PROPERTIES OF FRONT-END ELECTRONICS OF THE COMPACT HIGH ENERGY CAMERA PROTOTYPE FOR GAMMA-RAY ASTRONOMY

A THESIS SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN PHYSICS

OF

THE UNIVERSITY OF NAMIBIA

BY

MWASHANA SUSANA NEKWAYA

201212416

MAY 2022

Main-Supervisor: Dr Michael Backes (Department of Physics, University of Namibia)

Co-Supervisor: Dr Steve Leach (Space Research Centre, University of Leicester, UK)

Abstract

Electronics are a key component for all modern detector systems. The Compact High Energy Camera (CHEC) is a full-waveform camera designed for the dual-mirror Small-Sized Telescopes (SST) of the Cherenkov Telescope Array (CTA). Some of the main components of the front-end electronics for CHEC are photo-sensors, preamplifiers and TARGET modules. CTA is planned to be a large observatory with up to 70 SSTs, with thousands of photosensor pixels per telescope (2048 camera channels per SST). The readout electronics for an array with such a high number of channels will certainly require a highly integrated and modular recording technology. In this study pulse waveforms generated by the camera channels and output by the preamplifier channels were analysed and parameterised in order to identify the different characteristics of the pulse waveforms. Characterising the pulse waveforms will help to better understand the behaviour of different electronic channels during camera mass production and during observations as well. An automated analysis procedure that will be able to identify faulty camera channels has been established. A cubic spline has been fit to the data in order to get a good representation of the data and for better comparison of the behaviour of the different electronics channels. The interpolated pulse waveform has a gradually rounded peak, while the actual pulse waveform has a rather sharp peak. In addition, for comparing the behaviour of the different electronic channels, the frequency distribution of the different characteristics are determined. In this study, it was

found that the rise time of the pulse waveform is on average (7.0 ± 0.3) ns , with an average fall time of (15.0 ± 0.8) ns, while the interpolated fit has an average rise time of (6.0 ± 0.6) ns, thus showing consistency in the behaviour of the different channels.

Contents

Acknowledgements

This thesis would not have been possible without the support of some great minds. Firstly, I would like to express my sincere gratitude to my supervisor, Dr Michael Backes, who read many drafts of my project and helped me make some sense out of the confusion. I would also like to thanks the CTA group at the University of Oxford, and the University of Leicester, United Kingdom, for accepting me and making me feel like one of them during the student exchange program I spend at those respective institutions. My special thanks go to Dr Steve Leach of the University of Leicester, as well as to Prof. Garret Cotter of the University of Oxford, who gave me the golden opportunity to do this wonderful project. I am sincerely grateful to them. I will strive to use gained skills and knowledge in the best possible way in order to attain desired career objectives. I hope to continue cooperation with all of you in the future.

I must also express my very profound gratitude to my family and friends for providing me with their unfailing support and continuous encouragement throughout my years of study.

This accomplishment would not have been possible without you all.

All praises are for the ALMIGHTY GOD, who's blessings are unending.

Declarations

I, Mwashana Susana Nekwaya, declare hereby that this study is a true reflection of my own research, and that this work, or part thereof has not been submitted for a degree in any other institution of higher education.

Ms. Mwashana Susana Nekwaya

Signature:.....

Date:May 2022.....

A list of acronyms

***Fermi*-LAT** The Large Area Telescope is the principal scientific instrument on the Fermi Gamma Ray Space Telescope spacecraft which is used to perform gamma-ray astronomy observations near the Earth orbit

ASICs Application Specific Integrated Circuits

CT Cherenkov Telescope

CTA Cherenkov Telescope Array

DAQ Data Acquisition

EGRET Energetic Gamma Ray Experiment Telescope

FEE Front-End Electronics

H.E.S.S. High Energy Stereoscopic System

IACT Imaging Atmospheric Cherenkov Telescope

MAGIC Major Atmospheric Gamma Imaging Cherenkov

SiPM Silicon Photo Multiplier

UNAM The University of Namibia

VERITAS Very Energetic Radiation Imaging Telescope Array System

List of Figures

Chapter 1

Introduction

Front-tend electronics (FEEs) and preamplifiers are a key component for all modern detector systems. Although experiments and their associated electronics can take very different forms, the same basic principles of the electronic readout and optimisation of signal to noise ratio apply to all [1]. The past quarter century has brought about a revolution in front-end electronics for large scale detector systems. The front-tend electronics, which process the signal at these detectors , has likewise developed in complexity and performance [2]. Nowadays, the frontiers of physics are calling for FEE which can cope with high count rates of several MHz. The solution is frequently based on custom designed Application-Specific Integrated Circuits (ASICS), with small physical size, low power dissipation, and high radiation tolerance. Custom integrated circuits specifically tailored to the requirements of large detector systems have provided unprecedented performance and enabled systems that once were deemed impossible [3]. The Compact High Energy Camera (CHEC) for the Gamma Cherenkov Telescope is a camera development project proposed to be part of the dual-mirror Small-Sized Telescopes (SST) of the Cherenkov Telescope Array (CTA) [4]. CTA is the next generation ground-based Imaging Atmospheric Cherenkov Telescope (IACT) array and will operate as an open observatory, building on the strengths of current IACTs. CTA is

designed to achieve an order of magnitude improvement in flux sensitivity, with unprecedented energy and angular resolution. Importantly, CTA will also increase the energy reach of ground-based gamma-ray astronomy, with sensitivity to astrophysical photons in the energy range of 0.02 TeV to beyond 100 TeV [5].

Statement of the Problem

The Cherenkov Telescope Array is planned to be a large observatory with up to 70 SSTs. During camera mass production and during observations there will be a need to inspect the camera channels (2048 camera channels per SST) in order to assure that they will be capable of full operation . Inspecting individual camera channels manually is going to be challenging and thus an automated method will be mandatory. Currently there is no method to automatically point out faulty camera channels. Hence there is a need to develop an analysis procedure that will point out faulty channels based on their behaviour, thus making inspecting camera channels less challenging.

1.1 Objectives of Study

As already mentioned in the introduction, preamplifiers are important components of the FEE and this study will mainly focus on analysing the pulses generated by the camera channels and outputted by the preamplifiers. Thus, the main objectives of this study were to parameterise the pulse shape outputted by the preamplifiers, and to establish an automated waveform analysis procedure that will highlight any variation from the expected channel response and thus help ensure that each assembled CHEC will have 2048 validated and uniform FEE channels.

1.2 Significance of Study

The University of Namibia (UNAM) is one of the member institutes of the Cherenkov Telescope Array. Studying the properties of the front-end electronics of CHEC, that may be used by CTA, gives the UNAM CTA group a better understanding of the working principles of the electronics of the camera to be used. The automated analysis procedure can also be used in the future during camera mass productions and during the analysis of waveform data.

1.3 Limitation of the Study

The FEEs are composed of various electronic parts and due to time limitation, this study will only focus on studying the behaviour of the camera channels in conjunction with the preamplifiers.

1.4 Delimitation of the Study

This study will only focus on characterising the pulses output by the preamplifiers, and no other measurements and analysis will be done on the preamplifiers.

This thesis is divided into six chapters. Chapter 1 gives a short introduction to the study, while in Chapter 2 a brief introduction to VHE gamma-ray astronomy will be presented in order to explain the main science motivations that make this field of research so active. Then, an overview on the Cherenkov Telescope Array project will be provided, highlighting its main advantages with respect to present-generation instruments. Technological aspects required for CTA application will also be described, giving particular attention to signal recording electronics and trigger performances.
The different front-end electronics (FEEs) for detectors, as well as their main purpose and basic working principles and the front-end electronics for the GCT camera called CHEC will be discussed in Chapter 3. Furthermore, the detailed characterisation of the

pulse waveform generated by the Silicon Photo Multipliers (SiPM) and output by the preamplifiers is described in Chapter 4. The description and results of normalised data and that of the interpolated data are presented in Chapter 5, and lastly the conclusions and recommendations are discussed in Chapter 6.

Chapter 2

Gamma-ray Astronomy and the Cherenkov Telescope Array

2.1 Introduction to Gamma-ray Astronomy

For over a century, physicists around the world have tried to uncover the mystery of the origin of ionising radiation inducing extended air showers. In 1912, Austrian-American scientist Victor Franz Hess has proved that the radiation has cosmic origin by measuring the ionisation in air [6]. Over the years, scientists developed stationary and mobile experiments to measure ionising radiation particles.

Very high-energy electromagnetic radiation reaches the Earth from an outer part of the cosmos carrying crucial and unique information about the most energetic phenomena in the Universe. Yet it has solely been within the last 30 years that we have had instruments to detect this radiation. The case improved with the development of Imaging Atmospheric Cherenkov Telescopes (IACTs) [7] and air shower detectors, that have advanced to open a new window for exploring the high-energy universe.

The current generation of ground-based gamma-ray instruments (H.E.S.S., MAGIC, VERITAS) enable the imaging, measurement and spectroscopic analysis of sources of high energy radiation. Data collected by these experiments within the latest years have ensured that VHE gamma-ray studies will have a large discovery potential in key areas

of astroparticle physics and basic physics research.

In the near future, the primary instruments that will be used to study the phenomenas from the very-high energy universe will be the Cherenkov Telescope Array (CTA). The Cherenkov Telescope Array will be an advanced facility for ground-based very-high-energy (VHE) gamma-ray astronomy.

The CTA observatory, which is presently under development, will be an experiment consisting of IACTs defined by a substantial increase in sensitivity (about 10-times) within the 10 GeV to 100 TeV energy range with respect to the prevailing VHE gamma-ray experiments. CTA will be a sizeable ground-based observatory collecting and recording Cherenkov light from particle showers initiated by VHE gamma-rays entering Earth's atmosphere.

This technique, referred to as Imaging Atmospheric Cherenkov Technique, requires an array of several telescopes, each one consisting of thousands of photo-sensor pixels corresponding to electronics channels to read out.

2.2 VHE Gamma-Ray Astronomy with CTA

Understanding the origin of cosmic rays and the search for dark matter may be thought of as two main science drivers of the CTA project. Cosmic rays (CRs) consists of charged particles that travel across the Universe and eventually interact with Earth's atmosphere producing showers of secondary particles. They can be classified in primary and secondary CRs. The former are originally created in numerous astrophysical processes and are composed primarily of protons and heavier nuclei (98%), with about 2% of electrons. Secondary cosmic rays, originated by the interaction of primary CRs with the Earth's atmosphere, embody neutrons, pions, positrons, and muons [8]. The CRs spectrum extends on a large energy range, from 10^9 to 10^{21} eV, as shown in Fig. 2.1.

Since gamma rays can created in hadronic cascades and may travel long distances while not being absorbed or deflected by interstellar magnetic fields, they provide a useful tool to trace back CRs and investigate their acceleration and propagation mech-

anisms. Radiation created within the gamma-ray energy range plays a crucial role so as to seek out proof of accelerated particles, since it cannot be generated by thermal emission from celestial objects.

High energy gamma-rays can be generated when highly relativistic particles collide with ambient gas, or interact with low-energy photons of the Cosmic Microwave Background, of the infrared dust emission and with the Galactic magnetic field. Since the flux and energy spectrum of gamma-rays corresponds to the flux and the spectrum of high energy particles, they can be used to trace cosmic rays in distant regions of our Galaxy or in other galaxies. Typically, cosmic rays do not reach directly to Earth because of the interaction with interstellar magnetic fields. Thus CR sources cannot be observed directly through CR measurements, since they lose the directional information during propagation due to interstellar and interplanetary magnetic fields.

Therefore gamma-ray astronomy is a powerful way to directly probe and image the cosmic particle accelerators responsible for the high-energy particle population observed. Another possible origin for high energy gamma-rays is represented by the decay of heavy unknown particles, which could represent dark matter candidates. Therefore, gamma-ray astronomy can play a crucial role conjointly within the discovery of the constituents of dark matter.

2.3 Gamma-ray Production Mechanisms

High energy gamma-rays have been detected from emissions of astronomical sources such as supernova explosions, neutron stars, and quasars. However, these gamma rays are not produced directly by the sources of the cosmic radiation, they can be created as secondaries of high-energy cosmic rays. Gamma rays can be produced for instance during particle-particle collisions, where a high-energy proton, or cosmic ray, interact another proton or atomic nucleus. This collision produces secondary particles, which may include one or more neutral π-mesons (or pions) [10–12]. The resulting pions can further decay into a pair of gamma rays ($\pi^0 \rightarrow 2\gamma$). Also, if a primary accelerator generates a beam of high-energy electrons, these electrons may undergo bremsstrahlung in

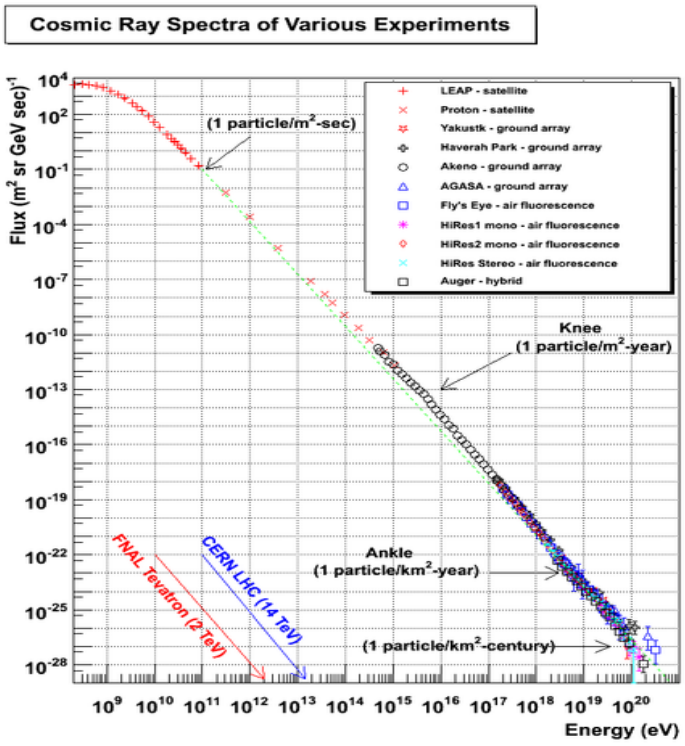


Figure 2.1. Cosmic rays spectrum from various experiments [9].

the ambient medium, may suffer synchrotron radiation losses in local magnetic fields, or may, via the inverse Compton scattering process, transfer a significant part of their energy to an ambient photon, which then emerges as a high energy gamma ray [13].

2.3.1 Electromagnetic Shower Formation

When a very-high-energy gamma ray (primary) interacts with the atmospheric nuclei (this can happen at an altitude of about 10 – 20 km above sea level), it initially generates an electron and positron pair, via pair production. Each high-energy $e^{\mp}$ radiates secondary gamma rays through bremsstrahlung, which further convert into e^{-} + e^{+} pairs of lower energies (see Fig. 2.2). This process continues until the average energy

per particle drops below 80 MeV [14]. At this point, the interactions lead to the absorption of particles and the cascade begins to die. This altitude is known as the shower maximum, occurring typically at an altitude of $\sim$10 km above sea level [10].

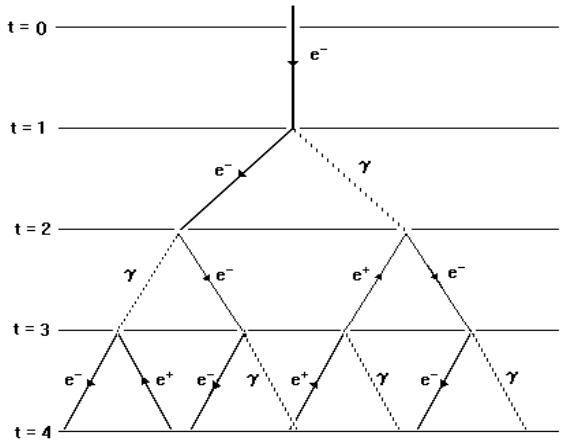


Figure 2.2. Electromagnetic shower: A schematic view of an electromagnetic cascade [14]. It should be noted that this is just an illustrative model but not the state-of-art of current air shower knowledge.

Fig. 2.2 is the simple model that illustrates the characteristic longitudinal shower development of a photon induced air-shower. The model assumes a cascade, starting with the pair production of an electron and positron, which radiate bremsstrahlung in the second step, and so forth. E_0 is the energy of the primary gamma ray. X_0 is the distance over which an electron or positron loses, on average $\langle E \rangle$, half of it's energy by radiation.

$$\langle E \rangle = E_0 \, e^{-\frac{X}{X_r}}, \tag{2.1}$$

After traveling the same distance, X_0 , a photon splits into an electron-positron pair. In either instance, the energy E_0 of a particle (electron or photon) is assumed to be equally divided between two outgoing particles, which lead to an energy,

$$E_n = 2^{-n} E_0, \tag{2.2}$$

of the particles in a shower after *n* radiation lengths. The number of particles *N* in the shower after *n* radiation lengths is

$$N = 2^n. \tag{2.3}$$

Multiplication of particles ceases when their energies are too low for pair production or bremsstrahlung, at around $E < 80$ MeV [14, 15].

2.3.2 Cherenkov Radiation

In a dielectric medium, the speed of light *c* is lower than in vacuum. The charged particles (electrons and positrons) traversing a medium with refractive index $n > 1$ at speed *v*, higher than the speed of light in the medium $\frac{c}{n}$, causes the emission of a characteristic electromagnetic radiation, the so-called Cherenkov radiation [16].

Before starting the description of the Cherenkov effect, it must be noted that the Cherenkov radiation is entirely unrelated to the bremsstrahlung, which is emitted by the moving electron itself when it collides with atoms in the medium. The Cherenkov effect involves radiation emitted by the medium under the action of the field of the particle moving in it [17].

If a charged relativistic particle travels through a medium, which can be polarized, it moves other charged particles in the medium out of their rest positions, inducing dipoles along its path. Upon relaxation of those dipoles, spherical waves are emitted. However, if the speed of the charged relativistic particle becomes bigger than the speed of light in the medium, the elementary waves interfere constructively and Cherenkov radiation is emitted [18].

Cherenkov light is emitted into a forward cone of aperture $\cos(\theta_c) = \frac{1}{n \cdot \beta}$, where $\beta = \frac{v}{c}$. The condition for Cherenkov radiation emission is that $\beta \geq \frac{1}{n}$. The Cherenkov angle θ_c depends on the refractive index *n* and grows with increasing particle speed *v*. If the particle is ultra-relativistic, $\beta \approx 1$, θ_c = constant and has maximum value.

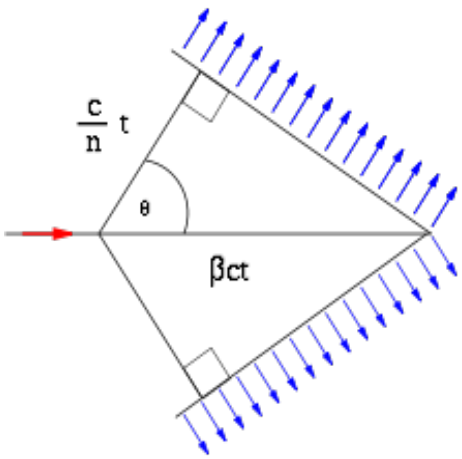


Figure 2.3. Cherenkov wave front: Illustration of the formation of the Cherenkov wave front [18].

The wavefront (analogous to shock wave for supersonic aircraft) is emitted at a well defined angle θ_c with respect to the trajectory of the particle. In Fig. 2.3, a particle (arrow) traveling from point A to B in time *t* has velocity βc. The distances AB and AC are given by $AB = \beta ct$ and $AC = \frac{c}{n}t$. Since $\cos\theta = \frac{AB}{AC}$, making the right substitution gives $\cos\theta = \frac{1}{\beta n}$, which is the criterion for Cherenkov light emission.

If the medium is transparent the Cherenkov light can be detected. This light can be collected by special telescopes even if the electrons and positrons of the shower do not reach the ground. The effective detection area is comparable to that of the light pool on the ground, i.e. about 10^4 m^2, and it is suited to the very-low gamma-ray fluxes above 100 GeV [18].

The intensity of the Cherenkov radiation (number of Cherenkov photons emitted per unit path length and per wavelength interval) by a particle with charge *Z* can be calculated with the Frank-Tamm formula:

$$\begin{aligned}\frac{d^2N}{dxd\lambda} &= \frac{4\pi^2Z^2e^2}{hc\lambda^2}(1-\frac{1}{\beta^2n^2(\lambda)}) \\ &= \frac{2\pi Z^2}{\lambda^2}\alpha(1-\cos^2(\theta_c)) \\ &= \frac{2\pi Z^2}{\lambda^2}\alpha\sin^2(\theta_c), \end{aligned} \tag{2.4}$$

where ($\alpha = e^2\hbar, c = 1/37$ is the fine structure constant, $\alpha = \frac{e^2}{\hbar\cdot c}$ = $\frac{1}{137}$) [19].
Equation 2.4 shows that the number of photons per unit length and radiation wavelength depends on charge and velocity of particle. Cherenkov light is proportional to a λ^{-2} power law, where λ is the wavelength. Due to the factor $\frac{1}{\lambda^2}$, the maximum of the distribution $\frac{d^2N}{dxd\lambda}$ is reached for short wavelengths. That is, the greater part of Cherenkov photons are emitted in the blue and ultraviolet range, and the spectrum has a peak at around 330 nm [18].

2.3.3 The Imaging Atmospheric Cherenkov Technique

High-energy gamma rays are too difficult to reflect or focus like other forms of light (e.g. optical light and soft X-rays), instead, they are imaged indirectly.
The technology of ground-based Cherenkov telescopes developed in the latest years provides relevant progresses and advances in VHE gamma-rays astronomy. In the IACT technique the Earth's atmosphere is used as a calorimeter to absorb high energy particles. When a VHE gamma-ray enters the atmosphere, it interacts with atmospheric nuclei generating a secondary electromagnetic shower of electrons, positrons and photons.
Moving through the atmosphere at speed higher than the speed of light in air, electrons and positrons emit a beam of Cherenkov light, mainly concentrated in the near

UV and optical region of the electromagnetic spectrum. Cherenkov light reaches the ground almost unattenuated and is reflected by means of telescope mirrors in its focal plane where a multi-pixel camera records the shower image. The shape of the showers recorded by the camera is approximately elliptical, and its orientation depends on the angle it makes with the optical axis of the telescope [20]. If the axis of the telescope and of the shower coincide, the observed image will be a circle centered on the axis itself. The reconstruction of the shower axis in space and the trace-back into the sky allow to determine the incoming direction of the gamma-ray.

A high energy photon produces a cone of Cherenkov light confined to 1° of the original photon direction [21]. Cherenkov radiation ($\sim$100 photons/m^2/TeV) created in the electromagnetic air showers almost uniformly illuminates a circle of radius $r \approx 120$ m on the ground in a time typically of the order of 10 ns [12, 22]. A telescope (like the H.E.S.S. telescopes) located at some altitude above the sea level will detect this Cherenkov light pool and reflect it to its camera, located at the focal point of its reflecting surface (mirror).

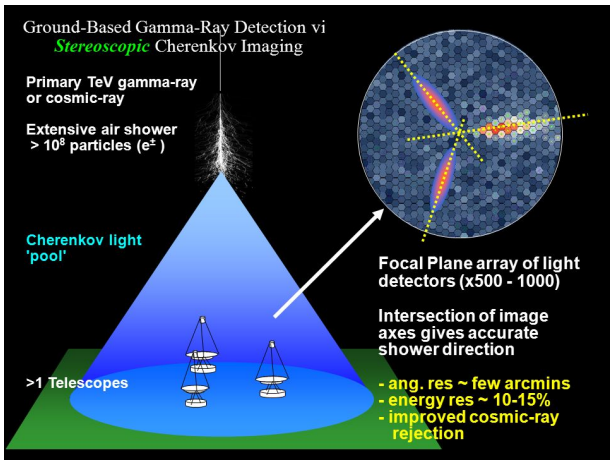


Figure 2.4. Production of Cherenkov light and its detection by the Cherenkov telescopes [21].

In Fig. 2.4, Cherenkov light gets imaged by a large mirror onto a fast camera. Electromagnetic air showers generate elongated, roughly elliptical images. The major axis of the ellipse represents the shower axis, pointing towards the image of the source on one side. The ellipse is parameterised according to Hillas parameters [23], and this parameterisation is used for reconstruction of the shower direction and separation of electromagnetic from hadronic showers.

The use of several telescopes observing the same shower in coincidence allows a unique determination of the shower direction (see Fig. 2.4) by projecting the images in all triggered telescope cameras onto one camera. Then the intersection point of the image major axes yields the shower direction. The energy of the primary gamma ray can be determined from the intensity of the Cherenkov light detected [23].

Unfortunately, not all air showers seen are initiated by gamma rays. In fact, only a small fraction of them are gamma-ray induced showers. This is because the showers can also be formed when cosmic ray particles (mainly protons) interact with the atmosphere, but these showers can be distinguished based on the shape of the image of the light flash [24]. When compared to proton and nuclei induced air showers, photon induced air showers contain only electrons, positrons and photons (see Fig. 2.2) [10, 16]. Also, gamma ray induced air showers are more compact, more uniform and decay faster. Having more than one telescope simultaneously observing the same air shower can also help reject cosmic ray induced air showers..

2.4 The Cherenkov Telescope Array

2.4.1 CTA performance

The next-generation of VHE Cherenkov telescopes will overcome a series of limitations affecting the current experiments and will advance the state of the art in astronomy at the highest energies. The improvements in performances expected from CTA can be summarised as follows:

- **Improved sensitivity:** CTA will be about 10 times more sensitive than any existing instrument in its energy range, as shown in Fig. 2.5. In the core range CTA will achieved milli-Crab (mCrab) sensitivity.
- **Improved angular resolution:** CTA can reach angular resolution in the arc-minute range, 5 times better than the typical values for current instruments.

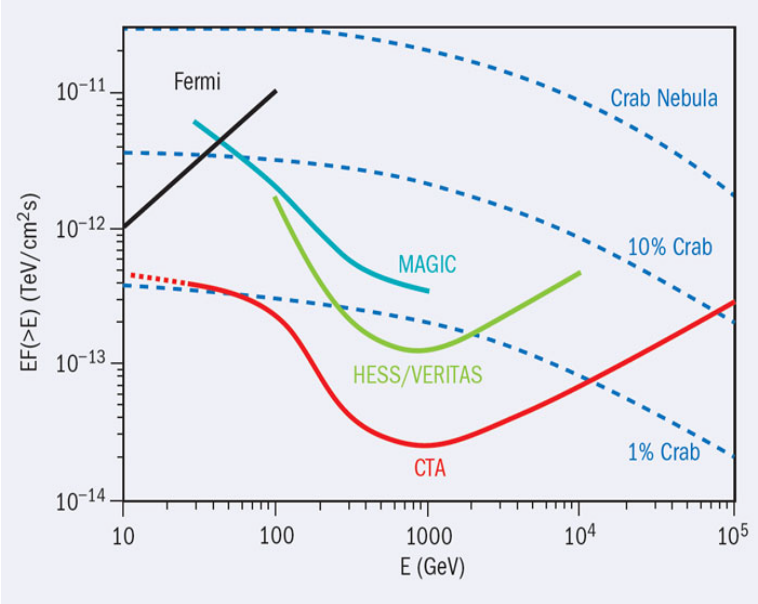


Figure 2.5. Sensitivity of the CTA compared with other air-Cherenkov telescopes (H.E.S.S., VERITAS and MAGIC) and the Fermi Gamma-ray Space Telescope [25].

2.4.2 Cherenkov Telescope Array Layout

Taking into account the wide energy range which needs to be covered by the CTA telescopes, a uniform array of identical devices equally spaced is not the most efficient solution. On the contrary, three different-sized telescopes will be used, but their arrangements and characteristics within the array are still being tested [20]. The array will include:

- **Large Size Telescopes (LST):** the large collecting area allows the detection of low energy gamma-ray (down a few tens of GeV) which produce fainter Cherenkov signals.
- **Medium Size Telescopes (MST):** to perform the core energy search from 100 GeV to about 10 TeV several tens of 12m diameter telescopes will be used.
- **Small Size Telescopes (SST):** a great number of 4 m-diameter telescopes will complete the array to perform the super-TeV search, thanks to the large effective area covered by this array.

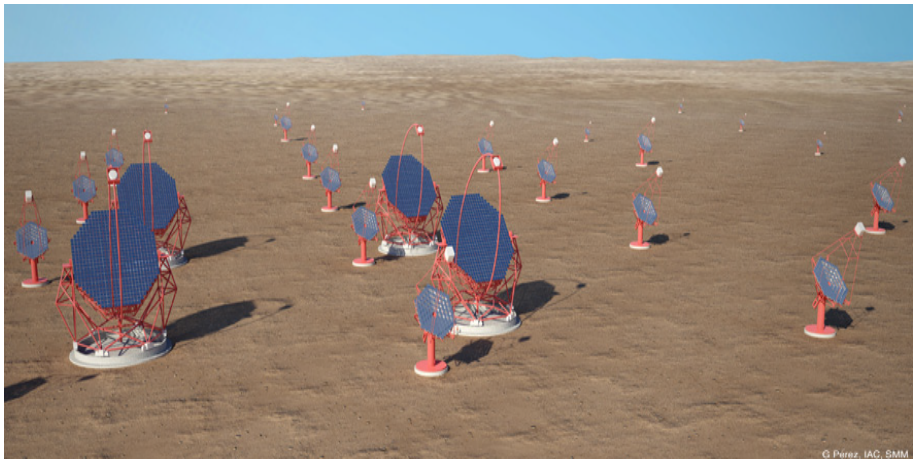


Figure 2.6. A schematic figure of CTA telescopes array [25].

2.5 CTA Telescopes

A Cherenkov telescope is composed of the mirror plane characterised by its light collection capability and the detecting camera in the focal plane followed by front-end electronics. The pixel size in the focal plane limits the size of image features that can be resolved, while the electronics for signal capture and triggering should provide a bandwidth compatible with the time duration of Cherenkov pulses of a few nanoseconds. In the following sections candidate detectors for camera requirements on the readout and trigger systems will be reviewed.

2.5.1 Photon Detection

The photon detectors most ordinarily utilised in IACTs cameras are ordinarily photomultiplier tubes (PMTs) with dynodes performing the electron multiplication [26]. The comparatively high quantum efficiency (up to 30%), together with the high gain ($\sim 10^6$) and a spectral response which perfectly matches the Cherenkov light spectrum in the 300-600 nm wavelength range, make them a well-working solution in detecting low-level light during a quick process. However, differing kinds of photon detectors are presently under study as potential candidates for CTA camera construction [26].
Silicon photomultipliers (SiPMs) and Multi-Anode photomultipliers (MAPMTs) are two of the foremost quoted alternatives to PMTs, particularly in MST and SST cameras for which PMTs are not offered within the applicable physical sizes [27]. In this work, signals from SiPMs are going to be acquired using one of the proposed front-end electronic solutions for CTA. Silicon detectors will give a better photon detection efficiency with reference to current photomultipliers at low price, do not need high voltage supply and do not suffer from the influence of magnetic fields.
On the other hand, these devices usually need cooling to scale back their dark count rate[1] and suffer from optical cross-talk[2] .

[1]The **dark count rate** is the average rate of registered counts without any incident light.This determines the minimum count rate at which the signal is dominantly caused by real photons.

[2]**optical crosstalk** is the probability that an avalanching microcell will cause an avalanche in a second microcell.

2.5.2 Signal Recording Electronics

Photodetector signals evoked by Cherenkov light have a pulse width of a couple of nanoseconds, superimposed on a night sky background with typical rates between 10 MHz and 100 MHz, depending on mirror and pixel size. An optimised data acquisition system for these signals implies high analogue information measure, short integration time and a dynamic range such that single photoelectron signals are resolved, and signals of a couple of hundred of photoelectrons are captured without distortion at an equivalent time.

Moreover, the recording electronics ought to delay or store the signal until a trigger is generated, indicating that an occasion of interest must be acquired and processed. The estimated time for generation of a trigger signal is from 0.1 to few μs within one telescope, and $\pm$ 10 s if a trigger logic among many telescopes is developed [28].

Typically two differing kinds of recording systems are used to sample and process photodetector signals: the primary system performs the integration of pulse amplitude returning the charge corresponding to that pulse, whereas the secondary one consists of the storage of the complete sampled wave shape over a fixed time window. The second solution is the one utilised in the TARGET chip sampler. Recording many parameters of the pulse form instead of simply the integrated charge, permits to get additional information concerning signal timing. An adaptive integration window is utilised in order to scale back the integration of noise.

Most techniques used for signal acquisition in existing IACT arrays are supported by either of two totally different technologies, the **Flash Analogue-to-Digital Converters** (FADCs) or also the analogue sampling recollections, known as **Switched capacitor Array**.

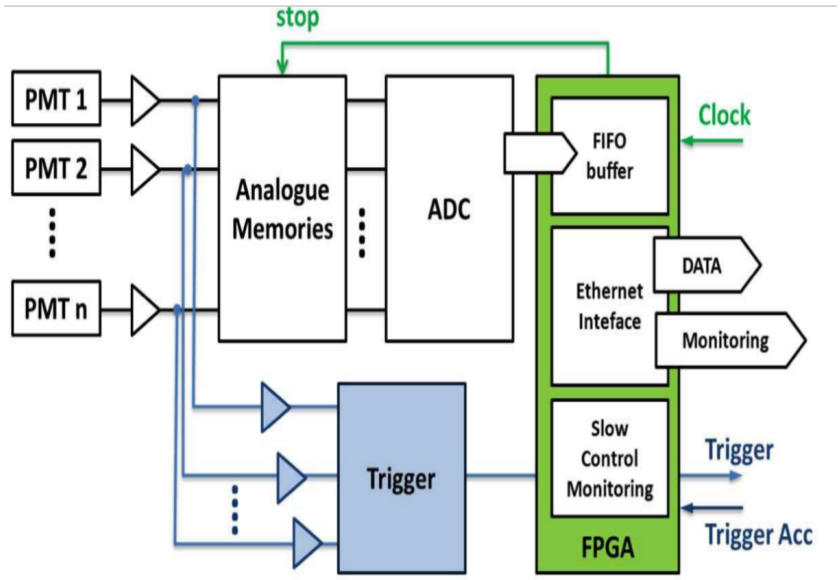


Figure 2.7. Block diagram of recording system based on analogue memory [27].

FADCs modify the detector signals at rates from about 100 MSa/s to a couple of GSa/s, writing the output into a digital ring buffer. The modest cost of such digital memories allows to realise wide buffers getting a large trigger latency (tens of μs). However FADCs typically have a restricted dynamic range of no more than 8-10 bits, and globally represent a quite costly solution with respect to other recording devices. Additionally, they are large and characterised by a high-power consumption, permissive to the possibility of integrating the readout electronics into the camera. Despite this, in recent years low-power and affordable FADCs have been developed at the expenses of digitising speed: the so called FlashCam system, that performs pulse reconstruction with a sampling rate of 250 MSa/s, is presently under development for CTA [27].

However, analogue sampling memories encompass banks of switched capacitors utilised in turn to record the signal waveform at a given sampling rate. The maximum recording depth out there is given by the sampling time interval multiplied by the number of storage capacitors, and represents the main limit for the trigger latency value. The trigger signal is produced by an additional analogue trigger circuit, whereas the digitisation of sampled waveform is performed using external ADCs. The signal converted

in digital format can be stored in a local Field-Programmable Gate Array (FPGA) before being transferred to the processing system, as shown in Fig. 2.7. Examples of such sampling systems include H.E.S.S. and MAGIC electronics, which are based on ARS and Domino Ring Sampler (DRS) ASIC [28] respectively. Different analogue switched capacitor systems are under study for CTA; among these, the TARGET ASIC will be considered and discussed in terms of structure and sampling performances in next sections.

In conclusion, while FADCs system may offer slightly superior performances, analogue samplers permit lower costs and power consumption. These are two vital aspects that ought to be taken under consideration within the development of front-end electronics for such a high number of readout channels, as in CTA application.

Chapter 3

Front-end Electronics for Detectors

The past quarter century has led to a revolution in front-end electronics for large scale detector systems. Front-end electronics (FEEs) and preamplifiers are a key element for all modern detector systems. Though experiments and their associated electronics can take totally different forms, the same basic principles of the electronic readout and optimisation of signal to noise ratio apply to all [1].

3.1 Multi-purpose Front-end Electronics

The FEE is the interface between the sensing elements of particle detectors and the Data Acquisition (DAQ). It provides the first step of the signal processing (see the diagram in Fig. 3.1).

The output of detectors is usually a brief current or voltage pulse, $V(t)$, whose time response is processed by the FEE. The logic behind the electronics determines the information carried by the output signal. Discriminators, for instance, return a rectangular pulse which might be proportional to the width of the input signal, whereas integrators return a pulse proportional to the integral of $V(t)$, thus to the deposited charge. Several alternative quantities can be encoded within the response, like event rate, time of arrival, minimum /maximum signal, or some combination of those. But the simultaneous optimisation of various measurements within the same FEE is usually difficult. As a consequence, some compromises in design can not be avoided, e.g.

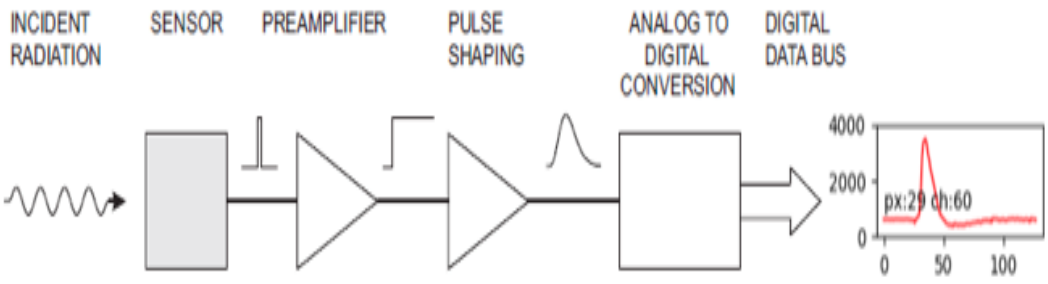


Figure 3.1. Basic detector functions: Radiation is absorbed in the sensor and converted into an electrical signal. This low-level signal is then integrated in a preamplifier, fed to a pulse shaper, and then digitised for subsequent storage and analysis [1].

a precise integrator has a low timing resolution. This does not represent a disadvantage for single-purpose FEE, because their response is tailored to the specific detector. In addition, Custom Integrated Circuits specifically tailored to the requirements of large detector systems have provided unprecedented performance and enabled systems that were once deemed impossible [3].

The minimum detectable signal and the precision of the amplitude measurements are limited by fluctuations. For instance, electronic noise induces baseline fluctuations, which are superimposed on the signal and alter the peak amplitude. Superimposing a constant amplitude signal on a noisy baseline will yield a Gaussian amplitude distribution and, often the width of the waveform is expressed as the full width at half maximum (FWHM) [1]. Strip detectors and custom Integrated Circuits (now called Application-Specific Integrated Circuits (ASICs) are routine in high-energy physics [7].

3.2 The Gamma-ray Cherenkov Telescope

The Gamma-ray Cherenkov Telescope (camera) called CHEC is a dual-mirror camera development project for the small sized telescope components of CTA to explore the energy range from a few TeV to hundreds of TeV with a field of view 9° and an angular resolution of a few arcminutes. The GCT design features dual-mirror Schwarzschild-Couder optics and a compact camera based on densely-pixelated photodetectors as well as custom electronics. The dual-mirror optical design demagnifies the air shower image, allowing a reduction of the telescopes plate scale, and thus enabling small form-factor photosensor elements to be used in the camera [5]. The Compact High Energy Camera is designed to record flashes of Cherenkov light, lasting from a few to a hundred nanoseconds, with a field of view of 9°.

CHEC consists of 32 module prototypes, each one able to read 64 photosensor channels. A photograph of a single module is shown in Fig. 3.2. It is composed of four TARGET chipS and one FPGA, and then 64 input channels can be readout by a single compact module. The prototype is compatible to interface with 64 SiPM pixels.

3.3 The Front-end Electronics of The Compact High Energy Camera

The CHEC camera is composed of several components as part of its FEE. The main components are listed below, also see the digram in Fig. 3.2 [29]:

Photosensors - these are electronic components that detects the presence of visible light, infrared transmission (IR), and/or ultraviolet (UV) energy. Silicon Photomultipliers will be used for CHEC.

Preamplifiers - for signal amplification and shaping to optimise camera triggering and

readout.

TARGET Modules - 64-channel signal capture modules based on the TARGET ASIC for data digitisation, read out, pixel-level triggering and slow control.

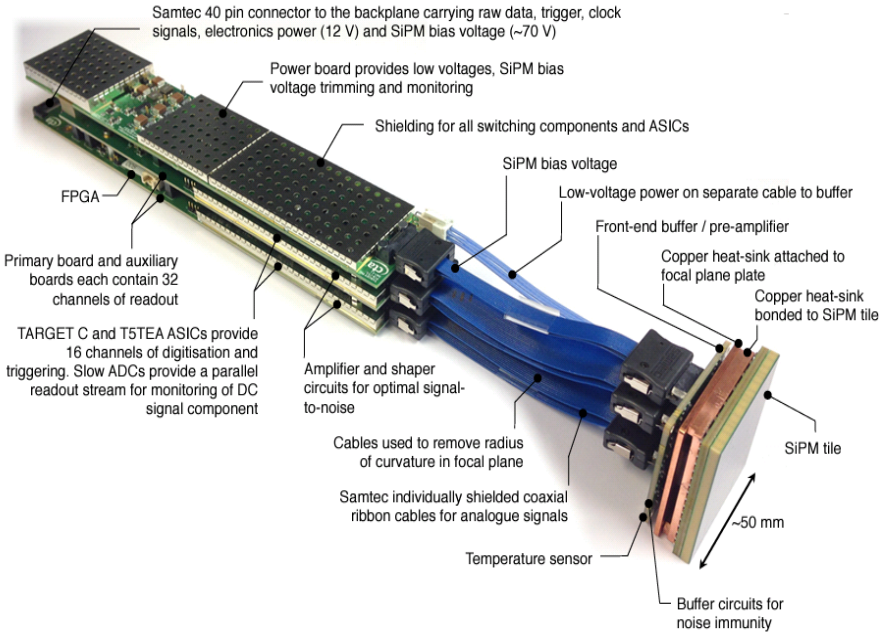


Figure 3.2. A characteristic Front-end module of the Compact High Energy Camera [30].

3.3.1 Silicon Photomultipliers

Silicon Photomultipliers are radiation detectors with extremely high sensitivity and efficiency. They consist of a matrix of reverse biased p/n junctions connected in parallel, in which each cell works beyond the breakdown voltage voltage V_{BD} as a Geiger-Mode Avalanche Photo-Diode. The SiPM is a sensor that addresses the challenges of sensing, timing and quantifying low light signals down to the single photon level. The SiPM is an array of microcells. Each CHEC SiPM tile is composed of 256 (3 x 3) mm^2 pixels combined in groups of four on a bias board directly mounted to the SiPM to provide the desired camera pixel size. Thus, each FEE has 64 (6 x 6) mm^2 camera pixels and hence 64 camera channels and 1 pulse per channel for a single photon [31].

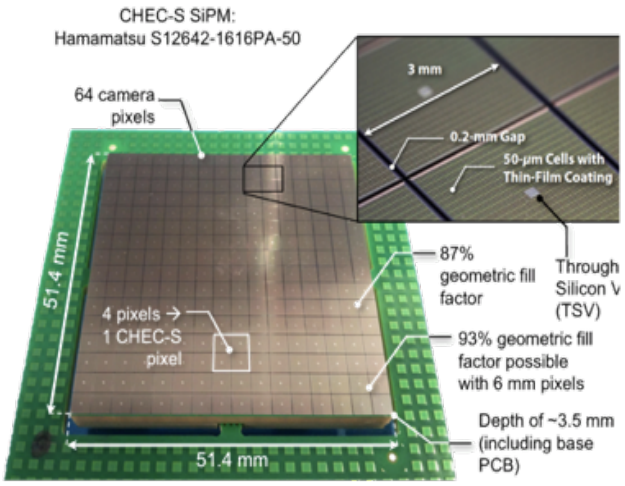


Figure 3.3. The microcell structure of the SiPM surface [31].

The SiPM integrates a dense array of small, independent Single Photon Avalanche Diodes (SPAD) sensors, each with its own quenching resistor. Each independently operating unit of SPAD and quench resistor is referred to as a "microcell". Once a microcell within the SiPM fires in response to an absorbed photon, a Geiger avalanche is initiated inflicting a photocurrent to flow through the microcell. This results in a voltage drop across the quench resistor, which in turn reduces the bias across the diode to a value below the breakdown, thus quenching the photocurrent and preventing further Geiger-mode avalanches from occurring [27]. Once the photocurrent has been quenched, the voltage across the diode recharges to the nominal bias value. The time it takes for the microcell to recharge to the full operating voltage is called the recovery time. It is important to note that the Geiger avalanche will be confined to the single microcell it was initiated in. During the avalanche process, all other microcells will remain fully charged and ready to detect photons.

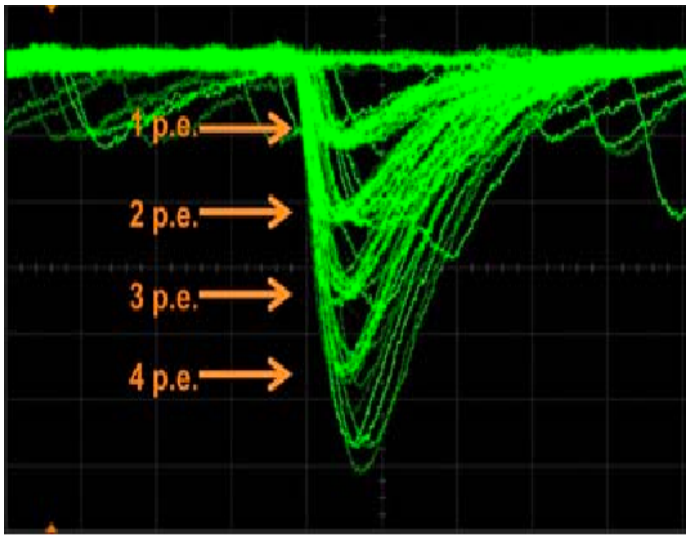


Figure 3.4. The figure shows the discrete nature of the SiPM output when illuminated by brief pulses of low-level light [27].

Each microcell detects photons identically and independently. The sum of the photocurrents from each of these individual microcells combines to form a quasi-analog output[1], and is thus capable of giving information on the magnitude of an instantaneous photon flux. The response to low-level light pulses of different photon content is shown in Fig.3.4, and a charge spectrum of these pulses is shown in Fig.3.5.
The SiPM features low-voltage operation, insensitivity to magnetic fields, mechanical robustness and uniformity of response.

3.3.2 TARGET Module

CTA will consist of an array of many telescopes, with thousands of photosensor pixels per telescope. The readout electronics for an array with such a high number of channels require a highly integrated and modular recording technology. Therefore in order to improve camera electronics performances, systems based on multi-channel waveform sampling circuits are leading candidates as front-end solutions for fast recording electronics.

[1]The conversion of a digital signal to a form that is suitable for transmission over a specified analog channel. This specification should include frequency, range, bandwidth, signal-to-noise ratio and envelope distortion.

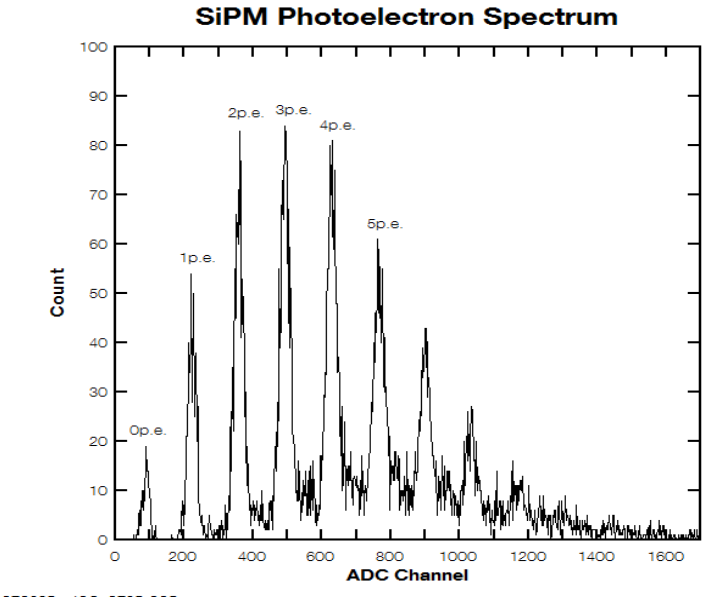


Figure 3.5. Photoelectron spectrum of the SiPM, achieved using brief, low-level light pulses, such as those from Fig.3.4 [27].

The TARGET application-specific-integrated circuit series has been designed and optimised specifically for the readout of Cherenkov cameras [32], such as the GCT. The TARGET C ASIC is used for sampling and digitisation in the CHEC camera. The main functional blocks of the ASIC are triggering, analog sampling and storage buffers, analog-to-digital converters (ADC) for digitising the signal. The TARGET ASICs record the full waveform for every event a good charge reconstruction is also guaranteed in the saturated regime of the ASIC transfer function using waveform parameters like the rising or falling edge of the pulse [32]. The first functional camera made of SiPMs for Cherenkov telescopes is used in FACT.

3.4 Signal Characterisation

The first step for the electronic design is the detailed study of the expected input signals. The main target of the FEE development was originally the readout of Imaging Cherenkov Counters. The challenging requirements of this type of detectors include high performance on the detection of Single Photon (SP) signals. These are usually small and fast voltage pulses which demand careful characterisation.

3.4.1 Single Photon Signals

The SP detection capability is of fundamental importance for a large number of particle detectors (e.g Cherenkov PID and fast time-of-flight detectors, among others). In general, the use of standard SiPM precludes the achievement of high precision timing for SP events. To accomplish such a task, characteristics such as transit time spread and rise time of the SiPM need to be as small as possible (< 100 ps). High gain ($\sim 10^5$ to 10^6) and high signal-to-noise ratio are of crucial importance as well.

.

3.4.2 Silicon Photomultiplier Pulse Shape

A typical SiPM anode-cathode output pulse is as shown in Fig.3.6. The rise time of the SiPM is determined by the rise time of the avalanche formation and the variation in the transit times of signals arriving from different points on the sensor's active area. The recovery time of the sensor, or fall time of the pulse, is determined by the microcell recharge time constant, which is given by,

$$T_R = C_d(R_q + R_s x N) \tag{3.1}$$

where C_d is the effective capacitance of the microcell, R_q is the value of the microcell quench resistor, N is the total number of microcells in the sensor and R_s is any resistance in series with the sensor (e.g. sense resistance, signal tracks on the sensor).

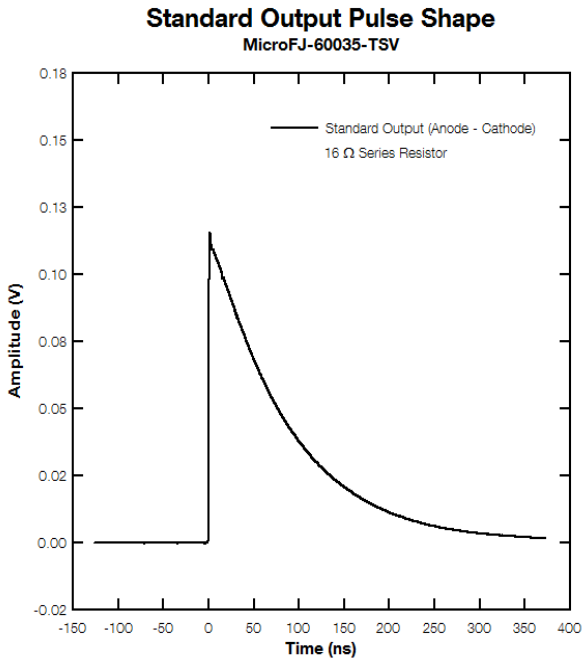


Figure 3.6. Output pulse read out from the anode-cathode of a 6mm, 35μm microcell SiPM, showing the recovery time of the microcell [27].

Chapter 4

Materials and Methods

As the goal is to measure properties of FEEs specifically pulse waveforms generated by the camera channels and outputted by the preamplifiers under monitored conditions. the setup has to comply with several requirements. Firstly, it was insured that the SiPM was supplied with its operation voltage and to collect the signals generated. The second major task was to create an environment in which the SiPM could be exposed to a controllable and monitored amount of light. The setup was placed into a dark box to block out any light from sources other than the ones used during the experiment. These signals then need to be processed properly to make them usable in data analysis.

4.1 Experimental setup and procedure

The experimental setup used in this thesis consists of a 650 nm pulsed laser emitting at a frequency of 1 kHz, incident onto an SiPM photodetector (64 pixels), through a preamp-buffer board, onto TARGET module and the data was captured in Python. The FEE module as well as the pulsed laser were placed into a dark box (within a dark room) and properly biased by means of a voltage supply.

Rigorous studies for testing the SiPM response to SP required the use of a dedicated setup. A metal light-tight box (dark box) is used to shield the photodetector from ambient light and to reduce possible electrical noise. The interior is furnished with an optical board which is used to place sensors, electronics and laser head for systematic studies. The side opening allows operation inside the box.

The system consists of a control unit and an optical head with an electrical driver and a laser diode. Trigger input and output are available as Transistor-Transistor Logic (TTL) signals. The chosen optical head produces short laser pulses with a full width half maximum (FWHM) of 35 ps and a wavelength of 650 nm. The repetition rate can be varied up to 1 kHz The laser is directly projected onto the photosensor, a collimator, fixed on an adjustable support, focused the laser light on the SiPM. The initial intensity of the laser I_0 can be attenuated by using a Neutral Density Filters (NDFs) with an optical density d:

$$\frac{I}{I_0} = 10^{-d}, \tag{4.1}$$

where I is the output intensity

The CHEC front-end module A fully mounted front-end module (drawer), where the pulses can be injected from the pulse generator onto the SiPM photon detector. These input analog signals are routed to the MUSIC chips where they are stored at high rates. The CHEC front-end module is made up of the components listed below as shown in Fig. 3.2.

SiPM sensor and amplification stage Silicon Photomultipliers consist of a matrix of reverse biased p/n junctions connected in parallel, in which each cell works beyond the breakdown voltage V_{BD} as a Geiger-Mode Avalanche Photo-Diode. When a photon hits one cell a hole-electron pair will be created. Then, because of the electric field within the active area of the cell itself, these carriers can trigger an avalanche multiplication process that eventually leads (after proper amplification) to a detectable signal. A single cell operated in Geiger-mode works as a photon-triggered switch, in either an on or off state, and therefore cannot provide a signal proportional to the incident light

intensity. Regardless the number of photons interacting within a diode at the same time, it will produce a signal of the same amplitude, corresponding to the one obtained with a single incident photon.

To overcome this lack of proportionality Geiger-APD are arranged in a matrix structure, and the signals from each single microcell are summed to form the output of the SiPM. The sum of signals from each single photodiode combines to form a quasi-analog output, and is thus capable to give information on the magnitude of an incident photon flux.

The pulser To pulse an LED a pulse generator was used. Throughout this thesis it will be referred to as the pulser. It can produce pulses with a maximum height of 5.3 V and a minimum pulse width of about 2 ns. The baseline of the pulses can be shifted by 2.1 V up and down. The pulser can generate pulses up to a rate of about 10 MHz. The device is completely analog, all settings have to be done by hand. As this is rather inaccurate when it comes to the frequency, it is advisable to use an external trigger to fix a frequency from the outside. For this function the pulser has an internal delay of 18 ns fixed plus an additional selected delay of minimum 2 ns.

The **robot arm** helps to focus the **laser pointer** onto the detector.

4.2 Data Analysis

In this section, the data analysis method is described and, the amplitude distributions of acquired SiPM signals are presented for different pixels.

The signal outputs of the SiPM (after amplification stage) consists of a negative pulse, with an amplitude proportional to the number of p.e. detected and a width of few ns.

The waveforms are output as un-processed waveforms by the preamplifiers as shown in Fig. 4.2 for all 64 channels.

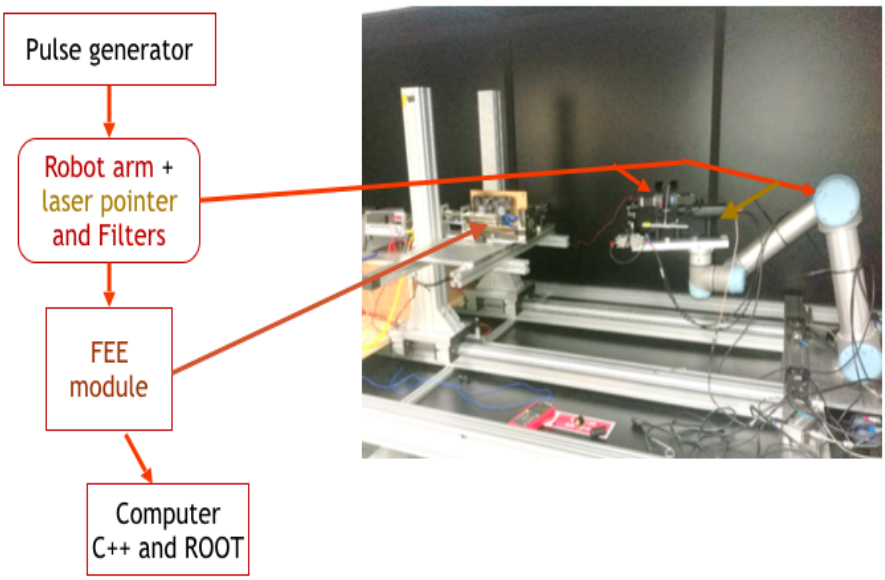


Figure 4.1. The experimental setup of the FEE Lab module in test at the university of Leicester, UK.

Where the first diagram in Fig.4.2. shows an average pulse waveform for all the 64 pixels.

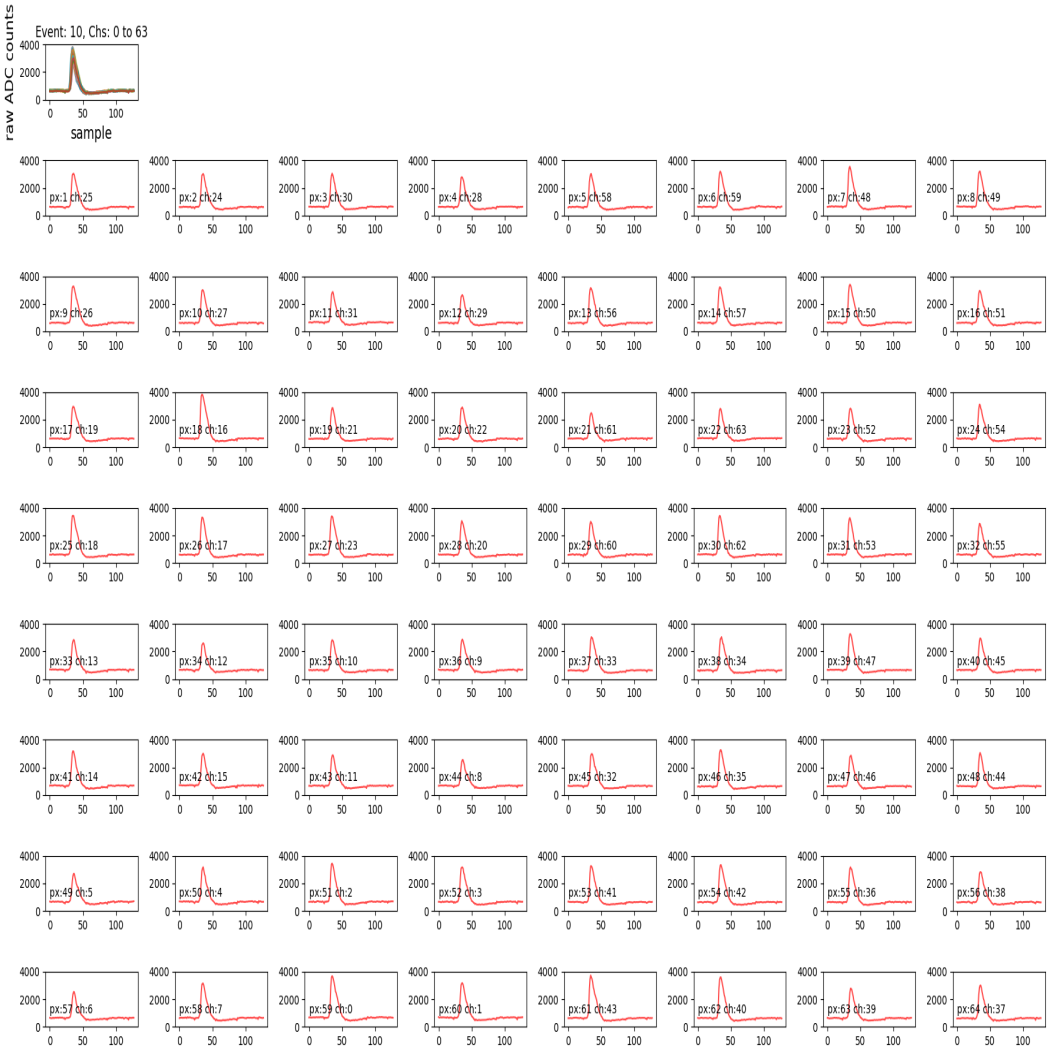


Figure 4.2. Raw data waveforms as output by the preamplifiers.

In order to get a better fit to the data, and for better comparison, a cubic spline was fitted to the data as shown in Fig. 5.2. The cubic spline was fitted to the data by generating 1000 points between the data points. The same procedure as described above was followed in order to obtain the threshold as well as the rising-edge of the interpolated data. It can clearly be seen in Fig. 5.2 (b). that the interpolated pulse waveform has a gradually rounded peak compared to Fig. 5.1. (b).

For this study, an analysis procedure (Python code) has been developed. For each camera channel, a threshold was calculated from the actual data by taking data points that are within 3 standard deviations of the baseline. The rising edge of the pulse wave was then determined by choosing from data points that are less than or equal to the threshold.

The uncertainty on the values presented in Chapter 5 under Section 5.2 where obtained by calculating the average and the standard deviation for the different parameters as shown below:

```
L = [ i - j for i,j in zip(W,w)]  # rise time difference

risetimediff_ave = sum(L)/len(L)  # average rise time difference

#Calculating the standard deviation
risetimediff_diff = [i - risetimediff_ave for i in L]
risetimediff_sqdiff = [i**(2.0) for i in risetimediff_diff] #average
                                    value
risetimediff_avesqdiff = (sum(risetimediff_sqdiff)/(len(L) - 1))
risetimediff_stdev = np.sqrt(risetimediff_avesqdiff) #standard
                                    deviation value
```

the full analysis procedure is shown in the Statistical Analysis Python code in the appendices section.

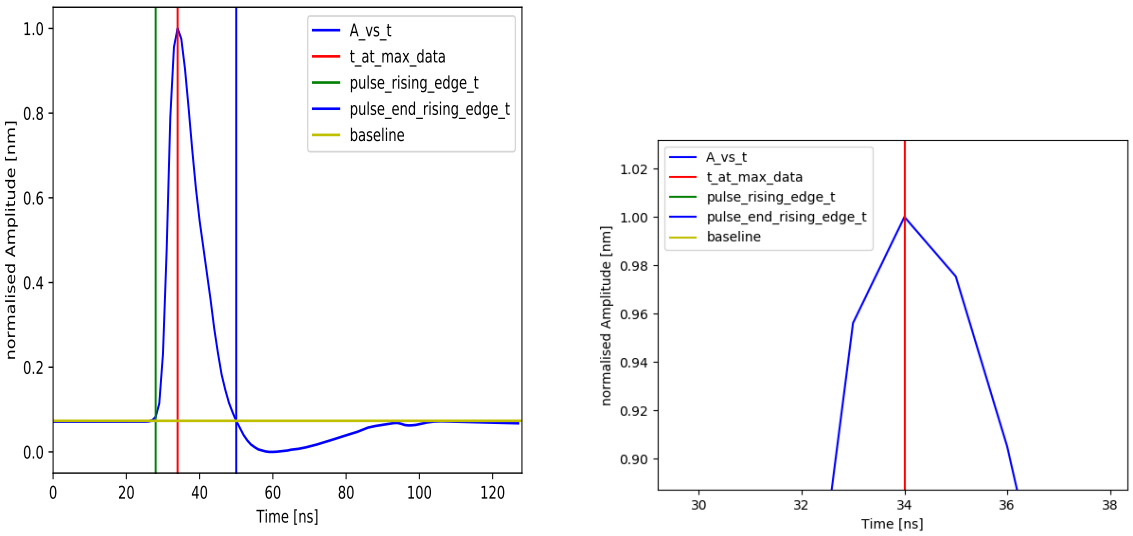
Chapter 5

Results

5.1 Pulse-Waveform Characterisation

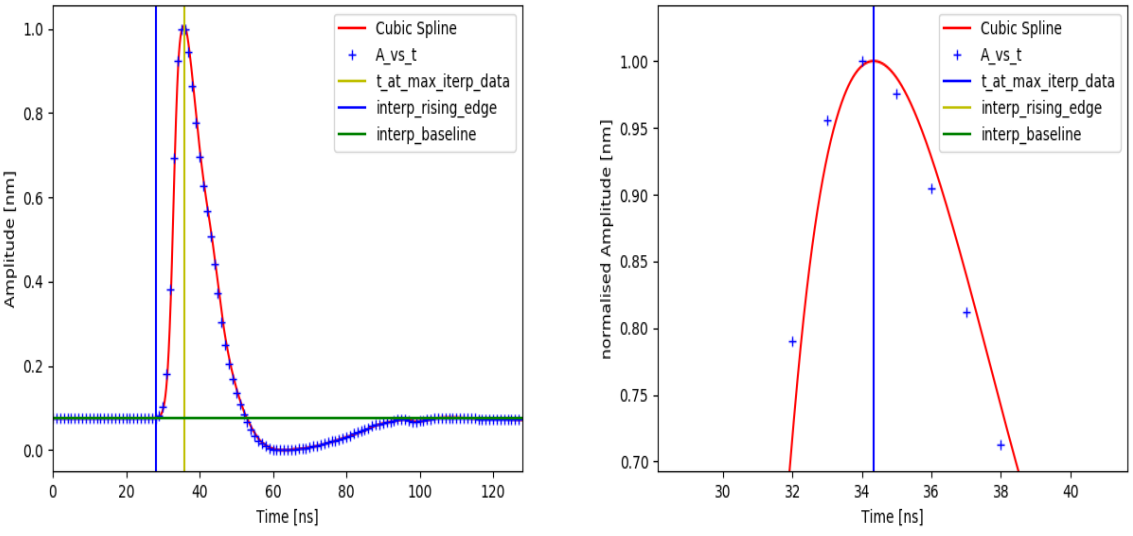
In order to be able to study and compare the behaviour of the different preamplifier channels, the waveforms from the 64 camera channels have been parameterised. The parameters that will be used in this study are: the Full Width at Half Maximum of the pulse, the pulse's rise and fall times, the maximum amplitude (peak) and peak-time values, the start and end baseline values.

The **FWHM** is the width of the pulse, the **rise time** is the time it takes the pulse to reach the peak, while the **fall time** is the time it takes the pulse to return to the baseline. Fig. 5.1 shows a characterised pulse waveform of a single camera channel. After taking a closer look at the pulse waveform, it was observed that the data does not form a smooth curve, as mentioned in section 3, but has a rather sharp peak.
The Fig. 5.3 below shows a fully parameterised pulse waveform. This is a summary of the various parameters that will be used in comparing the behaviour of the different camera channels as mentioned in 5.1. The frequency distributions for these parameters are discussed in Sec. 5.2.



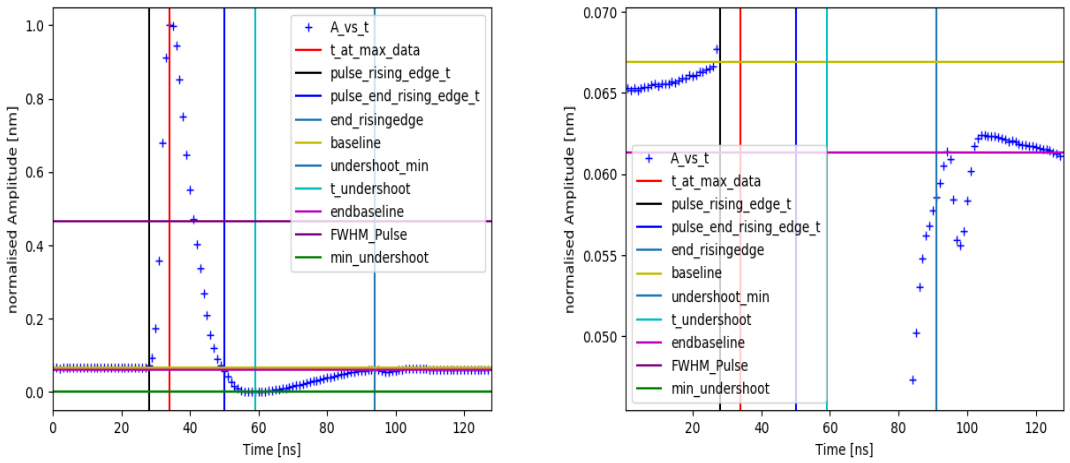
(a) *(b)*

Figure 5.1. Fig. (a) shows a characterised pulse waveform and Fig. (b) shows a blow-up of the peak of Fig. (a) and it can be seen that the pulse waveform has a rather sharp peak.



(a) *(b)*

Figure 5.2. Fig. (a) shows a characterised interpolated pulse waveform and Fig. (b) shows a blow-up of Fig. (a) and it can be seen that it has a rather rounded peak.



(a) *(b)*

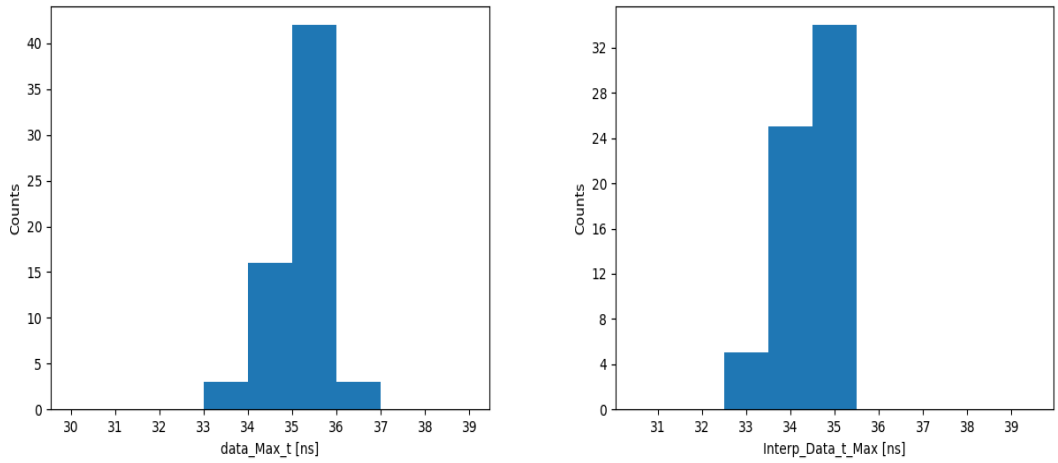
Figure 5.3. A fully parameterised waveform (a) and a blow-up of the start and end baselines (b).

5.2 Statistical Analysis of the Various Characteristics

For comparing the behaviour of the different electronic channels, the amplitude data was normalised to have a maximum of 1.00 and, the frequency distribution of the different pulse waveform parameters discussed in 5.1 were determined from the histograms shown below. The results discussed in this chapter are those obtained for 64 camera channels (64 pixels).

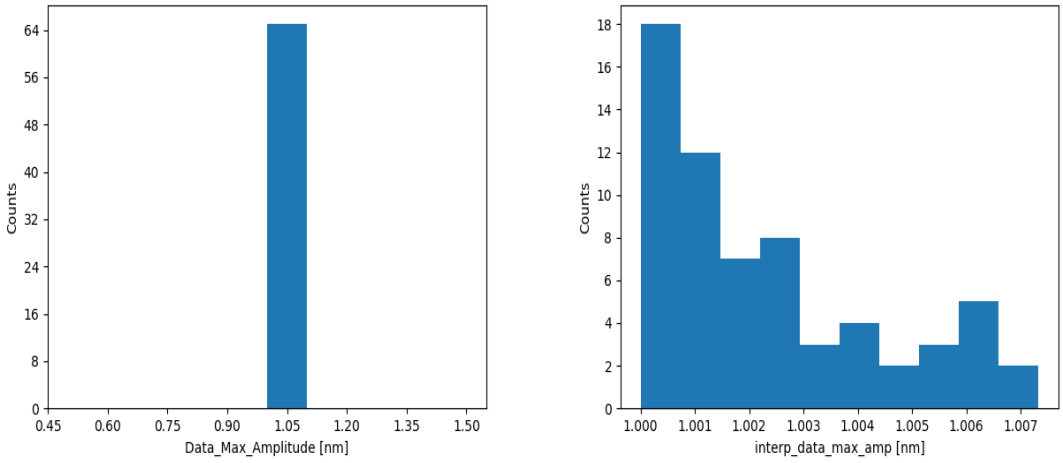
Fig. 5.4 shows the frequency distribution of the time at the peak, the average peak time for the pulse waveform is (34.7 ± 0.6) ns, while that of the interpolated pulse waveform is on average (34.5 ± 0.6) ns, with an average difference of (0.25 ± 0.44) ns. Hence showing that the difference between them is not significant.

It can be seen in Fig. 5.5 (a) that the maximum amplitude of the pulse waveform is on average (1.0 ± 0.0) nm and that of the interpolated pulse waveform is on average (1.002 ± 0.002) nm with an average difference of (0.002 ± 0.002) nm.



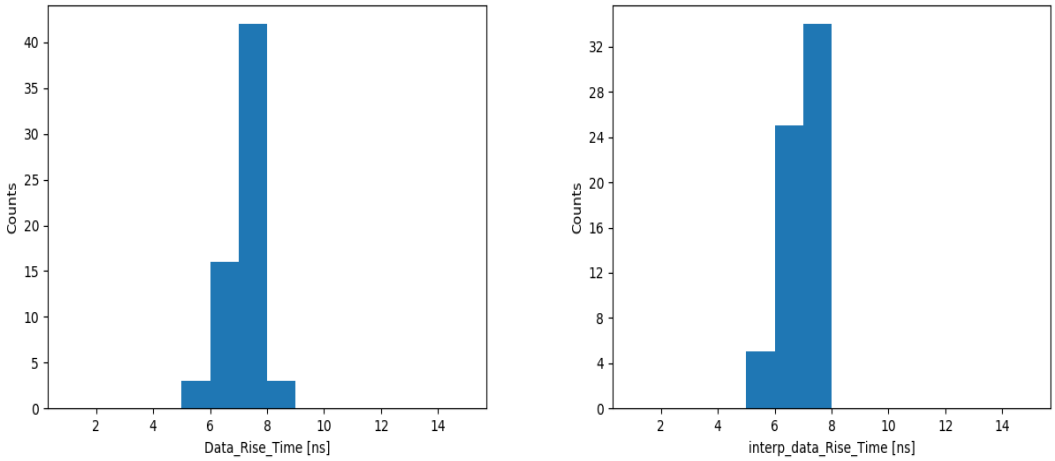
(a) *(b)*

Figure 5.4. The frequency distribution of the time at the peak for the pulse waveform (a) and that for the interpolated pulse waveform (b).



(a) *(b)*

Figure 5.5. The frequency distribution of the maximum amplitude (peak) for the pulse waveform (a) and that for the interpolated pulse waveform (b)



(a) *(b)*

Figure 5.6. The frequency distribution for the rise time for the data (a) and interpolated data (b) .

The histograms in Fig. 5.6 show the time it takes the pulse to reach its peak, which is referred to as the rise time. The first histogram shows the rise time of the data. It can be seen that the pulse waveform takes on average (6.78 ± 0.54) ns to reach it's peak with the mode rise time frequency distribution centred at 7 ns. The second histogram shows the time it takes for the interpolated pulse waveform to reach its peak, the interpolated pulse wave takes on average (6.54 ± 0.54) ns to reach its peak with the mode frequency distribution centred at ~ 7 ns.

The significant difference between the data and interpolated data rise times was calculated and the results are as shown on Fig. 5.7. The results show that the difference between the rise time is on average (0.24 ± 0.43) ns, this means that there is no significant difference between the rise times and that the results are closely related.

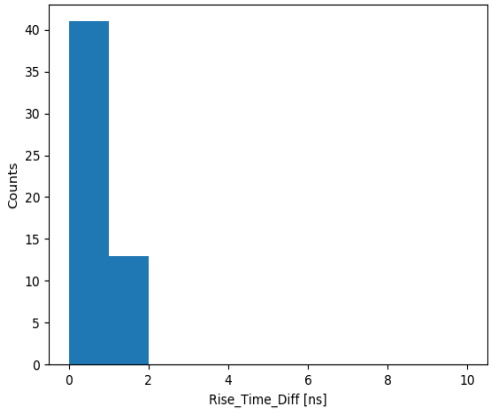
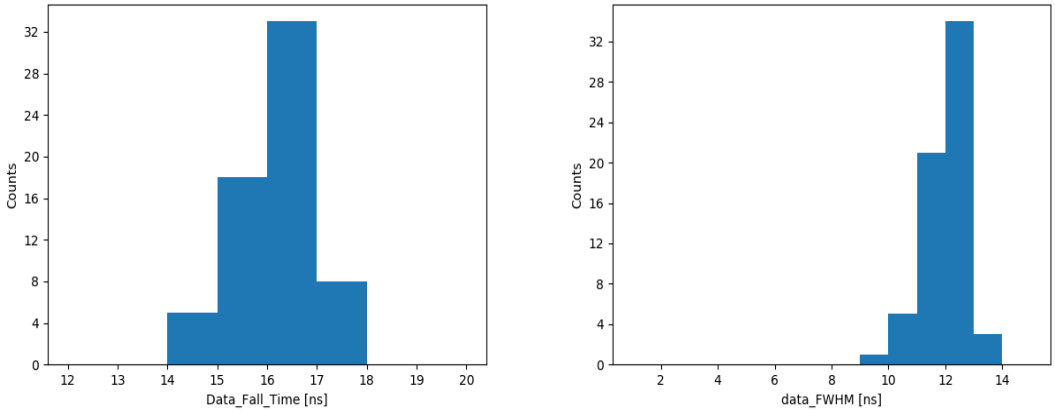


Figure 5.7. The frequency distribution of the difference in rise time for the data and interpolated data.

The time it takes the pulse waveform to return to the baseline is referred to as the fall time, and the frequency distribution is as shown Fig.5.8. (a).



(a) *(b)*

Figure 5.8. Frequency distribution of the fall time time (a) and the FHWM (b) for the data.

The fall time is on average (15.7 ± 0.76) ns, which clearly shows that the pulse waveforms do not suddentlly drop to the baseline, but take a reasonable amount of time. While the FWHM which describes the width of the pulse waveform is shown in Fig. 5.8 (b), with an average width of (11.5 ± 0.8) ns.

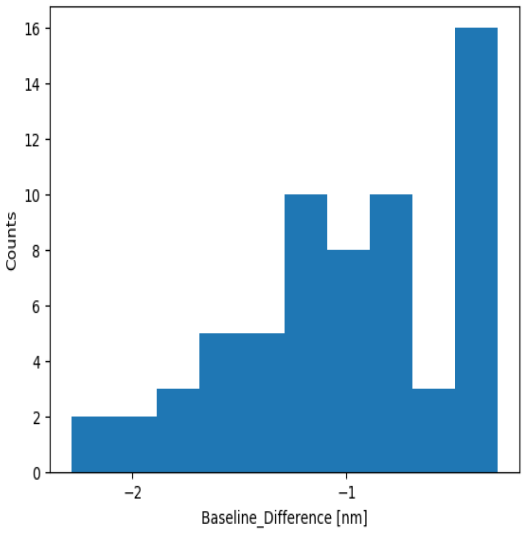


Figure 5.9. The difference in the baseline value for the 64 channels.

The histogram in Fig. 5.9 shows the difference between the start baseline value and the end baseline value for the 64 channels as can be seen in Fig. 5.3 (b). The difference was calculated in order to see whether the undershoot recovers back to the start baseline value. The difference is on average (-0.995 ± 0.518) nm, being a negative value, which clearly shows that the undershoot does not recover to the start baseline value.

Chapter 6

Conclusion

The study presented in this thesis was aimed to the characterisation of pulse waveforms generated by the SiPM and output by the preamplifier. Since CTA is planned to be a large observatory that will featuring tens of Imaging Atmospheric Telescopes, each with thousands of photosensor pixels, a highly integrated data analysis procedure to identify faulty camera channels was established in the course of this project.

The study has shown that the rise time of the pulse waveform is on average (6.78 ± 0.54) ns while the fall time is on average (15.7 ± 0.76) ns, thus showing consistency in the behaviour of the different electronic channels. On the other hand, the cubic spline turned out to be a better representation of the data as it exhibits a gradually rounded peak as observed for pulse waveforms generated by SiPMs, while the pulse waveform from the data in study has a rather sharp peak.

The difference between rise time of the data pulse waveform and the interpolated pulse waveform was calculated to be very small (0.24 ± 0.43) ns, thus showing that the interpolated data fit well with the data in use. These results imply that, there is consistency in the behaviour of the different electronic channels. A camera channel is considered to be faulty when, the pulse waveform rises too fast $\sim$ ($<$ 5.16 ns) and has a sudden drop (fall) of $\sim$ ($<$ 13.42 ns).

All in all, none of the pixels in this study fell below the limits mentioned above, thus none of the channels were found to be faulty.

Bibliography

[1] Spieler H 2003 *Proceedings of the 2003 ICFA School on Instrumentation, Itacuruca, Brazil*

[2] Cardinali M 2015 *Fast frontend electronics for high rate particle detectors* Ph.D. thesis Universitätsbibliothek Mainz

[3] Spieler H 2007 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **581** 65–79

[4] Daniel M, White R, Berge D, Buckley J, Chadwick P, Cotter G, Funk S, Greenshaw T, Hidaka N, Hinton J *et al.* 2013 *Power* **10** 4

[5] Brown A M, Abchiche A, Allan D, Amans J P, Armstrong T, Balzer A, Berge D, Boisson C, Bousquet J J, Bryan M *et al.* 2016 The gct camera for the cherenkov telescope array *Ground-based and Airborne Telescopes VI* vol 9906 (International Society for Optics and Photonics) p 99065K

[6] Hess V F 1912 *Phys. Z.* **13** 1084–1091

[7] Spieler H 2004 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **531** 1–17

[8] Longair M S 2011 *High energy astrophysics* (Cambridge university press)

[9] Schneider P 2014 *Extragalactic astronomy and cosmology: an introduction* (Springer)

[10] Cui W 2009 *Research in Astronomy and Astrophysics* **9** 841

[11] Garmire G and Kraushaar W 1965 *Space Science Reviews* **4** 123–146

[12] Völk H J and Bernlöhr K 2009 *Experimental Astronomy* **25** 173–191

[13] Protheroe R and Kazanas D 1982 Origin of relativistic particles and gamma-rays in quasars Tech. rep. National Aeronautics and Space Administration, Greenbelt, MD (USA). Goddard ...

[14] Cheng J 2009 The principles of astronomical telescope design *Astrophysics and Space Science Library* vol 360

[15] Matthews J 2005 *Astroparticle Physics* **22** 387–397

[16] Chadwick P 2005 *Astronomy & Geophysics* **46** 6–12

[17] Degrange B and Fontaine G 2015 *Comptes Rendus Physique* **16** 587–599

[18] Balzer A 2010 *Verhandlungen der Deutschen Physikalischen Gesellschaft*

[19] Fülöp L and Biró T 1992 *International journal of theoretical physics* **31** 61–74

[20] Hillas A 2013 *Astroparticle Physics* **43** 19–43

[21] de Naurois M and Mazin D 2015 *Comptes Rendus Physique* **16** 610–627

[22] Aleksić J, Babić A, Godinović N, Hrupec D, Lelas D, Puljak I, Šnidarić I, Terzić T and Zanin R 2016 *Astroparticle physics* **72** 76–94

[23] De Naurois M and Rolland L 2009 *Astroparticle Physics* **32** 231–252

[24] Holder J 2015 Atmospheric cherenkov gamma-ray telescopes Tech. rep.

[25] 2019 URL `https://cerncourier.com/a/cherenkov-telescope-array-is-set-to-open-new-windows/`

[26] Knoll G F 2010 *Radiation detection and measurement* (John Wiley & Sons)

[27] Francesca S 2014 *Study and characterization of TARGET-7 chip for a CTA camera* Master's thesis Universita degli Studi di Bari "Aldo Moro"

[28] Actis M, Agnetta G, Aharonian F, Akhperjanian A, Aleksić J, Aliu E, Allan D, Allekotte I, Antico F, Antonelli L *et al.* 2011 *Experimental Astronomy* **32** 193–316

[29] Daniel M and Veberič D 2013 *A Compact High Energy Camera for the Cherenkov Telescope Array* Ph.D. thesis Univerza v Novi Gorici

[30] White R and Schoorlemmer H 2017 A compact high energy camera (chec) for the gamma-ray cherenkov telescope of the cherenkov telescope array Tech. rep. SISSA

[31] White R 2017 *Journal of Instrumentation* **12** C12059

[32] Funk S, Jankowsky D, Katagiri H, Kraus M, Okumura A, Schoorlemmer H, Shigenaka A, Tajima H, Tibaldo L, Varner G *et al.* 2017 Target: A digitizing and trigger asic for the cherenkov telescope array *AIP Conference Proceedings* vol 1792 (AIP Publishing LLC) p 080012

Appendix A

Pulse Waveform Python Analysis Code

```
"""
    Plotting Graphs with data from multiple files
 """
import glob
from scipy.interpolate import interp1d
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.backends.backend_pdf import PdfPages

filenames = sorted(glob.glob('/Users/garret-l/Dropbox/
                                    LeicesterVisit2018/Mar22_all_pbs/
                                    pb001/data_0_7_r1_waveforms/
                                    data_waveform_p*.txt')) # the glob
                                     module allows you to iterate over
                                     multiple files

filenames = filenames[0:65]
t = range(128) # defines a list of numbers

#print (t)
for f in filenames:

    data = np.loadtxt(fname=f)
##    print (data)
```

```
# calculating averages
mean_data = np.mean(data)
#print 'mean data = %f' %mean_data

# opens a textfile file for appending
dfile = open('pb018_0_1_results.txt', 'a')

#Normalizing the Data
normalized_data = (data-min(data))/(max(data)-min(data))
#print normalized_data

max_data = max(data)  # Find the maximum y value from the
                                    normalized data
max_t = t[data.argmax()]  # Find the x value corresponding to
                                    the maximum y value

# Calculating the baseline
lenbaseline = 26
baselinedata = (data[0:lenbaseline])

#print (baselinedata)
sumbaseline = sum(baselinedata)
#lenbaseline = len(baselinedata)
#print lenbaseline
avebaseline = sumbaseline/lenbaseline # This is the baseline
                                    value
print 'baseline_ave =', avebaseline

diffbaseline = [i - avebaseline for i in baselinedata]
sqdiffbaseline = [i**(2.0) for i in diffbaseline]
avesqdiffbaseline = (sum(sqdiffbaseline)/len(sqdiffbaseline))
stdevbaseline = np.sqrt(avesqdiffbaseline)
print 'stdevbaseline =', stdevbaseline
```

```
# Calculating threshold
thresholdnumber = 3.0
threshold = avebaseline + (thresholdnumber*stdevbaseline)
rising_edge_t = 0.
print 'threshold =', threshold

''' PULSE PARAMETERS '''

# Finding the start rising edge
for i in range(100):
    #print i,data[max_t-i]
    if data[max_t - i] <= threshold:
        rising_edge_t = (max_t - i)
        baseline_val = data[max_t - i]
        #print "found edge ", rising_edge_t
        #print "Amplitude", data[max_t - i]
        break

'''
Calculating Full Width at Half Maximum of pulse
'''


peak = max_data - avebaseline
halfpeak = peak/2.0 # calculates half the maximum value
#print 'half peak', halfpeak

# Obtain the time t at point1 of half the peak
for i in range(20):
    #print i,data[max_t+i]
    if data[max_t - i] <= halfpeak:
        point1_t = (max_t - i)
        point1 = data[max_t - i]
        #print "found point1_t ", point1_t
        #print "point1", point1
```

```
            break
    # Obtain the time t at point2 of half the peak
    for i in range(30):
        #print i,data[max_t+i]
        if data[max_t + i] <= halfpeak:
            point2_t = (max_t + i)
            point2 = data[max_t + i]
            #print "found point2_t ", point2_t
            #print "point2", point1
            break

    # Calculating Full Width at Half Maximum
    FWHM = point2_t - point1_t
    #print 'FWHM = %d' %FWHM



    # Calculting the rise time
    rise_time = max_t - rising_edge_t
    #print 'Rise Time = ', rise_time

    # Calculating the end baseline and undershoot

    end = len(data) - lenbaseline

    #endbaselinedata = data[100:128]
    endbaselinedata = data[end:len(data)]
##    print (endbaselinedata)

    sumendbaselinedata = sum(endbaselinedata)
##    print (sumendbaselinedata)
    lenendbaseline = len(endbaselinedata)
##    print (lenendbaseline)
    aveendbaseline = sumendbaselinedata/lenendbaseline
##    print (aveendbaseline)
    diffendbaseline = [ i - aveendbaseline for i in endbaselinedata]
```

```
##    print (diffendbaseline)
    sqdiffendbaseline = [i**(2.0) for i in diffendbaseline]
##    print (sqdiffendbaseline)
    avesqdiffendbaseline = (sum(sqdiffendbaseline)/len(
                                        sqdiffendbaseline))
##    print (avesqdiffendbaseline)
    stdevendbaseline = np.sqrt(avesqdiffendbaseline)
##    print (stdevendbaseline)



    # end baseline threshold
    endthreshold = aveendbaseline + (5*stdevendbaseline)
##    print (endthreshold)
    endrising_edge_t = 128

    '''Calculation of the undershoot'''
    min_data = min(data)
    undershoot_min = min_data
    undermin_t = t[data.argmin()]
##    print (under_t, undershoot)


    ''' UNDERSHOOT PARAMETERS'''

    ''' Finding undershoot start time and amplitude '''
    for i in range(100):
        #print (i,data[max_t+i])
        if data[max_t + i] <= threshold:
            understart_t = (max_t + i)
            understart_ampval = data[max_t + i]
            #print "found edge ", understart_t
            #print "Amplitude", data[max_t + i]
            break
```

```
    '''Finding end rising edge'''
    for i in range(0,100,33):
        #print i,data[max_t+i]
        if data[undermin_t + i] > min_data and data[undermin_t + i]
                                            <= endthreshold:
            endrising_edge_t = (undermin_t + i)
            endbaseline_val = data[undermin_t + i]
            #print "found edge ", endrising_edge_t
            #print "End_Rising_Edge_Amplitude", data[undermin_t + i]
            break



    undershoot_startamp = list(data).index(understart_ampval) #
                                        finds the index of the
                                        undershoot start amplitude
                                        value in a list
##    print 'undershoot_startamp',undershoot_startamp

    undershoot_endamp = list(data).index(endbaseline_val) # finds
                                        the index of the undershoot
                                        end amplitude value in a list
##    print 'undershoot_endamp',undershoot_endamp

    ''' Undershoot data values '''
    undershoot_data = data[undershoot_startamp:undershoot_endamp]
    #print undershoot_data

    ''' Calculating FWHM of Undershoot '''
    minimum = min(data)
    #print 'min undershoot = %f' %minimum
    half_minimum = minimum/2.0
    #print 'half_minimum = %f' %half_minimum
```

```
    for i in range(10):
        if data[undermin_t - i] <= half_minimum:
            point11_t = (undermin_t - i)
            point11 = data[undermin_t - i]
##            print "found point11_t ", point11_t
##            print "point11", point11
            break


    for i in range(30):
        if data[undermin_t + i] >= half_minimum:
            point22_t = (undermin_t + i)
            point22 = data[undermin_t + i]
##            print "found point22_t ", point22_t
##            print "point22", point22
            break

    FWHM_undershoot = point22_t - point11_t
    #print "FWHM_undershoot= %f" %FWHM_undershoot


    ''' CALCULATIONS'''

    # calculating baseline difference
    baseline_diff = aveendbaseline - avebaseline
##    print (baseline_diff)

    # Calculating the relative baseline difference
    rel_baselinediff = (baseline_diff)/(avebaseline)
##    print ("relative baseline", rel_baselinediff)

    # Significant baseline calculation
    sig_baselinediff = np.abs((baseline_diff)/(stdevbaseline))
##    print ("significant baseline", sig_baselinediff)
```

```
    # Calculating the time it takes for the pulse to return to the
                                        baseline
    falltime = understart_t - max_t
##    print ("fall time", falltime)


    # Calculating difference between Max_amplitude and baseline
    actual_data = max_data - avebaseline

    # Calculating signal integral
    signal_data = data[31:97]
    signal_int = sum(i - avebaseline for i in signal_data)
##    print ("signal integral", signal_int)



    ''' SPLINE CHARACTERISATION '''

    ''' Cubic Spline Interpolation'''

    xm = np.linspace(0, 128, 128)
    #print len(xm)
    ym = data
    # Create the interpolating function

    xnew = np.linspace(0, 128, 100000)
    ynew = np.interp(xnew, xm, ym)


    max_ynew = (max(ynew))  # Find the maximum y value from the
                                        interpolated data
    max_xnew_1 = (xnew[ynew.argmax()])
    max_xnew = int(max_xnew_1)

    # Calculating the baseline
```

```
baseline_len = 16
interp_baseline = (ynew[0:baseline_len])


#print (baselinedata)
interp_baseline_sum = sum(interp_baseline)
#lenbaseline = len(baselinedata)
#print lenbaseline
interp_baseline_ave = interp_baseline_sum/baseline_len # This is
                                    the interpolated baseline
                                    value
print 'interp_baseline_ave =', interp_baseline_ave

interp_baseline_diff = [i - interp_baseline_ave for i in
                                    interp_baseline]
interp_baseline_sqdiff = [i**(2.0) for i in interp_baseline_diff
                                    ]
interp_baseline_avesqdiff = (sum(interp_baseline_sqdiff)/len(
                                    interp_baseline_sqdiff))
interp_baseline_stdev = np.sqrt(interp_baseline_avesqdiff)
print 'interp_baseline_stdev =', interp_baseline_stdev
# Calculating threshold
interp_thresholdnumber = 3.0
interp_threshold = interp_baseline_ave + (interp_thresholdnumber
                                    *interp_baseline_stdev)
interp_rising_edge_t = 0.
print 'interp_threshold =', interp_threshold
''' INTERPOLATED PULSE PARAMETERS '''

# Finding the start rising edge
for i in range(100):
    #print i,data[max_t-i]
    if ynew[max_xnew - i] <= interp_threshold:
        interp_rising_edge_t = (max_xnew - i)
        interp_baseline_val = ynew[max_xnew - i]
```

```
        print "found edge ", interp_rising_edge_t
        print "Amplitude", ynew[max_xnew - i]
        break

''' Calculating Full Width at Half Maximum of interpolated Pulse
                                    '''

interp_peak = max_ynew - interp_baseline_ave
interp_halfpeak = interp_peak/2.0 # calculates half the maximum
                                    value
print ('half interp_peak', interp_halfpeak)

# Obtain the time t at point1 of half the peak
for i in range(50):
    #print i,data[max_t+i]
    if ynew[max_xnew - i] <= interp_halfpeak:
        point3_xnew = (max_xnew - i)
        point3 = ynew[max_xnew - i]
        print "found point3_xnew ", point3_xnew
        print "point3", point3
        break

# Obtain the time t at point2 of half the peak
for i in range(2000):
    #print i,data[max_t+i]
    if ynew[max_xnew + i] <= interp_halfpeak:
        point4_xnew = (max_xnew + i)
        point4 = ynew[max_xnew + i]
        print "found point4_xnew ", point4_xnew
        print "point4", point4
        break


# Calculating Full Width at Half Maximum
interp_FWHM = point4_xnew - point3_xnew
```

```
print ('interpolated FWHM = %d' %interp_FWHM)


# Calculting the rise time
interp_rise_time = max_xnew - rising_edge_t
print 'Interpolation Rise Time = ', interp_rise_time


sfile = open('pb018_0_1_spline_results.txt', 'a')


'''
Write the important interpolated pulse values to a textfile file
'''
sfile.write('%d\t %f\t %d\t %f\t %d\t %f\n'%(
                                    interp_rising_edge_t,
                                    interp_threshold,max_xnew,
                                    max_ynew, interp_rise_time,
                                    interp_FWHM))
sfile.close()

'''
Write the important pulse values to a textfile file
'''
dfile.write('%d\t %f\t %d\t %f\t %d\t %f\t %f\t %f\t %f\t %f\n'
                                    %(rising_edge_t,threshold,
                                    max_t,max_data,rise_time,
                                    minimum,FWHM,falltime,
                                    sig_baselinediff,actual_data))
dfile.close()


''' PLOTTING GRAPHS '''

plt.title('Amplitude vs Time')
```

```
    plt.ylabel('Amplitude')

    plt.xlabel('Time')

    plt.plot(xnew, ynew, 'r-', label='Cubic Spline')
    plt.plot(t,data, 'b+',label='data')


    plt.axvline(max_t, c='r', label='t_max_Amp')
##
##    plt.axhline(max_data)

    plt.axvline(max_xnew_1, label='interp_data_max')

##    plt.axvline(rising_edge_t, c='c' , label='t_rising_edge')
##    plt.axvline(interp_rising_edge_t, c='green', label='
                                    interp_rising_edge')
##    plt.axvline(understart_t, c='blue', label='understart_amp')
##    plt.axvline(endrising_edge_t, c='purple', label='
                                    end_risingedge')
    plt.axhline(avebaseline, c='y', label='start_baseline')

##    plt.axhline(interp_baseline_ave, c='k', label='
                                    interp_start_baseline')

##    plt.axhline(minimum, label="undershoot")
##    plt.axvline(undermin_t, c='c', label='t_undershoot')
##    plt.axhline(aveendbaseline, c='green', label='endbaseline')
##    plt.axhline(threshold, c='b', label = 'threshold')
##    plt.axhline(interp_threshold, c='c', label='interp_threshold')
##    plt.axhline(halfpeak, c="purple", label = 'FWHM_Amp')
##    plt.axhline(interp_halfpeak, c='r' , label='FWHM_Spline')
##    plt.axhline(half_minimum, c='g', label= 'FWHM_undershoot')
##    plt.axhspan(0,baseline_val, xmin=0, xmax=rising_edge_t)
```

```
    plt.xlim(0,128)

    plt.ylabel('Amplitude [nm]')

    plt.xlabel('Time [ns]')

    plt.legend()

    plt.show()

##    plt.savefig('pulse')




pp = PdfPages('multipagepb001.pdf')
plt.savefig(pp, format='pdf')
#pp.savefig()
pp.close()
```

A.1 Statistical Analysis Python Analysis code

```
import matplotlib.pyplot as plt
import numpy as np
import random
from matplotlib.ticker import MaxNLocator

fr = open('pb018_0_7_results.txt', 'r')

S = [] # rising_edge_t
N = [] # average baseline value
T = [] # t_threshold
```

```
U = [] # max_t
V = [] # max_amplitude
W = [] # rise_time
X = [] # undershoot
Y = [] # FWHM
Z = [] # Fall time
R = [] # Significant baseline
Q = [] # Change in max_amp and start_baseline

for line in fr:
    A = int(line.split()[0]) # rising_edge_t
    P = float(line.split()[1])
    B = float(line.split()[2]) # t_threshold
    C = float(line.split()[3]) # max_t
    D = float(line.split()[4]) # max_amplitude
    E = float(line.split()[5]) # rise_time
    F = float(line.split()[6]) # undershoot
    G = float(line.split()[7]) # FWHM
    H = float(line.split()[8]) # fall time
    I = float(line.split()[9]) # Significant baseline
    J = float(line.split()[10]) # change in max_amp and
                                    start_baseline

#print E
    S.append(A)
    N.append(P)
    T.append(B)
    U.append(C)
    V.append(D)
    W.append(E)
    X.append(F)
    Y.append(G)
    Z.append(H)
    R.append(I)
    Q.append(J)
```

```
fr.close()

risetime_ave = sum(Z)/len(Z)
risetime_diff = [i - risetime_ave for i in Z]
risetime_sqdiff = [i**(2.0) for i in risetime_diff]
risetime_avesqdiff = (sum(risetime_sqdiff)/(len(Z) - 1))
risetime_stdev = np.sqrt(risetime_avesqdiff)
print len(Z)
print (risetime_ave)
print (risetime_stdev)


#FOR INTERPOLATED DATA

fs = open('pb018_0_1_spline_results.txt', 'r')

s = [] # interp_rising_edge
n = [] # interp_average_baseline
t = [] # interp_threshold
u = [] # max_xnew
v = [] # max_ynew
w = [] # interp_rise_time
x = [] # interp_FWHM

for line in fs:
    a = float(line.split()[0]) # interp_rising_edge
    q = float(line.split()[1])
    b = float(line.split()[2]) # interp_threshold
    c = float(line.split()[3]) # max_xnew
    d = float(line.split()[4]) # max_ynew
    e = float(line.split()[5]) # interp_rise_time
    f = float(line.split()[6]) # interp_FWHM
```

```
    s.append(a)
    n.append(q)
    t.append(b)
    u.append(c)
    v.append(d)
    w.append(e)
    x.append(f)

fs.close()

interp_risetime_ave = sum(w)/len(w)
interp_risetime_diff = [i - interp_risetime_ave for i in w]
interp_risetime_sqdiff = [i**(2.0) for i in interp_risetime_diff]
interp_risetime_avesqdiff = (sum(interp_risetime_sqdiff)/(len(x) - 1
                                    ))
interp_risetime_stdev = np.sqrt(interp_risetime_avesqdiff)
print w
print (interp_risetime_ave)
print (interp_risetime_stdev)

O = [i - j for i,j in zip(s,S)] # Difference in data and
                                    interpolated data rising edge
p = [ i - j for i,j in zip(T,t)] # Difference in data and
                                    interpolated data Threshold
K = [i - j for i,j in zip(v,V)] # Difference in data and
                                    interpolated data maximum
L = [ i - j for i,j in zip(W,w)]  # difference in rise time
M = [i -j for i,j in zip(N,n)] # baseline difference


risetimediff_ave = sum(L)/len(L)
risetimediff_diff = [i - risetimediff_ave for i in L]
risetimediff_sqdiff = [i**(2.0) for i in risetimediff_diff]
risetimediff_avesqdiff = (sum(risetimediff_sqdiff)/(len(L) - 1))
```

```
risetimediff_stdev = np.sqrt(risetimediff_avesqdiff)
print L
print (risetimediff_ave)
print (risetimediff_stdev)

# Plotting the histogram
##binwidth = 0.5

ax = plt.figure().gca()
bins_1 = [0,1,2,3,4,5,6,7,8,9,10] # bins for rise time
bins_2 = [27,28,29,30,31,32,33,34,35,36]
bins_3 = [1,2,6,7,8,9,10,11,12,13,14,15,16,17,18,19]
bins_4 = [0.000,0.001,0.002,0.003,0.004,0.005,0.006,0.007,0.008]



plt.hist(N)



ax.yaxis.set_major_locator(MaxNLocator(integer=True))
ax.xaxis.set_major_locator(MaxNLocator(integer=True))

##plt.grid()
#plt.title('pb07_Rising_Edge_Diff_Hist')
plt.ylabel('Counts')
plt.xlabel('Data_Baseline [nm]')
plt.savefig('Data_Baseline')
```

A.2 Tabular Report For The Various Parameters

The pulse waveform parameters for each of the 64 channels

are shown in the tables below.

Table A.1: Pulse waveform parameters

Rising edge [ns] Max amplitude [nm]	Baseline [nm] Rise time [ns]	Max time [ns] Fall time [ns]			
28	0.075912	35	1.00	7	17.00
28	0.076984	36	1.00	8	16.00
28	0.071804	35	1.00	7	15.00
28	0.076841	35	1.00	7	16.00
28	0.067005	35	1.00	7	14.00
28	0.078309	35	1.00	7	16.00
28	0.077723	35	1.00	7	16.00
28	0.070576	35	1.00	7	16.00
28	0.068884	35	1.00	7	16.00
28	0.064224	34	1.00	6	15.00
28	0.069584	35	1.00	7	16.00
28	0.070760	35	1.00	7	16.00
28	0.065511	35	1.00	7	15.00
28	0.063032	35	1.00	7	15.00
28	0.063060	35	1.00	7	14.00
28	0.065972	35	1.00	7	15.00
28	0.069730	35	1.00	7	15.00
28	0.066340	35	1.00	7	16.00
28	0.072220	35	1.00	7	17.00
28	0.074411	35	1.00	7	17.00
28	0.068038	35	1.00	7	16.00
28	0.074488	35	1.00	7	16.00
28	0.071118	35	1.00	7	15.00
28	0.072668	35	1.00	7	16.00
28	0.074142	35	1.00	7	15.00
28	0.075949	35	1.00	7	16.00
28	0.069248	35	1.00	7	16.00
28	0.070013	35	1.00	7	17.00
28	0.068623	35	1.00	7	16.00
28	0.063295	34	1.00	6	15.00
28	0.074774	35	1.00	7	16.00
28	0.070880	35	1.00	7	16.00
28	0.072940	35	1.00	7	16.00

Interpolated pulse waveform parameters are shown below

Table A.2: Pulse waveform parameters

Rising edge [ns] Max amplitude [nm]	Baseline [nm] Rise time [ns]	Max time [ns] Fall time [ns]			
28	0.069324	35	1.00	7	16.00
28	0.072774	35	1.00	7	16.00
28	0.065332	35	1.00	7	15.00
28	0.067853	35	1.00	7	16.00
28	0.069342	35	1.00	7	16.00
28	0.068480	35	1.00	7	16.00
28	0.069847	35	1.00	7	14.00
28	0.075928	35	1.00	7	16.00
28	0.067120	35	1.00	7	15.00
28	0.066939	34	1.00	6	16.00
28	0.071264	34	1.00	6	16.00
28	0.073463	34	1.00	6	16.00
28	0.078188	36	1.00	8	16.00
28	0.072680	34	1.00	6	16.00
28	0.074653	34	1.00	6	17.00
28	0.070090	34	1.00	6	15.00
28	0.071602	34	1.00	6	14.00
28	0.073896	34	1.00	6	17.00
28	0.074638	34	1.00	6	16.00
28	0.070970	34	1.00	6	15.00
28	0.073697	33	1.00	5	15.00
28	0.073684	34	1.00	6	16.00
28	0.080194	34	1.00	6	15.00
28	0.069799	36	1.00	8	15.00
28	0.072380	34	1.00	6	15.00
28	0.075165	34	1.00	6	14.00
28	0.074471	33	1.00	5	15.00
28	0.077029	33	1.00	5	16.00
28	0.070376	35	1.00	7	16.00
28	0.076868	35	1.00	7	17.00
28	0.077781	35	1.00	7	17.00

Table A.3: Pulse waveform parameters

Rising edge [ns]	Baseline [nm]	Max time [ns]		
Max amplitude [nm]	Rise time [ns]			
17	3.991	35	2592.239	7
20	3.966	35	2414.170	7
35	3.588	35	2438.127	7
35	3.785	35	2435.263	7
18	3.103	34	2564.878	6
35	3.781	35	2422.596	7
35	3.764	35	2459.953	7
20	4.013	35	2431.490	7
35	3.425	35	2718.641	7
34	2.777	34	2759.181	6
35	2.956	35	2518.059	7
21	2.890	35	2521.356	7
34	3.595	34	2711.068	6
20	2.873	35	2642.590	7
34	2.323	34	2478.335	6
21	2.770	34	2517.430	6
34	2.820	34	2448.871	6
35	2.689	35	2468.365	7
35	2.980	35	2373.909	7
35	2.869	35	2377.227	7
35	2.642	35	2525.867	7
20	2.787	35	2371.848	7
34	2.423	34	2490.541	6
35	4.015	35	2756.677	7
19	2.740	35	2341.755	7
35	3.663	35	2486.734	7
35	3.674	35	2701.645	7
35	3.150	35	2515.492	7
20	3.005	35	2598.574	7
19	2.840	34	2827.872	6
35	3.219	35	2460.537	7
21	2.714	35	2333.951	7
35	3.053	35	2274.475	7
21	2.834	35	2513.068	7
35	3.892	35	2583.199	7
20	3.230	34	2689.771	6
19	3.080	35	2633.467	7

Table A.4: Interpolated pulse waveform parameters

Rising edge [ns] Max amplitude [nm]	Baseline [nm] Rise time [ns]	Max time [ns]		
19	3.215	34	2698.514	6
18	3.272	34	2715.585	6
19	2.482	35	2115.870	7
34	3.238	34	2504.272	6
18	2.959	34	2530.721	6
17	3.306	34	2667.889	6
19	3.693	34	2708.539	6
20	3.122	34	2536.508	6
35	3.581	35	2269.622	7
19	3.274	34	2719.063	6
20	3.279	34	2491.905	6
20	2.912	34	2533.001	6
19	3.082	34	2617.202	6
18	3.213	34	2285.216	6
18	3.089	34	2481.110	6
18	2.811	34	2539.597	6
18	2.616	33	2682.716	5
19	2.862	34	2492.722	6
20	3.193	34	2472.995	6
35	3.164	35	2230.106	7
20	3.029	33	2753.783	5
19	3.030	33	2581.613	5
17	2.660	33	2667.485	5
17	3.637	33	2674.783	5
35	3.807	35	2568.024	7
35	4.229	35	2446.014	7
35	3.889	35	2448.214	7

A.3 Parameterised Pulse Waveforms

Fully parameterised pulse waveforms for the 64 pixels are shown below. The figures are in chronological order according to the pixel number. For a description see Fig. 5.1 (a)

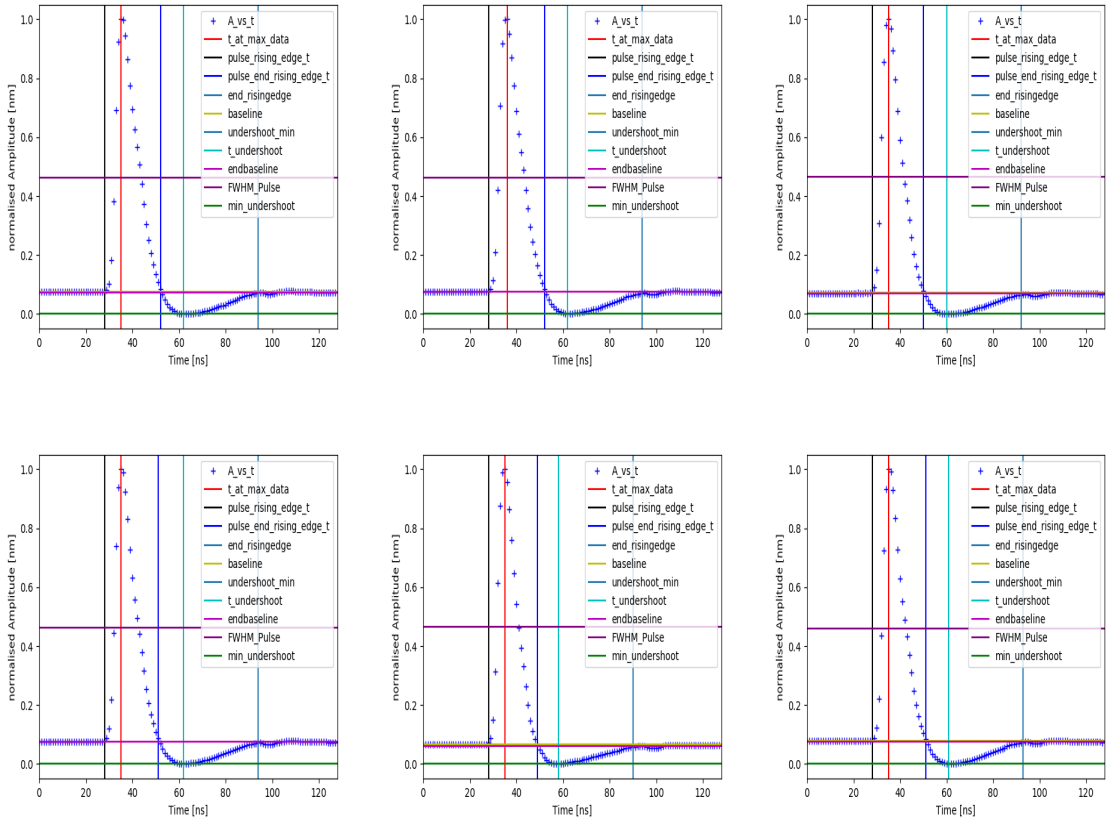


Figure A.1. Parameterised pulse waveforms from pixel 1 to pixel 6

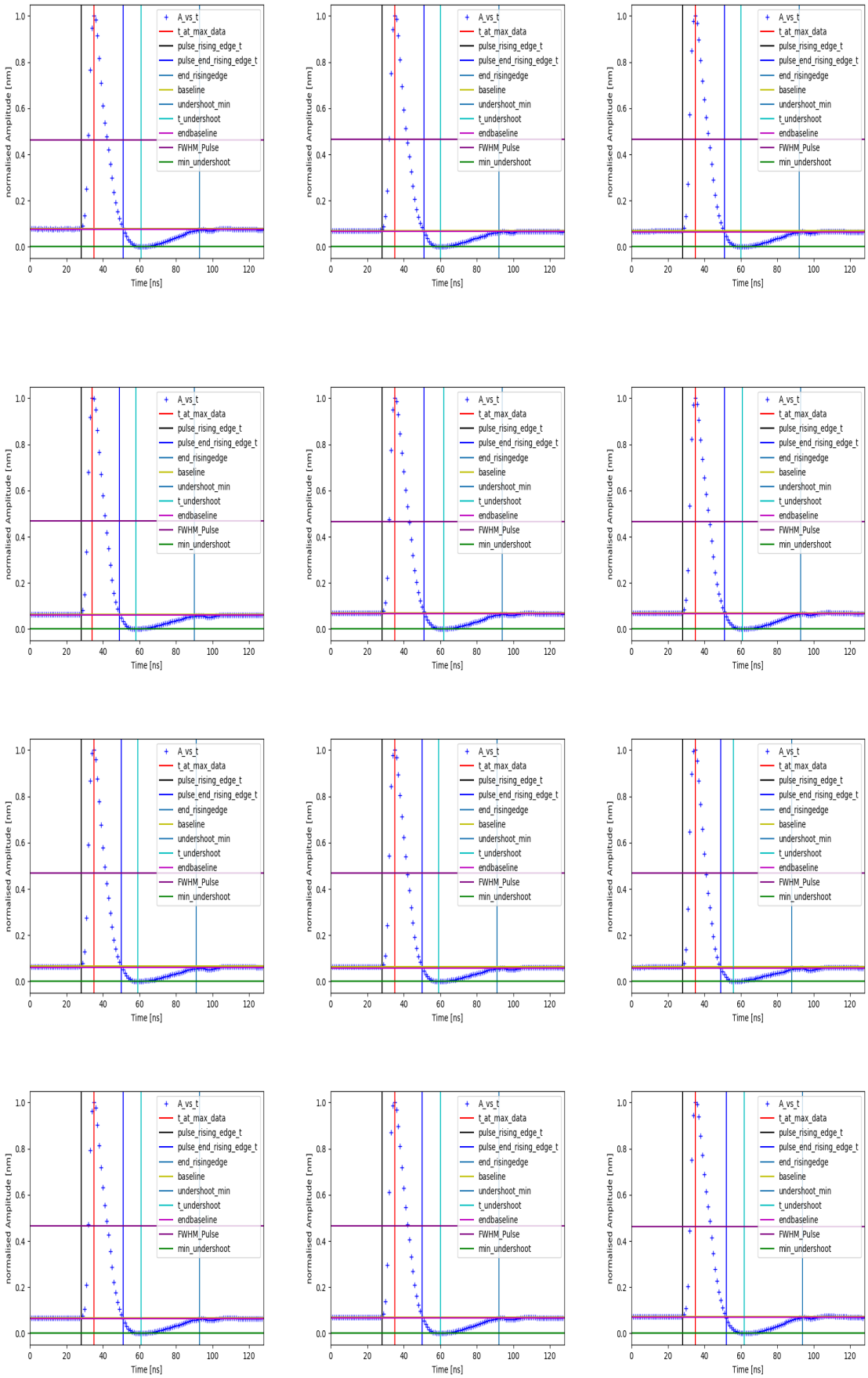


Figure A.2. Parameterised pulse waveforms from pixel 7 to pixel 18

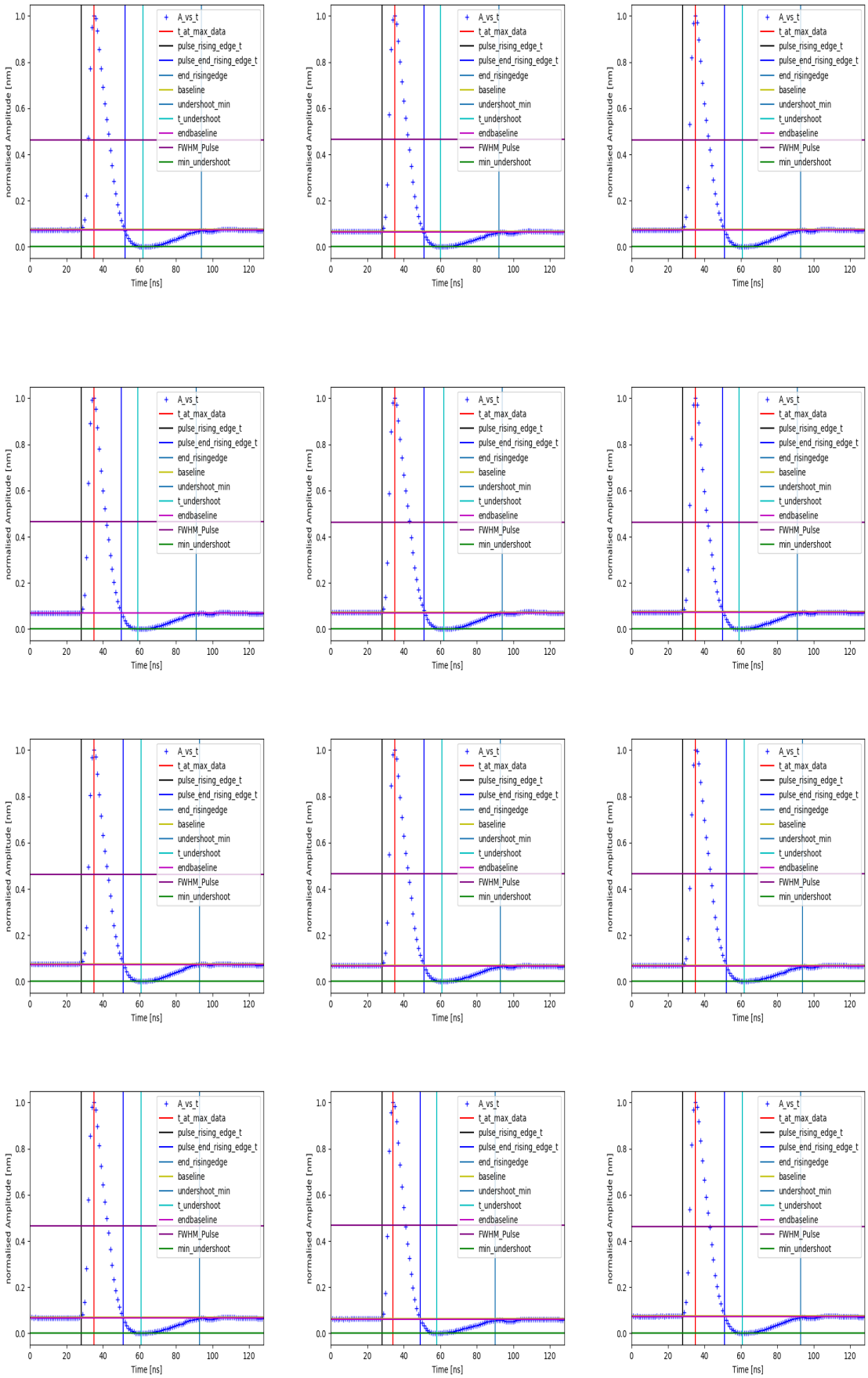


Figure A.3. Parameterised pulse waveforms from pixel 19 to pixel 30

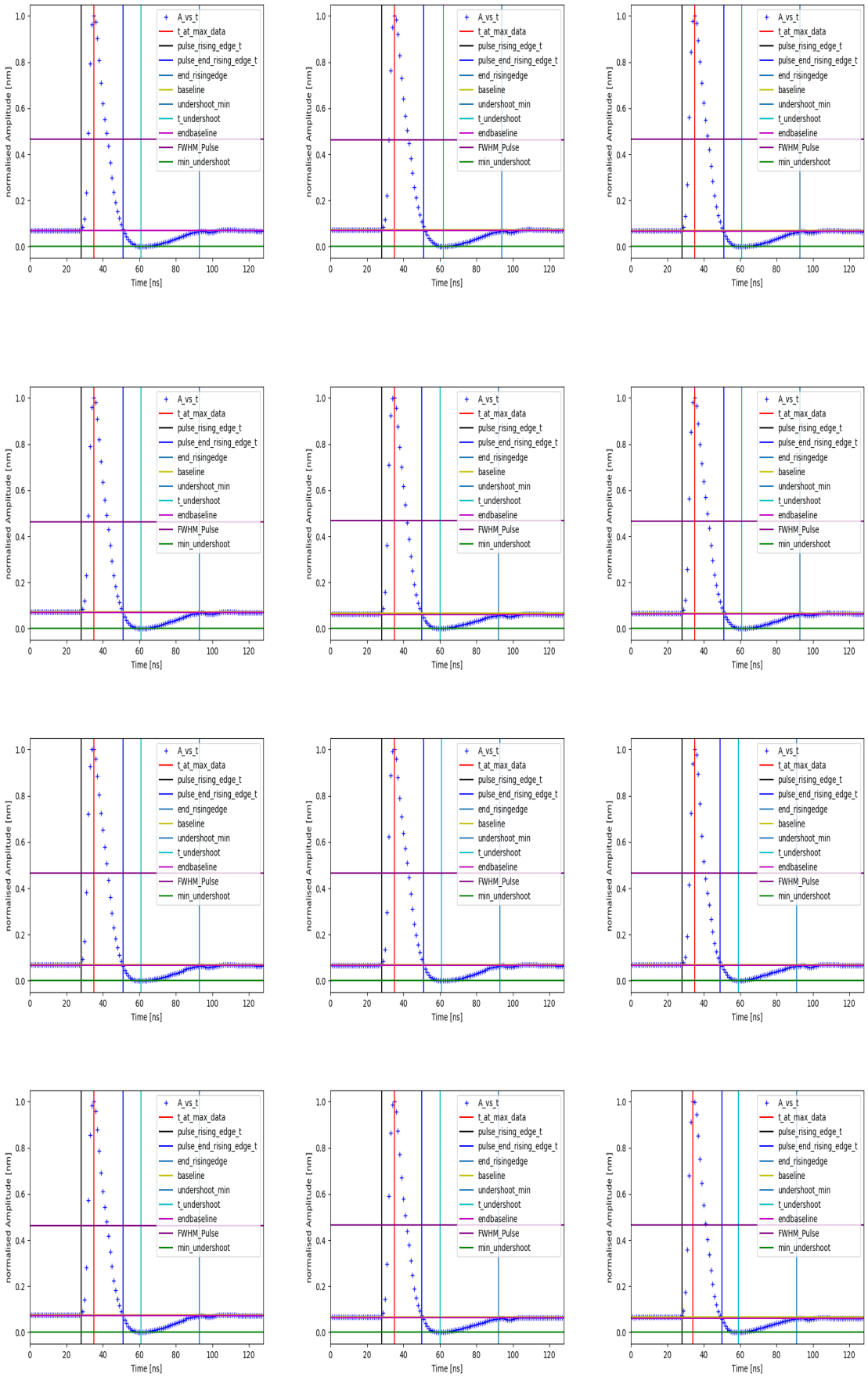


Figure A.4. Parameterised pulse waveforms from pixel 31 to pixel 42

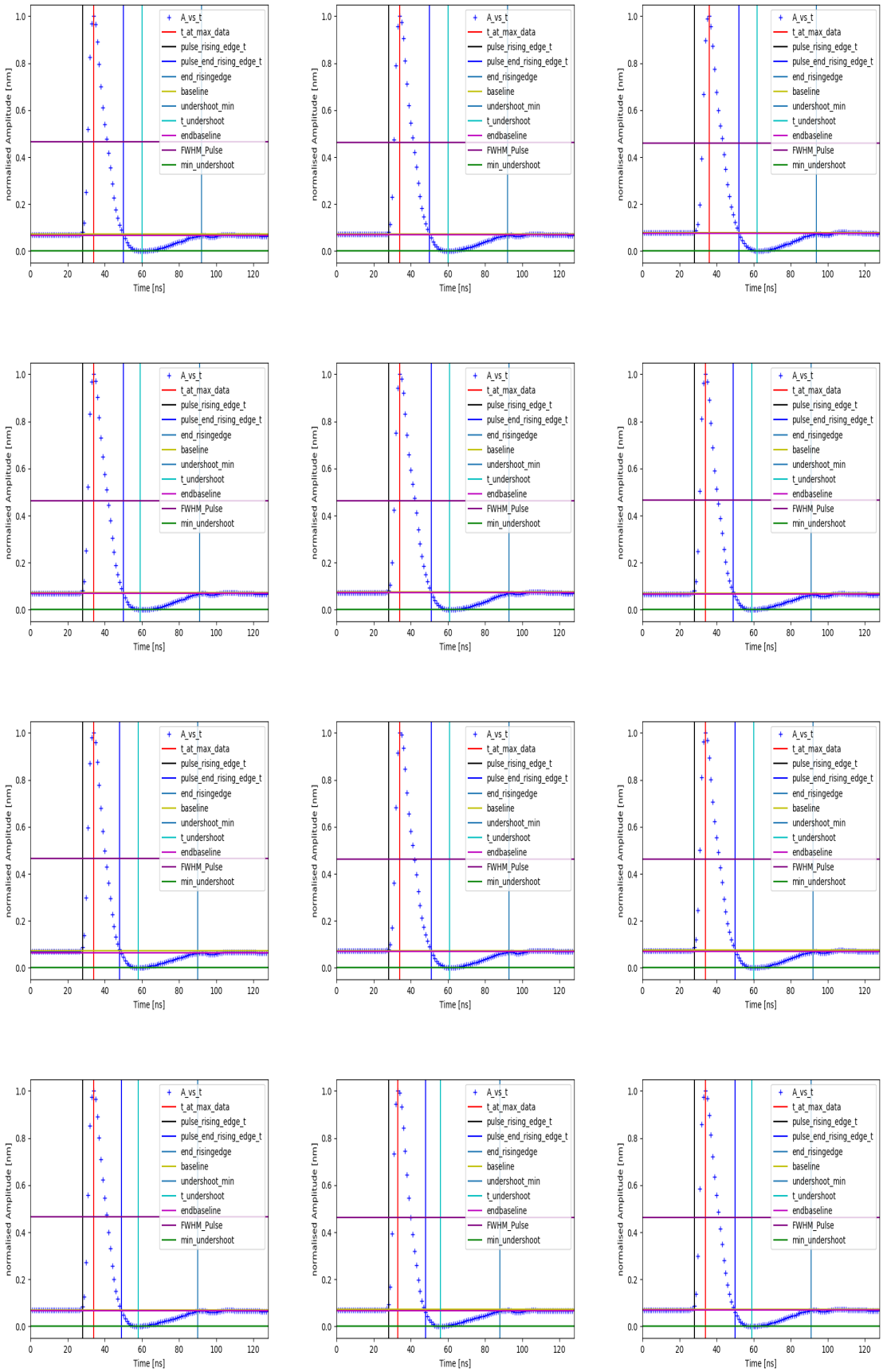


Figure A.5. Parameterised pulse waveforms from pixel 43 to pixel 54

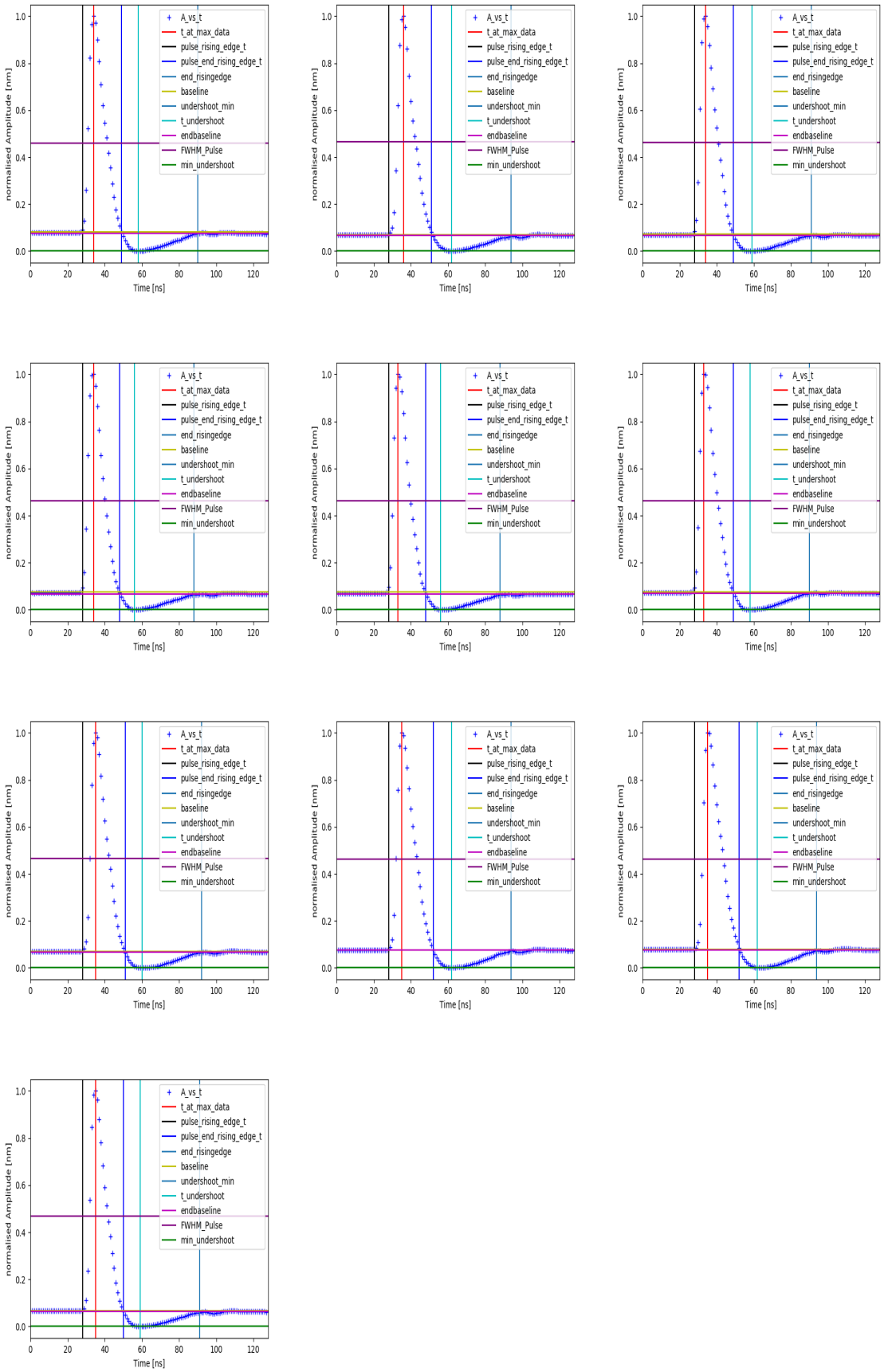


Figure A.6. Parameterised pulse waveforms from pixel 55 to pixel 64