in the universe. For this purpose LSST will use a 3.2 Gigapixels camera characterized by a high segmentation, low noise, fast read out, and sensitive to light in the range of 0.3 to 1.1 Micron. University of Pennsylvania's instrumentation group is involved in the development of the front end electronics for the LSST camera readout. The goal is to achieve a highly compact low power read out with a signal sensitivity of 4 microVolts per electron and a rate of 500 Kilopixels per second. To prepare for the tests, a pulser driven electronic signal is used to mimic the CCD signal in order to understand the noise level achieved, the integration time sensitivity, and the channel-to-channel interference. We performed read out consistency checks in order to determine a pedestal background and electronic noise baseline, in addition to determining the overall gain of the system. Then, introduced pulses to monitor the time evolution of the response and the crosstalk. My work aided in performing these studies by developing analysis tools based on the high energy physics data analysis interface ROOT that allowed the creation of histograms, Gaussian fits, and time changes of means and standard deviations. The end goal of this project is to determine the current characteristics of the electronic set up in terms of noise, gain, and crosstalk to prepare for the next electronic upgrade that will read CCD signals of the LSST.

1 Introduction

The recent years have known large developments in the fields of astrophysics, particle physics and cosmology which led to the conclusion that only 4% of the energy density present in the universe is baryonic. The rest is made out of invisible matter that accounts for 26% of the energy density called dark matter, and the remaining 70% is composed of some unknown substance driving the accelerating expansion of the universe, referred to as dark energy. The scientific community started investigating the origins of dark matter and dark energy by designing experiments to probe deep within the universe at its small dimensions as well as its large scales.

These efforts led to the construction of the Large Hadron Collider at CERN which is expected to offer some new insights on the particle nature of dark matter. In astrophysics, there was a need for a mission that would map the distribution of dark matter in the universe by collecting images at multiple optical bands, over wide fields and to faint magnitudes. The Large Synoptic Survey Telescope (LSST) was designed as an 8.4m ground based telescope that will be used to produce deep astronomical survey of the southern sky with the capacity to collect imaging data with very large sky coverage, cadence, and depth so that it can provide with precise images of the sky. One of the scientific methods that will be used to determine the distribution of mass in the universe is Gravitational Lensing; the deformation of images of distant light sources by massive objects. This method requires images with a high spatial resolution, and accurate flux ratios, light curves, and time delays. These measurements are necessary to perform studies on distant luminous objects, such as quasars, in order to infer the distribution of Dark Matter in the universe.

The scientific goals of LSST impose challenging requirements on the design of the telescope. One of the sensitive pieces of the instrument is the CCD Camera that will be used to capture images of the sky. LSST will use a 3.2 Gigapixel camera characterized by a high segmentation, low noise, fast read out, and sensitive to light in the wavelength range of 300 to 1100 nanometers with a dynamic range from about 10 to about 100,000 photons per pixel.

The work presented in this thesis is part of the vertical slice tests of the LSST camera electronics that was done with the University of Pennsylvania's instrumentation group. As it is described in the camera overview of **Section 2**, the electronics of the camera are composed of two main parts; the Front End Electronics and the Back End Electronics that reads out the voltage output of a camera pixel. It was crucial to perform a noise analysis of the electronics in order to determine the different sources of noise affecting the system, and the parameters that minimize it in order to match the desired noise of 4 electrons for the camera. A background understanding of noise is presented in **Sections 3 and 4**. The work carried out involved gain measurements, noise spectrum analysis, and channel crosstalk correlations which are detailed in **Sections 5, 6, and 7**. My work aided in performing these studies by developing analysis tools based on the High Energy physics data analysis interface ROOT described in **Appendix A**.
Finally, the science of LSST is addressed in **Section 8** where a case study of using LSST's high resolution of flux ratios and time delay measurements will be used to study gravitationally lensed quasars to probe dark matter subhalos.
This study instructed me on electronics, signal processing, computational analysis, and statistics in addition to the physics behind gravitational lensing.

2 The Telescope

The Large Synoptic Survey Telescope (LSST) is an 8.4m ground based telescope that is expected to cover half of the southern night sky in 6 optical bands (u, g, r, i, z, y) during a survey period of ten years. LSST will be located on Cerro Pachon mountain at an altitude of 2,715m in Northern Chilean Andes which is home for two other large telescopes; Gemini-South (8.2m in diameter) and Southern Astrophysical Research (SOAR) telescopes (4.3m in diameter). The selection of the site was based on a detailed weather study done by the Cerro Tololo Inter-American Observatory (CTIO), located some 10km away, which showed that 80% of the nights have favorable seeing conditions.

2.1 Optics

The LSST is composed of a three-mirror system with three refractive lenses and a filter. The primary mirror (8.4m in diameter) and tertiary mirror (5m in diameter) face the night sky and are both fabricated on the same substrate. The secondary mirror (3.4m in diameter) is placed in the opposite side with an opening of 1.8m. in the center to allow the camera to access the focal plane. The tertiary mirror focuses the beam of light to the lenses and corrects for any field aberrations. The filter is placed between the second and third lens, where the later acts as a vacuum barrier to the detector's cooling system; the cryostat. Figure 1 shows the configuration of the mirrors and lenses of LSST.

The optical focal length of LSST is 10.3 m; given its primary mirror size of 8.4m, the f/number is then f/1.23 (f/# = f/D, f = focal length and D the mirror diameter). A small f/# suggests that the telescope is fast because it collects more light in less time compared to a larger f/#, since light is spread over less pixels on the focal plane. The flat focal surface of the camera is 64-cm in diameter which corresponds to a 3.5° or 9.6 deg^2 field of view (FOV) (note the moon's FOV is .5°). The on-axis collecting area is $35m^2$ which gives an etendue of 319 m^2deg^2 (etendue = $A\Omega$, A = collecting area and Ω = FOV). The plate scale is defined as the number of degrees separating two points in the sky that corresponds to the distance separation on the focal plane . For LSST, this

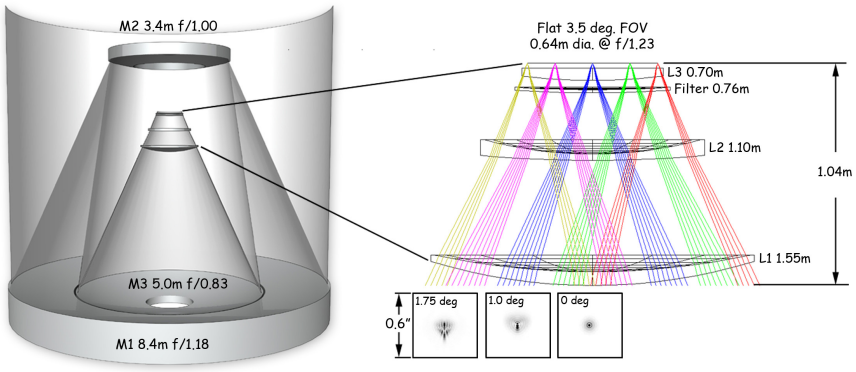

Figure 1: LSST's mirrors on the left, and lenses on the right. The lower image shows the spread of a point source being at an angle (0 deg, 1.0 deg, or 1.75 deg) from the primary axis of the telescope [2].

value is .2″/pixel (or 50μm/″).

LSST is expected to visit 20,000 deg^2 of the sky in 5.6 million 15 second exposures measured back-to-back by each filter. The allocated reading time is 2 seconds, with with an additional 2 seconds allocated for opening and closing the shutter. There are 189 CCD sensors that have 16 outputs (figure 3) which gives 3024 channels that will be read in parallel. Each output will read 1 million pixels during 2 seconds which imposes a readout rate of 500kHz.

2.2 Camera Electronics

The LSST camera is subdivided into independently operating sub-cameras composed of three main parts: The CCDs, the Front End Electronics (FEE), and the Back End Electronics (BEE). FEE is the generic term for the electronics package that reads the signals from the CCD and provides the CCD timing clocks to shift the charge from pixel to pixel until it reaches a read-out point. BEE is the generic term for the electronics package that performs the digitization of the signal coming from the FEE and controls the CCD timing clock generator of the FEE. An array of 3×3 CCDs are read by a set of connected FEE and BEE, forming a structure called a Raft. A description of the three pieces is given below.

2.2.1 CCD

A Charged Coupled Device or CCD operates by first accumulating charge as a result of exposing the surface of the CCD to photons of light which gets converted to electrons by the photoelectric effect. Next, the charge in sequentially moved to an output amplifier where the charge contained in individual pixels is read in order to reconstruct the image. Through manipulation of the substrate potentials, charges are shifted off of the CCD through two sets of operations: parallel registers move rows of charge in parallel to the edge of the sensor, and serial registers move individual pixels sequentially to a readout point. Figure 2 shows an example of a four output sensor where the parallel registers move the pixels contents from Image Section B to Image Section A and from Image Section C to Image Section D until the image sections reach the Registers GH and EF where the pixels are read one at the time. The individual pixels are read from registers GH and EF by

moving the content of each pixel to its neighbor until it reaches an amplifier at E, F, G, or H, based on the location. The parallel transfer of the charge will not occur until the serial readout is complete.

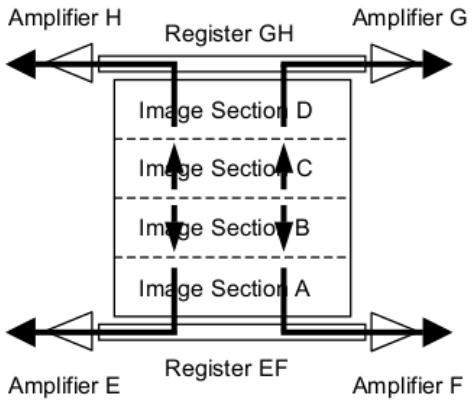

Figure 2: A four output sensor showing the parallel transfer of charge between image sections and the serial readout at the amplifiers of the registers[4].

The focal plane of the LSST camera has an area of 3200 cm^2 that will be covered by 189 highly segmented CCD sensors of 4K $\times$ 4K pixels 10 μm in size. Each CCD sensor (figure 3) is subdivided into 16 segments of 500$\times$2000 pixels, each connected to a readout point. The parallel register shifts the 500 pixels segment over 2000 lines by a three phase clock, while the serial register where the individual charge is read shifts 500 pixels in a four phase clock.

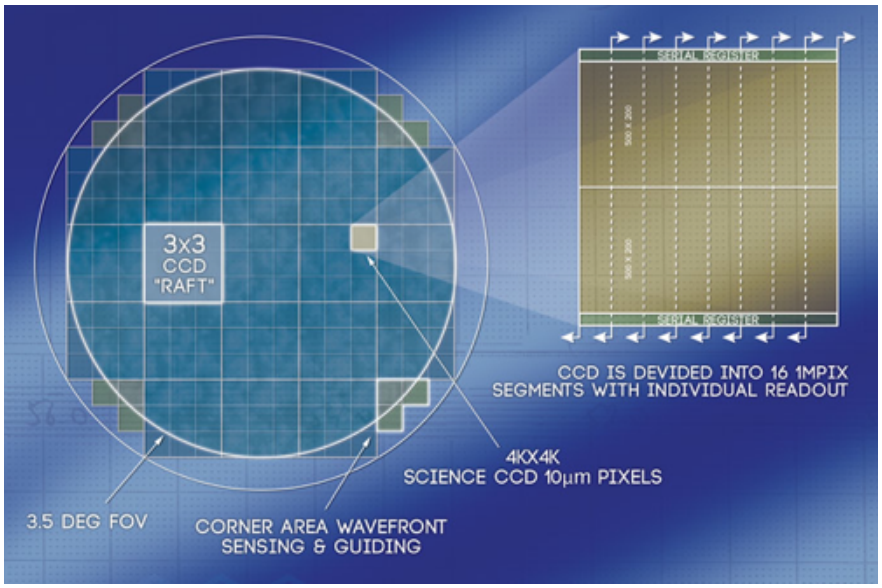

Figure 3: Focal plane of the camera showing the 3$\times$3 CCD rafts and a magnified sensor with 4K $\times$ 4K pixel subdivided into 16 segments each with a readout point[3].

Figure 4 shows the electronics of the CCD read out sequence which consists of two Field Effect

Transistors (FETs); the output FET is in a source-follower configuration with constant current source that maintains a constant gate-source voltage that permits a high responsivity to small signal changes at the gate. The second FET is a reset that sets a reference voltage at the gate of the main FET. The charge is read one pixel at the time by initially raising the potential of the last serial register to keep the charge in it, and maintaining the potential of the output gate (OG in Figure 4) at a lower value to prevent the charge from moving to it. The CCD reset sets a reference voltage at the output gate (OG), when the electronics is ready for reading, the potential of the last serial register is lowered to permit the charge to move to the read out sensor. A voltage drop of the amount of Q/C is generated at the output gate, where C referes to the total capacitance due to the FETs, is followed by a similar drop at the output source. Recall that the voltage difference between the gate and the source stays constant due to the constant current drawn. The voltage drop at the source stays constant until the voltage reference is established by the reset, which allows multiple read outs of the voltage of the pixel if needed. The signal is read by means of the ASPIC chip that performs the integration of the signal as it will be described.

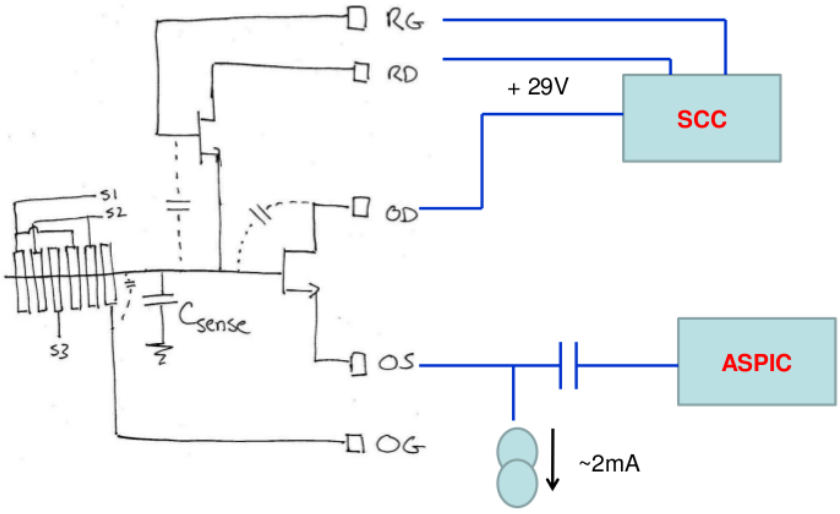

Figure 4: The CCD electronics showing a pixel with three serial registers and the ouput gate of the FET where a charge q will be converted to a voltage by q/C. The reset FET sets the output gate at a reference value. The SCC (Sensor and Control Chip) sets the different timing clocks and voltages of the CCD, while the ASPIC (Analog Signal Processing Integrated Circuit) performs the readout of the CCD signal.

The characteristics of the sensors set a limit on the device's sensitivity. For this reason the LSST has high requirements on its sensors some of which are listed below:

- High quantum efficiency (the number of photons producing an electron/hole pair) in the wavelength range from 320 to 1080 nm (u, g, r, i, z, y).
- Minimal contribution to the point spread function (charge diffusion of a point source).
- Surface flatness less than 10μm peak-to-valley (the difference between the highest and lowest part of the focal plane surface).

- The sensors cover a maximum focal plane area with a filling factor of 93%.
- Dark current of the focal plane is less then 0.2 electrons per second.
- Maximum charge per pixel 10^5 electrons.
- Read noise is about 5 electrons.

The work in this thesis is centered around determining the current noise levels of the prototype electronics of the camera that are downstream of the CCD.

2.2.2 Front End Electronics (FEE)

The Front End Electronics is mounted on a mechanical housing, called a raft tower, that consists of a set of six Front End Boards (FEB) connected to cooling planes as shown in Figure 5.

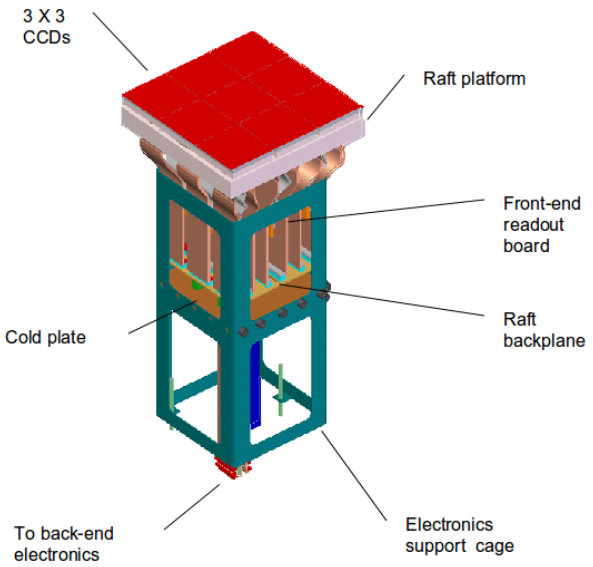

Figure 5: The figure shows how each CCD sensor is packaged into the 3×3 array of CCDs connected to the FEE[12].

Each FEB is responsible for reading a half row of CCDs in a raft tower requiring a set of 3×8 reading inputs. Each FEB integrates the signal received from one side of a row of the CCD array via three 8-channels ASPIC dual slope integrators. The board also contains sensor and control chips (SCC) that provide clock signals to the CCD.

2.2.3 ASPIC

The ASPIC chip, short for Analog Signal Processing Integrated Circuit, is the essential part of the FEE that will process the signal coming from the LSST science CCDs. It amplifies the CCD signal which improves the signal-to-noise ratio, and performs Dual Slope Integration which combines Correlated Double Sampling (CDS) with CCD signal integration, and to provide differential output signals for the Back End Board for digitization. Each channel of the ASPIC chip begins with an

amplifier that applies a gain to the input signal. The ASPIC allows the selection of one of three capacitors on the chip which give rise to three different nominal gain values of 2.5, 5, and 7.5 shown in figure 6 as programmable gains. The second stage transforms the single ended signal into a differential signal. In the last stage, both signals are integrated for programmable RC constants (500ns, 1μs, 1.5μs), where both signals are integrated. The use of a differential signal allows the subtraction of any noise referenced to ground. Finally, the CDS switches manage the polarity of integration for both signals in order to ensure opposite polarities during different integration times as it will be explained next.

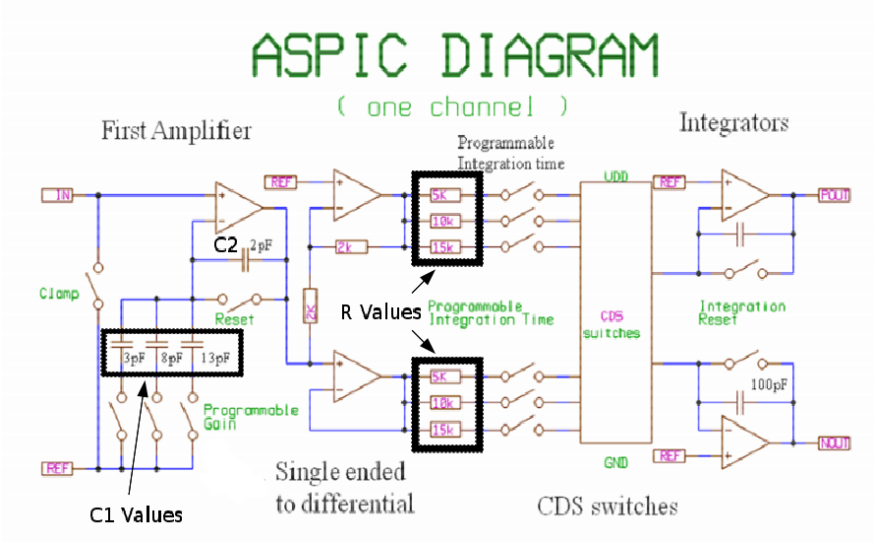

Figure 6: ASPIC Diagram: The CCD signal is transfered to the first amplifier that has programmable input amplifier gains of 2.5, 5, or 7. The signal is then converted to a differential signal that will be integrated by two integrators that have three integration time constants; 500ns, 1μs, and 1.5μs[10].

2.2.4 Dual Slope Integrator

The sampling process is performed in two steps. First, in a period designated "Ramp Down", background noise is integrated in order to get a baseline for the noise. Then charge is transfered to the input of the ASPIC, and a second integration is performed with equal length and opposite polarity; this period is designated "Ramp Up". The double integration with inverted polarity between Ramp Down and Ramp Up results in the integration of noise in one direction, then the integration of noise and signal in the opposite direction which results in a "double slope" integration. This process is also called Correlated double sampling and eliminates the reset noise, kT/C (capacitive) noise, and 1/f noise, while integration filters thermal noise (high frequency noise). One of the goals of the current study is to determine how well the dual slope integration eliminates noise in the system and to which frequencies are we least sensitive. Figure 7 displays a scope image showing the Ramp Down signal, Ramp Up signal, and the input pulse, as well as the differential output of the ASPIC.

The second version of the ASPIC chip that was used in this setup was developed by the Paris

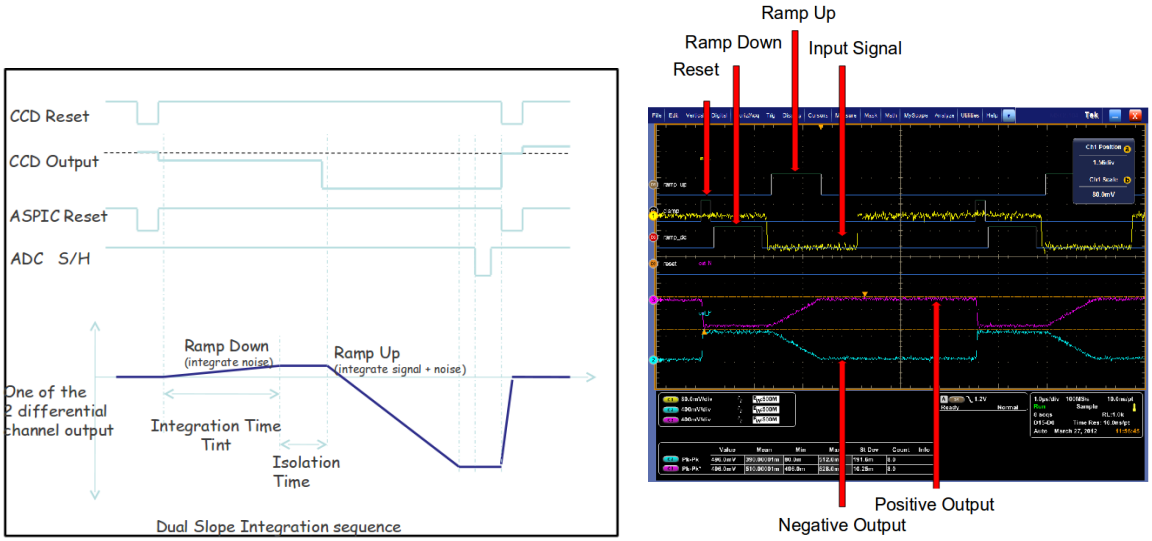

Figure 7: Left: Diagram of a correlated double sampling showing one of the two differential outputs of the ASPIC with the CCD output signal and the ADC digitization. Right: Scope image showing a Dual Slope Integration with a Reset, Ramp Down, Ramp Up, ASPIC input signal, and the differential ASPIC output signal which will be digitized.

IN2P3 group, which is working on the third version. A picture of ASPICII is shown in upper left corner of Figure 8.

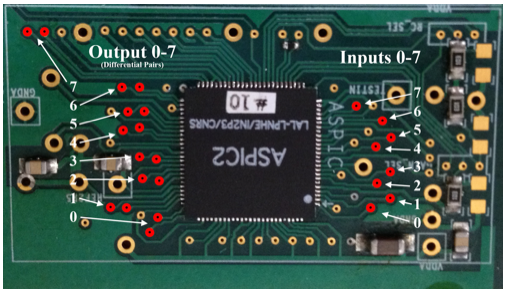

Figure 8: Picure of the ASPIC chip with the 8 input signals on the right, and 8 differential output signals on the left. One of the tests performed consists of sending pulses through channel 6 and measuring the differential output signal to get an approximate gain of the ASPIC.

2.2.5 Back End Electronics (BEE)

The Back End Electronics consists of a mechanical crate that connects six Back End Boards (BEB) to timing and readout controllers (called Raft Control Module RCM) The crate houses six Back End Boards and a module that provides with a path between the different boards of the camera (FEB and BEB) and the Raft Control Module (RCM) that controls timings and readouts of the different chips. The Front End Board and the Back End Board are connected via a Video Flex

Cable that guarantees the transmission of 24 differential video signals from the FEB to the BEB, and a Bias & Control Cable that carries timing, power, and bias signals to the FEB.

BEB has three functionalities. First, the BEE provide timing signals to the FEBs to control the CCDs gates and Reset switches as well as real time signals for control of the ASPIC. Second, the BEE receives ASPIC output signals (the video stream) and digitizes them for transmission to the Science Data System (SDS) via optical fiber. Finally, it handles numerous "housekeeping" tasks such as providing programmable voltages to different chips, monitoring temperature sensors distributed on the FEE and sensor packages, and preparing meta-data packets that refer to the conditions of operation of the telescope at data taking time (time, temperature, ...).

Each of the BEBs interfaces to a corresponding FEB in the Raft Tower where two complete FEB/BEB pairs service one row of CCDs in each Raft.

There are 24 channels of digitizer sub-circuits on each BEB arranged in three columns of eight circuits each. These correspond to eight of the segments in each of three CCDs in the Raft Tower. Each digitizer circuit consists of a differential amplifier to receive the differential video signals from the FEE and present them to an ADC. The differential amplifier is the THS4130 from Texas Instruments chosen for its low noise and high speed, and the analog-to-digital converter is the 18-bit AD7982 from Analog Devices.

In the tests performed in this study, one FEB was used which contains one operating ASPIC chip with an 8 channel BEB. The CCD signals were emulated by a Pulse generator.

We will refer to the prototype camera electronics consisting of a FEB and a BEB connected to RCB and RCM as the readout setup.
Taking a reading means performing the process of double integration and ADC conversion for each of the 8 channels outputted to a computer.

3 Vertical Slice Tests Overview

The LSST camera will read the 3.2 Gigapixels of the focal plane in 2 seconds with a total read noise of 7 electrons rms (root mean square value of the voltage fluctuations at the output of the CCD in absence of a signal), less than 4 electrons of which are allocated to the camera electronics (~ 3.36 electrons). The work performed in this thesis consists of characterizing the state of the camera's prototype electronics by running noise and timing tests on the multichannel readout setup that will perform the digitization of a CCD signal. The set of tests that starts from an input signal and ends with a digitized readout of the signal is referred to as the first vertical slice tests. These tests are performed on a single set of a Front and Back End Board driven by a pulse generator to mimic the camera CCD signals. Our investigation consists of studying the gain of the system through one channel (channel 6), the noise characteristic of the system through a noise spectrum analysis, the effect of integration time over noise, and finally the correlation between different channels. The elements of the electronic setup under investigation are the ASPIC chip (reads the CCD signal) and the ADC (digitizes the CCD signal).
We would like to measure the inherent noise in the system, its bandwidth, and the integration time that minimizes its value. Starting with basic assumptions, we know that the process of integration eliminates all high frequency noise (white noise) generated by passive devices in the circuit, while the

process of CDS [1] eliminates some of the low frequency noise. It would be interesting to determine what frequency bandwidth our system is least sensitive to. In order to study the noise levels achieved by the electronic setup, we use a white noise source, such as a resistor that has a flat frequency distribution. We would like to learn about the spectral density at the output given the fact that the ASPIC chip will eliminate a portion of the noise. By using the fact that white noise has a flat spectral distribution, the intrinsic noise and bandwidth of the readout setup will be determined through the use of a resistor.
The integration time will be varied in order to determine the time window that leads to the least noise contribution while maintaining the readout timing requirements.

4 Noise in the System

We motivate the study of noise in our system by its importance in determining the lower limit of the size of the signal we are able to amplify while maintaining a good quality readout [2] . In the current study, the readout noise is defined as the random voltage fluctuation at the CCD output in absence of a signal. This noise affects the output signal by adding some random fluctuations to it. The value of the noise at the input is obtained from the standard deviation of the output fluctuations converted to an equivalent input voltage of noise by applying the gain of the readout setup. The equivalent input noise is given in terms of number of electrons based on the CCD amplifier responsivity. A knowledge of the noise levels enable us to subtract its effect from the signal readout so that a limit can be set on the magnitude of the faintest objects that we can see in the sky.

This parameter is very important in the LSST camera design since the specifications require the camera to be sensitive to signals as low as 5 electrons. We start by giving an overview of the major types of noise we are expecting in the current camera readout prototype.

4.1 Probing Noise

The noise levels in the electronic setup can be probed by the Mixed Signal Oscilloscope (Tektronics MSO5104) used for the rest of the measurements. The noise we are probing is in the range of few tens of microvolts, the amplification raises the voltage to few millivolts at the output which is within the sensitivity of the scope. We short the inputs of the FEB and look at the differential output of the BEB opamp when we take a full integration cycle (Ramp Down and Ramp Up). Since there is no signal input, in an ideal case we expect that the integration of the noise present in the system during Ramp Down period should be completely canceled out during the Ramp Up period. However, our observation shows that there is some voltage fluctuations between the reference value (voltage at reset) and the end of integration as shown in the right frame of Figure 9. The variation is in the order of 1mV at the output. We have computed the gain of the setup to be in the order of 11, thus the noise voltage level is in the order of 100μV at the input, since it is a differential voltage we divide by 2 to get 50μV. Given the specification of the camera, each electron will contribute to about 5μV, thus we are seeing noise equivalent to 10 electrons. This is a rough estimate to illustrate the problem we are trying to address in this thesis; minimizing the camera electronics noise to 4

[1] Correlated Double Sampling: The process by which a sampling with a signal present is subtracted from a sampling without signal in order to eliminate the noise.

[2] The noise discussion was based on Chapter 11: Noise in Integrated Circuits of Gray's and Meyer's text [8].

electrons.

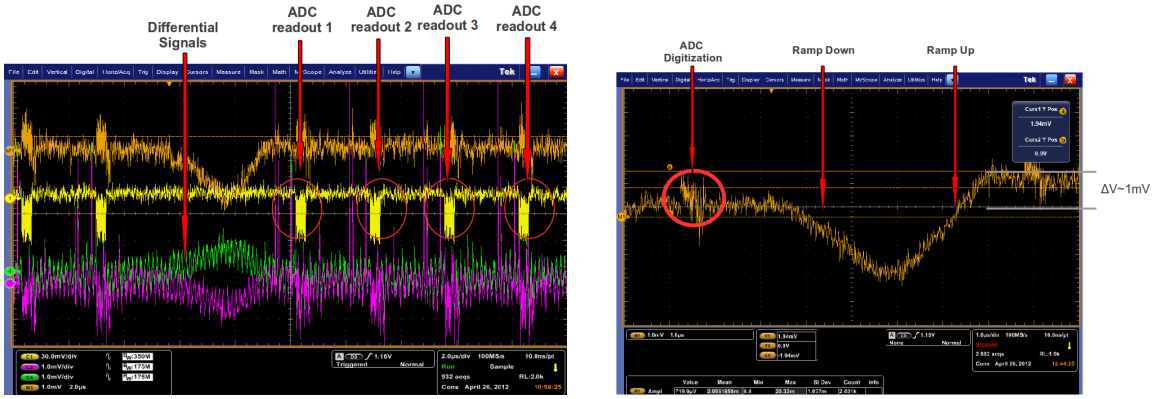

Figure 9: Left: Screenshot showing the differential output signals (green and pink traces) at the output of the BEB opamp during the Ramp Down and Ramp Up phases. The difference of the two signals (brown trace) gives the net output. Notice the ADC readouts. Right: The differential output of the opamp shows the double slope integration of the noise where the voltage nearly comes back to its original value.

4.2 Types of Noise

4.2.1 Thermal and Shot Noise

In any electrical circuit with a linear passive resistor, the random thermal motion of electrons generate electrical signals referred to as thermal noise (Noise was discovered by J.B. Johnson and explained by H. Nyquist). Thermal noise is always present in the system even if a current is applied since the thermal velocity of the electrons is higher than any drift velocities due to a current. The Nyquist theorem gives an expression for the thermal noise voltage generated by a resistor in thermal equilibrium that is proportional to the resistance R, the temperature T, and the frequency bandwidth Δf within which the voltage fluctuations are measured, all frequency components outside of the given range are ignored. The power of the noise signal is the same at each frequency; the larger the bandwidth, the more power we get, and thus the higher the noise signal.
Because of the random values the noise signal takes, we define the mean-square variation, $\overline{v^2}$, about the average voltage, V_M, over a time period T as

$$\overline{v^2} = \overline{(V - V_M)^2} = \lim_{T\to\infty} \frac{1}{T} \int_0^T (V - V_M)^2 \, dt \tag{1}$$

The relation between $\overline{v^2}$, T, R, and Δf is given by

$$\overline{v^2} = 4kTR\Delta f, \tag{2}$$

where k $= 1.3806 \times 10^{-23} J/K$ is Boltzmann's constant. This is the expression of the Nyquist theorem derived in Appendix B. The Norton equivalent is

$$\overline{i^2} = \frac{\overline{v^2}}{R^2} = 4kT\frac{1}{R}\Delta f \tag{3}$$

From expression (2), there is a linear relation between the mean-square noise value $\overline{v^2}$ and the bandwidth Δf. The spectral density is a function of frequency that is defined as the ratio of the mean-square noise fluctuation and the bandwidth, $\overline{v^2}/\Delta f$ (square volts per hertz). For thermal noise at a constant temperature T, the spectral density is constant as a function of frequency. This type of noise that displays a constant spectral density is called white noise. A 1-kΩ resistor at room temperature T = 300K, has a spectral density value of $\overline{v^2}/\Delta f = 16 \times 10^{-18}\text{V}^2/\text{Hz}$. The root-mean-square (rms) noise value can be written as v = 4 nV/$\sqrt{Hz}$. As we will discuss later, the rms noise of our prototype electronic setup have a value of 11.55μV and a bandwidth of .801 *MHz* which gives a value of 14nV/$\sqrt{Hz}$. The ASPIC noise specification has a maximum noise density of $5nV/\sqrt{Hz}$ [10] at a temperature of 173K.
There is another type of white noise similar to thermal noise known as shot noise. In a diode or a bipolar transistor with a depletion region having a certain potential, shot noise results from the random passage of charge carriers (electrons and holes) through the junction when they have sufficient energy. This process is arbitrary and the cumulative of this process give rise to a current I with small fluctuations around the mean value of direct current I_D. We define the mean square value of the current $\overline{i^2}$, about the average current, I_D, over a time period T as,

$$\overline{i^2} = \overline{(I - I_D)^2} = \lim_{T\to\infty} \frac{1}{T} \int_0^T (I - I_D)^2 \, \mathrm{d}t \tag{4}$$

The noise current has a mean-square value

$$\overline{i^2} = 2qI_D\Delta f \tag{5}$$

where the electronic charge q = 1.6 $\times 10^{-19}$C and the bandwidth Δf in units of hertz. Again, the noise-current spectrum density $\overline{i^2}/\Delta f = 2\text{q}\text{I}_D$ is constant as a function of frequency. From equation (3) and (5), we can determine the equivalent value of I_D to a 1kΩ resistor,

$$2qI_D = 4kT\frac{1}{R},$$

the direct current value is $\text{I}_D = 50$ μA. Thermal and shot noise are indistinguishable in a circuit since they both have a flat frequency spectrum and a Gaussian amplitude distribution (Figure 16).

4.2.2 Flicker Noise (1/f Noise)

Flicker noise is generated by the flow of direct current in active devices such as transistors. Its spectral density is expressed as

$$\overline{i^2} = K_1\frac{I^a}{f^b}\Delta f \tag{6}$$

where K_1 is a constant that depends on the device, a is a constant between 0.5 and 2, I is the direct current, and Δf bandwidth at frequency f.

When b = 1, the spectral density has a 1/f dependence, and hence the name 1/f noise. This type of noise is especially significant at low frequencies as it will be shown in Section 6.3.

4.2.3 Theoretical Gain of the Electronic Setup

The gain of the readout setup refers to the input equivalent voltage for a signal read at the output. In this section, we describe the different stages a signal goes through before being read at the output. In our measurements we will use two circuit configurations set by the appropriate choice of the capacitance C1 of the pre-amplifier and the resistance R of the integrator of Figure 6:

First test (Nominal gain 5, RC 1.5 us): C1 = 8pF, R = 15kΩ
Second test(Nominal gain 7.5, RC 500ns): C1 = 13pF, R = 5kΩ

The experimental setup consists of a Front End Board, attached to a Back End Board via flex cables, which is mounted to a module (RCM) that permits a computer data collection. When a pulse is sent through one of the inputs of the FEB, we get a digital output number between 131072 and 262144 (2^{17} and 2^{18}). Figure 10 shows a picture of the experimental setup.

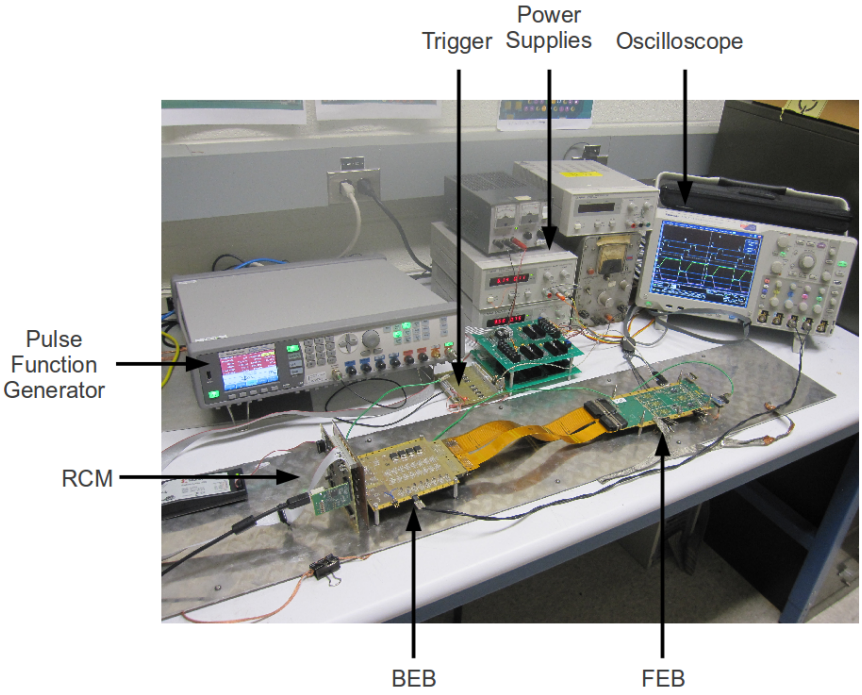

Figure 10: Experimental Setup

The first task is to calibrate the electronic setup by determining the gain factor by which an input signal V_{in} gets amplified to give an ADC readout value. Once the overall gain of the setup is obtained, the voltage input corresponding to each ADC value can be determined.

The electronic system performs the following processes on the initial voltage V_{in}; an amplification by an amount α_{ASPIC} in the ASPIC chip and by an amount α_{opamp} in the BEB opamp, an integration of the initial signal in the ASPIC chip, and an ADC conversion in the BEB.

Amplification

The ASPIC chip contains a preamplifier that sets the initial amplification of the signal by a programmable gain. Figure 6 shows a layout of the chip, with three capacitor choices in the inverting input of the operational amplifier. The gain at this stage is obtained by:

$$\alpha_{preamp} = \frac{C_1 + C_2}{C_2}$$

Where C_1 can be selected from one of the three values shown, and C_2 is fixed to 2pF. The first choice of $C_1 = 8$pF gives a gain of 5, and the second choice $C_1 = 13$pF gives a gain of 7.5. The amplification at the level of the BEB opamp is set to unity $\alpha_{opamp} = 1$.

Integration

Each of the differential signals is sent through an integrator that sets an amplification of α_{int} [3]:

$$\alpha_{int} = \frac{\Delta t}{RC}$$

The value of C is 100pF, and the value of R is set to 15kΩ in the first test and 5kΩ in the second test. At this stage, the overall amplification, α of the initial voltage V_{in} is

$$\alpha = \alpha_{preamp} \times \alpha_{opamp} \times \alpha_{int}$$

$$\alpha = 2 \times \frac{C_1 + C_2}{C_2} \times \frac{\Delta t}{RC} \tag{7}$$

The factor 2 is due to the contribution from each integrator which nearly outputs the same voltage. Note that the gain depends on the integration time Δt, in other words the length of Ramp Up or Ramp Down (they are expected to be equal in order to cancel equivalent noise contribution).

ADC

The voltage output αV_{in} will be digitized by the ADC converter in the BEB to a digital signal , i.e. ADC counts. The calibration consists of determining the voltage input corresponding to each ADC count (V/ADC count). Since the differential input range of the ADC is 10V, and it has a 2^{17} precision, the conversion factor is $7.6 \cdot 10^{-5}$V/ADC count.

For the noise measurements, we are interested in fluctuations of the voltage given by equation (1). As it will be shown, the noise distribution is very close to a normal distribution, we thus take the rms value $\bar{v}$ to be equal to the standard deviation σ. Equation (2) becomes

$$\sigma^2 = 4kTR\Delta f \tag{8}$$

[3] The voltage output of an Op-Amp integrator is $V_{out} = \frac{1}{RC}\int_0^t V_{in} dt$, since V_{in} is a constant,

$$V_{out} = \frac{V_{in}\Delta t}{RC}$$

When this value is multiplied by the experimental gain factor, the equivalent of the input noise is obtained. It should be noted that the noise gets amplified throughout the electronic chain, nevertheless, we are only interested in the input equivalent noise.

4.3 Noise Spectrum of the Electronic Setup

The noise spectrum of the electronic setup is characterized based on the previous discussion.
The system contains multiple sources of noise; passive devices (resistors, capacitors,..) and active devices (bipolar and FET transistors, opamps, ...) contained in the different chips of the electronics. For the purpose of our study, we distinguish between 4 noise sources; FEB input, ASPIC chip, BEB operational amplifier (opamp), and the ADC converter.
When we short the input channels of the electronic setup, and take multiple readings, the histogram obtain has a bell shape that can be fit by a Gaussian (Figure 16). We assume that the noise contribution from each category of the electronic setup follows a normal distribution with mean μ_i and standard deviation σ_i of the independent random variables X_i, where i=input, ASPIC, opamp, ADC.
From statistics, the total noise is obtained by adding in quadrature the contribution from each noise source in the electronic setup:

$$V(X) = V(X_{input}) + V(X_{ASPIC}) + V(X_{opamp}) + V(X_{ADC})$$

$$\sigma^2 = \sigma^2_{input} + \sigma^2_{ASPIC} + \sigma^2_{opamp} + \sigma^2_{ADC}$$

Since we know that the ASPIC, the BEB opamp, and the ADC are constituents of the electronic setup, we can fairly assume that the input resistance is the only noise variable in the system. We can rewrite the previous equation as

$$\sigma^2 = \sigma^2_R + \sigma^2_{setup}$$

where $\sigma^2_{setup} = \sigma^2_{ASPIC} + \sigma^2_{opamp} + \sigma^2_{ADC}$.

Our goal is to determine the value of the noise in the electronic setup,σ^2_{setup}, and the range of frequencies we are least sensitive to. A white noise source, a resistor, is placed at the input of the FEB, and several runs are taken in order to get a large enough sample to perform statistics on it.

Next, we allow the values of the resistance to vary, and by using the Nyquist theorem equation (8) for σ^2_R, we obtain

$$\sigma^2 = 4k_B RT\Delta f + \sigma^2_{setup} \tag{9}$$

The slope of this curve is proportional to the bandwidth Δf, while the y-intercept determines the value of the noise of the electronic setup σ^2_{setup} which is expected to be constant when we repeat the tests. We obtain the following relation:

$$\sigma^2 = AR + B,$$

with $A = 4k_B T\Delta f$, $B = \sigma^2_{setup}$

In summary, we would like to learn about the frequency range we are not sensitive to. We obtain this information by using a white noise source that generates noise at all frequencies, then we look

at the bandwidth obtained from the change of voltage fluctuations as a function of resistance, R, which also allows us to determine the overall noise in the system (due to the ASPIC, the BEB opamp, and the ADC converter).

5 Procedure

The first vertical slice tests that will be presented consist of characterizing the gain of the readout chain, approximating the gain of the ASPIC chip, determining the noise spectrum, evaluating the effects of integration time over noise, and finally determining the crosstalk correlation. The parts of the experimental setup that are of interest to us are the FEB inputs where we will be sending signals or connecting noise sources (resistors) shown in Figure 11, the single input and differential output of the ASPIC chip shown in Figure 8.
A Pulse Function Arbitrary generator (81150A) was used to produce square pulses with an adjustable amplitude, pulse width, time delay, and output impedance. The goal is to send pulses that mimic signals coming from a real CCD during the read out time window. The reset signal from the ASPIC chip was sent to the external trigger of the pulse generator in order to generate a pulse at each read out sequence.

An oscilloscope used was a Mixed Signal Oscilloscope (MSO5104) that has a 9 bit precision, an AC and DC coupling, a functionality to display digital signals used for showing ASPIC clocking signals, and analog probes.
The RCM contains a state machine that controls the timing, width, and number of signals based on a user defined set of commands. These commands are compiled into an input text file and uploaded to the RCM for execution where the timing signals get transfered to different parts of the electronic setup (ASPIC timing, ADC timing, ...).
The pulse is delayed so that it falls in between Ramp Down and Ramp Up and extends until the end of Ramp Up so that a signal is present during the integration period. It should be noted that in a real CCD, the output of a serial sensor is connected to the gate of an FET. When the charge is moved to the output gate, a voltage drop is induced at the output source of the FET which is then transfered to the ASPIC for readout. The voltage drop stays constant allowing multiple readouts until the output gate is reset to the reference value. Taking multiple readouts will allow a statistical analysis of the noise in the readout chain, however, the LSST camera in normal operation will read a pixel charge only once in order to stay within the readout time norms.
The oscilloscope of Figure 7 shows the three timing signals; Reset, Ramp Down, and Ramp Up. The pulse at the channel 6 input of the ASPIC is shown, along with the differential output of channel 6 (Negative and positive outputs). Note that the reset initializes the voltage output for a new reading. The effect of Ramp Down is not visible in this scope picture since the signal sent during Ramp Up is very large but it is seen clearly in Figure 9 when no signal is sent.
We chose our Ramp Up and Ramp Down signals to be 1μs in length, with a 20ns gap between the two periods. We used the scope to take measurements of the input and differential output of the signal by using an AC coupling, with an internal impedance of 50MΩ, and triggered on the Ramp Down pulse.

The output data from the RCM is a stream of ADC readouts from 8 channels. As we increase the number of ADC samples, the ADC distribution becomes more Gaussian. For this reason we

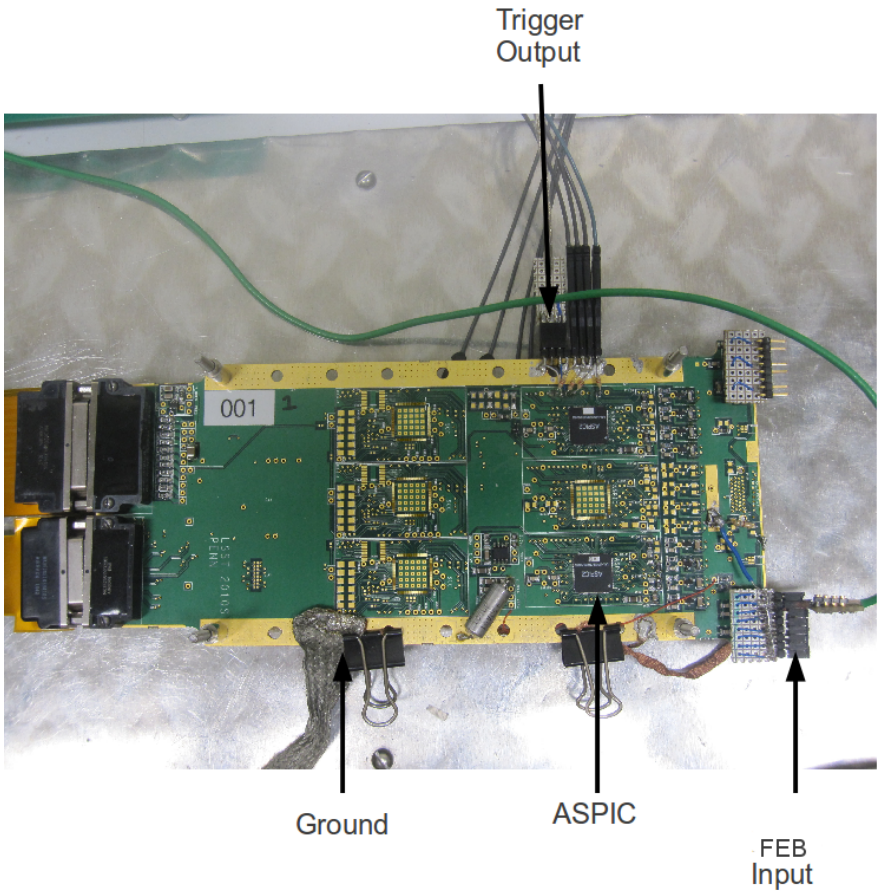

Figure 11: Picture of the FEB showing the 8 inputs of the FEB with a pulser on channel 6 (green wire), the ASPIC chip used for the signal readout, and timing signals used to supply an external trigger signal to the pulse function generator.

need to accumulate enough ADC readouts to get the mean and standard deviation of the ADC distribution that will be used for gain and noise measurements. However, the number of ADC readouts to be taken is limited to 517 readouts due to the hardware storage capacity. In order to determine the mean and standard deviation of an ADC distribution, the mean and standard deviation was computed first from the raw data. Then a histogram was obtained based on ADC readouts 5 standard deviations from the mean. This histogram was fit by a Gaussian to obtain a new mean ADC output and standard deviation (See Appendix C). This procedure was used for all tests to determine the mean and standard deviation of each ADC distribution.

The gain is determined by sending a pulse of a known amplitude, and obtaining the mean ADC output as described above. The signal coming from the Pulse Function Arbitrary generator is attenuated by 10 by a voltage divider before being injected in the FEB and measured at the input of the ASPIC chip. By incrementing the pulse amplitude by 500mV, and measuring the input pulse at the ASPIC, we measure that there is a linear relationship between the voltage input and the mean ADC output. The coefficient of linearity is the gain of the readout chain. In our tests, a single run consists of sending a sample of 257 input signals of the same amplitude that produce 257 ADC

readouts in each of the eight channels. We are interested in channel 6, so we plot the input voltage as a function of mean ADC output for this channel. The results are in the next section.
We also check the linearity of the ASPIC chip by measuring its output voltage as a function of the input voltage.

To measure the noise spectrum as described in section 4.3, we place different resistances at the desired FEB input while shorting all the other inputs (Figure 11). We then take 50 runs for each resistance value, where each run consists of 257 ADC readouts, to obtain a sample large enough to do a statistical analysis of the noise. Channel 6 is chosen with the following resistance values; a short (0 Ω), 1kΩ, 2kΩ, 5kΩ, 10kΩ, and 15kΩ. All external sources of noise should be minimized; for instance we turn off the oscilloscope and the pulse generator, and ensure an appropriate grounding of the mechanical setup, and short to ground all the other channels that are not being used. The results are shown in the next section.

6 Results and Analysis

In this section we present the results related to the gain measurement, the noise spectrum, the integration time effects on noise, and the crosstalk correlation between the different channels.

6.1 Gain Measurement

The gain is determined through plotting a graph of input voltages as a function of their mean ADC values. First, we histogram the ADC outputs of each run with a histogram bin width of 1 ADC count. Figure 12 shows histograms collected at different input voltages. It is clear that the histograms follow a normal distribution from which we deduce the standard deviation of the voltage fluctuations. We proceed by fitting a Gaussian to the histograms with tails going to $\pm 5\sigma$, which captures almost all data points. From this fit, we obtain a value for the sample mean and the sample standard deviation of the Gaussian for each particular voltage. Since the distribution is normal, the standard deviation σ and the rms value $\bar{v}$ are identical.

We then plot the voltage input at the ASPIC chip as a function of the mean value of ADC counts. The slope of this graph will give us the gain of the electronic setup determined as voltage per unit ADC count as shown in Figure 13.
The gain of the ASPIC is determined by measuring the ASPIC input and the differential ASPIC output using the oscilloscope (MSO).

At some measurements, we encountered a droop in the first values of the ADC counts, the cause of which were not identified. In such cases, we eliminated the first 80 ADC counts to minimize the droop effect as shown in Figure 14. This phenomena especially appeared in noise measurements addressed next.

We repeat similar steps for ASPIC nominal gain of 7.5. Figure 15 shows the graphs and Table 1 summarizes the results.

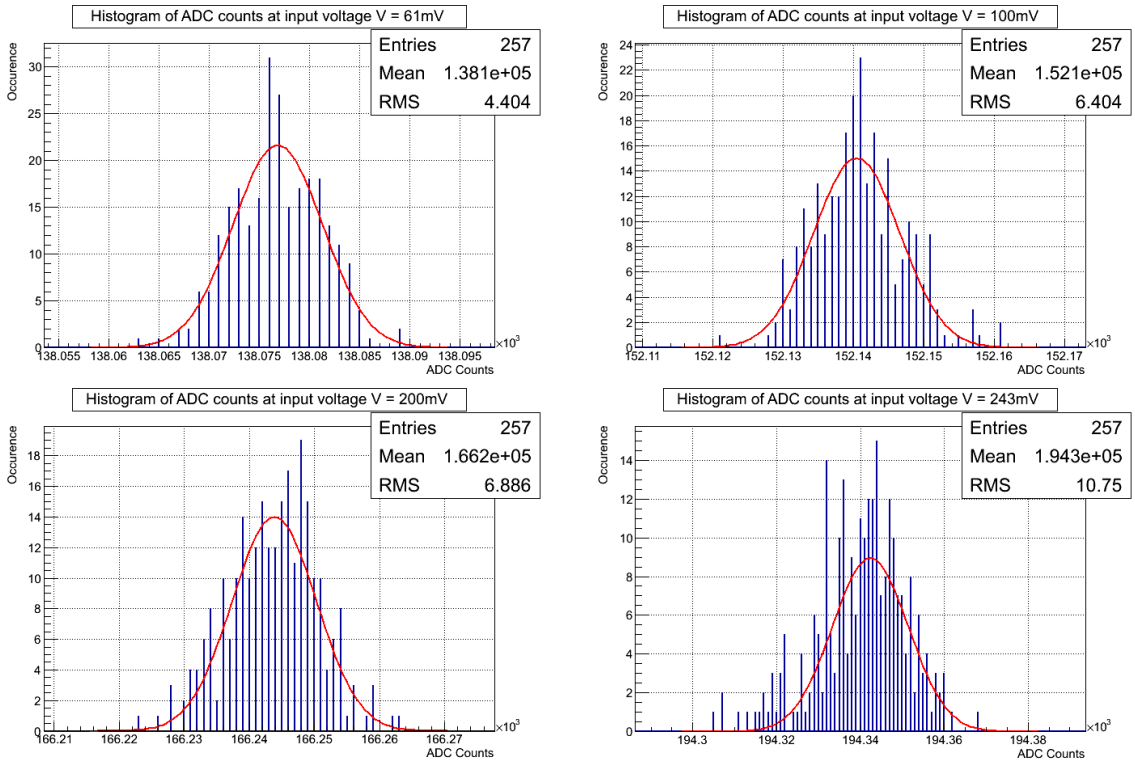

Figure 12: Histograms showing the increase in the mean ADC values as the voltage input increases.

	Overall Gain (μV/ADC count)	**ASPIC measured using Oscilloscpe**
Nominal Gain 5	4.832	7.849
Nominal Gain 7.5	3.465	10.96

Table 1: Gain measurements summary.

6.2 Noise Spectrum

In order to determine the noise spectrum, we follow the steps outlined in section 4.3 for the nominal gain of 5 to get an approximate noise bandwidth and noise level. Figure 16 shows a histogram of the output ADC values with shorted FEB inputs (note the reduced number of ADC counts 177) in the first ASPIC configuration (Nominal gain 5). The histogram has a Gaussian distribution which confirms our earlier assumptions.

In this example, the mean and RMS readout have values of 131000 and 3.66 counts respectively. By using the gain and the offset parameter (Par0) shown in Figure 13 (for Nominal Gain 5) we get an equivalent voltage input of -8.3mV (approximately 0V), and an equivalent RMS voltage of 17.6μV. The RMS voltage represents the random noise fluctuation superimposed on the detected signal. We will use this method to characterize the readout noise for each of our runs in order to determine the overall noise of the readout chain as well as the noise bandwidth.

We took 4 runs at each resistance value and determined the mean and standard deviation for each. The square of the standard deviation obtained from the Gaussian distribution approximates

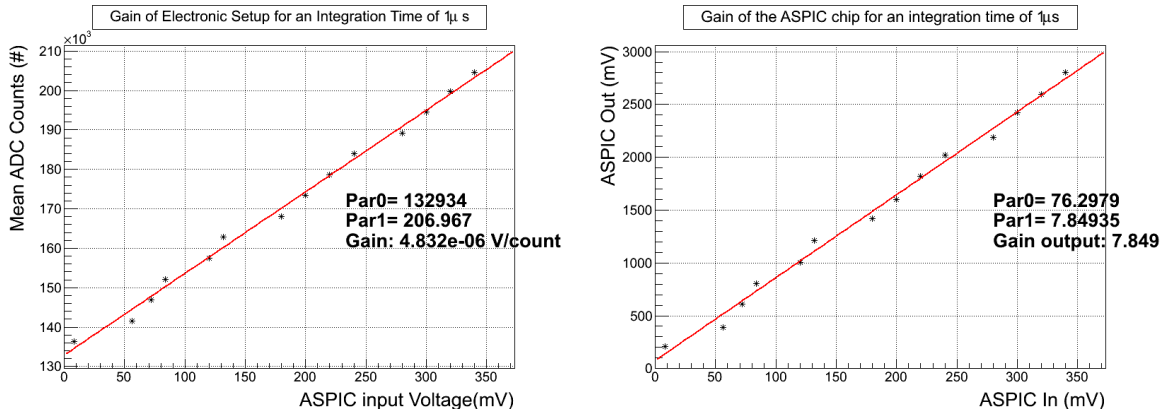

Figure 13: The graphs show the gain of a signal integrated for 1μs using channel 6 with a nominal ASPIC gain of 5. Left: The slope of the graph gives the overall gain of the electronic setup, i.e. voltage per unit ADC count. Right: The slope of the graph gives the gain of the ASPIC.

the value σ^2 that appears in equation (9). Upon plotting the 4 points, we removed the value that had the largest deviation and fit the results linearly as shown in Figure 17.

To improve upon the noise sensitivity and statistics of the previous measurement, we increased the value of the nominal gain to 7.5 and the number of runs to 50. We also dismissed the first 80 data outputs to get more stable ADC readouts.

The standard deviation of the ADC readouts was obtained for each of the 50 runs. Then the mean of these standard deviations and the standard error was computed based on the definitions given in Appendix C.

A graph of σ^2 referring to the readout noise as a function of resistance is constructed and fitted linearly in Figure 18. The noise results are summarized in Table 2.

	Bandwidth (MHz)	**Noise (μV)**
Nominal Gain 5	0.874	13.85
Nominal Gain 7.5	0.801	11.55

Table 2: Noise measurements summary.

6.3 Integration Time Dependence

The noise present in the readout setup fluctuates at different frequencies where each type of noise dominates at a certain frequency range as described in Section 4.2. Learning about the noise levels at different frequencies will instruct us on the types of noise present in the setup as well as the frequency range that minimizes the noise values. To do so, we change the integration time window during the correlated double sampling process, and evaluate the readout noise in each case. The readout noise is again the voltage fluctuation seen at the output determined by the standard deviation of the ADC readouts which are converted to a voltage using the gain of the readout chain. It should be noted that the gain will change as a function of integration time as shown in Equation 7.

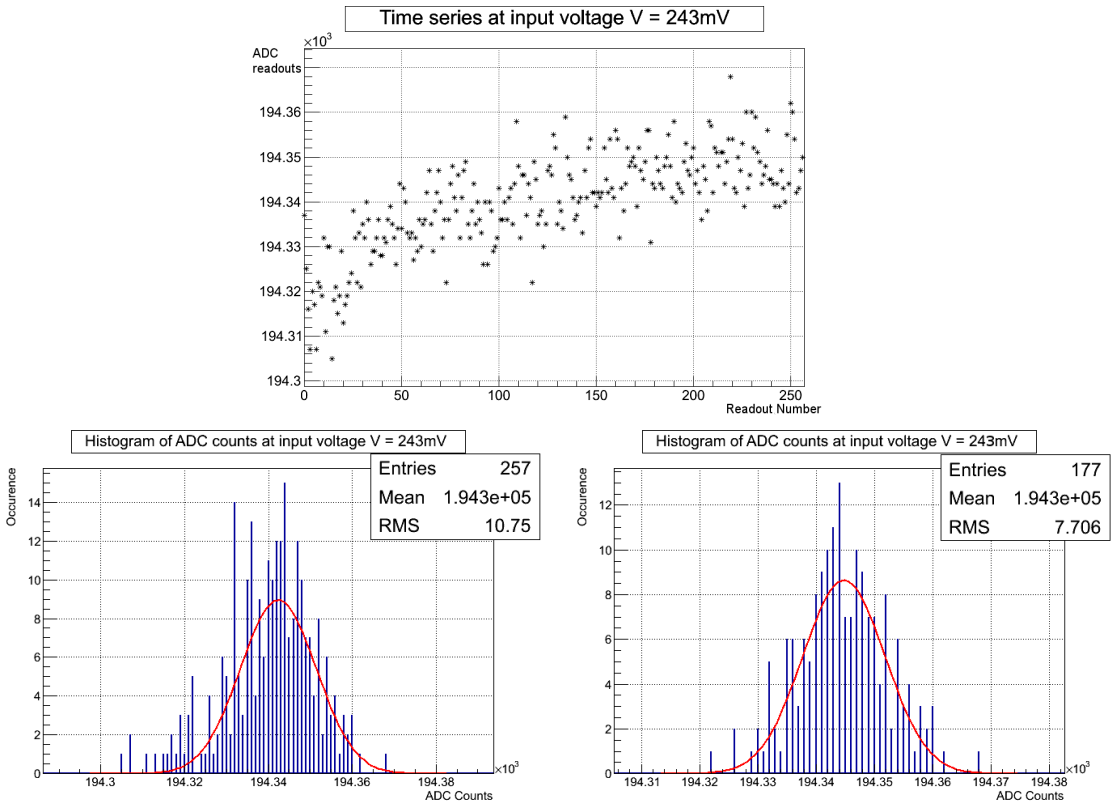

Figure 14: The top figure represents a time series of 257 ADC readouts showing an initial droop. The left histogram was obtained for the raw data, while the right one was obtained by removing the first 80 counts which is expected to minimize the effect of the droop. The left distribution seems to be more Gaussian.

Thus a new gain needs to be computed following the procedure of Section 6.1 for each integration time length Δt. The integration time was gradually increased by time steps of $200ns$ where 10 runs of 177 ADC readouts (257 readouts with the first 80 removed) were taken at each step. The mean and error of the noise voltage was obtained based on the method described in Appendix C. The frequency range was approximated by taking the inverse of the integration time. For example, an integration time of $\Delta t = 1\mu$s corresponds to a frequency of about $1MHz$. The results are shown in Figure 19.

The behavior of this curve can be understood when we refer back to the integration time window and noise frequency. Because of its high frequency, white noise is expected to approach zero by a simple integration of the signal. However, a small integration time window sets a lower limit on the frequency range that will be eliminated. An integration time $\Delta t = 200ns$, for example, corresponds to a lower frequency limit of $5MHz$ (all frequencies below $5MHz$ will not be eliminated). For this reason, we see a high noise level at high frequencies (i.e. small integration time window). Conversely, at a low frequency limit most of the white noise is filtered, however, a new type of noise emerges; the 1/f noise that is expected to be eliminated by correlated double sampling. As the frequency decreases, the Ramp Down and Ramp Up integration periods do not capture the same

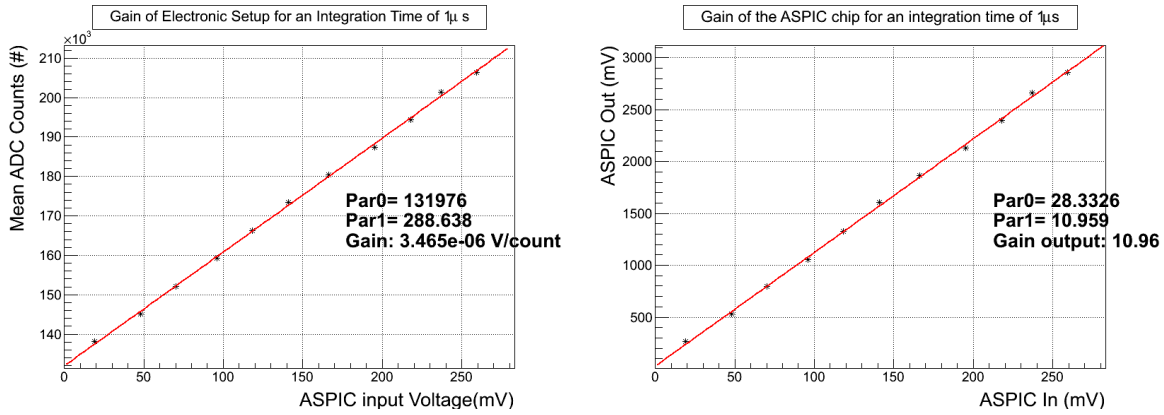

Figure 15: The graphs show the gain of a signal integrated for 1μs using channel 6 with a nominal ASPIC gain of 7.5. Left: The slope of the graph gives the overall gain of the electronic setup, i.e. voltage per unit ADC count. Right: The slope of the graph gives the gain of the ASPIC.

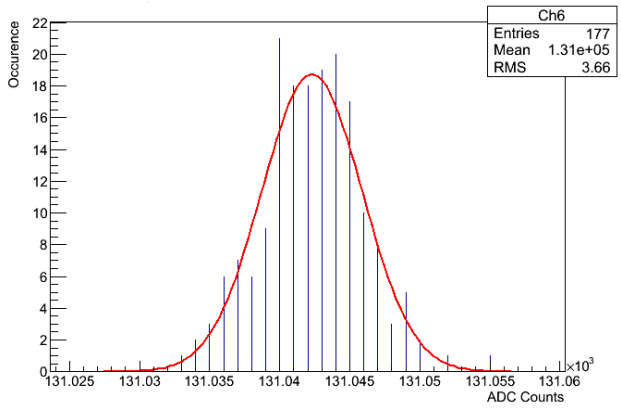

Figure 16: Histogram of 257 ADC values with no input voltage, fit by a Gaussian with mean 131000 and standard deviation 3.66. This represents a pedestal for the ADC readouts.

noise signal, which results in an incomplete cancellation of the noise. Equations 8 and 6 confirm the shape of our curves; the linear dependence of white noise and the 1/f dependence of flicker noise.

6.4 Crosstalk Correlation

Crosstalk refers to the effect that a signal passing through one channel has on adjacent channels. Since the FEB contains 8 channels that are compactly mounted next to each other through out the FEB and BEB circuits, we expect some signal interference from one channel to the next. In order to investigate this effect, we sent repeated pulses through one channel for several runs, averaged out the ADC outputs from all runs and subtracted them from the pedestal ADC counts obtained when no pulse was sent. The ADC outputs of each channel were averaged and divided by the ADC output of the pulsed channel. The sample used consisted of five runs with 512 ADC readouts (the hardware storage limitation of ADC values) for 8 channels taken for each pulsed channel and for

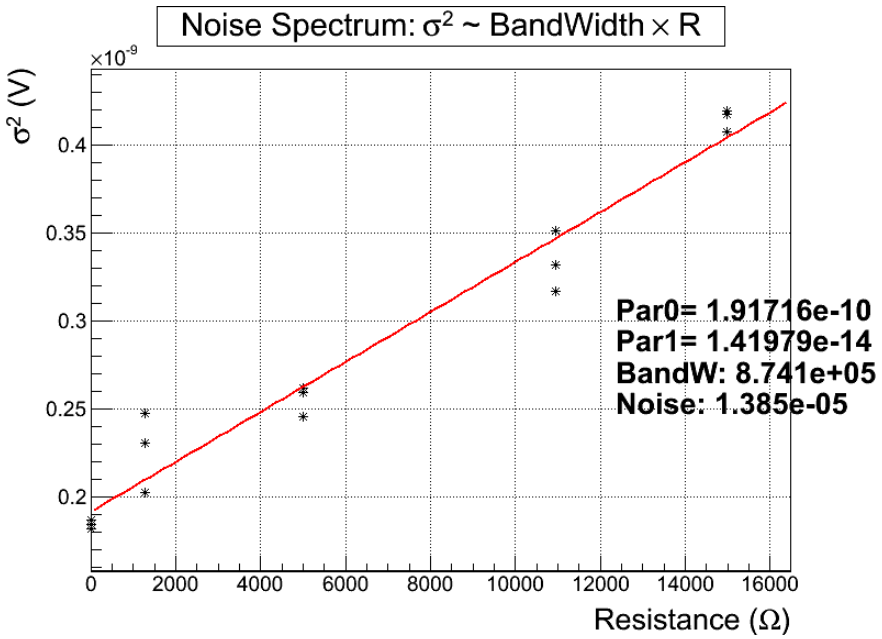

Figure 17: Noise Spectrum for an integration time of 1μs showing a bandwidth of 1 Mhz and a noise level of 14 μV

the pedestal readout. The result was an 8 × 8 matrix that represents the fraction of the input pulse that was transmitted to adjacent channels. The result is shown in Table 3 and plotted in Figure 20 as a percentage.

	Ch0	**Ch1**	**Ch2**	**Ch3**	**Ch4**	**Ch5**	**Ch6**	**Ch7**
Ch0 pulsed	1.000000	0.000224	0.000094	0.000037	0.000022	0.000035	-0.000028	0.000019
Ch1 pulsed	0.000349	1.000000	0.000098	0.000116	0.000071	-0.000007	0.000020	0.000015
Ch2 pulsed	0.000117	0.000229	1.000000	0.000280	0.000131	0.000110	0.000014	0.000052
Ch3 pulsed	-0.000375	0.000408	0.000666	1.000000	0.000266	0.000373	-0.000098	-0.000614
Ch4 pulsed	-0.000579	0.000658	0.001080	-0.000033	1.000000	0.001706	-0.000530	0.000257
Ch5 pulsed	0.000003	0.000091	0.000185	0.000142	0.001169	1.000000	0.000170	0.000199
Ch6 pulsed	-0.000048	0.000114	0.000207	0.000063	0.000152	0.000486	1.000000	0.000401
Ch7 pulsed	-0.000045	0.000123	0.000217	0.000046	0.000137	0.000308	0.000340	1.000000

Table 3: Channel-to-Channel interference for pulses sent through each channel. The numbers represent the fraction of the channel readout to the pulsed channel.

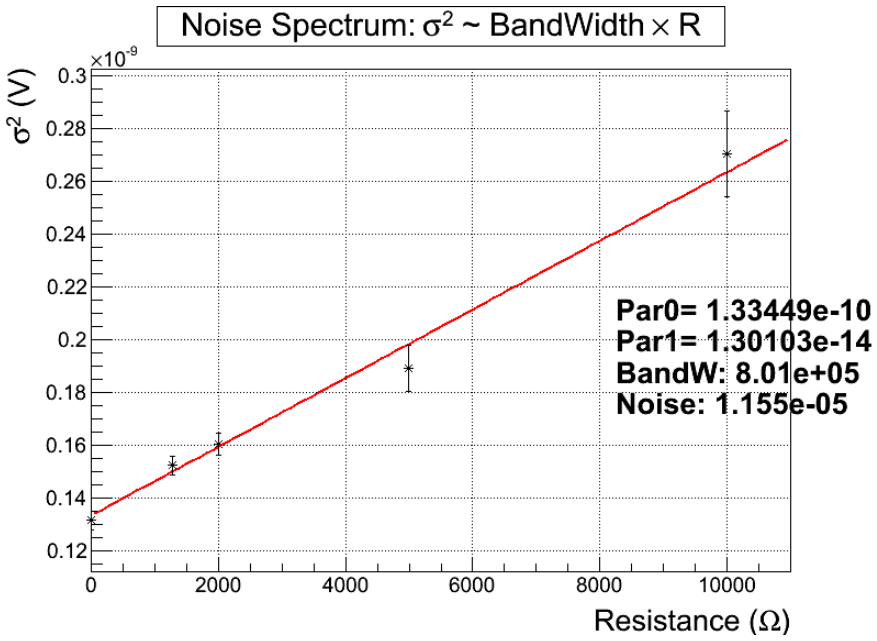

Figure 18: Noise Spectrum for an integration time of 1μs showing a bandwidth of .8 Mhz and a noise level of 11 μV

7 Summary of Tests

The tests of the prototype electronics of the LSST camera presented in the current study determined values for the gain, the readout noise, and the channel-to-channel interference. The readout noise refers to the random noise of a CCD output amplifier, as the one shown in Figure 2, that introduces a voltage fluctuation at the output superimposed with the CCD pixel signal. Determining the value of the readout noise is important in order to take it into account while reconstructing the charge contents of each pixel and thus the image. In order to measure this noise, the method used consisted of determining the standard deviation of the voltage fluctuations at the output, and by using the gain of the readout setup, the equivalent noise fluctuation at the input was found to be in the order of 11μV by using the noise spectrum generated by a white noise source at the input. The noise spectrum also provides with the bandwidth of voltage fluctuations to be in the order of 1*MHz*. Additionally, the noise frequency plot gives the integration time that minimizes the readout noise which is around 800*ns*. To match the total readout time of 2 seconds, the LSST camera will operate with an integration time of 500*ns* which is close to the minimum noise range. The CCD specifies the amplifier responsivity value to be 5.75μV/e$^-$, the conversion gives an equivalent of 2 electrons of readout noise in the current setup that consists of one FEB and BEB pair. One important parameter that affects noise is temperature where all our tests were done at room temperature ($T_{room} \sim$ 298K). It can be assumed that the square of noise term is proportional to T as given in equation (8). The noise value at the expected operational temperature of T = 173K is given by $\sqrt{\frac{T}{T_{room}}}\sigma$,which reduces the noise further. The values are summarized in Table 4

The noise levels achieved are below the allocated noise value of 4e$^-$ to the camera. However, assembling the complete raft may contribute to more readout noise in the system. The tests de-

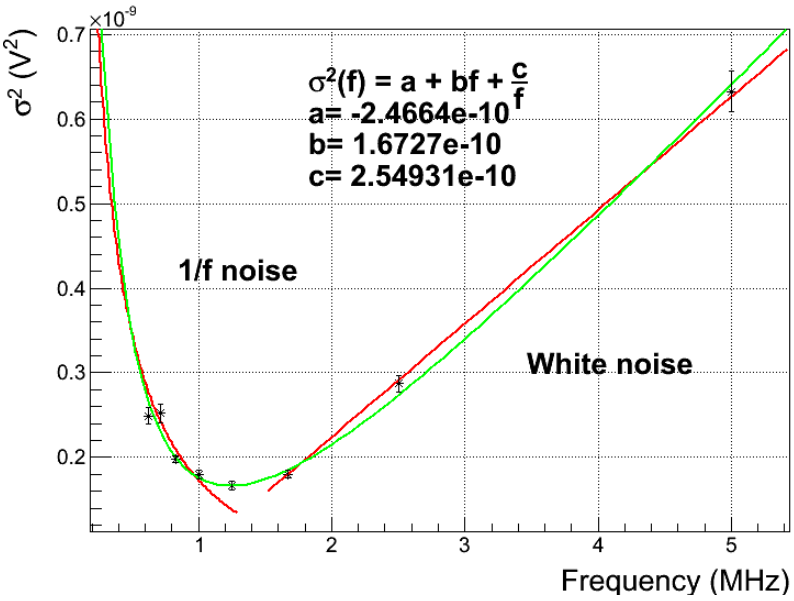

Figure 19: Noise as a function of frequency. The graph shows two behaviors; At low frequencies, 1/f noise dominates the noise response, while at high frequencies, white noise is dominant.

	Noise @ 296K (μV)	**Noise @ 173K (μV)**	**Noise @ 173K (electrons)**
Nominal Gain 5	13.85	10.58	1.84
Nominal Gain 7.5	11.55	8.83	1.54

Table 4: Noise measurements summary at room temperature and the corresponding expected noise values for the operating temperature T = 173K.

scribed in this study will have to be repeated when the full raft is assembled in order to test the noise levels again for an operating temperature of 173K.
The goal of the crosstalk levels are to be under 0.05% which were mostly met with few exceptions, such as pulsing channel 4, that needs to be studied separately in order to understand the crosstalk level.
A further test on the complete raft (6 FEB/BEB pairs) should be done in order to ensure that the LSST camera electronics meet the noise and readout speed requirements specified for it.

8 LSST Science

8.1 Overview

The scientific goals of LSST are to take an inventory of the Solar System, map the Milky Way, explore the transient optical sky, and essentially probe dark energy and dark matter. LSST is expected to operate over a period of 10 years to visit each patch of the sky 1000 times as part of the main survey which should cover 18000 deg^2 of the southern sky in all filters (u,g,r,i,z,y). Each night, LSST will take two pairs of 15 seconds exposures completely read by the camera in a given filter.

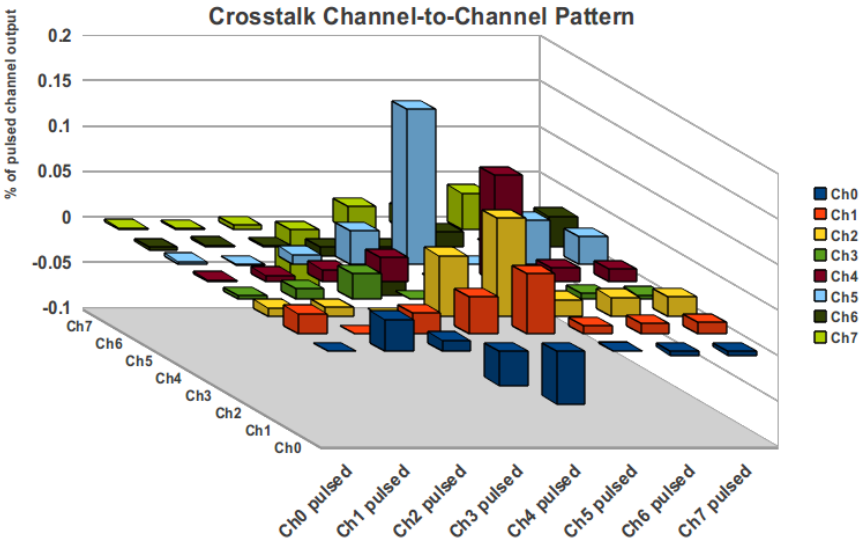

Figure 20: Channel-to-Channel interference

The survey will result in collecting images for millions of objects, producing some 30 terabytes of data each night. Based on the projects to be studied, the LSST science group has set some basic requirements on the telescope. One basic criteria is how dim the observed objects are. In AB magnitude, a standardized magnitude system that measures the brightness of objects in the sky, the depth of a single exposure should reach a magnitude of r $\sim$ 24.5. Over the ten years period of the survey, multiple images will be obtained for the same patch of the sky. By stacking all these images together, a depth of r $\sim$ 27.5 will be reached, which will aid in detecting very faint objects necessary for weak lensing studies. The image quality , with a seeing of about .7 arcseconds, should only be limited by the atmosphere of the observation site and to a minimum hardware systematics error. An improved image quality over existing surveys is required by all studies, especially the ones dealing with point sources. Additional observables include photometric repeatability that is important in determining redshifts and astrometric precision that gives good measurements of proper motion of stars and parallax.

The science requirements are directly dependent on the hardware characteristics developed for LSST. The noise study presented in this work is important in measuring the amount of light incident on the camera, which is directly correlated to the image quality among other observables.

Since a main mission of the LSST is to probe the distribution of dark matter in the universe, we give an example of how such study would benefit from the LSST's ability to extend the 'edge's of the observable universe.

One of the techniques that will be used consists of studying the deformation of images of distant luminous objects and the appearance of multiple images of the same object due to the presence of some mass distribution in the foreground. This later process is called Gravitational Lensing. In this section, we will discuss the importance of the LSST's resolution and depth in taking images of gravitationally lensed objects, such as quasars, in mapping the distribution of clumps of dark matter in the universe.

8.2 Gravitational Lensing

Gravitational lensing is the process by which light coming from a luminous object, referred to as the source, get deflected by a massive foreground object, referred to as the lens. We differentiate between two types of lensing; Strong lensing occurs when the source and the lens is almost perfectly aligned with the observer (earth) which results in the formation of multiple images of the source and rings due to the strong gravitational field encountered. This type of lensing is rare, however it provides detailed information about the source as well as the lens. For instance, the time delays between images and the flux ratios allow us to constrain the mass distribution which gives us indicators on the presence of dark matter. To date, only $\sim$ 100 strongly lensed quasars are known. LSST will increase this number by an order of magnitude, which would allow detailed studies of the distribution of dark matter in galaxies and clusters of galaxies.
The second type of lensing is weak lensing that slightly disturbs the initial random orientations of galaxy sources. This type is more common since most of the light that reaches us from distant objects gets perturbed with variable degrees by the gravitational wells of the lenses. However an increased number of images is required in order to provide with sufficient constraints on the distribution of dark matter. LSST images of r $\sim$ 27.5 with a 20000 deg^2 map coverage will be needed to provide with images of nearly 3 billions galaxies which should improve the statistics of weak lensing observations.

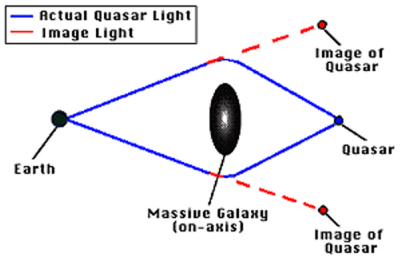

Figure 21: Strong doubly lensed quasar [1].

One of the strong lensing sources LSST will be looking for are quasars. Quasars are compact regions in the center of a massive galaxy surrounding a super-massive black hole that is accreting new material. This material creates an accretion disc that radiates due to frication as a multi-temperature blackbody in the soft X-ray, ultra-violet, and optical. The luminosity of this region often times outshines the whole galaxy making quasars easy to detect by most telescopes. LSST will enable identifying quasars through their time variability.

Gravitational lensing requires good spatial resolution of in order to differentiate between the various components of lenses. This will enable a good measurement of image positions, flux ratios, and time delays between images which will be described in the following sections.

8.2.1 Spatial Resolution

The current data available from operating telescopes features a small number of strong gravitationally lensed objects (30 quadruply imaged quasars, and about 200 lensed quasars known in total) with a limited quality. While the seeing sizes of these telescopes are comparable to the typical angular scales of strong lensing, their spatial resolutions are not enough to distinguish lensed images. Figure 22 shows the importance of spatial resolution by comparing two images of the same double lensed quasar that has a separation of 1.14". The first, taken by SDSS (J1332+0347) with a median seeing of 1.4", was not able to resolve the lens as well as the second image taken by Suprime-Cam on Subaru with seeing of .7", comparable to that of LSST. The improvement on spatial resolution permits the identification of more lenses with small image separations, many of which are expected to exist. It also provides with a better measurement of observables such as positions and brightness of lensed images and lensing galaxies.

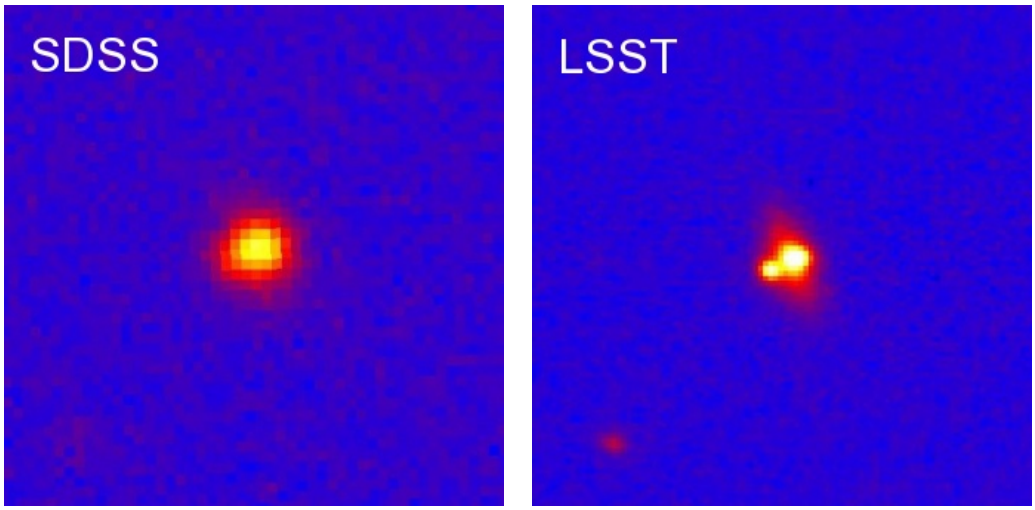

Figure 22: Gravitationally double lensed quasar J1332+0347 taken by SDSS with seeing of 1.4"(left), and Suprim-Cam (right) with seeing of .7" comparable to that of LSST[2].

8.2.2 Flux Ratios

Gravitational lensing can be a good probe for dark matter substructure in the foreground lensing galaxy. Based on the disturbing potential of the lens, we distinguish between three different lensing scales; First, macrolensing ($\sim$ 1 arcseconds) is due to the overall mass distribution of the lens; Millilensing ($\sim 10^{-3}$ arcseconds) results from the dark matter substructure perturbations. Last, microlensing ($\sim 10^{-6}$ arcseconds) is induced by stars sweeping across the lensed image.
Strongly lensed quasars come into two and four magnified images. The position of the quasar in the projected plane of the lens determines the number of lensed images as well as their locations. Four imaged quasars are of a particular interest because they offer more constraints on the gravitational potential of the lens. In the case of four lensed quasars, we expect certain relations between the brightness of images if there is no substructure that would disrupt the image fluxes. Figure 23 shows two examples of four-image lenses A, B, C, and D. In panel 23a the lensing galaxy is represented as the flux E, and the sum of fluxes A and C are expected to be equal to B. However, the figure

clearly shows that B is dimmer than the sum of the two. In 23b we expect B to be as bright as A which is contrary to our observation. These two examples illustrate the flux "anomaly" problem where our predictions of fluxes based on a smooth lens model (a galaxy mass distribution model without subhalos) do not agree with our observations.

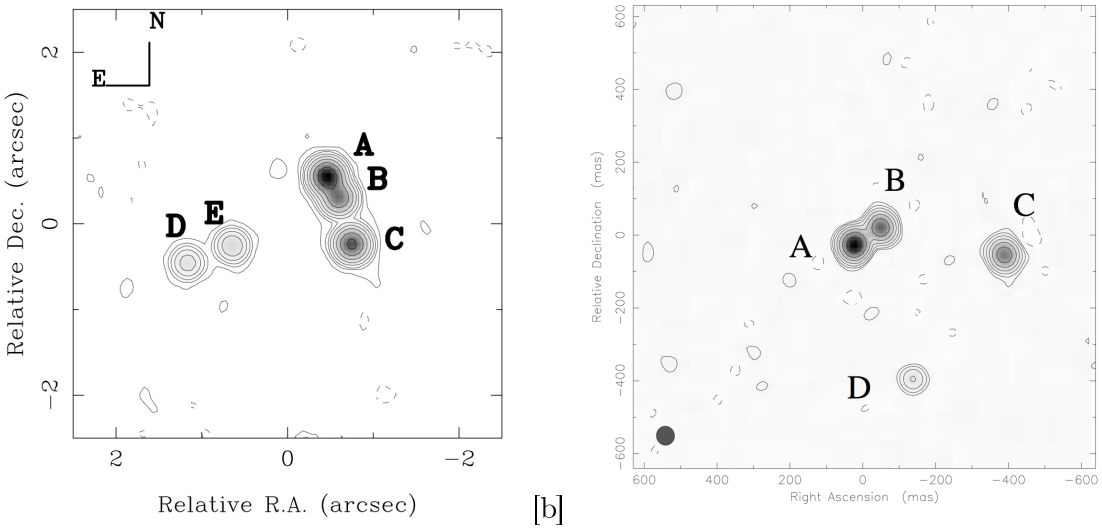

Figure 23: Four-image gravitational lenses. [a] Quasar B2045+265 with images A-D and E the lensing galaxy with expected flux $A + C \sim B$ [7]. [b] CLASS B1555+375 with images A-D with expected flux $A - B \sim 0$ [6]. The lensing galaxy is too faint to appear in [b].

This suggested that the smooth lens models was perturbed by some additional substructure that gave additional magnification to the lens images. These anomalies could be due to subhalo perturbations just as they could be due to microlensing by stars or propagation effects of light (absorption, scattering, or scintillation) by the interstellar medium of the lens. To rule out some of these hypotheses, microlensing in the optical range should be observed over time since it is expected to vary while millilensing should stay constant, or look into the microlensing-free range of mid-infrared or radio range. The later has been used because of limitations in the optical. The results of the four-image radio-flux ratios showed that the 'anomalies' are due to subhalos in the lens galaxy which agrees with the predictions of Cold Dark Matter (CDM) theory and provides some important tests to it. LSST is expected to measure microlensing well enough in the optical to remove its effects from the four-image lenses and correct for the flux ratios. An improved measurement of millilensing via image flux ratios, positions, and time delays as well as an increased lens sample will present good constraints in the study of CDM substructure.

8.2.3 Time Delays

Light beams coming from a strongly lensed source pass through different gravitational potentials that affect the timing of the beam due to relativistic effects (clocks running at a slower rate in gravitational fields) and the length of the photons path (deflection results in a longer path). The light beams reach an observer at different times, thus engendering time delays between the lensed images. This effect is especially observable for nontransient light sources that display flux variability over

time such as quasars. Time delays can give us useful information on cosmological parameters and the lensing mass distribution. The time delay between two images A and B, $\Delta t_{A,B}$, is proportional to the comoving distance (or "time delay distance") that separates the two images, D, and a factor related to the gravitational potential of the lensing mass, $f_\Omega(\psi)$. Since the comoving distance is related to the Hubble constant by $D \propto H_0^{-1}$, we obtain a relation between the time delay, the Hubble constant, and a factor dependent on the lensing potential;

$$\Delta t_{A,B} \propto \frac{f_\Omega(\psi)}{H_0}$$

By constraining the gravitational potential ψ by a cosmological model Ω, and obtaining measurements of the time delay, a value for the Hubble constant can be obtained based on the cosmological model chosen. On the other hand, if the Hubble constant and time delays of the images are determined independently, information about the gravitational lensing potential can be inferred. In this respect, accurate time delays in the order of hours and upwards will provide with better constraints on the lens potential (ψ) and the Hubble constant(H_0) can also be used to probe the CDM subhalos of the lens galaxy. The time delay of an image perturbed by a subhalo will show an anomaly compared to the time delay predictions of a smooth lens model. The advantage of time delays is that they present a clear signature of the existence of subhalos that cannot be mimicked by dust or stars as in the case of flux ratios. Time delays perturbations due to subhalos are in the order of a fraction of a day, which again requires an accurate measurement down to this limit to ensure constraints on CDM subhalo population. Figure 24 shows a macrolensed quasar where one of its images is further millilensed by a subhalo. The flux of the second image shows a time lag between the two image splittings.

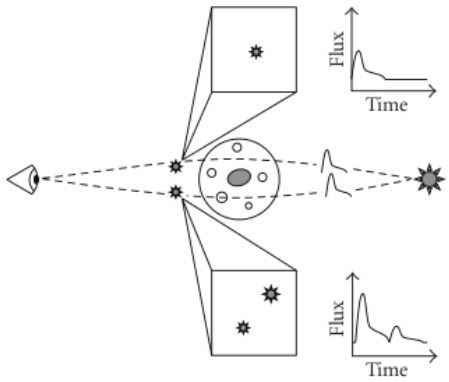

Figure 24: Light beams of a variable source (i.e. quasar) passing by a galaxy with subhalos. The upper image is a macrolensed image with normal flux variation. The lower macrolensed image experiences a small scale image splitting due to the subhalo in its path. A time lag between the small image splittings is observed[5].

9 Conclusion

LSST is expected to see about 100 high quality time delay lenses for the cosmographic study. Currently, 20 lensed quasars have measured time delays ($\Delta t \sim 0.1 - 400$ days). LSST is anticipated

to find 2600 quasars with well measured time delays[2]. These accurate measurements will rely on LSST fast and precise instrumentation. The visits depth should reach r $\sim$ 24.5 (5σ, point source) which is one magnitude improvement over the current operating telescopes. Finally, the hardware used should maintain a good image quality and astrometrics so that the only constraints would be set by the atmosphere which imposes limits on the readout time, the slew time, and the read noise.

The current work that is being performed by the LSST's camera electronics development groups is geared towards matching the science requirements of LSST that will start providing the astronomical community with high quality images of the night sky by 2015.

Acknowledgment

I would like to express my gratitude to professor Gordon Richards, my advisor from Drexel, Mitch Newcomer and Rick Van Berg, my advisors from the University of Pennsylvania, who were patient with my questions and provided insightful feedback. I would like to thank my lab partner Isaac Pedisich for being helpful and open to any questions. I also would like to thank Mike Reilly and Godwin Meyers for setting up the readout electronics and helping us troubleshoot problems and taking measurements. John Alison saved me a lot of time and effort by helping to write the ROOT codes in Python. I would like to thank Ross Fadely for few discussions we had about gravitational lensing and dark matter.
Finally, I would like to thank the faculty and staff of the the physics department at Drexel University for their support and help during the three years I spent in the department.

References

[1] Goddard Space Flight Center. Gravitational lensing. `http://imagine.gsfc.nasa.gov/docs/features/news/grav_lens.html`, April 2007.

[2] LSST Science Collaborations. *LSST Science Book*. LSST Corporation, 2009.

[3] LSST Corporation. Camera subsystem. `http://www.lsst.org/lsst/science/concept_camera`, 2011.

[4] e2v technologies Limited 2007. CCD231-84 BI NIMO Data Sheet, July 2007.

[5] Teresa Riehm Erik Zackrisson. Gravitational lensing as a probe of cold dark matter subhalos. *Advances in Astronomy*, 2010(478910):14, 2010.

[6] C. D. et al. Fassnacht. B2045+265: A new four-image gravitational lens from class. *The Astronomical Journal*, 117(2):658–670, 02 1999.

[7] D. R. et al. Marlow. Class b1555+375: A new four-image gravitational lens system. *The Astronomical Journal*, 118(2):654–658, 08 1999.

[8] Paul R. Gray and Robert G. Meyer. *Analysis and Design of Analog Integrated Circuits*. Wiley, second edition, 1984.

[9] J. B. Johnson. Thermal agitation of electricity in conductors. *Physical Review*, 32:97–109, July 1928.

[10] C. Juramy and J. Jeglot. Aspic 2 testing report. Technical report, IN2P3 LSST electronics group, December 2010.

[11] H. Nyquist. Thermal agitation of electric charge in conductors. *Physical Review*, 32:110–113, July 1928.

[12] John Olivier. Development of a 3.5 Gigapixel Camera for the Large Synoptic Survey Telescope (LSST). Technical report, Harvard Univeristy, March 2005.

Appendix

Appendix A: ROOT Interface

Part of my work with the University of Pennsylvania's instrumentation group was to develop analysis tools based on the high energy analysis interface ROOT. All plots and fits presented in this thesis were done using ROOT in combination with Python. The following is a description of the codes used to compute the gain, the noise spectrum, etc...
The following libraries were imported to Python:

```
 from ROOT import TH1F, TF1, TGraph, TCanvas,TLatex
import string, array, sys
```

ROOT daq

The goal of the program is to compute the mean and standard deviation of a channel dataset read-out from a series of files. An initial file is read which contains the filenames of the desired data files. Each file is opened at the time and read in order to determine the mean and std. dev. of a channel selected by a user input. The mean and std. dev. values are determined via three different methods:

Computed: From the basic definitions of mean and standard deviation of a sample.
ROOT: ROOT gives a value of the mean and RMS values of the dataset.
Gaussian: Fit a Gaussian to the histogram and determine the mean and std. dev. from it. This method has been used for our tests.

An input data filename has the following form:
20120329T115633.data
20120329T115637.data
20120329T115702.data

An input data file from this file has the following form:

B0Ch0	**B0Ch1**	**B0Ch2**	**B0Ch3**	**B0Ch4**	**B0Ch5**	**B0Ch6**	**B0Ch7**
131087	131036	131047	131020	131063	131038	131037	134411
131114	131067	131070	131048	131088	131062	131050	134463
131098	131056	131064	131036	131077	131050	131045	135593
...							

In the command line, enter: python root_daq.py <arg1> <arg2> <arg3><arg4> <arg5>
arg1: <folder path>,default folder path is: /home/HEP/LsstDaq
arg2: <input_files>, file containing filenames
arg3: <channel_number>, select a channel number
arg4: <output_type>
r for ROOT Mean and RMS values in output
g for Gaussian fit
c for the computed value of the mean and stdev based on the whole data set

arg5: <filename_out>, the output filename

Example:

```
python root_daq.py GainNoisedata/Noise_data5 noisech6filesall.dat 6 g noisech6gauss.dat
```

```
# ROOT analysis program using a Python interface: root_daq.py
# Command Line Arguments
arguments = sys.argv

if len(sys.argv) > 1:
        folder = sys.argv[1]
if len(sys.argv) > 2:
        input_files = sys.argv[2]
if len(sys.argv) > 3:
        channel_num = int(sys.argv[3])
if len(sys.argv) > 4:
        output_type = sys.argv[4]
if len(sys.argv) > 5:
        filename_out = sys.argv[5]
if len(sys.argv) > 6:
        start = int(sys.argv[6])


file_num = 0 # number of files

# Read in Filenames
source = open("/home/HEP/LsstDaq/%(folder)s/%(input_files)s"%vars(), 'r')
for file in source.readlines():
    filename = map(str,string.split(file))
    # Open Files one by one
    for x in filename:
        sum1 = 0
        sum2sq = 0
        i=0
        channel = []
        ind = []
        print "\n", x    # Print filename
        f = open('/home/HEP/LsstDaq/data/%(x)s'%vars(), 'r')
        f.readline()    # Skip one line (ch0 ch1 ...)
        if start > 0:
            for i in range(0,start):
                f.readline() # Skip start lines
            print "Skipping ", start, " lines"
        for line in f.readlines():
            ch = map(int,string.split(line)) # List of one data output row
            channel.append(ch[channel_num]) # Select desired channel
            ind.append(i)
            sum1 = sum1 + ch[channel_num] # For mean
            sum2sq = sum2sq + ch[channel_num]*ch[channel_num] # For std.
```

```
        numEntries = len(channel)

        mean = sum1/(numEntries)
        rms = sum2sq/(numEntries)
        stdev = 0
        var = 0

        for i in range(0,numEntries):
            var = var + pow(channel[i] - mean, 2)
        var = var/(numEntries)
        stdev = sqrt(var)

        # Converting lists to arrays:
        indarr = array.array("d",ind)
        channelarr = array.array("d",channel)

        # Construct a Historgrm with 4 sigma
        NBINS = 262144
        myHist = TH1F("Channel", "Histogram of Channel", NBINS,\
mean -4*stdev, mean +4*stdev)
        for i in range(0,numEntries): myHist.Fill(channel[i])

        # Histogram Mean, Std. dev., RMS:

        # ROOT:
        meanh = myHist.GetMean()
        RMSh =  myHist.GetRMS()
        print "Histogram ROOT:"
        print "Mean R: "  + repr(meanh) + '  RMS:' + repr(RMSh)

        # Computed:
        print "Computed: "
        print "Mean: "  + repr(mean) + '  Stand. Dev.:' + repr(stdev) +\
'  RMS:' + repr(rms)

        # Gaussian Fit:
        # create fit funciton, gaus, to 5 sigmas
        lowh = mean-5*stdev
        highh = mean+5*stdev
        gfit = TF1("Gaussian","gaus", lowh, highh)
        myHist.Fit("Gaussian","RQ")
        # collect parameters
        amp = gfit.GetParameter(0)
        eamp = gfit.GetParError(0)
        meanf = gfit.GetParameter(1)
        emean = gfit.GetParError(1)
        stdevf = gfit.GetParameter(2)
        estandev = gfit.GetParError(2)
        FW2 = gfit.GetX(amp/2,meanf,meanf+stdevf)
        FW1 = gfit.GetX(amp/2,meanf-stdevf,meanf)
```

```
        FWHM =.5*(FW2-FW1) # full width at half maximum
        print "Gaussian Fit: "
        print "Mean: %.3f \t Stand. Dev.: %.3f \t Amplitude: %.3f" %(meanf,stdevf,amp)
        print "FWHM: %.3f" %FWHM

        # Write in file:
        w = open("/home/HEP/LsstDaq/%(folder)s/%(filename_out)s"%vars(), 'a')
        # output type
        if output_type == 'g':
                meanx = meanf
                stdevx = stdevf
        else if output_type == 'r':
                meanx = meanh
                stdevx = RMSh
        else output_type == 'c':
                meanx = mean
                stdevx = stdev
        w.write('%(x)s  %(meanx)f  %(stdevx)f\n'%vars())
        file_num = file_num + 1
        print filename_out + " written"
print "Number of files: %d" %file_num
```

Gain

The goal of this program is to compute the gain of the overall setup and the ASPIC chip based on a set of data of voltage input and mean ADC counts having the form:

vol_in	vol_out	filename	Mean	std.
4.4	265.1	20120329T124100.data	138079.822958	4.023337
46.8	535.1	20120329T124147.data	145118.954547	5.505548
72	808.2	20120329T124200.data	152164.428083	4.305813
96	1080	20120329T124317.data	159257.761916	4.764126
120	1322	20120329T124333.data	166287.464997	5.729193

In the command line, enter: python compute_gain.py <arg1> <arg2> <arg3> <arg4> <arg5><arg6>
arg1: <folder path>,default folder path is: /home/HEP/LsstDaq
arg2: <filename>, desired file
arg3: <output_file>, output graph (*.png, *.jpg, *.eps)
arg4: <vol_col>, voltage input column (usually 0)
arg5: <out_col>, output voltage measured of the ASPIC
arg6: <mean_col>, mean counts column (usually 2)

Example:

```
python compute_gain_system.py GainNoisedata/Gaindata7 gain7rampgaus.dat gain7ch6.png 0 2 1
```

```
# ROOT analysis program using a Python interface: compute_gain_system.py
# Command Line Arguments
if len(sys.argv) > 1:
        folder = sys.argv[1]
if len(sys.argv) > 2:
        filename = sys.argv[2]
if len(sys.argv) > 3:
        output_file = sys.argv[3]
if len(sys.argv) > 4:
        vol_col = int(sys.argv[4])
if len(sys.argv) > 5:
        out_col = int(sys.argv[5])
if len(sys.argv) > 6:
        mean_col = int(sys.argv[6])


# Initialize variables
mean = []
voltage = []
output = []

# Open file
f = open('/home/HEP/LsstDaq/%(folder)s/%(filename)s'%vars(),'r')
for line in f.readlines():
    file = map(str,string.split(line))
    mean.append(float(file[mean_col]))
    voltage.append(float(file[vol_col])) # Voltage in
    output.append(float(file[out_col]))  # Voltage out


numEntries = len(mean)
# Lists to arrays
meanarr = array.array("f",mean)
voltagearr = array.array("f",voltage)
outputarr = array.array("f",output)

# Graph of overall gain
c1 = TCanvas("c1","",800,600)
func = TGraph(numEntries, meanarr, voltagearr)
func.SetTitle("Gain of Electronic Setup for an Integration Time of 1#mus")
func.GetXaxis().SetTitle("Mean Counts (#)")
func.GetXaxis().SetTitleSize(.05)
func.GetYaxis().SetTitle("Voltage (mV)")
func.GetYaxis().SetTitleSize(.05)
func.Draw("AP*")
# Fit function to a polynomial of order 1
func.Fit("pol1")
fit = func.GetFunction('pol1')
# Parameters
par0 = fit.GetParameter(0)
par1 = fit.GetParameter(1)
```

```
# Add labels to graph
lat = TLatex()
lat.SetNDC()
lat.DrawText(0.6,0.5,"Par0= %g"% par0)
lat.DrawText(0.6,0.45,"Par1= %g"% par1)
lat.DrawText(0.6,0.4,"Gain: %.4g V/count" % float(par1*1e-3))
c1.Update()
# Save Graph
c1.SaveAs("/home/HEP/LsstDaq/%(folder)s/gain.png"%vars())

# Graph of overall gain
c2 = TCanvas("c2","",800,600)
func2 = TGraph(numEntries, voltagearr, outputarr)
func2.SetTitle("Gain of the ASPIC chip for an integration time of 1#mus")
func2.GetXaxis().SetTitle("ASPIC In (mV)")
func2.GetXaxis().SetTitleSize(.05)
func2.GetYaxis().SetTitle("ASPIC Out (mV)")
func2.GetYaxis().SetTitleSize(.05)
func2.Draw("AP*")
# Fit function to a polynomial of order 1
func2.Fit("pol1")
fit2 = func2.GetFunction('pol1')
# Parameters
par0 = fit2.GetParameter(0)
par1 = fit2.GetParameter(1)
# Add labels to graph
lat = TLatex()
lat.SetNDC()
lat.DrawText(0.7,0.5,"Par0= %g"% par0)
lat.DrawText(0.7,0.45,"Par1= %g"% par1)
lat.DrawText(0.7,0.4,"Gain output: %.4g" % float(par1))
c2.Update()
# Save Graph
c2.SaveAs("/home/HEP/LsstDaq/%(folder)s/gain_aspic.png"%vars())
raw_input('Press Return key to quit: ')
```

Noise

The goal is to create a noise spectrum by plotting voltage fluctuations (std. dev.) as a function of resistance to estimate the noise and bandwidth. This program can be used to compute the noise by giving the datafile containing the resistance values and the std. dev. of the voltage. The input data file has the form shown below.

In the command line, enter: python compute_noise.py <arg1> <arg2> <arg3> <arg4> <arg5><arg6> <arg7>
arg1: <folder>, default folder path is: /home/HEP/LsstDaq
arg2: <filename>, desired file
arg3: <output_file>, output graph (*.png, *.jpg, *.eps)

R	filename	Mean	std.
0	20120329T115625.data	131040.952212	3.372669
0	20120329T115630.data	131040.847304	3.177026
0	20120329T115633.data	131041.054013	3.647218
...			
1995	20120329T120006.data	131025.291146	3.629866
1995	20120329T120028.data	131025.061189	3.599267
1995	20120329T120036.data	131025.479100	3.638749
...			

arg4: <resistance column>, resistance column (usually 0)
arg5: <Std. Dev. column>, Std. Dev. column (usually 3)
arg6: <Temp.>, Temperature for 4kT value
arg7: <Gain>, Gain of the system for conversion

Example:

```
python compute_noise_gauss.py GainNoisedata/noise7 noisech6gaus.dat noisech6error.png 0 3 70 3.462e-6
```

```
# ROOT analysis program using a Python interface: compute_noise.py
# Default Values for T and gain
k_b = 1.38e-23
T = 297.444
gain = 4.647e-06

# Command Line Arguments
if len(sys.argv) > 1:
        folder = sys.argv[1]  # Folder Containing File
if len(sys.argv) > 2:
        filename = sys.argv[2]
if len(sys.argv) > 3:
        output_file = sys.argv[3]
if len(sys.argv) > 4:
        r_col = int(sys.argv[4])
if len(sys.argv) > 5:
        std_col = int(sys.argv[5])
if len(sys.argv) > 6:
        T = (float(sys.argv[6])+459.67)*5/9.
if len(sys.argv) > 7:
        gain = float(sys.argv[7])

# Initialize
mean = []
std = []
var = []
resistance = []

# Gain used:
```

```
print "Gain: " + repr(gain)

# Open file with resistance values and std. devs:
f = open('/home/HEP/%(folder)s/%(filename)s'%vars(),'r')
for line in f.readlines():
    file = map(str,string.split(line))
# Store variances converted to square voltage
    var.append(pow(float(file[std_col])*gain,2))
# Store resistance values
    resistance.append(float(file[r_col]))

numEntries = len(var)

# Lists to arrays
vararr = array.array("f",var)
resistancearr = array.array("f",resistance)

# Create Graph:
c1 = TCanvas("c1","",800,600)
func = TGraph(numEntries, resistancearr,vararr)
func.SetTitle("Noise Spectrum: #sigma^{2} ~ BandWidth #times R")
func.GetXaxis().SetTitle("Resistance (#Omega)")
func.GetXaxis().SetTitleSize(.05)
func.GetYaxis().SetTitle("#sigma^{2} (V)")
func.GetYaxis().SetTitleSize(.05)
func.Draw("AP*")
c1.Update()
# Fit the function:
func.Fit('pol1')
fit = func.GetFunction('pol1')
# Get the parameters
par0 = fit.GetParameter(0)  # Slope
par1 = fit.GetParameter(1)  # Y-intercept
# Show in the graph
lat = TLatex()
lat.SetNDC()
lat.DrawText(0.7,0.5,"Par0= %g"% par0)
lat.DrawText(0.7,0.45,"Par1= %g"% par1)
lat.DrawText(0.7,0.4,"BandW: %.4g" % float(par1/(4*k_b*T))) # Bandwidth
lat.DrawText(0.7,0.35,"Noise: %.4g" % pow(par0,.5)) # Noise value

# Print results
print "Bandwidth: %.4g" % float(par1/(4*k_b*T))
print "Other noise: %.4g" %pow(par0,.5)

c1.Update()

# Save the output
c1.SaveAs("/home/HEP/LsstDaq/%(folder)s/%(output_file)s"%vars())
raw_input('Press Return key to quit: ')
```

Integration time

This program is a complement for compute_gain_ramp.py to create a frequency dependent noise graph by obtaining multiple (4) noise voltage fluctuation corresponding to each integration time frame. The program reads in a file containing the time window, gain, and std dev. of ADC counts. The format of the file is:

Dt	Gain	filename	Mean	std.
200	0.016762	20120416T111559.data	131051.720849	1.450839
200	0.016762	20120416T111604.data	131051.553509	1.402020
200	0.016762	20120416T111607.data	131051.602783	1.545993
400	0.008590	20120416T111702.data	131048.213196	1.973611
400	0.008590	20120416T111706.data	131047.883970	1.964095
400	0.008590	20120416T111709.data	131048.117817	2.042898

The corresponding gain is used for each noise spectrum evaluation.

In the command line, enter: python compute_gain.py <arg1> <arg2> <arg3> <arg4> <arg5>
arg1: <folder path>,default folder path is: /home/HEP/LsstDaq
arg2: <filename>, desired file
arg3: <output_file>, output graph (*.png, *.jpg, *.eps)
arg4: <time_col>, integration time width (usually 0)
arg5: <gain_col>, gain for each integration time (usually 1)
arg6: <std_col>, std dev in ADC counts (usually 4)
arg7: <COUNT>, the number of data reads at each integration time (10)
arg8: <TRIALS>, the number of gains to compute (example 8)

Example:

```
python compute_noise_ramp.py GainNoisedata/noiseramp2 set2 sigmafreq.png 0 1 4 10 8
```

```
# ROOT analysis program using a Python interface: compute_noise_ramp.py
# Default
TRIALS = 8

# Command Line Arguments
if len(sys.argv) > 1:
        folder = sys.argv[1]
if len(sys.argv) > 2:
        filename = sys.argv[2]
if len(sys.argv) > 3:
        output_file = sys.argv[3]
if len(sys.argv) > 4:
        time_col = int(sys.argv[4])
if len(sys.argv) > 5:
        gain_col = int(sys.argv[5])
```

```
if len(sys.argv) > 6:
        std_col = int(sys.argv[6])
if len(sys.argv) > 7:
        COUNT = int(sys.argv[7])
if len(sys.argv) > 8:
        TRIALS = int(sys.argv[8])

# Initialize variables
value = ""
mean = []
std = []
var = []
time = []
xvar = []
xtime = []
etime = []
varf = []
evar = []

# Open file to read DT, gain, std. dev
f = open('/home/HEP/LsstDaq/%(folder)s/%(filename)s'%vars(),'r')
for line in f.readlines():
    file = map(str,string.split(line))
    DT = float(file[time_col])
    gain = float(file[gain_col])
    xstd = float(file[std_col])*(gain*1e-3)
    xvar.append(pow(xstd,2))
    xtime.append(DT)

# Compute the mean, and error of the of noise at each integration time
for i in range(0,TRIALS):
    sum1 = 0
    sum2sq = 0
    for j in range(i*COUNT,(i+1)*COUNT):
        var.append(xvar[j])
        sum2sq = sum2sq + pow(xvar[j], 2)
        sum1 = sum1 + xvar[j]
    time.append(1/xtime[j]*1e3)
    varh = sum1/float(COUNT)
    varf.append(varh)
    evar.append(sqrt((sum2sq/COUNT - varh*varh)/COUNT))
    etime.append(0.)

# Lists to arrays
vararr = array.array("f",varf)
evararr = array.array("f",evar)
timearr = array.array("f",time)
etimearr = array.array("f",etime)
numEntries = len(time)

# Graph points
```

```
c1 = TCanvas("c1","",800,600)
func = TGraphErrors(numEntries, timearr, vararr,etimearr,evararr)
func.SetTitle("")
func.GetXaxis().SetTitle("Frequency (MHz)")
func.GetXaxis().SetTitleSize(.05)
func.GetYaxis().SetTitle("#sigma^{2} (V)")
func.GetYaxis().SetTitleSize(.05)
func.Draw("AP*")
c1.Update()

# Fit two curves to each part of the graph, then an overall curve
par = []

# Define fitting functions:
func1 = TF1("fit1","[0]+[1]*1/pow(x,2)") # 1/f noise
func2 = TF1("fit2","pol1") # white noise
total = TF1("total","[0]+[1]*1/x+pol1(2)")
total.SetLineColor(3) # different color for the total

# Setting initial parameters for the fitting functions
func1.SetParName(0,"a")
func1.SetParName(1,"b")
func1.SetParameter(0,-2e-6)
func1.SetParameter(1,1e-4)

func2.SetParName(0,"a")
func2.SetParName(1,"c")
func2.SetParameter(0,1e-6)
func2.SetParameter(1,1e-9)

total.SetParameter(0,-1e-6)
total.SetParameter(1,1e-4)
total.SetParameter(2,-1e-6)
total.SetParameter(3,1e-8)
total.SetParName(0,"a")
total.SetParName(0,"b")
total.SetParName(0,"a")
total.SetParName(0,"c")

# Fit the functions for appropriate functions and intervals
func.Fit(func1,"R","",0,1.4)
func.Fit(func2,"R+","",1.4,6)
func.Fit(total,"R+","",0,6)

# Collecting the fitted parameters
par1 = func1.GetParameters() # list of paramters (2)
par2 = func2.GetParameters()
part = total.GetParameters()

# Drawing labels in the graph
```

```
lat = TLatex()
lat.SetNDC()
lat.DrawLatex(0.4,0.80,"f_{total}(T) = a + b#dotf + #frac{c}{f}")
lat.DrawText(0.4,0.75,"a= %g"% part[0])
lat.DrawText(0.4,0.70,"b= %g"% part[1])
lat.DrawText(0.4,0.65,"c= %g"% part[3])
lat.DrawText(0.2,0.55,"1/f noise")
lat.DrawText(0.6,0.3,"White noise")

# Save Graph
value = '_fit'
c1.Update()
c1.SaveAs("/home/HEP/LsstDaq/%(folder)s/%(output_file)s%(value)s.png"%vars())
raw_input('Press Return key to quit: ')
```

Appendix B: Nyquist Theorem

The formulation of the Nyquist theorem appeared in H. Nyquist 1928 paper[11] in order to explain J. B. Johnson's discovery of thermal noise [9]. A version of the argument is described here based on a statistical mechanics result along with some basic electronics [4].

The statement of the theorem:

$$\overline{v^2} = 4kTR\Delta f.$$

The current generated on a circuit with a resistance load R and R', shown in the figure below, is P = $\frac{V}{R+R'}$.

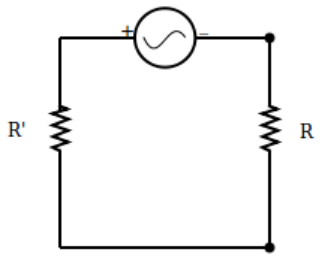

Figure 25: A noise generator with a resistance R delivers power to a load R′.

The power dissipated across the matching load R' is P = $\overline{i^2}R'$. The power exchange between the two resistors R and R' is,

$$\mathrm{P} = \frac{\overline{v^2}R'}{(R+R')^2}$$

[4] A discussion of thermal noise appears in *Thermal Physics* by C. Kittel

The power is maximum when R = R' [5]. The thermal noise power per unit frequency is

$$\mathrm{p} = \frac{\overline{v^2}}{4R} \tag{10}$$

From statistical mechanics, the thermal energy spectrum in a solid is given by the Planck distribution function. The energy at each state s, with an oscillation frequency ω comes into packets of energy ϵ_s per unit frequency,

$$\epsilon_s = s\hbar\omega$$

Thus the total energy per mode is found to be,

$$\overline{\epsilon} = \frac{\hbar\omega}{\exp\left(\frac{\hbar\omega}{kT}\right) - 1}$$

In the classical limit where most experimental values lie; $\hbar\omega << kT$. By using a Taylor expansion of $\exp\left(\frac{\hbar\omega}{kT}\right)$, we conclude that the total energy per mode,

$$\overline{\epsilon} = kT \tag{11}$$

The expression for the power (energy per unit of time) in an interval Δf is the product of ϵ (energy per unit mode) and Δf (modes per unit of time), from (1) and (2),

$$\mathrm{p} = kT\Delta f = \frac{\overline{v^2}}{4R}$$

We thus obtain an expression for the Nyquist theorem,

$$\overline{v^2} = 4kTR\Delta f \tag{12}$$

Appendix C: Gaussian Distribution and Elementary Statistics

In this study, we used some basic definitions from statistics such as the mean, the standard deviation, and the standard error of our measurements. The Gaussian distribution was used to fit the ADC readouts and determine a Gaussian mean and standard deviation. In this section we present these basic definitions along with a description of the Gaussian distribution.
If we have random variables X_1, X_2, X_3, ..., X_n, we define the mean and variance respectively as:

$$E(X) = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$V(X) = E\left((X_i - E(X))^2\right)$$

$$= E(X^2) - E(X)^2$$

The standard deviation is $\sigma = \sqrt{V(X)}$, and the standard error of n measurements is

$$\sigma_{err} = \frac{\sigma}{\sqrt{n}}$$

[5] $\frac{\mathrm{dP}}{\mathrm{dR'}} = 0$ when R = R'.

In Section 6.2, we obtained a standard deviation value from each of the 50 runs. We would like to compute the mean and standard error of these standard deviations which represents the noise equivalent at the input. We denote the mean as μ_i and the standard deviation as σ_i for each one of the 50 runs where i = 1,2,...,50.
The mean and standard deviation of the 50 values of σ_i are

$$\mu_\sigma = E(\sigma_i)$$
$$\sigma_\sigma = \sqrt{V(\sigma_i)}$$

The standard error is

$$\sigma_{se} = \frac{\sigma_\sigma}{\sqrt{50}}$$

The result of each point is plotted in Figure 18.

The ADC readouts collected by the readout setup are best considered as random variable that are described by some probability distribution. In this study, the histogram of the ADC readouts of a certain run had a bell like shape representing a Gaussian distribution (also called normal distribution). Our Gaussian fits follow closely the shape of the histogram as shown in Figure 16 which confirm the assumption about the random distribution of noise to be a Gaussian distribution of the parameters. A Gaussian distribution of the parameter x is given by the probability density function p(x)

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(x-m)^2}{2\sigma^2}$$

where σ is the standard deviation of the distribution and m is the mean of x.
This distribution implies that if one ADC readout is taken at random from a large ADC readout measurements, the probability of the parameter having values between x and (x+dx) is p(x)dx. The probability that x has a value between a and b is

$$P(a < x < b) = \int_a^b p(x)\mathrm{dx}$$
$$= \int_a^b \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)$$

Figure 26 shows the probability distribution function for a Gaussian distribution.

The peak value of the distribution occurs at the mean value x=m, and the standard deviation σ represents the spread of the distribution. The distribution extends to infinity on both sides, however most of the area is contained within x = m $\pm$ 3σ. We can normalize the variable x by defining z as

$$z = \frac{x-m}{\sigma}$$

The probability of finding z between z_a and z_b is

$$P(z_a < z < z_b) = \int_{z_a}^{z_b} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$

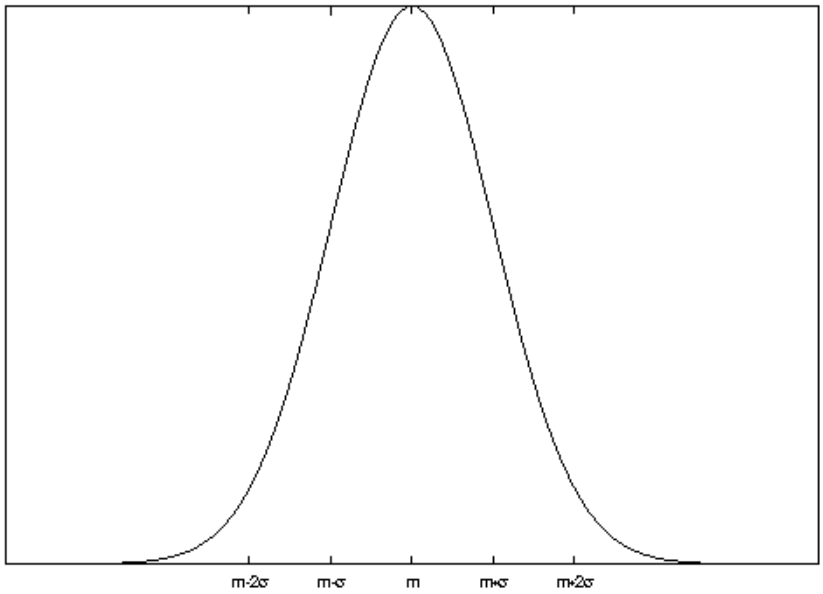

Figure 26: A Gaussian probability distribution showing the mean m and standard deviation σ. Note as we get away from the mean, the tails of the distribution drop to zero.

By carrying the integration of this expression, the probability of finding z between z_a = -3 and z_b = 3 is .9973. Extending the z range to ± 5 will give almost 1 which would account for most of the distribution.

If a random variable X is written as a sum of some other random variables a, b, and c from a normal distribution such as

$$x = a + b - c$$

The mean of the random variable X is

$$E(x) = E(a) + E(b) - E(c)$$

While the variance is

$$\begin{aligned} V(x) &= V(a) + V(b) + V(c) \\ &= \sigma_a^2 + \sigma_b^2 + \sigma_c^2 \end{aligned}$$

which was used to determine the noise of the overall setup by taking the noise contributions of each of its parts.