

修士論文

CTA 大口径望遠鏡 2-4 号機における波形記録チップ DRS4 の
サンプリング時間幅較正

茨城大学大学院理工学研究科理学専攻

学生番号 21NM154A

服部 勇大

2023 年 4 月 16 日

概要

超高エネルギーのガンマ線が地球大気に入射すると空気シャワーを生じ、生成された荷電粒子が大気中の光速を超えた際にチェレンコフ光と呼ばれる可視光を放射する。このチェレンコフ光をカメラへ集光することで間接的にガンマ線を観測することが出来る。Cherenkov Telescope Array(CTA) 計画は口径の異なる三種の望遠鏡を南半球と北半球に建設し、20 GeV から 300 TeV 領域のエネルギーのガンマ線を観測することを目指す国際共同実験計画である。三種の望遠鏡のうち最も口径の大きい大口径望遠鏡 (LST) は CTA の中で 20 GeV から 3 TeV の低エネルギー側に感度を持ち、7つの光電子増倍管と波形読み出し回路からなる光検出器 265 モジュールで構成されたカメラを持つ。2018 年 10 月にはラパルマで LST 初号機が完成し、LST2-4 号機も準備が進められている。

数 ns に一回の頻度で混入する夜光由来のバックグラウンドを除去するため、LST のカメラでは DRS4 と呼ばれる GHz でのサンプリングを可能とするアナログメモリを搭載している。DRS4 チップは 1 チャンネルあたり 1024 個のキャパシタを 8 チャンネル有しており、約 1 GHz の周期でスイッチを切り替えて波形情報をキャパシタに電荷として保存し、約 30 MHz の低速で読み出しを行う。また、入力チャンネルから読み出すキャパシタの範囲 (ROI) を指定でき、LST では ROI=40 で読み出ししている。DRS4 チップは波形サンプリングの時間間隔が不均一になるサンプリング時間幅特性を持つ。この特性により望遠鏡で読み出した信号が歪み、エネルギー推定や到来時間を誤ってしまうためこれらの較正が必要である。

本研究では LST2-4 号機でサンプリング時間幅較正を行うためにまず、波形読み出し回路へ波形を入力しない状態での電気ノイズを測定した。また、そのペDESTALデータに関してサンプリング時間幅較正前に必要な補正を行い波形を入力しない状態での LST1 用ボード及び LST2-4 用ボードそれぞれの電気ノイズを評価した。その結果、ペDESTAL補正後のペDESTALノイズがサイン波やテストパルスの波高値と比べて非常に小さいためサンプリング時間幅較正ではそれらの影響は無視できるくらい小さいと考えられる。ただし、特定のピクセルではスパイクのような立ち上がりが混入することがあるため今回はそのピクセルを省いて時間幅較正を行うこととする。

先行研究では実験室で ROI=100 のサイン波での較正を行っていたが、実際の望遠鏡のでは ROI=40 と設定されており、データ取得の設定を自由に変えられない。なので実際の望遠鏡の ROI=40 でサンプリング時間幅較正を行う必要がある。そこで ROI=40 でのサイン波較正を検証する。ローカルキャリブレーションではサイン波によるフィッティングを行うが、従来の全セルのデータ点を用いる方法ではフィット関数とデータ点のズレが大きい。そこでフィッティングに使用するキャパシタについて波高値がバラつきやすい最初の 2 セルと最後 1 セルを取り除くことでサイン波フィッティングが改善した。またグローバルキャリブレーションでは従来のサイン波 1 周期使用する方法からサイン波半周期使用する方法に変更した。これは先行研究で行われていた実験室での ROI=100 から実際の望遠鏡の ROI=40 へと変更したことでサイン波 1 周期を確保できるとは限らないからである。ROI=40、ROI=100 でのサイン波較正に関して電荷分解能に着目して較正精度を調べた。ROI=40 で電荷分解能が 1.86% であり、従来の ROI=100 でのサイン波による較正の電荷分解能が 1.96% であることから同程度の較正精度であることが確かめられた。今回用いた二枚の波形読み出し回路ボード A ボード B について、ボード A では較正前が電荷分解能が $3.466 \pm 0.029\%$ から較正後には $1.992 \pm 0.018\%$ 、ボード B では電荷分解能が $3.375 \pm 0.026\%$ から較正後に $1.818 \pm 0.014\%$ と各ボードそれぞれで電荷分解能が較正後に約 1.5 ポイント程度改善し、電荷分解能を 3% 以下に抑えるという目標を達成することが出来た。これにより、LST2-4 号機でのサンプリング時間幅較正の較正手法を確立することが出来た。

Abstract

When ultra-high energy gamma rays enter the earth's atmosphere, they produce an air shower, and when the charged particles generated exceed the speed of light in the atmosphere, they emit visible light known as Cherenkov light. By focusing this Cherenkov light onto a camera, gamma rays can be observed indirectly. The Cherenkov Telescope Array (CTA) project is an international collaboration to observe gamma rays in the energy range of 20 GeV to 300 TeV by constructing three IACT telescopes with different apertures in the southern and northern hemispheres. The largest of the three telescopes, the Large Scale Telescope (LST), has sensitivity in the low energy side of the CTA, from 20 GeV to 3 TeV, and has a camera composed of 265 modules of photodetectors, consisting of seven photomultiplier tubes and a waveform readout circuit. In October 2018, the first LST was completed and the LST2-4 is being prepared.

In order to eliminate backgrounds of noctilucent origin that are introduced every few ns, LST's cameras are equipped with an analog memory called DRS4 that allows sampling at GHz. The DRS4 chip has 8 channels with 1024 capacitors per channel, which are switched at a period of about 1 ns. The DRS4 chip has eight channels with 1024 capacitors per channel, which are switched at a period of about 1 ns to store waveform information as a charge in the capacitors and read it out at a low speed of about 30 MHz. The range of capacitors to be read out from the input channels (ROI) can be specified, and LST reads out at ROI=40. The DRS4 chip has a sampling time width characteristic that results in non-uniform time intervals for waveform sampling. This characteristic distorts the signal readout by the telescope, resulting in incorrect energy estimation and arrival time, so calibration is necessary.

In order to perform sampling time width calibration on LST2-4, we first measured the electrical noise without waveform input to the waveform readout circuit. The pedestal data was corrected as necessary before the sampling time width calibration, and the electrical noise of the LST1 board and the LST2-4 board were evaluated without waveform input. As a result, the pedestal noise after pedestal correction is very small compared to the wave height of the sine wave and test pulses, so their influence is considered negligible in the sampling time width calibration. However, since spike-like noise may be introduced at certain pixels, we will omit those pixels for the time width calibration.

In LST2-4, ROI=40 is set, and since the data acquisition settings cannot be freely changed, it is necessary to perform the sampling time width calibration with ROI=40. Therefore, sine wave measurement at ROI=40 is performed to verify the time width calibration by sine wave. In local calibration, sine wave fitting is performed, but the conventional method of using the data points of all cells results in a large discrepancy between the fitting function and the data points. Therefore, the sine wave fitting was improved by removing the first two cells and the last cell of the capacitor used for fitting, which are prone to variations in wave height values. In addition, the global calibration method was changed from the conventional method of using one sine wave period to the method of using half a sine wave period. The calibration accuracy of the sine wave calibration at ROI=40 and ROI=100 was examined focusing on the charge resolution. The charge resolution of the conventional sine wave calibration at ROI=100 was 1.96%, which was confirmed to have a similar calibration accuracy. For the two waveform readout circuit boards A and B used in this study, the charge resolution of board A went from $3.466 \pm 0.029\%$ before calibration to $1.992 \pm 0.018\%$ after calibration, and that of board B went from $3.375 \pm 0.026\%$ to $1.818 \pm 0.014\%$ after calibration, respectively. The charge resolution improved significantly by about 1.5% after calibration, achieving the LST requirement of keeping the charge resolution below 3%. This allowed us to establish a calibration method for the sampling time width calibration of LST2-4.

目次

第 1 章	序論	5
第 2 章	ガンマ線天文学	8
2.1	ガンマ線天文学	8
2.2	ガンマ線バースト	9
第 3 章	Cherenkov Telescope Array(CTA) 計画	11
3.1	解像型大気チェレンコフ望遠鏡	11
3.2	Cherenkov Telescope Array(CTA) 計画	12
3.3	CTA 大口径望遠鏡 (LST)	12
3.4	焦点面検出器	12
第 4 章	大口径望遠鏡 2 から 4 号機における DRS4 諸特性の較正	19
4.1	目的	19
4.2	DRS4 の諸特性の較正方法	19
4.3	測定及び解析環境	22
4.4	各較正後のペDESTAL分布	24
4.5	議論	32
第 5 章	大口径望遠鏡 2 から 4 号機用におけるサンプリング時間幅の較正	34
5.1	目的	34
5.2	LST1 でのテストパルスを用いたサンプリング時間幅較正	36
5.3	LST2-4 でのサイン波を用いたサンプリング時間幅較正	37
5.4	測定および解析環境	39
5.5	大口径望遠鏡初号機におけるサンプリング時間幅較正	41
5.6	大口径望遠鏡 2 から 4 号機用ボードでのサンプリング時間幅較正	42
第 6 章	まとめと今後の展望	55
6.1	ROI=40 でのサイン波によるサンプリング時間幅を計算するスクリプト	56
6.2	サイン波でのサンプリング時間幅較正の実験及び解析マニュアル	58
	謝辞	61
	参考文献	62

目次

2.2.1	BATSE で検出された GRB の継続時間の分布 [13]	9
3.3.1	完成した LST1 の写真 [9]	12
3.3.2	CTA 北サイトの完成予想図 [17]	13
3.4.1	ライトガイド	13
3.4.2	PMT の基本構造 [20] 入射窓に入射光が入射することで光電効果し、飛び出した光電子が後段のダイノードで増幅され読み出される。	14
3.4.3	LST1 の波形読み出し回路のブロック図 [18]	15
3.4.4	アナログメモリの回路図のイメージ [21]	16
3.4.5	DRS4 チップのブロック図 [22]	17
4.2.1	DRS4 の 2019 年 5 月に LST1 で測定されたペDESTAL の平均値の分布 [30]	20
4.2.2	dt の定義 [18]	21
4.2.3	2020 年 6 月に LST1 で測定されたペDESTAL にスパイクが混入した際の波形。縦軸が adc 値で横軸がキャパシタ id。この終えですたる波形はペDESTAL 補正及び dt 補正語の波形である。×はスパイクを記録したキャパシタとそれに対応する adc 値を示しており周囲のキャパシタより 50adc 値ほど高くなっていることが分かる。[30]	22
4.3.1	測定のセットアップ	23
4.3.2	実験及び解析に使用したツール等	24
4.3.3	解析に使用するペDESTAL 測定データ	24
4.4.1	図 4.4.1 に横軸キャパシタ id、縦軸読み出された回数	25
4.4.2		25
4.4.3	左図は LST1 号機用 (1 万イベント)、右図は LST2-4 号機用 (10 万イベント) の読み出し回路での残留時間 dt と各キャパシタに記録された adc 値の関係。時間が経過するほど各キャパシタの adc 値は小さくなることが分かり、LST1 号機用と 2-4 号機用では異なる分布であることが分かる。switching point は残留電荷がほぼ 0 となる時間を表しており、LST1 では約 100ms、LST2-4 では約 10ms 程度である。また上部に見えるもやのような分布はスパイク特性によるものである。	26
4.4.4		27
4.4.5		28
4.4.6		29
4.4.7		30
4.4.8		31

4.4.9	一つのキャパシタに現れるスパイクのような立ち上がりを記録したキャパシタ iD	32
4.5.1	32
5.1.1	LST で時間幅の不均一性を完全に補正した際の限界。Kobayashi, et al.(ICRC 2021)[29] より一部改変。赤線が high gain での青線が Low gain でのモンテカルロシミュレーションで調べられた入力電荷に対する電荷分解能の結果	35
5.1.2	サンプリング時間幅の不均一性のイメージ [30]。横軸が時間、横軸の紫色の線がサンプリング時刻を表している。サンプリング時間間隔を一定として再構成すると右図のように入力信号の波形を歪めてしまう。	35
5.1.3	電荷計算のイメージ	36
5.2.1	テストパルスを用いた較正方法のイメージ [30]。横軸が時間、縦軸が adc 値。赤線はサンプリング時刻を表しており、隣り合う赤線の間隔をサンプリング時間幅とする。橙色の線は隣り合うサンプリング時刻の midpoint を表している。	37
5.3.1	ローカルキャリブレーション [30]	38
5.3.2	グローバルキャリブレーション [30]	39
5.3.3	ROI=40 でのグローバルキャリブレーション	39
5.4.1	測定のセットアップ	40
5.5.1	テストパルスを使用したサンプリング時間幅較正について、較正前 (左図)、較正後 (右図) の高ゲイン系統 1 ピクセルでの積分電荷のヒストグラム	42
5.6.1	ROI=40 でのサイン波測定のデータ点とフィット関数	43
5.6.2	ROI=100 及び ROI=40 でのローカルキャリブレーション後の電荷分解能	43
5.6.3	ROI=40 でのパワースペクトル解析	44
5.6.4	ROI=1024 でのパワースペクトル解析	45
5.6.5	ROI=40 でのローカルキャリブレーション	45
5.6.6	ROI=40 でのローカルキャリブレーション。左図が較正前の電荷量のヒストグラム。電荷分解能は $3.48 \pm 0.03\%$ 、右図が ROI=40 でのローカルキャリブレーションから時間幅較正を行って求めた電荷量のヒストグラムである。電荷分解能は $1.89 \pm 0.02\%$	46
5.6.7	各ピクセル毎の時間幅のヒストグラム	47
5.6.8	pix=0 でのキャパシタ iD に対する較正後の時間幅	48
5.6.9	ROI=40 でのローカルキャリブレーション+グローバルキャリブレーション。左図が較正前の電荷量のヒストグラム。電荷分解能は $3.48 \pm 0.03\%$ 、右図が ROI=40 でのローカルキャリブレーションから時間幅較正を行って求めた電荷量のヒストグラムである。電荷分解能は $1.87 \pm 0.012\%$ で 1.6 ポイントほど電荷分解能が改善した。	49
5.6.10	ROI=40 でのローカルキャリブレーション。左図がローカルキャリブレーション後の電荷量のヒストグラム。電荷分解能は $3.48 \pm 0.03\%$ 、右図が ROI=40 でのローカルキャリブレーション+グローバルキャリブレーション後の電荷量のヒストグラムである。電荷分解能は $1.87 \pm 0.02\%$	50
5.6.11	二枚の波形読み出し回路 AB それぞれについて各 7 ピクセルでのサイン波による時間幅較正後の電荷分解能の平均値と誤差。それぞれ 10 回測定し 10 回分の電荷分解能の平均値をプロットしており、エラーバーは標準誤差としている。	50

5.6.12	左図が ROI=100 のサイン波での時間幅較正後の電荷量のヒストグラム、右図が ROI=40 のサイン波での時間幅較正後の電荷量のヒストグラム	51
5.6.13	52
5.6.14	横軸がピクセル番号、縦軸が電荷分解能。青線が時間幅較正前であり、黄色線が時間幅較正後の結果である。各データ点は 10 回測定分の平均値でありエラーバーは標準誤差を用いている。	53
5.6.15	各ボードごとのサイン波によるサンプリング時間幅較正前後の電荷分解能の変化	54
6.1.1	ROI=40 でのサイン波を用いた較正を行うスクリプトの簡易的なプログラム構成	57
6.2.1	58
6.2.2	59

第 1 章

序論

宇宙で起きる高エネルギー現象の解明にガンマ線は適している。何故ならば、ガンマ線は電荷をもたないために磁場に曲げられることなく直進して地球に飛来するからである。これによりガンマ線の到来方向の情報は失われにくく、どのような天体でガンマ線が放射されたか知ることが出来る。また、検出したガンマ線のエネルギーを測定することが出来ればどのような過程で放射されたか推定することも期待できる。故にガンマ線の観測は宇宙線陽子の加速機構やガンマ線バーストの放射機構などの現代の天文学の問題の解決の鍵となることが期待されており、今まで衛星や望遠鏡によって観測が進められてきた。

高エネルギーガンマ線が地球大気へ入射すると大気中の原子核と電子陽電子対生成し、そうして生じた荷電粒子が大気中の原子核の近傍を通過する際に制動放射によってガンマ線を放射し再び電子陽電子対生成を起こす。こうした相互作用を繰り返すことでそのたびに荷電粒子が増加し空気シャワーを形成する。この連続的な反応は制動放射の臨界エネルギーである約 84 MeV にまで荷電粒子のエネルギーが小さくなることで対生成が生じなくなり電離損失が卓越する。その後は荷電粒子が大気に吸収され荷電粒子の数は減少に転じていく。この空気シャワーの形成の過程で生じた荷電粒子が大気中の光速を超える速度で運動するとチェレンコフ光とよばれる可視光を放射し、ガンマ線の入射軸から約 1 度の広がり地上に降り注ぐ。

ガンマ線は地球大気による吸収のために宇宙ガンマ線そのものを地上で観測することは難しいが、空気シャワーから生じたチェレンコフ光を地上に設置した望遠鏡で観測することで間接的にガンマ線を検出することが出来る。このような望遠鏡を大気チェレンコフ望遠鏡と呼ぶ。現行の大気チェレンコフ望遠鏡は焦点面に設置した多数の光検出器モジュールからなるカメラへと集光しカメラ上での分布や光子数を記録している。このような大気チェレンコフ望遠鏡を特に解像型大気チェレンコフ望遠鏡 (Imaging Atmospheric Cherenkov Telescope, IACT) と呼ばれている。地上の望遠鏡で超高エネルギーガンマ線由来のチェレンコフ光を検出するにはハドロン由来のシャワーからの成分がバックグラウンドとして混入してしまうためハドロン由来のシャワーとガンマ線由来のシャワーを識別する必要がある。ハドロンシャワーとガンマ線由来のシャワーでは異なる相互作用によってシャワーが発達するためにシャワーの広がりなどが異なる。そうした地上で観測されるイメージの違いからガンマ線由来のシャワーとハドロン由来のシャワーを識別することが出来る。この識別手法をイメージング大気チェレンコフ法という。現行の IACT では複数の IACT を用いて同時に観測するステレオ観測が主流となっており、複数台による観測により検出有効面積を稼ぎつつガンマ線の到来方向をより精度よく推定することが出来る。

Cherenkov Telescope Array (CTA) 計画はこれまで確立された IACT の技術をさらに発展させ、大規模な望遠鏡アレイを設置しより広大な有効面積を達成することで既存の望遠鏡の 10 倍という高感度での観測を達成し 20 GeV から 100 TeV のエネルギー領域をカバーすることを目指す国際共同実験計画である。大中小の異なる口径の三種の望遠鏡をアレイ状に北半球のスペイン領カナリア諸島ラパルマ及びチリのパラナルに建設

することで全天観測を行う。現在は三種の望遠鏡の内、最も口径が大きい大口径望遠鏡初号機がラパルマに建設され現在試験運用されており、大口径望遠鏡 2 号機以降の建設の準備が進められている。大口径望遠鏡 (Large-Sized Telescope, LST) は 20 GeV から 3 TeV の CTA の中で低エネルギー領域のガンマ線の感度を持つ。LST の焦点面のカメラは 7 本の光電子増倍管 (Photo Multiplier Tube, PMT) と波形読み出し回路 1 枚から構成される光検出器モジュールが 265 モジュール (1855 ピクセル) で構成されている。反射鏡は 198 枚の分割鏡で構成されており放物面をなすように設計されている。放射面であることによりカメラで検出される光子の同時性を保つことが出来、チェレンコフ光の到来するタイミング情報からシャワーの再構成に用いられる。LST で観測をする際に LST が低エネルギー側に感度をもつため、数百 MHz の頻度で星の明かりなどに由来する夜光のバックグラウンドの混入が問題となる。そこで LST では 1 GHz での波形サンプリングが可能である Domino Ring Sampler version 4 (DRS4) と呼ばれるアナログメモリを波形読み出し回路に搭載している。

DRS4 チップは 1 チャンネルにつき 1024 個のキャパシタの入力チャンネルが 9 チャンネルあり、そのうち 8 つのチャンネルが PMT からの入力信号の波形サンプリングに使われ、残りの 1 チャンネルはサンプリング時間の揺らぎによる波形の歪みを補正する測定を行うために確保されている。DRS4 の入力部分のスイッチ切り替えは多数のインバータを連結した回路上をパルス信号が伝搬することで行われており、LST では DRS4 の 4 つのチャンネルをカスケード接続させることで約 $4 \mu s$ のメモリ深さを確保している。また入力チャンネルのうち読み出すキャパシタの範囲を指定して読み出しする。この範囲を Region of Interest (ROI) と呼び読み出すキャパシタの数をスライスと表現する。LST では ROI を 40 スライスと設定している。こうして DRS4 チップから得られた波形情報にはキャパシタ固有の特性をいくつか持っていることが知られている。各キャパシタ毎に固有のベデスタル波高値を持つベデスタル特性、一度電荷を蓄えたキャパシタが放電し終える前に再度電荷を蓄えてしまうことで本来よりも電荷が大きくなってしまう残留電荷特性、キャパシタに蓄えた電荷が周りのキャパシタよりも高くなるものが混入するスパイク特性、キャパシタの切り替えを行うスイッチ回路に用いている半導体の個性により波形サンプリングの時間間隔にバラつきが生じるサンプリング時間幅特性が挙げられる。これらの特性により、DRS4 チップから読み出された波形の電圧値を変化させてしまうため電荷量に変化し、望遠鏡で捉えた光子量の推定を誤ってしまう。サンプリング時間幅特性はさらに時間情報までも歪めてしまうため入射光量だけでなくガンマ線の到来方向の推定に影響を及ぼすと考えられる。各特性の補正方法は DRS4 チップを開発したスイス ボール・シェラー研究所や現行の IACT で DRS4 を用いている MAGIC や CTA で研究されてきた。CTA の日本チームでは信号波形のサンプリング時間幅特性の補正方法の確立を目指して検証が行われており、2011 年佐々木修論 (茨城大学) で読み出し回路に搭載している Field Programmable Gate Arrays (FPGA) から生成できるクロック信号 (矩形波) を用いた方法、2016 年の谷川修論 (京都大学) ではスローコントロールボードから生成できるようになったテストパルスを用いた方法、2017 年の野崎修論では LST2 号機以降用に改良された読み出し回路で生成できるサイン波を使った方法が挙げられる。また、2021 年の野上修論 (茨城大学) では LST1 のカメラ全 1855 ピクセルに対して行ったテストパルスを用いた較正及び、LST2 号機以降の読み出し回路において ROI=100 でのサイン波とテストパルスを用いた較正について電荷分解能と時間分解能の観点で補正精度の比較がなされた。本研究ではまず、LST2-4 号機でのサンプリング時間幅較正の較正手法を確立するためにまず、サンプリング時間幅特性を較正する前に補正するベデスタル特性、スパイク特性、残留電荷特性について述べた後に、LST1 号機及び 2-4 号機用の波形読み出し回路へ波形を入力しない状態での電気ノイズについて各特性の補正を行いベデスタルノイズを評価し、パルスを入力した際の電気ノイズの影響を評価する。その後、LST2-4 号機用の波形読み出し回路を用いて ROI=40 でのテストパルス及び LST2-4 号機でのサイン波を用いたサンプリング時間幅特性の較正を行う。また、LST2-4 号機でのサンプリング時間幅較正を達成するために電荷分可能を調べ LST の要求仕様を満た

しているかを調べる。

第 2 章

ガンマ線天文学

2.1 ガンマ線天文学

宇宙空間には $10^9 \text{eV} (\text{GeV})$ を超えるような超高エネルギーガンマ線が飛び交っており、それらの一部は地球大気へと降り注いでいる。ただし、ガンマ線は地球大気による吸収のため地上での観測が不可能であるため高度 40km 以上の気球での観測や衛星での観測が不可欠であった。そうした制約もありガンマ線天文学は X 線天文学と比べるとより長い歴史があるが進歩が遅れていた。天体からのガンマ線が初めて確実に捉えられたのは 1967 年アメリカによる Orbiting Solar Observatory 3 衛星による観測であった。[1][2]

一方で、地上核実験の監視のために打ち上げられた vela 衛星のデータから短時間に大量のガンマ線が放射されるガンマ線バーストを検出されていたことが報告された。[3] その後、1989 年に打ち上げられた 100keV から 1 MeV のガンマ線を観測した Granat 衛星や 1991 年に打ち上げられた数十 keV から 10GeV のガンマ線を観測したコンプトンガンマ線衛星 (CGRO) によってガンマ線天文学が目覚ましく進歩していった。[4]1990 年代に入ると大気チェレンコフ望遠鏡の技術が進展し地上から TeV 領域の超高エネルギーガンマ線の観測がなされるようになった。

21 世紀になると、20 MeV から 300 GeV のガンマ線に感度を持つフェルミガンマ線宇宙望遠鏡 (Fermi) が打ち上げられ、これまでに 3000 を超えるガンマ線天体を発見し、高エネルギーガンマ線天文学に貢献してきた [5]。ガンマ線のエネルギーが数十 GeV を上回るような場合、飛来するガンマ線の到来する数が減ってくるためにより大きな検出器が求められる。だが検出器が大きくなるに伴って人工衛星そのものも大きく設計せざるを得ず、費用の増加などの理由から実現が難しく、人工衛星による数十 GeV 以上のガンマ線の観測は困難であった。そこでエネルギーが数十 GeV を超えるようなガンマ線を観測する際は、ガンマ線と地球大気の子核との相互作用によって引き起こされる電磁カスケードシャワーを観測することによって間接的にガンマ線の検出をする大気チェレンコフ望遠鏡が用いられ、大気チェレンコフ望遠鏡である H.E.S.S.[6]、MAGIC[7]、VERITAS[8]、CTA[9] がガンマ線天文学の発展に貢献をしている。

2.2 ガンマ線バースト

ここで CTA の観測対象となる天体及び天体現象の内、ガンマ線バーストについて紹介する。

ガンマ線バースト (Gamma Ray Burst, GRB) とは数秒から数十秒という短い時間で 10^{53}erg ものガンマ線を放出する突発的な天体現象である [10]。典型的には地球から数十億光年と宇宙論的距離で発生することが分かっており、Swift 衛星の観測で地球から約 132 億光年の距離で発生したのも確認されている [11]。GRB はその放射時間の短い Short GRB と長い Long GRB に分類することが出来る [12]。図 2.2.1 に BATSE で検出された GRB の継続時間の分布を示す。この放射時間の長さで GRB を放射した中心駆動天体が異なると予想されており、Short GRB では中性子星同士の連星が重力放出でその軌道を徐々に縮めていき合体した際に起こるという説や、Long GRB では大質量星の重力崩壊による放射と考えられている。

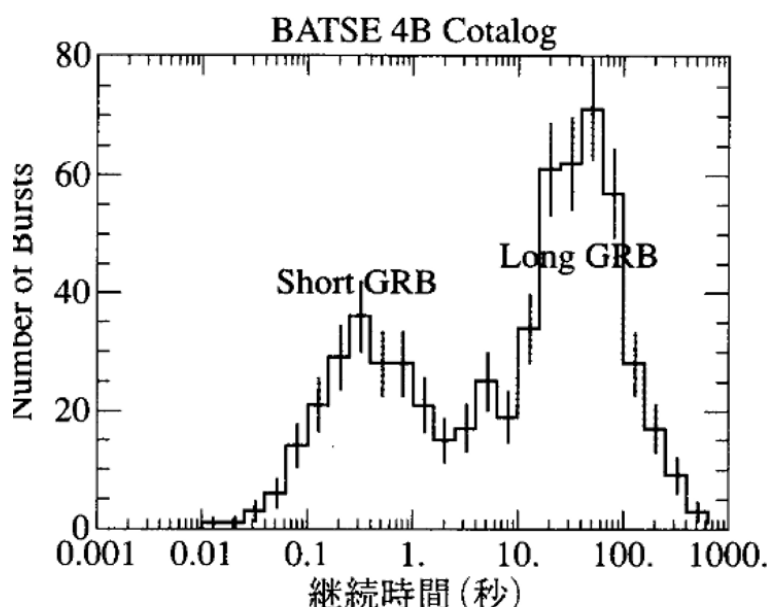


図 2.2.1: BATSE で検出された GRB の継続時間の分布 [13]

現在のガンマ線バースト (Gamma Ray Burst, GRB) の謎の一つが GeV 以上の高エネルギー放射である。GRB の数十 GeV 以上のガンマ線放射の例として GRB 130427A が挙げられる [14]。この GRB から放射されたガンマ線はバースト発生後 4 分で最大の 92 GeV に達し、その後 9 時間もの間 32 GeV のガンマ線が観測された。この放射を標準的な残光の放射機構である電子シンクロトロン放射と考えると極端な仮説が必要になる。残光とした数十 GeV 以上の放射は遅れて始まっていることから電子シンクロトロン放射ではなく、逆コンプトン散乱などの放射機構が考えられたが結論が出なかった。この問題を解決する糸口の一つが 100 GeV 以上の Very High Energy (VHE) 領域のガンマ線観測である。もし VHE ガンマ線が電子シンクロトロンで説明できないと結論づけられたのなら別の放射機構で解釈する協力的な根拠となる。

そこで、その VHE 領域の観測で活躍する望遠鏡こそが地上でガンマ線を間接的に観測することが出来る大気チェレンコフ望遠鏡である。MAGIC 望遠鏡は口径 17m の大気チェレンコフ望遠鏡 2 台によって約 50 GeV から 100 TeV のガンマ線を観測する望遠鏡で有効検出面積は人工衛星と比較して 4 桁大きい。ただし、

視野が数度と狭いため、GRB 発生時には衛星からの GRB の情報を受信して即座に望遠鏡を GRB へ向けて観測を行うこととなっていた。2019 年 1 月 14 日 20 時 57 分 03 秒 (世界時)、GRB 190114C が発生した [15]。衛星からの情報を得た MAGIC 望遠鏡は発生後 50 秒後に追尾を始め約 7 秒後に観測を開始した。月光による背景光が月のない空の約 6 倍であったためエネルギー閾値は 300 GeV と高かったが、今までのいかなる天体からも観測されたことのない高エネルギーガンマ線を検出することに成功した。このガンマ線のエネルギーは TeV にも達し、GRB からの放射エネルギーを 1 桁以上も更新することとなった。こうして史上初めて GRB からの TeV ガンマ線を地上のガンマ線望遠鏡で検出することに成功した。この観測結果から VHE ガンマ線放射の機構としてはシンクロトロン放射を作り出す電子によってシンクロトロン光子を逆コンプトン散乱でたたき上げる Synchrotron Self-Compton (SSC) と呼ばれる放射機構でよく説明できることが分かった。

ガンマ線バーストの更なる解明の鍵となることが期待されている次世代の大気チェレンコフ望遠鏡が Cherenkov Telescope Array (CTA) 計画である [9]。CTA は口径の異なる大中小の望遠鏡を複数設置しステレオ観測を行う望遠鏡であり、これまでの望遠鏡の 10 倍深い感度での観測を達成し 20 GeV から 300 TeV までのエネルギー領域を観測することを目指す国際共同実験計画である。TeV 領域で現在より 10 倍深い感度を達成することは CTA 以前の望遠鏡では不可能であり、Fermi 衛星より大型の衛星の実現もまた困難である。GRB に関しては口径の異なる望遠鏡のうち大口径望遠鏡 (LST) だけが衛星からの情報を受信し高速回転して観測をすることを可能としており GRB の観測にも適している。また GRB 190114C の観測を受けて GRB で TeV でのガンマ線が確認されたという知見もあるため TeV での GRB の観測が期待できる。また、GRB の放射機構の解明には GeV 領域の低エネルギー閾値観測も重要である。より大きな謎である即時放射や初期残光放射の起源を解明するためには 1 分程度以下での観測開始が求められ、また短時間での時間変化を追うためにはできるだけ低いエネルギー閾値でガンマ線の統計量を稼ぐ必要があるからである。つまり CTA のなかでも低エネルギー側に感度を持ち、高速回転により衛星からのアラートを受信しすぐに GRB の観測を開始できる LST は GRB にとって重要な天体現象である。また CTA では Fermi 衛星のデータと合わせて 20 MeV から 300 TeV までの広いエネルギー領域でのガンマ線観測が可能となるため、その他活動銀河核、超新星残骸など更なるガンマ線天文学の発展が期待されている。

第 3 章

Cherenkov Telescope Array(CTA) 計画

3.1 解像型大気チェレンコフ望遠鏡

3.1.1 空気シャワー

地球大気へと宇宙から高エネルギー粒子が飛来すると、大気中の原子核との相互作用により多数の二次粒子が生成される。この時飛来する粒子が高エネルギーガンマ線である場合には、まず電子陽電子対生成が生じ、二次粒子として電子陽電子が生成される。この時生じた電子陽電子は大気中の原子核の近傍を運動する際に制動放射しガンマ線を放射する。こうして放射されたガンマ線はそのエネルギーが電子陽電子の静止エネルギーより大きい限り再び電子陽電子対生成を生じこれらの過程を繰り返す。この繰り返しの過程によって荷電粒子の数が倍々と増えていく。このような現象を空気シャワーと呼び、一次粒子がガンマ線である場合を特に電磁シャワーという。このような生成過程は制動放射の臨界エネルギーである約 84 MeV ほどまで電子陽電子対生成で生成される荷電粒子のエネルギーが低下すると制動放射が生じなくなり、電離損失が卓越するようになる。こうしてそれ以降は荷電粒子が大気に吸収され荷電粒子数は減少していく。[2]

こうして生成された荷電粒子が大気中の光速を超えて運動すると荷電粒子の後方で大気中の原子で分極が生じ、荷電粒子が通過した後に分極から平衡状態へと移る際に電磁放射する。こうして放射された電磁波は強め合うことで荷電粒子の進行方向へとチェレンコフ光と呼ばれる可視光として放射される。1 TeV のガンマ線から生じる電磁シャワーによって生成されるチェレンコフ光は地上ではおよそ半径 140 m の円の範囲へと降り注ぐことになる。

3.1.2 解像型大気チェレンコフ望遠鏡

解像型大気チェレンコフ望遠鏡 (Imaging Atmospheric Cherenkov Telescope, IACT) 地上に降り注ぐチェレンコフ光を地上に設置した望遠鏡によって焦点面検出器へと集光しイメージを記録することによって一次粒子である高エネルギーガンマ線を間接的に観測する手法である。実際の観測では一次粒子はそのほとんどが宇宙線であるためにガンマ線観測の際にはバックグラウンドとして宇宙線とガンマ線を識別する必要がある。そこで IACT では宇宙線とガンマ線それぞれが生成するシャワーの形状が異なることを利用して識別する。この手法は A.M.Hillas によって考案されたため Hillas パラメーター [16] と呼ばれている。

こうして識別されたガンマ線由来のシャワーイメージよりその光量からガンマ線のエネルギーを、シャワーイメージの形状からガンマ線の到来方向を推定する。また複数の IACT を用いたステレオ観測により複数のシャワーイメージを取得可能となり、各イメージの軸の交点から 1 台の望遠鏡より良い精度でガンマ線の到来方向の推定が可能となる。

3.2 Cherenkov Telescope Array(CTA) 計画

Cherenkov Telescope Array(CTA) 計画とは口径の異なる三種の IACT を南半球と北半球に建設し、20 GeV から 300 TeV 領域のエネルギーのガンマ線を観測することを目指す国際共同実験計画である。三種の望遠鏡は最も口径の大きい口径 23 m の大口径望遠鏡 (Large-Sized Telescope, LST) と口径 12 m の中口径望遠鏡 (Middle-Sized Telescope, MST)、口径 4.3 m の小口径望遠鏡 (Small-Sized Telescope, SST) に分類される。現在は大口径望遠鏡初号機が建設され試験運用されており、2 号機以降の大口径望遠鏡、中口径望遠鏡、小口径望遠鏡の建設準備が進められている。

3.3 CTA 大口径望遠鏡 (LST)



図 3.3.1: 完成した LST1 の写真 [9]

LST は口径 23 m の IACT であり、CTA の中で最大の口径であるためよりガンマ線由来のチェレンコフ光をより多く集光することができるため CTA が観測可能となるエネルギー領域の内、低エネルギー側に感度を持つ。焦点面のカメラでは 7 本の光電子増倍管 (Photo Multiplier Tube, PMT) を 1 組とした光検出器モジュールが 265 モジュールでカメラを構成している。2018 年 10 月に LST 初号機 (LST1) が北サイトであるラ・パルマ島に建設され現在試験運用されている。また現在は LST2-4 号機の建設とその準備が進められている。

3.4 焦点面検出器

LST のカメラ部分にあたる焦点面検出器では、7 本の光電子増倍管 (Photo Multiplier Tube, PMT) と接続した波形読み出し基板 Dragon ボード、Slow Control Board(SCB) や Back Plane(BP) で構成される 1 モジュールが 265 モジュール分、全 1855 ピクセルで構成されている。以下に焦点面検出器を構成する各装置について述べる。



図 3.3.2: CTA 北サイトの完成予想図 [17]

3.4.1 ライトガイド

LST ではライトガイドと呼ばれる集光器を検出器である PMT 前面に配置している。ライトガイドの形状は六角形の入射口であり、この各辺に Winston Cone と呼ばれる形状を用いることで反射鏡から焦点面カメラへと入射するチェレンコフ光を効率的に集光することが出来る。また、視野外からの大きな角度でカメラに入射するような夜光等のバックグラウンドを混入することを防ぐことが出来、さらに PMT を並べた際に発生するデッドスペースを入射口が六角形のライトガイドであればデッドスペースを減らすことが出来る。図 3.4.1 にライトガイドの写真を示す。[19]

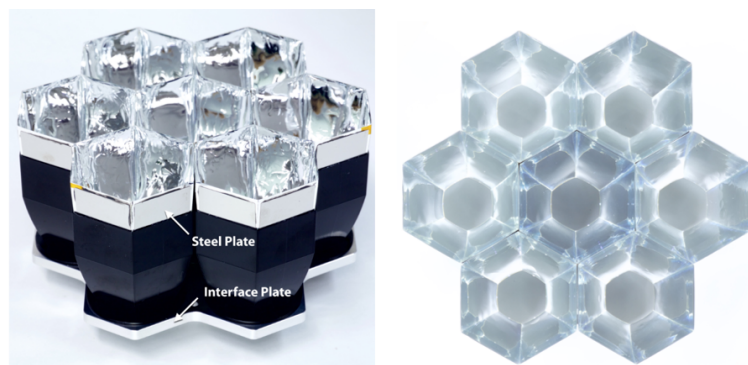


図 3.4.1: ライトガイド

3.4.2 光電子増倍管 (Photo Multiplier Tube, PMT)

図 3.4.2 に PMT の基本構造を示す [20]。光子が光電面に入射すると光電効果により PMT の真空管内に光電子をたたき出す。こうして飛び出した光電子は集束電極によって加速され第一ダイノードに衝突する。これにより、ダイノードから複数の電子が飛び出し電子数が増幅される。こうしてこれらの電子がその後のダイノードで増幅が繰り返されることで電子数が指数的に増幅されていく。こうして得られた電子を陽極から読み出すことで入射光を検出することが出来る。

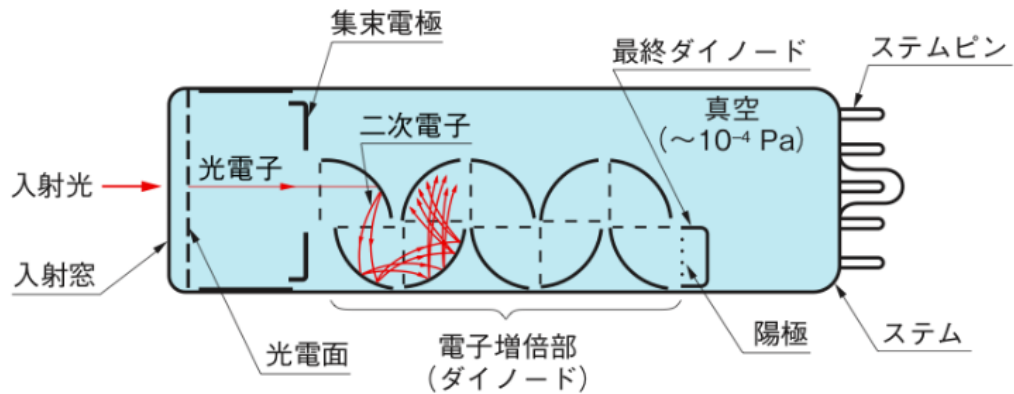


図 3.4.2: PMT の基本構造 [20] 入射窓に入射光が入射することで光電効果し、飛び出した光電子が後段のダイノードで増幅され読み出される。

3.4.3 スローコントロールボード (Slow Control Board, SCB)

SCB は光検出器モジュールと波形読み出し回路をつなぐ治具としての役割とともに、PMT に印加する高電圧の設定、PMT のアノードカレントや基板上の湿温度のモニター、波形読み出し回路へと出力するテルトパルスの生成などの機能が搭載されている。テストパルスは読み出し回路上の FPGA から矩形波を SCB へ送り、SCB に搭載されたテストパルス生成回路によって PMT 信号を模した疑似信号を生成し出力する。この疑似信号は実際の PMT 信号と同じようにプリアンプとメインアンプで増幅されたのち波形読み出し回路によって読み出される。

3.4.4 CTA 大口径望遠鏡用波形読み出し回路

空気シャワー由来のチェレンコフ光を観測する際に光検出器から出力される信号の幅は数 ns と非常に短く、観測天体以外の星などに由来する夜光が数百 MHz で混入するため、それらのバックグラウンドを取り除くためにチェレンコフ光の信号幅に合わせた時間幅で積分した電荷量の測定が必要になる。数 GHz での高速波形サンプリングによって信号の波形情報を記録できるような波形読み出し回路が要求される。また、LST では複数台の望遠鏡によるアレイ観測を行う際に波形読み出しを行うためのトリガー信号を各望遠鏡間でのコインシデンスをとることにより生成されるため、チェレンコフ光を捉えてから数 μ s 後に読み出し回路で読み出される。それ故に数 μ s の波形情報を記録可能なメモリ深さが要求される。

3.4.5 Dragon Board

高エネルギーガンマ線から形成された空気シャワーによって生成されたチェレンコフ光が主鏡によって焦点面検出器へと集光された後、検出器モジュール前面のによって光信号から電気信号へと変換されプリアンプによって増幅される。その後、0.2 光電子から 2000 光電子のダイナミックレンジを確保するために高ゲイン系統 (High Gain, HG) と低ゲイン系統 (Low Gain, LG) に分けた後に差動信号として出力する。検出器モジュールは 7 本の PMT からなるため、HG と LG 合わせて 14 チャンネル分の信号が SCB を通って波形読み出し回路へと入力される。

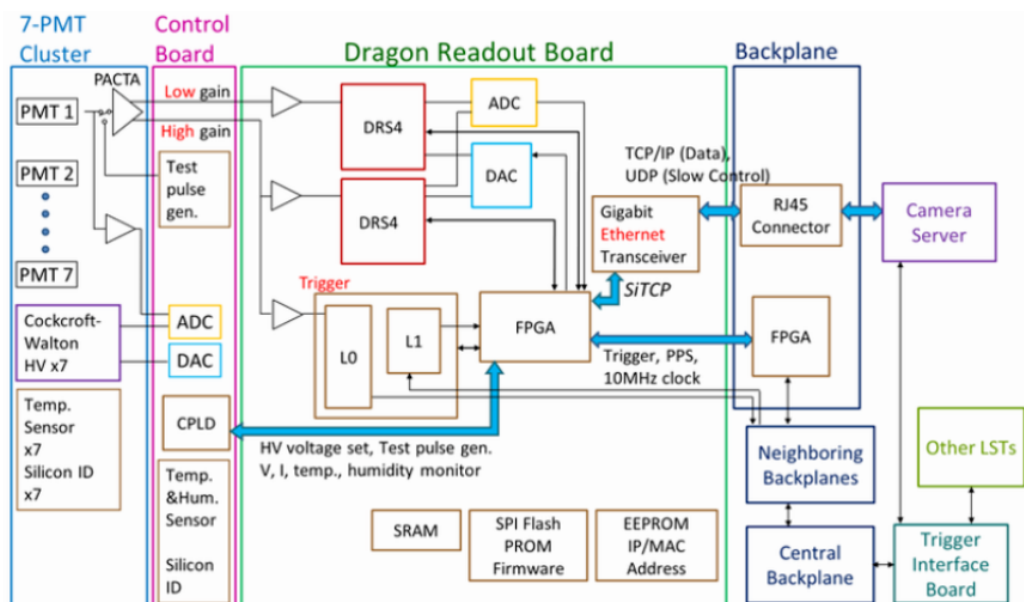


図 3.4.3: LST1 の波形読み出し回路のブロック図 [18]

3.4.6 プリアンプ

PMT から出力された信号は PACTA(PreAmplifier for the CTA camera) と呼ばれる CTA の光検出器用に開発されたプリアンプによって電圧信号への変換がなされ、増幅された後に差動信号として出力される。PACTA では入力された信号が HG で 24 倍、LG で 1.6 倍に増幅されるように設計されており、それらの信号が波形読み出し回路へと入力されるとメインアンプによってさらに増幅される。これらの増幅により LST の波形読み出し回路では 2000 光電子までのダイナミックレンジを確保されている。

3.4.7 Domino Ring Sampler version4 (DRS4)

アナログメモリ

空気シャワー由来のチェレンコフ光を測定する場合、チェレンコフ光のパルス幅は数 ns と非常に短いので高周波数帯域の読み出し回路によるサンプリングが求められる。また、測定したデータには数百 MHz 程度で星などに由来する夜光のバックグラウンドが混入する。このような夜光のバックグラウンドを取り除くために GHz ほどの速度による波形サンプリングによる短い時間幅での電荷積分を行う必要がある。また、LST1 台当たりには光検出器モジュールが 1855 本搭載しているため観測を行う際の LST 全体での消費電力及び発熱量を抑えることが求められる。さらに、波形を読み出すためのトリガー信号は望遠鏡間でコインシデンスをとることで生成されるため、実際のチェレンコフ光由来の信号が到達してから読み出し回路に入力されるまで数 μ s ほどかかる。なので、その間の波形情報を記録するために約 2.5 μ s 以上のメモリ深さが求められる。また、LST では 20 GeV から 3 TeV のエネルギー領域で観測を行うため、0.2 光電子から 1000 光電子以上のダイナミックレンジが求められている。

まず、高速でのサンプリングを実現する方法としてはフラッシュ ADC 方式とアナログメモリサンプリング方式の二つが挙げられる。このうち回路規模や消費電力、費用などの面から LST の読み出し回路ではアナロ

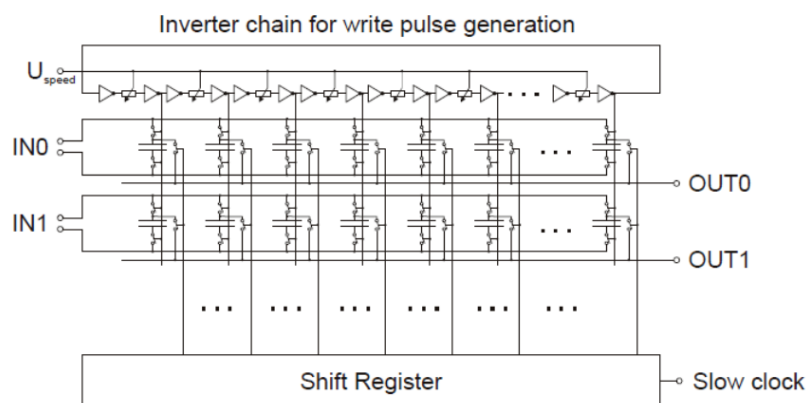


図 3.4.4: アナログメモリの回路図のイメージ [21]

グメモリサンプリング方式が採用されている。図 3.4.4 にアナログメモリの回路図のイメージを示す。アナログメモリ式サンプリング方式では、複数のキャパシタを並列に接続し、その入出力部をスイッチとなるトランジスタで接続したスイッチアレイを用いる。これらのスイッチを高速で切り替えることで各キャパシタに記録された電荷情報を保存し波形情報を記録することが出来る。メモリ深さはこのキャパシタ数とサンプリング時間幅の積で表せられるので数千個のキャパシタを用いる GHz サンプリングでかつ数 μs のメモリ深さを達成することが出来る。LST 用の波形読み出し回路にはスイスのポールシェラー研究所によってミューオンの電子光子崩壊を探索する MEG 実験のために開発された Dmomino Ring Sanmpler version4(DRS4) と呼ばれるアナログメモリを採用している。DRS4 は現行の IACT の 1 つである MAGIC 望遠鏡の波形読み出し回路でも用いられている。

アナログメモリ DRS4

DRS4 は 700 MHz から 5 GHz の高速サンプリングが可能であり、9 つの差動入力チャンネルを持ち 950 MHz の周波数帯域を持つ。図 3.4.5 にこのチップのブロック図を示す。各チャンネルには 1024 個のキャパシタが並列に接続され、9 つのチャンネルのうち 8 チャンネルを信号波形サンプリングに、1 チャンネルを各キャパシタのサンプリング時間幅較正用に用いられる。DRS4 の入力部分のスイッチ切り替えは Domino wave circuit と呼ばれる多数のインバーターを連結した回路上をパルス信号が伝搬することにより切り替えを行う。DRS4 のサンプリング速度は Domino wave circuit 上をパルス信号が一周する速度で決まり、phase locked loop(PLL) 回路により制御され、サンプリング速度が一定になるように調整されている。DRS4 外部から周波数 f_{ref} のクロックをリファレンスとして入力すると $2048 \times f_{ref}$ HZ で波形サンプリングが行われる [25]。DRS4 に入力された信号はこのように 1024 番目のキャパシタまで電荷情報を記録し、その後は同チャンネルの 1 番目のキャパシタの電荷を放電し、再び 1 番目のキャパシタから電荷を保存していく。この時例えば 1 GHz サンプリングでは PMT1 本あたり約 $1 \mu s$ がメモリ深さになる。ここで DRS4 の write shift register を書き換えることで、あるチャンネルで 1024 番目のキャパシタまで記録した後に次のチャンネルへ信号を入力させることが出来る。これをチャンネルカスケードと呼ぶ。このチャンネルカスケードを採用することで 1 チャンネル当たりのメモリ深さを増やすことが出来る。LST の読み出し回路ではメモリ深さを増やすために入力する差動信号を DRS4 の 4 チャンネル分用意している。こうすることで PMT1 本につき 4 チャンネルを用いて 1 GHz サンプリングで $4 \mu s$ のメモリ深さを確保することが出来る。また、こうして波形情報を記録した後に波形読み出しトリガーがかかると 33 MHz の低速クロックによって波形読み出しされる。

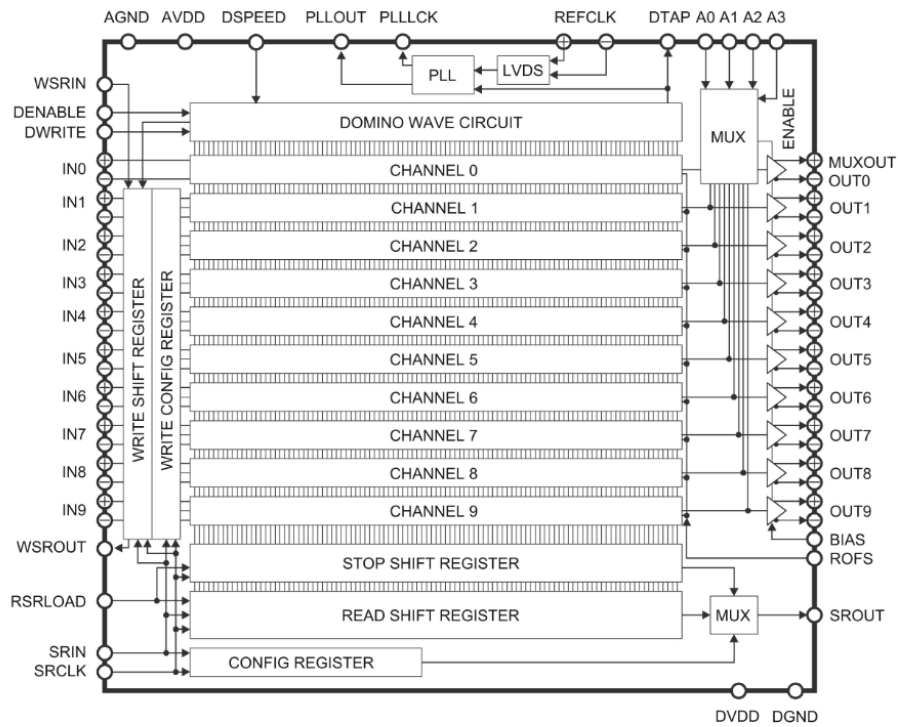


図 3.4.5: DRS4 チップのブロック図 [22]

ここで波形読み出しを行う際、転送するデータ量を抑えるためにチャンネル内から読み出すキャパシタの範囲を指定することが出来る。この範囲を Region of Interest(ROI) と呼ぶ。LST では ROI=40 と設定しており、1 イベントあたり 40 キャパシタ分の電荷情報を転送することになる。これは 1GHz でのサンプリングとすると 40ns 分の波形情報である。そうして DRS4 から読み出される信号は出力部のスイッチ切り替えの際に生じる高周波のノイズを除去するためにローパスフィルタを通してから ADC に入力される。こうして得られた波形情報にはキャパシタ固有の特性があるため読み出されたデータを取り扱う上で注意が必要である。この特性については 4 章で取り扱う。LST 初号機と LST2-4 号機の波形読み出し回路に搭載された DRS4 は新旧違いがありアップデートされている。後述するように DRS4 のもつキャパシタ固有の性質にも違いがみられるものもある。

3.4.8 ADC

DRS4 から出力されたアナログ信号は ADC へと入力されデジタル信号へ返還される。ADC へ入力された信号は 33 MHz の低速で 12bit のデジタル値に変換された後に 8 チャンネルの信号を FPGA へと入力される。

3.4.9 サイン波生成回路

DRS4 チップでは一定の間隔でキャパシタに電荷を蓄えることが理想的であるが、後述する DRS4 のキャパシタ固有の特性サンプリング時間幅の不均一性により、サンプリング時間幅にキャパシタ固有の揺らぎが存在していることが分かっている。このサンプリング時間幅を較正するには既知の波形を DRS4 チップに入

力しその入力波形と実際の DRS4 でサンプリングした波形を比較する較正テストを行うことでそのサンプリング時間幅のずれを求めることができる。実際に DRS4 チップ評価ボードでは 100MHz のサイン波を生成し搭載されている DRS4 チップへ時間幅較正を行う回路が搭載されている。LST2-4 号機用の dragon ボードでは 30MHz のオシレータから出力された矩形波をアナログ回路のローパスフィルタを通して高周波成分を取り除き、30 MHz の較正用サイン波を生成するサイン波生成回路が搭載されている。[18] こうして生成されたサイン波はシミュレーションでは 30MHz のほかに 30 MHz に対して 0.56% で 60 MHz、30 MHz に対して 0.19% で 90 MHz の成分が含まれていることが分かっている [18]。

3.4.10 バックプレーン (Back Plane, BP)

バックプレーンでは読み出し回路への 24V 電源の供給やカメラサーバーとのイーサネット通信を行っている。

第 4 章

大口径望遠鏡 2 から 4 号機における DRS4 諸特性の較正

4.1 目的

まず、サンプリング時間幅特性を較正の前に補正するペDESTAL特性、スパイク特性、残留電荷特性について述べた後に、LST1 号機及び 2-4 号機用の波形読み出し回路へ波形を入力しない状態での電気ノイズについて各特性の補正を行いペDESTALノイズを評価する。その後、パルスを入力した際の電気ノイズの影響を評価する。

4.2 DRS4 の諸特性の較正方法

DRS4 チップにはキャパシタ固有の特性がある。ペDESTAL特性、残留電荷特性 (dt 特性)、スパイク特性、サンプリング時間幅特性の 4 つである。4 章では 5 章で LST2-4 号機におけるサンプリング時間幅特性を較正を行う前段階として、まずペDESTAL特性、dt 特性、スパイク特性について述べる。その後 LST1 号機用ボードと LST2-4 号機用ボードそれぞれにおけるペDESTAL特性、dt 特性、スパイク特性を調べた結果を比較し LST2-4 号機用ボードにおけるペDESTALノイズを評価する。

4.2.1 ペDESTAL特性

図 4.2.1 は DRS4 の 2019 年 5 月に LST1 で測定されたペDESTALの平均値の分布である。縦軸が記録された ADC 値、横軸は DRS4 のキャパシタ id である。図 4.2.1 のようにキャパシタ id が 511 番目を境にペDESTALの電圧値が 50adc 値ほど異なっていることが分かる。なので波形を読み出す際にベースラインが ROI 内で異なってしまう電荷量を誤ってしまう。この特性は各ゲイン各ピクセル毎にペDESTAL測定をし、キャパシタ毎の電気ノイズの平均電圧値で差し引くことによって補正することが出来る。またこれによりキャパシタ毎の個性を補正する効果も期待できる。

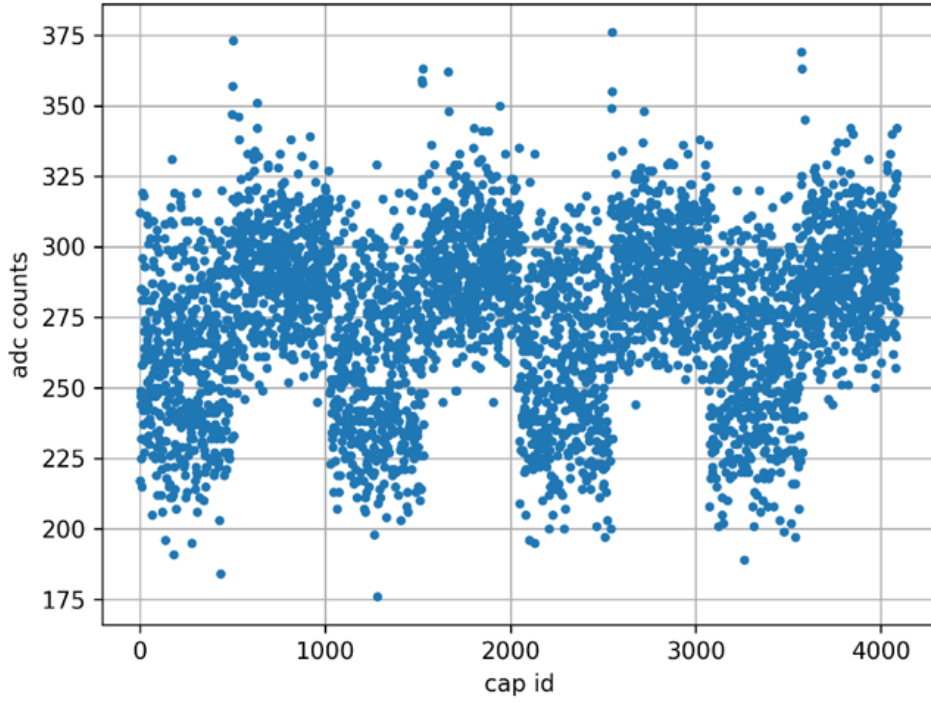


図 4.2.1: DRS4 の 2019 年 5 月に LST1 で測定されたペDESTALの平均値の分布 [30]

4.2.2 残留電荷特性 (dt 特性)

DRS4 では 1 ピクセルあたり 4096 個のキャパシタによって電荷を蓄えているが、そのうち読み出したいキャパシタの数つまり ROI 分のキャパシタの電荷だけを読み出している。なので、イベントごとに読み出されるキャパシタと読み出されないキャパシタが存在することになる。この時実際の観測ではトリガーは一定周期ではなくランダムであるために、あるキャパシタが読み出されてから再び読み出されるまでの残留時間 dt にはバラつきが生じる。この dt によって測定される電荷量に変動することが知られており、 dt が小さいほど読み出される電荷量が増加し、 dt が大きいほど電荷量が減少する。これを dt 特性と呼ぶ。この特性は一度読み出されたキャパシタに蓄えられた電荷が残留することに起因し、時間が経つに連れてこの残留電荷が小さくなる。LST1 で使用されている DRS4 においてこの残留電荷と dt の関係は以下の式 2.1 のような単純なべき関数で表せられることが分かっており、得られた波形データからキャパシタ毎に dt を調べてべき関数から残留している adc 値を求めて差し引くことで補正される。

$$ADC_{dt} = a \times (dt)^{-b} + C \quad (ADC_{dt} \text{は時間 } t \text{ で残留する電荷量、} a, b, c \text{ は定数}) \quad (2.1)$$

図 4.2.2 はイベント番号 m で読み出されたあるキャパシタ N が次にイベント番号 n で再び読み出される様子を表しており、イベント番号 m が読み出されてイベント番号 n が読み出されるまでの時間を dt とする。

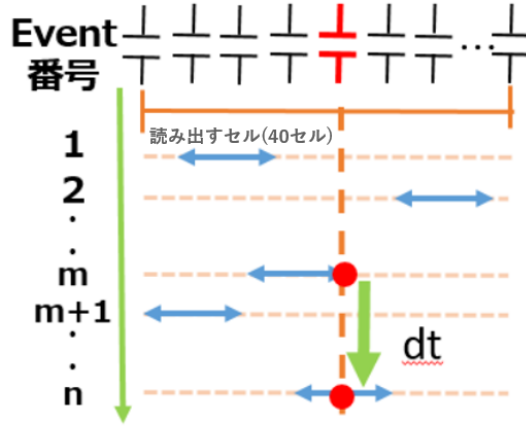


図 4.2.2: dt の定義 [18]

4.2.3 スパイク特性

DRS4 では波形読み出しをする際にある隣接する 2 つのキャパシタで電荷が 50adc 値ほど局所的に高くなる偽のパルスが現れることがある。これをスパイク特性と呼ぶ。スパイクが混入したイベントでは誤った値の電荷量を読み出すことになるため補正する必要がある。図 4.2.3 で実際にスパイクが発生した波形を示す。LST1 用ボードに搭載された DRS4 に関してスパイクが発生する条件は 2019 小林修論 [28] で調べられており以下の条件でスパイクの発生箇所を特定できる。

DRS4 のスパイク発生条件あるイベントを読み出した際の ROI のうち、最初のキャパシタを stop cell、最後のキャパシタを finish cell と呼び、各キャパシタを C_{stop}^n 、 C_{finish}^n と表すこととする。ここで C_{stop}^n 、 C_{finish}^n は 0 から 4095 の整数である。また n 番目のイベントでスパイクが発生するキャパシタ番号を C_{spike}^n と定義する。LST1 のボードで実装されている DRS4 では以下のスパイクの発生条件によりスパイクが発生する。

(1) 直前イベントの finish cell を読み出した際

1. $C_{spike}^n \equiv C_{finish}^{n-1} \bmod 1024$ あるいは $C_{spike}^n \equiv C_{finish}^{n-1} \bmod 1024$
2. $C_{finish}^{n-1} \bmod 1024 < 512$ かつ $C_{spike}^n \bmod 1024 \geq 512$
3. C_{spike}^n が偶数番目のリングにある (リングの数え方は 0~1023)

(2) 直前のイベントの finish cell と 1024 キャパシタのリングにおける対称的な位置のキャパシタを読み出した際

1. $C_{spike}^n \bmod 1024 = 1021 - C_{finish}^{n-1} \bmod 1024$ あるいは $C_{spike}^n \bmod 1024 = 1022 - C_{finish}^{n-1} \bmod 1024$
2. $C_{finish}^{n-1} \bmod 1024 < 512$ かつ $C_{spike}^n \bmod 1024 > 512$
3. C_{spike}^n が偶数番目のリングにある。

このようにスパイクが発生するキャパシタが予測可能である。なので、スパイクを記録しているキャパシタの

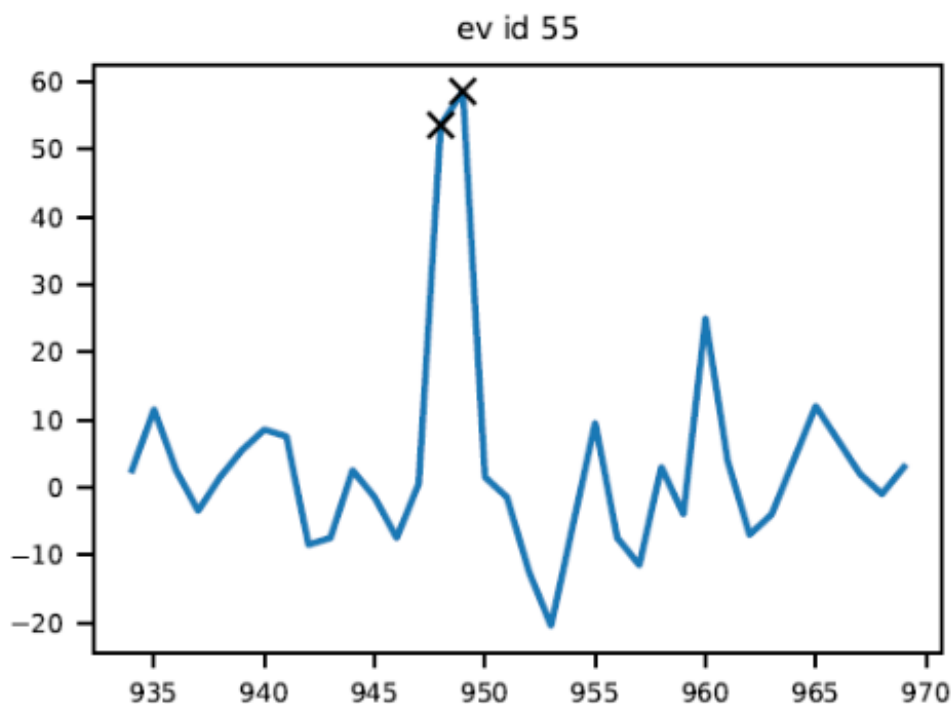


図 4.2.3: 2020 年 6 月に LST1 で測定されたペDESTALにスパイクが混入した際の波形。縦軸が adc 値で横軸がキャパシタ id。この終えですたる波形はペDESTAL補正及び dt 補正語の波形である。×はスパイクを記録したキャパシタとそれに対応する adc 値を示しており周囲のキャパシタより 50adc 値ほど高くなっていることが分かる。[30]

両隣のキャパシタの電荷量で線形補間もしくはペDESTAL測定から各パターンのスパイクの波高値を差し引くことで補正することが出来る。

4.3 測定及び解析環境

5 章でサイン波やテストパルス波形を読み出し回路へ入力する前段階として、4 章 4 節以降では LST2-4 号機用の読み出し回路に波形を入力せずにペDESTALを測定する。こうして得られた測定データには前述したとおりペDESTAL特性、残留電荷特性、スパイク特性があるため各特性について各較正後のペDESTALを調べる。また各較正後については LST1 号機用のボードのペDESTALとも比較する。

4.3.1 測定環境

図 4.3.1 に実際の測定を行う際のセットアップを示す。実際の LST2 号機以降の焦点面カメラに用いるものと同様の波形読み出し回路 (7 ピクセル)、信号増幅するプリアンプ、読み出し回路への電源の供給および測定した波形情報やコマンドの送受信を行うバックプレーン、読み出し回路への電源供給に用いる 24 V の電圧と 1 A 電流を供給できる直流安定化電源、読み出し回路の制御や測定データの解析にデスクトップ PC を使用している。波形読み出し回路には直流電圧が必要なため電源装置の正出力端子と GND 端子に BP を接続し出力電圧を +24V と設定し測定を行っている。また、波形読み出し回路でサンプリングをした後 AD 変換された

信号波形の情報やコマンドの送信はバックプレーンを介して実験室のデスクトップ PC と波形読み出し回路間で送受信する。

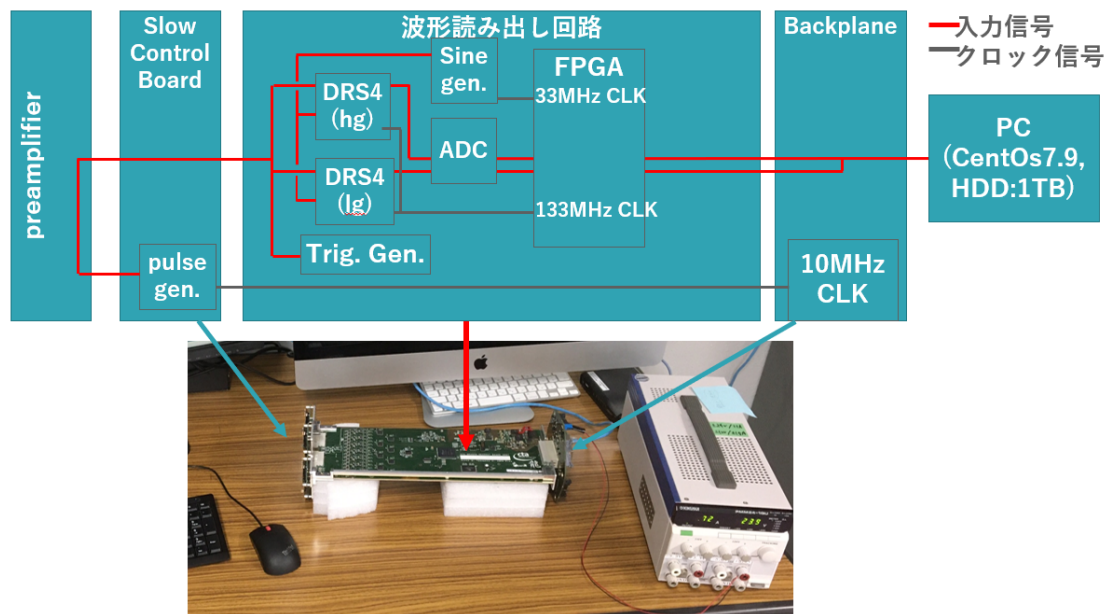


図 4.3.1: 測定のセットアップ

4.3.2 測定の設定

ROI が 40 スライス、トリガー周波数がランダムで平均 2 kHz、20 万イベントでペDESTAL測定を行った。また、DRS4 のサンプリング周波数 f_{smp} は DRS4 外部からのリファレンスクロック信号の周波数 f_{ref} 、クロック数 N_{clock} を用いて式 3.1 で表される。

$$f_{smp} = \frac{f_{ref} \times 1024[cell]}{N_{clock}} \quad (3.1)$$

LST1 でのサンプリング周波数 1 GHz を実現するために今回の測定は FPGA 内の 66.666 MHz のクロック信号をリファレンスとして使いクロック数を 66 とすることでサンプリング周波数を 1.034 GHz に設定している。

4.3.3 解析環境

4 章では 2 種類の解析ツールを使用して解析を行う。一つ目は LST で用いる波形読み出し回路の品質管理用の C、C++ 言語ベースのソフトウェア dragonqc である。測定時の読み出し回路の操作を行う。二つ目は CTA の python 言語ベースの解析ソフトウェアである ctapipe および LST 用の解析ソフトウェアのテストベンチである cta-lstchain でありペDESTAL測定データの解析に用いる。また、実験室での測定データはバイナリ形式であり、ctapipe 及び lstchain でのファイル読み出しは fits.fz 形式のファイル形式でなければならないため実験室でのバイナリ形式のデータファイルを fits.fz 形式へと変換するソフトウェア ConverToZfits を用い、dragonqc 上での解析に ROOT を用いている。図 4.3.2 に解析に使用したツールとそのバージョンをまと

めた。

名称	バージョン(時期)	用途
ctapipe	v0.11.0	pythonでの解析
cta-lstchain	v0.8.4	pythonでの解析
Anaconda	v4.10.3	pythonの使用
ROOT	v.5.34.34	Dragonqc上での解析
Dragonqc	2020年7月	読み出し回路の制御と解析
ConverToZfits.C	2022年10月	ファイル形式の変換

図 4.3.2: 実験及び解析に使用したツール等

また、図 4.3.3 に今回解析に使用したデータをまとめた。lst2-4 labo が今回測定したペDESTAL測定データである。比較に用いる LST1 用のボードを使用したペDESTAL測定データ lst1 labo、建設途中の LST2 で実際に使用されるボードを使用したペDESTAL測定データ lst2 tel は野崎誠也氏 (マックスプランク物理学研究所) から測定データをお借りして解析を行う。lst1 tel は現在試験運用中である LST1 でカメラを閉じて観測されたペDESTAL測定データを用いている。

使用する ボード	ピクセル数	イベント数	略称
LST 1	7	2万イベント	LST1 labo
LST1	1855(ラパルマの望 遠鏡)	5万イベント	LST1 tel
LST 2-4	7	20万イベント	LST2-4 labo

図 4.3.3: 解析に使用するペDESTAL測定データ

4.4 各較正後のペDESTAL分布

まず、2 万イベントのペDESTAL測定データ lst1 labo と 20 万イベントの lst2-4 labo についてキャパシタ id について 0 から 4095 まで偏りなく読み出されているかを確認する。図 4.4.1 に横軸キャパシタ id、縦軸読み出された回数でヒストグラムを作成した。図 4.4.1 より偏りなくランダムに読み出されていることが確認できた。

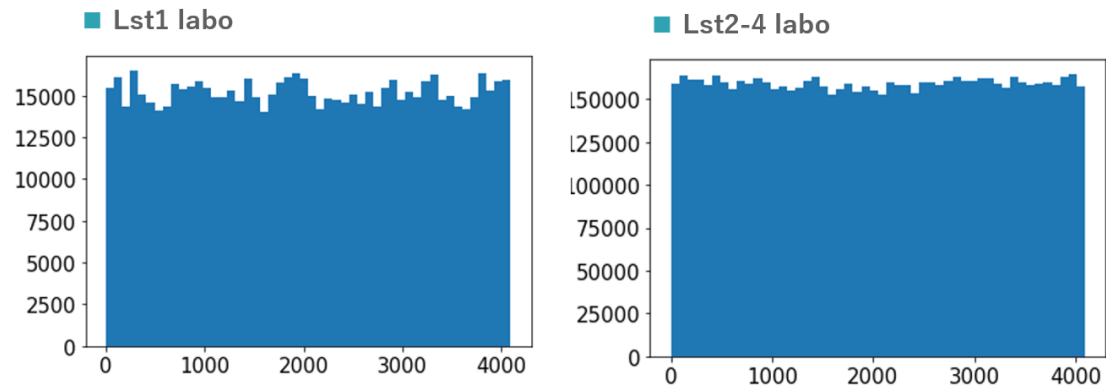


図 4.4.1: 図 4.4.1 に横軸キャパシタ id、縦軸読み出された回数

4.4.1 ペDESTAL補正

図 4.4.2 に 2-4 号機用ボードでのペDESTAL測定についてペDESTAL補正前と補正後の波形 50 イベントを重ねたものを示す。ペDESTAL補正によって電気ノイズの揺らぎが大きく解消されていることが分かる。この較正では較正されるペDESTAL測定データ自身から各ピクセル各キャパシタの ADC 値の平均値を求めて 2 ゲイン×ピクセル数× 4096 キャパシタのテーブルを作成しそれを用いてペDESTAL補正を行っている。このペDESTAL平均値記録したテーブルは lstchain を使用して作成した。

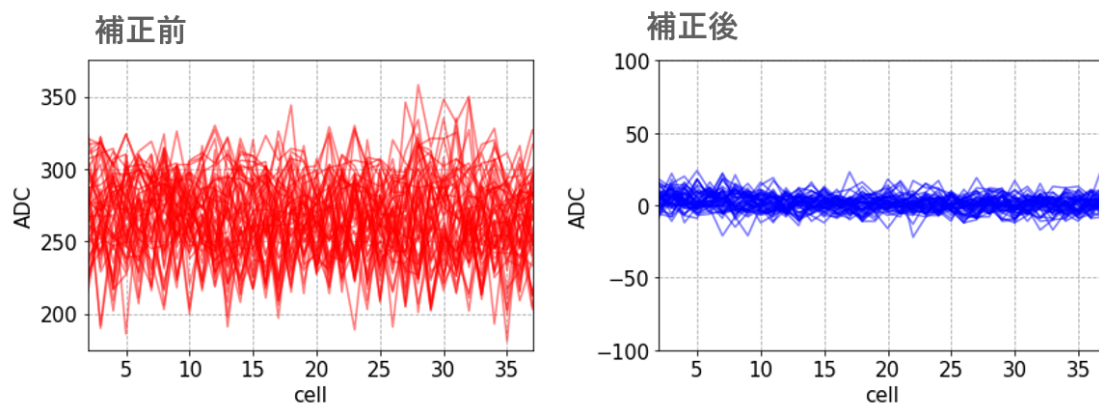


図 4.4.2

4.4.2 dt 較正

ペDESTAL測定データについてペDESTAL補正をした波形データについて dt 特性を補正する。まず、dt 特性を補正するのに使用する dt と残留電荷の関係を示すべき関数 dt カーブを調べる。まず LST1 号機用ボードで 2 万イベントでペDESTAL測定された LST1 labo を用いて残留電荷特性を調べた。図 4.4.3 の左の分布が LST1 用ボードにおける各イベント各キャパシタに記録された adc 値とそのキャパシタの dt の関係を plot

したものであり残留電荷と adc 値の関係を示している。dt が小さいほど adc 値が大きくなり dt が大きいほど adc 値が 0 付近に漸近することが分かる。残留電荷の分布のうち adc が 60 付近に見られる薄い分布はスパイク特性によるものである。こうして得られた dt カーブを用いて各 dt 毎に平均の adc 値を求めべき乗の関数でフィッティングを行いそれを dt カーブとする。こうして得られた dt カーブから各 dt に対応する残留電荷を決定しその adc 値分だけ補正する。lstchain に実装されている LST1 用の dt 補正でデフォルトでは dt カーブを $32.99 \times dt^{-0.18} - 11.9$ として補正しているがおおよそ一致していることが分かる。また残留電荷の分布より dt が 100 ms を超えると残留電荷特性がほぼ見られなくなる。故に dt が 100 ms より小さい場合に dt 補正を行うこととする。

図 4.4.3 の右図は 20 万イベントのペDESTAL測定データである lst2-4 labo を用いて同様に残留電荷特性を調べた結果である。図 4.4.3 の左図とは用いたイベント数が異なるためカラーバーが一致していない。LST1 号機用のボードでの残留電荷の分布とは dt カーブの関数が $13.5 \times dt^{-0.35} - 5.3$ と異なるため LST1 での結果より残留電荷が残留しにくく LST1 より速く残留電荷特性が見られなくなる。この dt カーブより LST2-4 での dt 補正では $13.5 \times dt^{-0.35} - 5.3$ を用い、dt が 10 ms より小さい場合に dt 補正を行う。図 4.4.3 に ROI=40 で実際に読み出されたペDESTAL補正済みのペDESTAL測定データについて dt 補正前後それぞれの波形を示した。dt 特性によってベースラインが補正されていることが分かる。

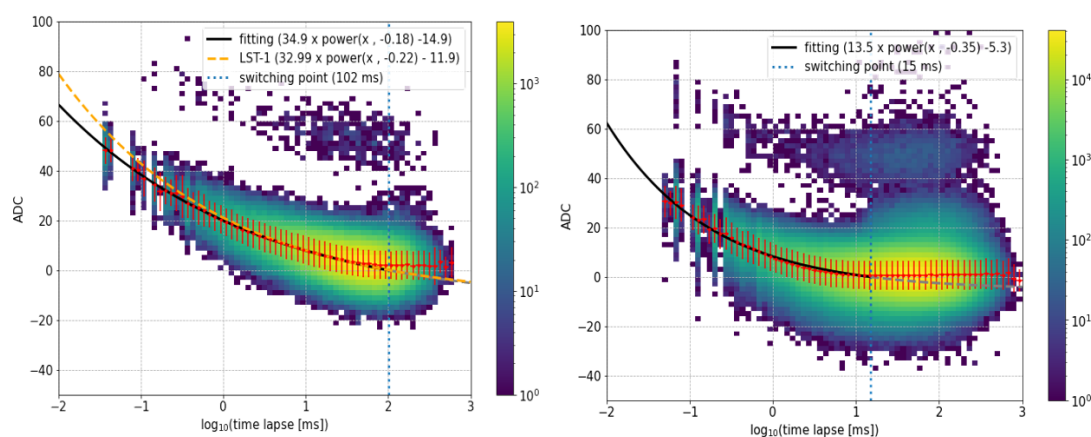


図 4.4.3: 左図は LST1 号機用 (1 万イベント)、右図は LST2-4 号機用 (10 万イベント) の読み出し回路での残留時間 dt と各キャパシタに記録された adc 値の関係。時間が経過するほど各キャパシタの adc 値は小さくなることが分かり、LST1 号機用と 2-4 号機用では異なる分布であることが分かる。switching point は残留電荷がほぼ 0 となる時間を表しており、LST1 では約 100ms、LST2-4 では約 10ms 程度である。また上部に見えるもやのような分布はスパイク特性によるものである。

4.4.3 スパイク較正

図 4.4.4 に LST2-4 用のボードで測定された 20 万イベントのペDESTALデータ lst2-4 labo についてペDESTAL補正及び dt 補正後の波形データを赤線で補正前、さらにスパイク補正をしたものを青線で補正後と示した。ただしスパイク補正前後で変化しない場合は紫線となっている。図 4.4.4 より隣接する二点で周囲よりも 50adc 値程局所的に高くなっているキャパシタがあることが分かり、スパイク補正後にはスパイクの波高

値分だけ補正されていることが分かる。

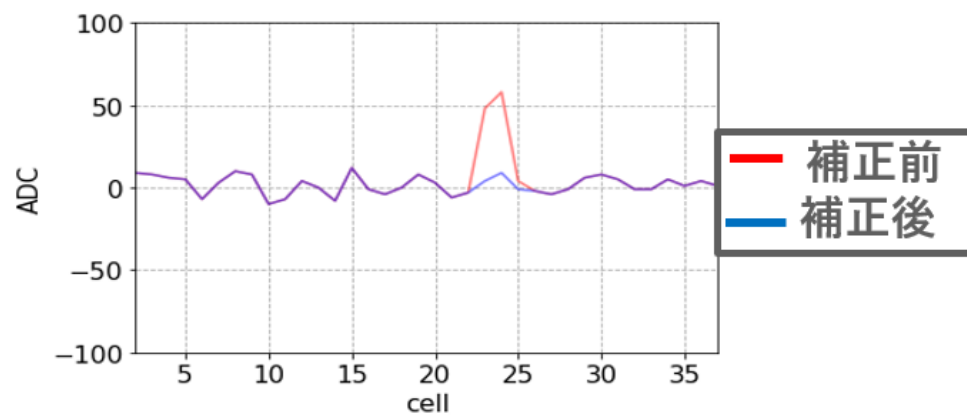


図 4.4.4

ここで予想されるスパイク波高値は各ピクセル各キャパシタ毎のペデスタル平均を計算する際に算出している。

4.4.4 各較正後のペデスタル分布

ここでペデスタル補正、dt 補正、スパイク補正の順で追加で補正を行った場合のペデスタル波形について測定された全イベントに関してピクセル毎に各キャパシタ毎の adc 値について積み上げたヒストグラムを作成した。図 4.4.5 に LST1 用のボードで測定された 2 万イベントのペデスタルデータ lst1 labo について各補正後のペデスタル分布 (pix=0) を示している。

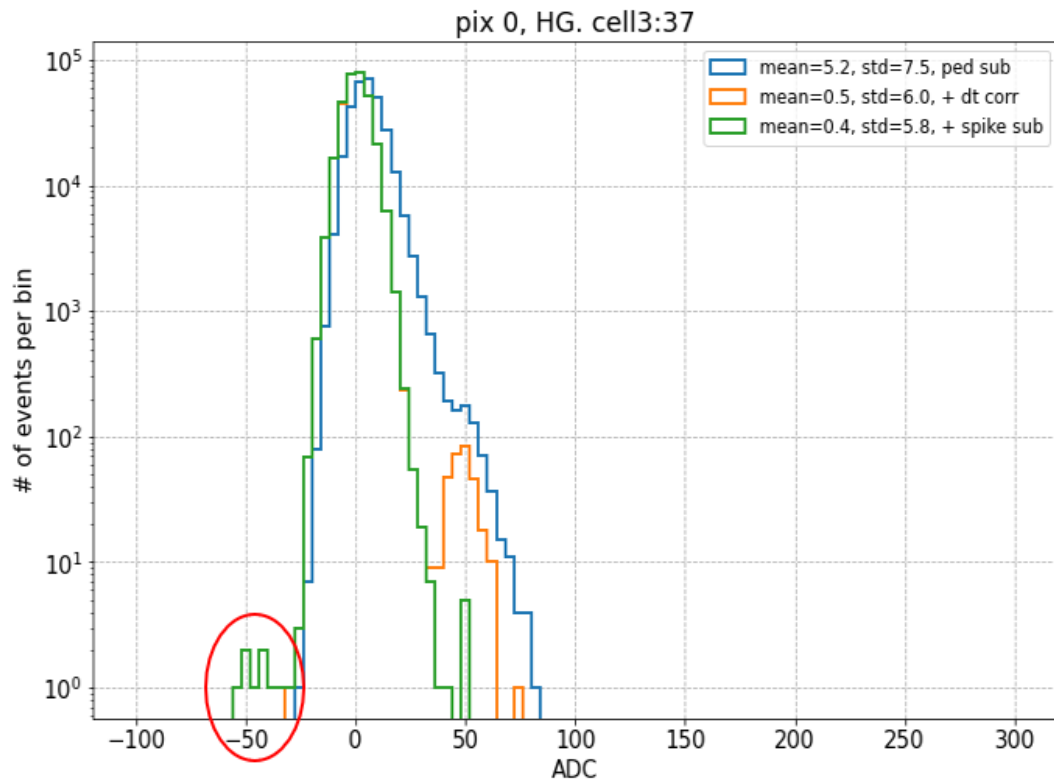


図 4.4.5

青線がペDESTAL補正後、黄線がペDESTAL補正 +dt 補正後、緑線がペDESTAL補正 +dt 補正 + スパイク補正後のペDESTAL分布である。各較正毎にペDESTAL分布が細くなっていることが分かる。ただし、-50adc 値付近の赤丸で示した箇所では小さな分布が見られることが見て取れる。実際にこの-50adc 値を記録したイベントの一部を図 4.4.6 で示した。

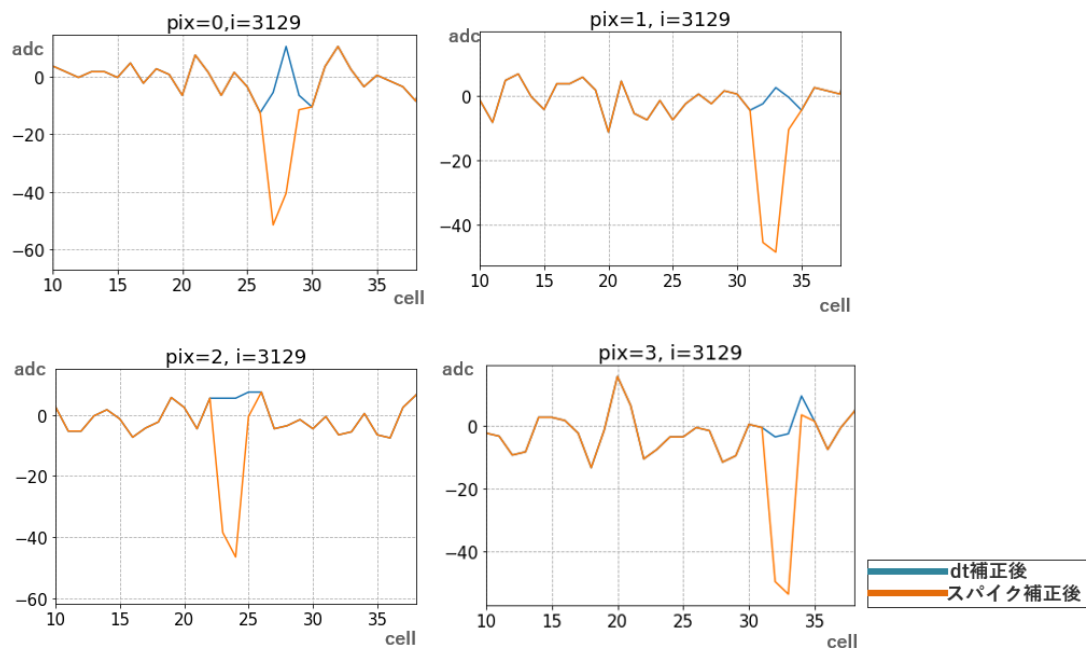


図 4.4.6

ペDESTAL補正後の波形についてさらに dt 補正をした波形を青線で、dt 補正後にさらにスパイク補正をしたものを黄線で表している。スパイク補正前の元の波形 (ペDESTAL補正 + スパイク補正) にはスパイクのようなものは見られないが、スパイク補正後にはスパイクが発生していない箇所でスパイクの波高値分の補正を行ってしまっていることが分かる。また 20 万イベントのペDESTALデータ lst2-4 labo についても同様に各補正後のペDESTAL分布 (pix=0) を図 4.4.7 のように作成した。

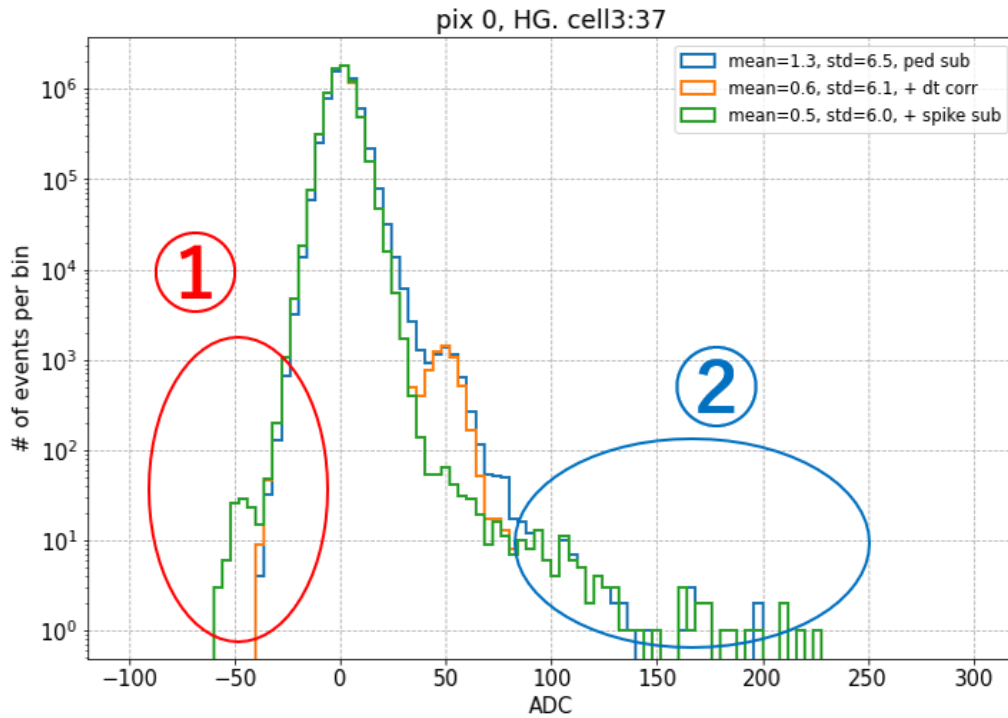


図 4.4.7

赤丸で示したように LST2-4 号機用ボードでも同様に -50adc 値付近に山が出ていることが分かるため、このスパイクの補正ミスはスパイクを補正するプログラムに問題があると考えられる。スパイクが発生すると誤って予測してしまったことで、スパイクの波高値 50adc 値分だけ補正を行ってしまい、波形がその分落ち込んでしまっている。このスパイクの補正ミスの発生確率を測定したイベントから計算するとは lst1 labo で 2 万イベント中約 0.089%、lst2-4 labo で 20 万イベント中約 0.0025%、lst1 tel で 5 万イベント \times 1855 ピクセル中 3 イベントとなっておりこのスパイクの発生ミスは非常に稀であることが分かる。4 章では 5 章に向けてサンプリング時間幅較正の前段階としての較正が達成していることが求められているが、このスパイクの補正ミスはその発生頻度の低さから時間幅較正への影響は非常に小さいことが予測できる。次に図 4.4.7 について青丸で示したように 50adc 値以上でテールのようなものが見えていることについて考える。

この 50adc 値以上で 80adc 値を超えるようなキャパシタが存在することに関して実際に該当するイベントの波形を調べた。図 4.4.6 にそのうちの 1 イベントの波形を示す。

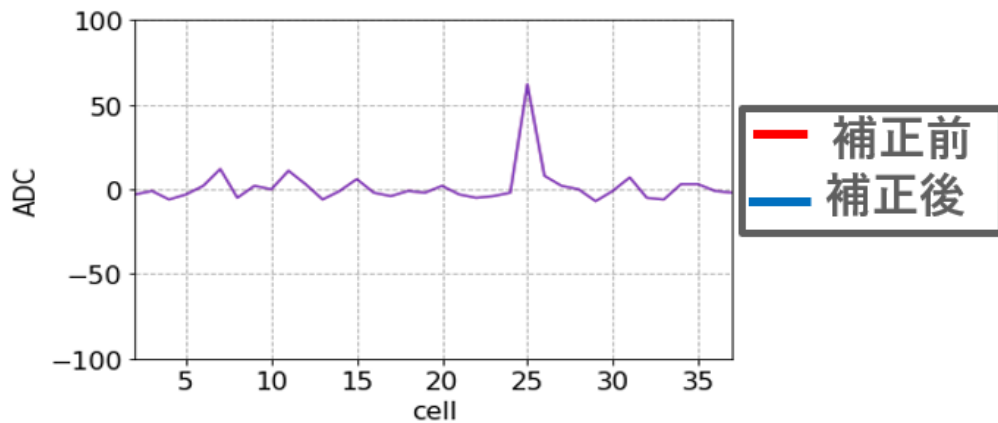


図 4.4.8

ペDESTAL補正 +dt 補正後を赤線で補正前、ペDESTAL補正 +dt 補正 + 後スパイク補正後を青線で補正後としている。図ではスパイク補正前後で波形が変化しておらず赤線青線が重なり紫色の線となっている。図 4.4.8 のように単一のキャパシタについて 50adc 値を超えるようなスパイクに見えるものが記録されていることが分かる。

このような現象が LST1 用のボードで発生していないか lst1 labo 及び lst1 tel のペDESTAL測定データを調べたがこのような現象は確認できず LST2-4 用ボード固有の問題もしくは lst2-4 labo で用いたこの 1 枚のボード固有の問題だと考えられる。ここで今回のピクセル 0 での結果がこのボードの他のピクセルでも発生しているかどうかを調べた。その結果、pix=0,3,5 で図 4.4.7 の青丸のような分布が確認されたが pix=1,2,4,6 では見られなかった。よって、pix=1,2,4,6 のデータのみを使うことでこの影響を無視することもできる。

次に、このようなスパイクのようなものが見られたキャパシタ iD を調べた。図 4.4.9 はその結果である。

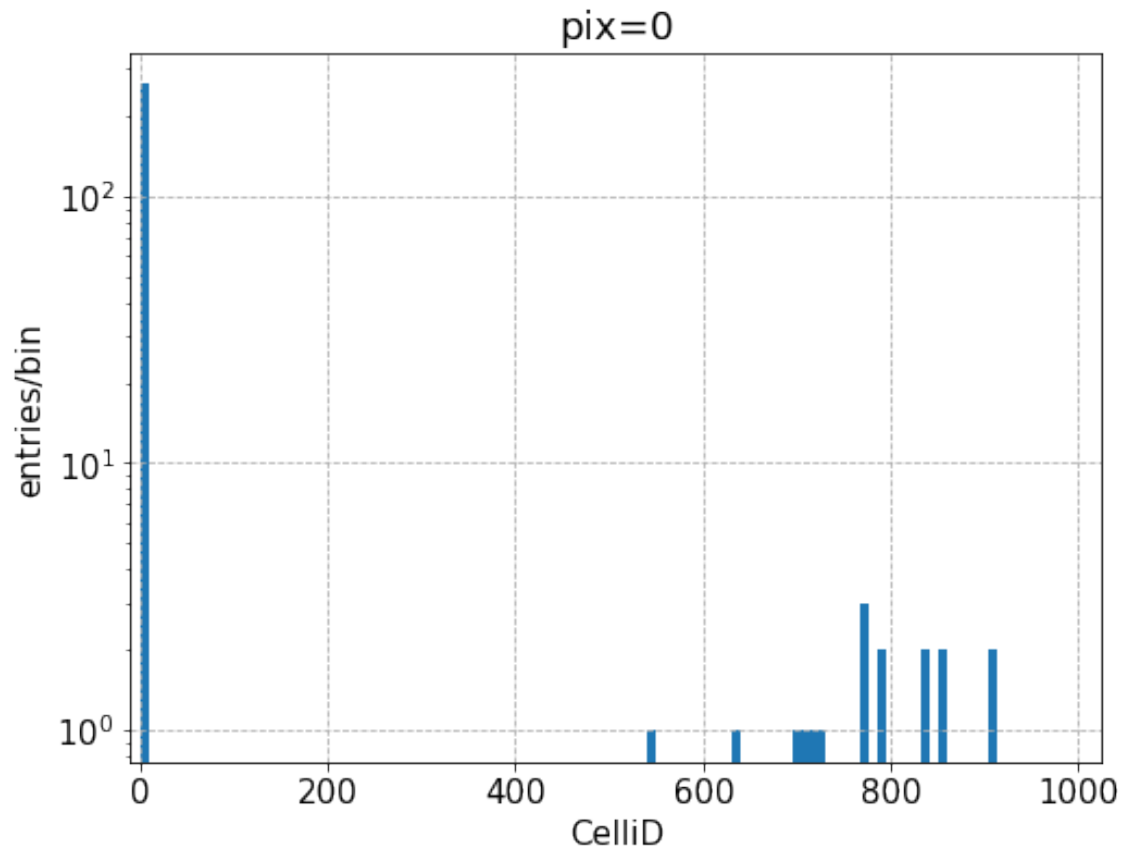


図 4.4.9: 一つのキャパシタに現れるスパイクのような立ち上がりを記録したキャパシタ iD

図 4.4.9 より、このような一つのキャパシタに現れるスパイクのような立ち上がりを記録したキャパシタ iD は軒並み cell iD=0 であった。cell iD=0 のデータがそのような立ち上がり起きるわけではないが、cell iD=0 のデータのみを解析から除くことでこの立ち上がりの影響を無視することが出来る。

4.5 議論

略称	ピクセル数	①スパイクの差引きミス -40adc以下の落ち込み	②単一のキャパシタに現れるスパイク のようなイベント
lst1 labo	7	全イベント中 約0.089%	なし
lst1 tel	1855	5万イベント×1855pix中3イベント	なし
lst2-4 labo	7	全イベント中約0.025%	Pix0,3,5だけで発生(pix1,2,4,6ではなし)

図 4.5.1

4 章では LST1 号機用ボードを用いた実験室で測定されたペDESTAL測定データ lst1 labo、ラパルマで試験運用中 LST1 で測定されたペDESTAL測定データ lst1 tel、LST2-4 号機用のボードを用いた実験室で測定さ

れたペDESTAL測定データそれぞれについてペDESTAL補正、dt 補正、スパイク補正と順に較正したペDESTAL分布を作成した。lst1 用のボードではペDESTAL補正後で 7.5adc 値ほどの揺らぎがみられ dt 補正スパイク補正を行うことで揺らぎが 5.8adc 値ほどまで小さくなる。lst2-4 号機用のボードではペDESTAL補正後で 6.5adc 値ほどの揺らぎが見られ、dt 補正スパイク補正を行うことで揺らぎが 6.0adc 値まで小さくなる。また、図 4.5.1 に今回の解析での問題点とその発生割合や判明している発生条件をまとめた。①については発生確率が非常に稀であることから時間幅較正等への影響は無視できる。②については $\text{pix}=0,3,5$ のデータを使用しない、あるいは $\text{cell iD}=0$ のデータを使わないことで時間幅較正への影響を無視することが出来る。ここで LST の波形読み出し回路に実装されてるサインは生成回路で生成するサイン波の振幅は約 $\pm 1500\text{adc}$ 値であり、またスローコントロールボードで生成可能なテストパルスの波高値は 2000adc 値である。ここで、今回のペDESTAL分布から波形読み出し回路へ波形を入力しない状態での電気ノイズについて各特性の補正を行いペDESTALノイズはペDESTAL補正後で $\pm 6.5\text{adc}$ 値とサイン波やテストパルスと比べて非常に小さいため実際の較正ではそれらの影響は無視できると考えられる。

第 5 章

大口径望遠鏡 2 から 4 号機用におけるサンプリング時間幅の較正

5.1 目的

LST の波形読み出し回路に搭載している波形記録チップ DRS4 での入力部分のスイッチ切り替えは Domino wave circuit と呼ばれる多数のインバータを連結した回路上をパルス信号が伝搬することによって行われている。このインバータに用いている半導体の特性によりスイッチの切り替わる時間間隔が各キャパシタ毎に少し異なることが知られている。なので、サンプリング時間間隔を一定としてしまうとサンプリングして波形を再構成した際に元の入力波形から歪んだ波形を再構成してしまうため入力信号の到来時間や電荷量を正しく求められない。図 5.1.2 にサンプリング時間幅を一定として波形が歪むイメージ示す。この特性は、DRS4 を搭載した波形読み出し回路を用いている MAGIC 望遠鏡では DRS4 の各キャパシタに対応した到来時間に対して補正を行っており [23]、D. Stricker-Shave et al.[24] ではサイン波を用いてサンプリング時間幅の較正が行われている。特に 2021 年野上修論では実際の LST2-4 に搭載されている波形読み出し回路を使用して ROI=100 でサイン波による較正を行っている。だが、実際の望遠鏡では ROI=40 をデフォルトにしており DAQ の設定を自由に変えることができないため自由に ROI を変更できない。故に ROI=40 でサンプリング時間幅較正を行う必要がある。ただし、ROI=40 でのサイン波によるサンプリング時間幅較正方法は確立していなかった。そこで本章では ROI=40 でのテストパルス及びサイン波を使ったサンプリング時間幅較正を行い、ROI=40 でのサイン波によるサンプリング時間幅較正を行う。図 5.1.1 は Kobayashi, et al.(ICRC 2021)[29] で調べられた入力電荷と電荷分解能についての結果のうち High gain(赤線) 及び Low gain(青線) でのモンテカルロシミュレーションの結果を示したものである。この時、モンテカルロシミュレーションでは時間幅の不均一性無しでの結果であるためこの二本の曲線は DRS4 の時間幅特性を完璧に補正することができれば達成できる限界値とみることが出来る。この電荷分解能はポアソン統計、夜光の揺らぎ、読み出し回路の電気ノイズの揺らぎ、PMT ゲインの揺らぎによる成分からなる。光電子数が多くなるとそれに伴ってポアソン統計による揺らぎの寄与が小さくなり約 3% ほどで頭打ちになる。そこで時間幅特性を較正し LST の性能の最大化を図るために統計以外の寄与を 3% 以下にする必要がある。そこで本研究では ROI=40 でのサンプリング時間幅較正を行い、サンプリング時間幅による寄与を 3% 以下に抑えることとし、要求仕様を達成し ROI=40 でのサイン波によるサンプリング時間幅較正を確立する。

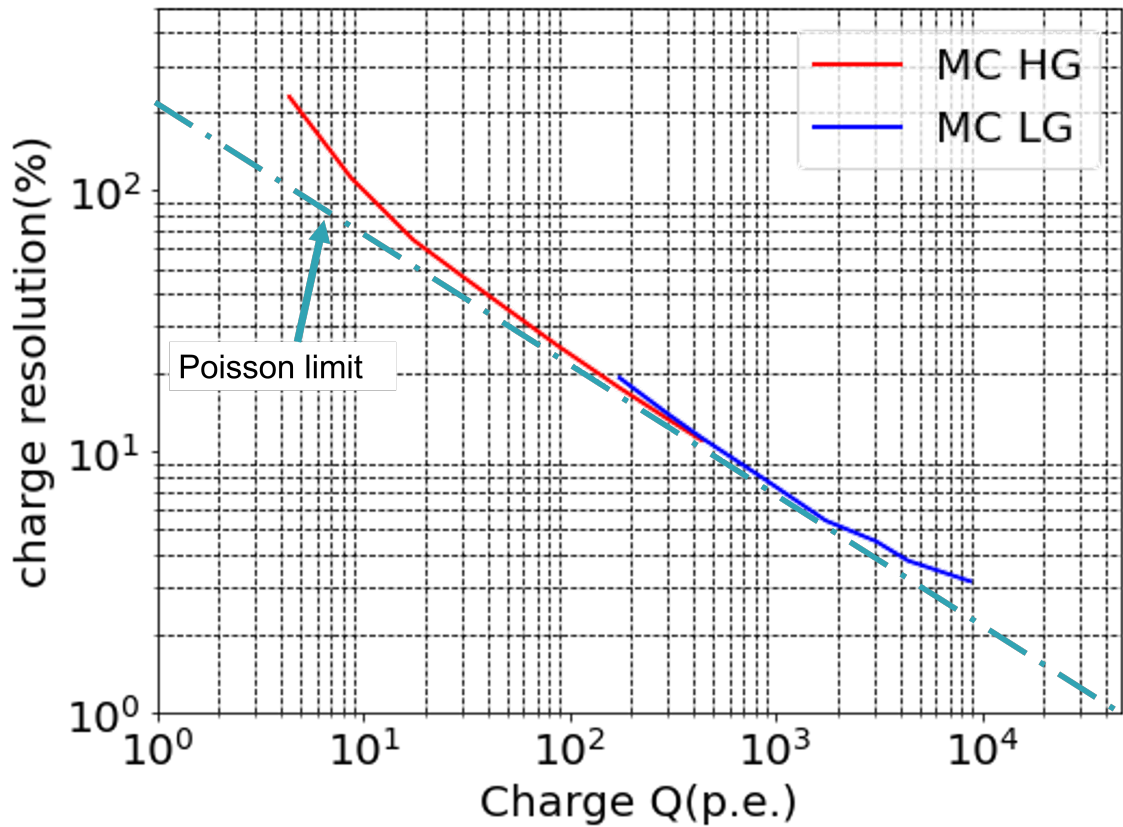


図 5.1.1: LST で時間幅の不均一性を完全に補正した際の限界。Kobayashi, et al.(ICRC 2021)[29] より一部改変。赤線が high gain での青線が Low gain でのモンテカルロシミュレーションで調べられた入力電荷に対する電荷分解能の結果

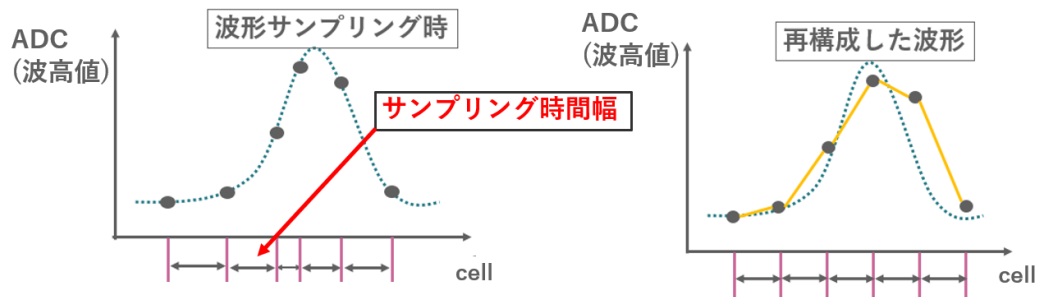


図 5.1.2: サンプリング時間幅の不均一性のイメージ [30]。横軸が時間、横軸の紫色の線がサンプリング時刻を表している。サンプリング時間間隔を一定として再構成すると右図のように入力信号の波形を歪めてしまう。

5.1.1 電荷分解能

まず、電荷量の求め方を述べる。図 5.1.3 に電荷計算の例を示す。電荷量はあるイベントにおいて、ROI 内で最大の波高値を記録したキャパシタを中心に 7 キャパシタ分の波高値 (adc 値) の積分値を電荷量 $Q[i]$ とする (i はイベント番号)。ここで使用したイベント中の平均電荷量を $\overline{Q[i]}$ 、電荷量の標準偏差を σ_Q とすると電荷分解能 $\epsilon_Q(\%)$ は以下の式で表される。また本研究では各種較正後に求まったサンプリング時間幅テーブルを用いてテストパルスに関して時間幅較正をし電荷分解能を調べる。この時使用するテストパルスの測定条件は前述したテストパルスを用いた較正で使ったテストパルス測定と同様であり、ペデスタル補正済みのものを使用する。

$$\epsilon_Q(\%) = \frac{\sigma_{Q[i]}}{\overline{Q}} \times 100 \quad (1.1)$$

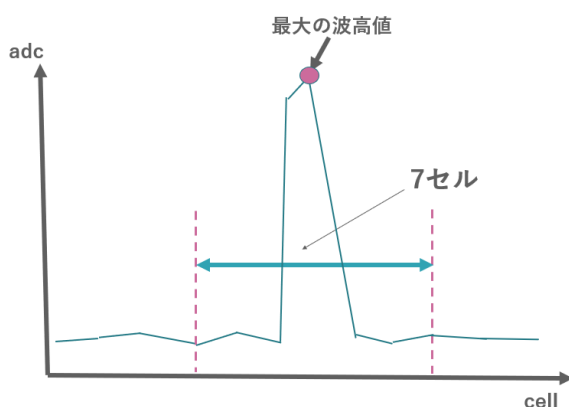


図 5.1.3: 電荷計算のイメージ

5.2 LST1 でのテストパルスを用いたサンプリング時間幅較正

現在試験運用されている LST1 において SCB より生成できるテストパルスを用いたサンプリング時間幅の較正が宇宙線研究所の齋藤隆之氏によって考案され 2016 年の谷川修論 (京都大学) [27] で実際にこの方法を用いて検証が行われている。また、2021 年野上修論 (茨城大学)[30] で実際の LST1 で使用されているボードを使用しテストパルスによる時間幅較正の検証が行われている。

LST では、波形読み出し回路と PMT を繋ぐ SCB にテストパルス生成回路が実装されておりこれにより波形読み出しボードへテストパルスを出力できる。この時テストパルスを DRS4 上の各キャパシタへと一様に記録させるために一定の周期でテストパルスを生成する。こうすることで波形読み出しされたキャパシタが一定間隔で遷移し各キャパシタへ一様にテストパルスを記録することが出来る。そうして得られた測定データより、最大波高値を記録した回数を DRS4 のキャパシタ毎に数えていく。これは補正前のサンプリング時間幅の大きさに比例してテストパルスの最大波高値を捉える確率が上がることを利用して較正している。入力されたテストパルスの総数を N_{total} 、DRS4 のキャパシタ id を $i(1 \leq i \leq 1024)$ 、テストパルスを測定したデータよりキャパシタ $id=i$ において最大波高値を記録した回数 $N[i]$ 、DRS4 チャンネルあたりのキャパシタ数を

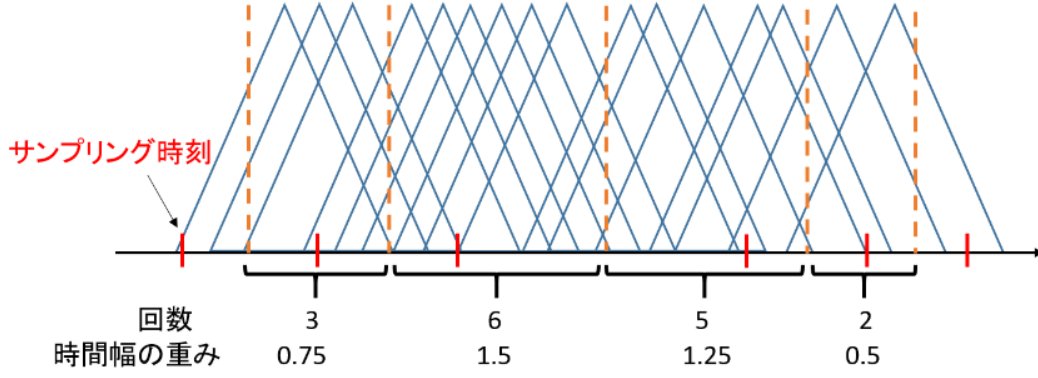


図 5.2.1: テストパルスを用いた較正方法のイメージ [30]。横軸が時間、縦軸が adc 値。赤線はサンプリング時刻を表しており、隣り合う赤線の間隔をサンプリング時間幅とする。橙色の線は隣り合うサンプリング時刻の中点を表している。

$N_{cap}(=1024)$ とし以下の式のように重みづけを行う。これによって各キャパシタのサンプリング時間幅に対応する補正係数 $\omega[i]$ とする。

$$\omega[i] = \frac{N[i]}{N_{total}} \times N_{cap} \quad (2.1)$$

この操作をゲイン系統、ピクセル、キャパシタ毎に行う。実際の望遠鏡であればピクセル数は 1855 ピクセルのため $2 \times 1855 \times 1024$ のテーブルとなる。

5.3 LST2-4 でのサイン波を用いたサンプリング時間幅較正

LST2 号機以降の波形読み出し回路ではその回路上に搭載されたサイン波生成回路によって周波数 30MHz のサイン波を DRS4 に入力することが出来る。今回は ROI を 40 と設定して 1 万イベント分の測定をした。ここで理想的なサンプリング時間幅はサンプリング周波数より 0.967 ns でありサイン波 1 周期の時間は 33.33ns である。D. Striker-Shaver et. al の較正手法 [24] を元に開発した野崎修論の較正手法 [18] を用いて較正を行う。具体的に二つの較正手順に分けられ、まず隣り合う 2 つのデータ点とサイン波の Fit 関数によって補正を行うローカルキャリブレーションを行い、その後入力サイン波の 1 周期ごとのぶれを補正するグローバルキャリブレーションを行う。

5.3.1 ローカルキャリブレーション

ローカルキャリブレーションは図 5.3.1 で示すように隣り合う 2 つのデータ点と Fit 関数との時間方向の差から Fit 点を求めてサンプリング時間幅を較正する方法である。ここでサイン波を測定したデータ点について以下のようなサイン関数でフィッティングを行い較正に使用する。

$$a \times \sin(2\pi b \times t + c) + d \quad (3.1)$$

この Fit 関数のパラメータのうち波高値に対応する a は 1500adc 値を初期値とし、入力周波数に対応する b は 30 MHz と既知なので固定している。また、キャパシタ i 、 $i+1$ がサイン波をサンプリングした時刻を $t[i]$ 、 $t[i+1]$ とすると較正前のサンプリング時間幅は $\Delta t[i]=t[i+1]-t[i]$ と表せる。図 5.3.1 のようにデータ点と同じ

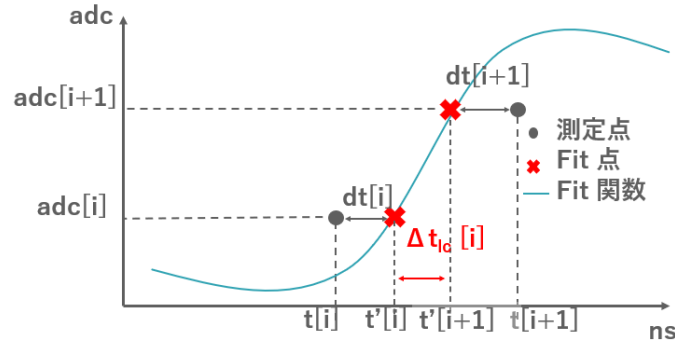


図 5.3.1: ローカルキャリブレーション [30]

波高値に該当する Fit 関数上の点における時刻をそれぞれ $t'[i]$ 、 $t'[i+1]$ とする。この時、キャパシタ i でのサンプリング時刻と Fit 関数上でのサンプリング時刻の差 $dt[i]$ は

$$dt[i] = t'[i] - t[i] \quad (3.2)$$

と表され予想されるキャパシタ i 、 $i+1$ 間のサンプリング時間幅 $\Delta t_{lc}[i]$ は以下の式で表せる。

$$\begin{aligned} \Delta t_{lc}[i] &= t'[i+1] - t'[i] \\ &= \Delta t[i] - dt[i] + dt[i+1] \end{aligned} \quad (3.3)$$

式 3.3 を各イベントのキャパシタ毎に行い、各キャパシタ毎に求めた時間幅の平均をとることで各キャパシタに対応するサンプリング時間幅とする。ここで N_{evcap} を全イベントキャパシタ i にデータが記録された回数とする。

$$\overline{\Delta t_{lc}[i]} = \frac{\sum \Delta t_{lc}[i]}{N_{evcap}} \quad (3.4)$$

5.3.2 グローバルキャリブレーション

グローバルキャリブレーションでは、ローカルキャリブレーションで得られたサンプリング時間幅を用いてサイン波 1 周期に含まれる複数のキャパシタに対して補正を行う。まず、ROI 内のサイン波 1 周期を求めるために測定波形と波高値 0 の直線との交点 (crossing point) を探す。この crossing point は図 5.6.6 のように測定波形について波高値が 0 になる前後のデータ点に関して線形補完することで決定する。これにより求めた 2 点の crossing point からサイン波 1 周期の時間を推定しこれを T_{est} 、入力サイン波の理想的な周期を $T_{true}(=33.33\text{ns})$ とする。こうして得られたサイン波 1 周期の時間と理想値の比を用いてサンプリング時間幅 $\overline{\Delta t_{lc}[i]}$ を以下のように補正する。ここで n_{ev} はイベント番号とする。

$$\Delta t_{gc} = \overline{\Delta t_{lc}[i]} \times \frac{T_{true}}{T_{est}[n_{ev}]} \quad (3.5)$$

式 3.5 を各イベントのキャパシタ毎に行い、各キャパシタ毎に求めた時間幅の平均をとることで各キャパシタに対応するサンプリング時間幅とする。ここで同様に N_{evcap} を全イベントキャパシタ i にデータが記録され

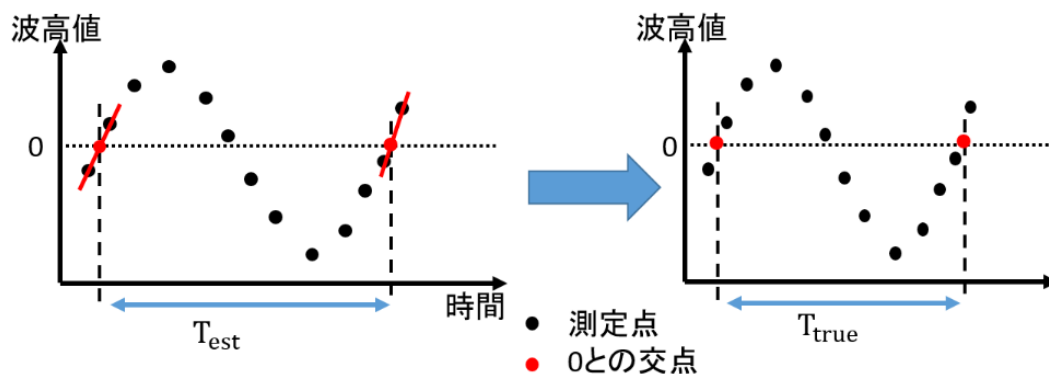


図 5.3.2: グローバルキャリブレーション [30]

た回数とする。

$$\overline{\Delta t_{gc}[i]} = \frac{\sum \Delta t_{gc}[i]}{N_{evcap}} \quad (3.6)$$

ただし、今回この手法でグローバルキャリブレーションを行う場合時間幅の計算を誤ってしまう。なぜなら ROI=100 のサイン波を使用する場合はサイン波 1 周期がおよそ 33 キャパシタに相当することから ROI 内に 3 周期分のサイン波が入ることになるため 1 周期分の crossing point を求められることが保証されているが、今回新しく使用する ROI=40 のサイン波の場合にはこの手法では 1 周期分の crossing point が求められるとは限らない。そこで、図 5.3.3 のようにサイン波の半周期分の crossing point を求めることにする。これにより半周期分の crossing point によって ROI=40 内にサイン波の半周期を求められることを保証できた。以降、ROI=40 の場合のグローバルキャリブレーションではサイン波半周期で行うこととする。

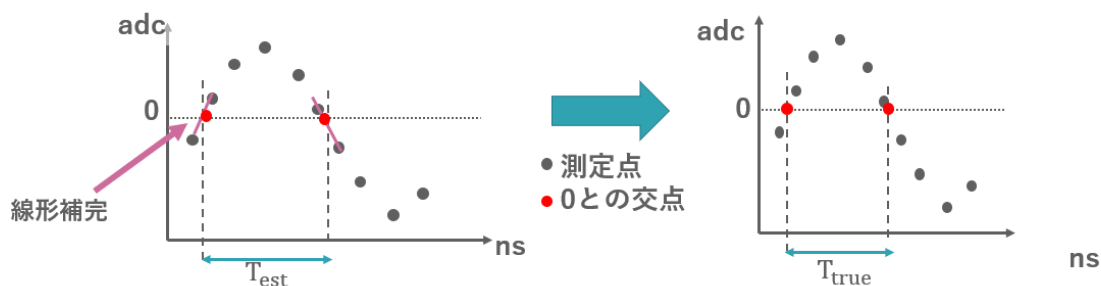


図 5.3.3: ROI=40 でのグローバルキャリブレーション

5.4 測定および解析環境

5.4.1 測定環境

図 5.4.1 に実際の測定を行う際のセットアップを示す。実際の LST2 号機以降の焦点面カメラに用いるものと同様の波形読み出し回路 (7 ピクセル)、テストパルスを生成するスローコントロールボード及び信号増幅す

るプリアンプ、読み出し回路への電源やテストパルス生成に用いるクロック信号の供給および測定した波形情報やコマンドの送受信を行うバックプレーン、読み出し回路への電源供給に用いる 24 V の電圧と 1 A 電流を供給できる直流安定化電源、読み出し回路の制御や測定データの解析にデスクトップ PC を使用している。波形読み出し回路には直流電圧が必要なため電源装置の正出力端子と GND 端子に BP を接続し出力電圧を +24V と設定し測定を行っている。また、波形読み出し回路でサンプリングをした後 AD 変換された信号波形の情報やコマンドの送信はバックプレーンを介して実験室のデスクトップ PC と波形読み出し回路間で送受信する。

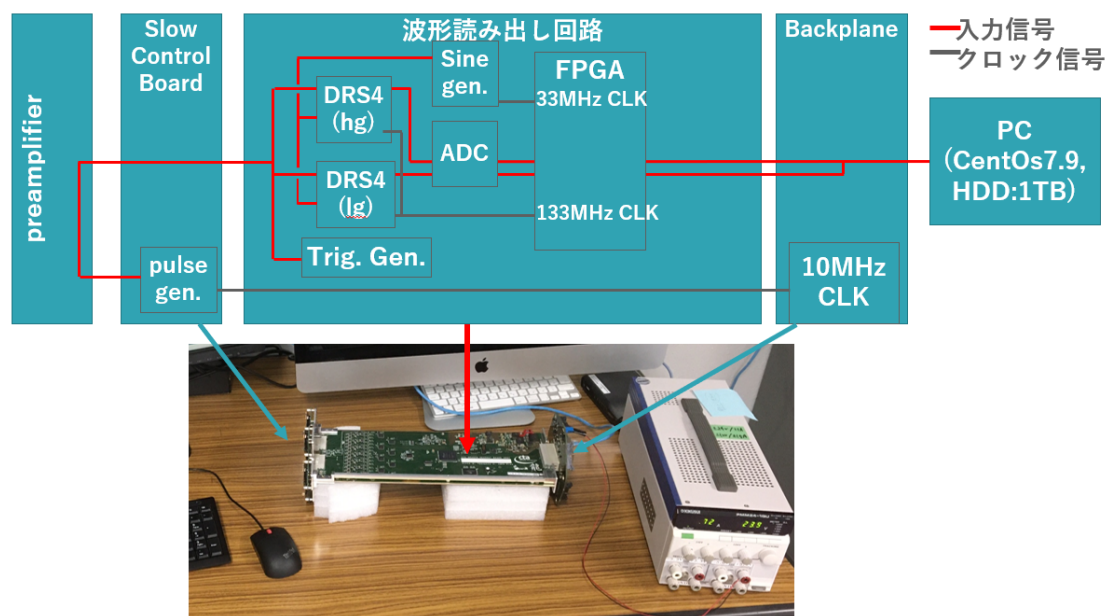


図 5.4.1: 測定のセットアップ

5.4.2 解析環境

5 章では 2 種類の解析ツールを使用して解析を行う。一つ目は LST で用いる波形読み出し回路の品質管理用の C、C++ 言語ベースのソフトウェア dragonqc。これは測定時の読み出し回路の操作や測定したサイン波の解析に用いた。二つ目は CTA の python 言語ベースの解析ソフトウェアである ctapipe および LST 用の解析ソフトウェアのテストベンチである cta-lstchain でありテストパルスの解析に用いる。また、実験室での測定データはバイナリ形式であり、ctapipe 及び lstchain でのファイル読み出しは fits.fz 形式のファイル形式でなければならないため実験室でのバイナリ形式のデータファイルを fits.fz 形式へと変換するソフトウェア ConverToZfits を用い、dragonqc 上での解析に ROOT を用いている。

5.4.3 解析に用いる波形データ

本章では DRS4 サンプリング時間幅較正に使用する読み出し回路上で生成したサイン波測定、SCB にて生成されるテストパルス測定及びそれらのペDESTAL補正に用いるペDESTAL測定を行う。以下にその詳細を示す。今回の測定では LST1 でのサンプリング周波数 1 GHz を実現するために FPGA 内の 66.666 MHz ク

ロック信号をリファレンスとし用い、クロック数を $N_{clock} = 66$ とすることでサンプリング周波数を 1.034 GHz と設定した。

・テストパルス測定

テストパルス測定では 4 章と同様に ROI は 40 スライスとした。トリガー周波数は 2kHz 固定とすることで DRS4 による読み出し始めるキャパシタの位置を約 7 スライスの間隔でずらしていき各キャパシタへテストパルスのピークが記録されるようにしている。またテストパルスの波高値はデスクトップのコマンドを介して読み出し回路上のファームウェアを書き換えることが可能で LST1 での設定と同じく高ゲイン系統で 40dB と設定している。これは 50 光電子に対応する波高値を実現している。テストパルスのイベント数は 20 万イベントとして測定を行った。テストパルスのペDESTAL補正に用いるペDESTAL測定は 4 章と同様に ROI は 40 スライス、トリガー周波数はランダムに平均で 2 kHz となるように設定し 20 万イベント測定を行った。ここでランダムなトリガー生成は読み出し回路のファームウェア上に実装されている線形帰還シフトレジスターによって行っている。

・サイン波測定

サイン波測定では通常の望遠鏡での観測やテストパルスの測定とは異なり DRS4 にかけるオフセット電圧を 2000 ADC counts ほど高く設定している。これは生成するサイン波の波高値が ± 450 mV であるのに対して DRS4 に入力できる電圧の範囲 -50 mV から +950 mV を超えてしまうからである。またこの設定はデスクトップ PC からコマンドを送ることで波形読み出し回路のファームウェアを書き換えることで行っている。サイン波測定で ROI は従来のサイン波較正で採用されていた 100 スライスと今回新たに 40 スライスでの測定を行う。これは実際の望遠鏡では ROI40 スライスをデフォルトにしておりこの設定を自由に変えられないため、40 スライスでのサイン波較正が求められているからである。二種類の ROI を用いた測定により 40 スライスでのサイン波較正が 100 スライスの場合と遜色なく較正することが出来るか確かめる。イベント数は 1 万イベントで行う。これは後述するサイン波のイベント数とサイン波較正後の電荷分解能の関係より 1 万イベント十分であると結論付けたためである。トリガー周波数は LST2 号機以降の品質管理テストを参考に 300 Hz としている。

5.5 大口径望遠鏡初号機におけるサンプリング時間幅較正

まず LST1 でのサンプリング時間幅較正で使用されているテストパルスを用いた時間幅較正を行う。テストパルス 20 万イベントを用いて時間幅計算をし、その時間幅を用いて電荷分解能を計算した。図 5.5.1 にテストパルスを用いたサンプリング時間幅較正前後の電荷量分布を示した。電荷分解能は較正前で 3.4%、較正後で 2.7% であり時間幅較正により 0.5% ほど電荷分解能が改善したことが分かる。

サイン波での較正と比べてテストパルスでの較正では主に二つの欠点がある。一つ目はイベント数がより多く必要である点である。テストパルスでは 20 万イベント程度必要なのに対してサイン波測定では 1 万イベントで較正が可能である。二つ目はクロックの問題である。テストパルスと DRS4 で異なるクロックをベースにしていたとしても、同期してしまうことがありそうだと全キャパシタに満遍なくパルスをふれなくなってしまいます。三つ目は較正の精度がサイン波較正の方が優れているということです。故に LST2-4 号機ではサイン波でのサンプリング時間幅較正を行う。

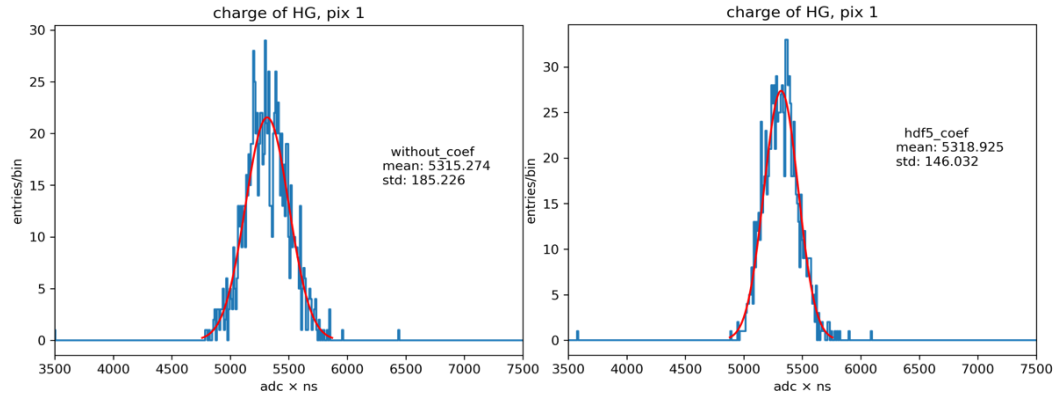


図 5.5.1: テストパルスを使用したサンプリング時間幅較正について、較正前 (左図)、較正後 (右図) の高ゲイン系統 1 ピクセルでの積分電荷のヒストグラム

5.6 大口径望遠鏡 2 から 4 号機用ボードでのサンプリング時間幅較正

次に LST2-4 号機用の波形読み出し回路に実装されているサイン波生成回路で生成したサイン波によるサンプリング時間幅較正を行う。

5.6.1 ROI=40 でのローカルキャリブレーション

従来の ROI=100 及び今回新たに行う ROI=40 のサイン波についてのローカルキャリブレーションを行った。まず、ROI=40,100 それぞれについて 30MHz のサイン波によるフィッティングを行った。図 5.6.1 は ROI=40 でのサイン波測定のデータ点とサイン波フィッティングの結果求めたサイン関数を赤線で示している。また、縦軸が ADC 値、横軸が時間 (ns) としている。このイベント以外にはフィット関数と概ね一致しているイベントもあるが、左図と中央の図の青丸で示したように波高値が高い場所でフィッティングとずれることがあると分かる。従来の ROI=100 での較正ではこの波高値の高いデータ点では時間幅を正しく求めることが出来ないため、時間幅を計算する際にサイン関数の振幅の 0.75 倍より低い波高値をもつデータ点の時間幅に関してのみ計算を行う。

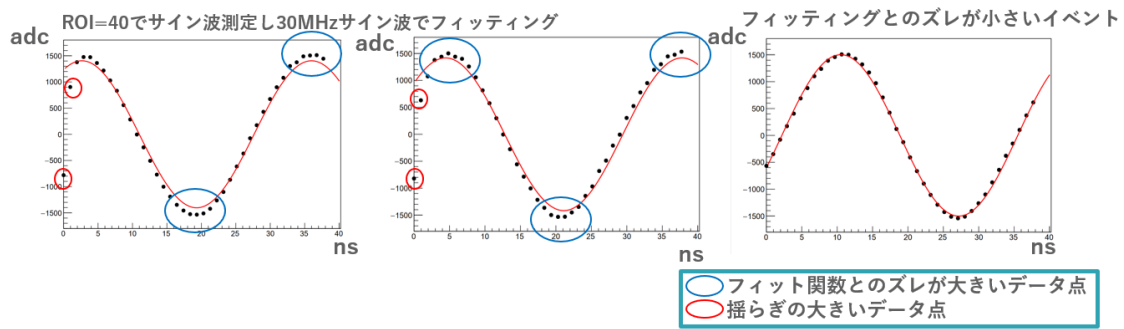


図 5.6.1: ROI=40 でのサイン波測定のデータ点とフィット関数

そこでまず、ROI=100、ROI=40 それぞれについてローカルキャリブレーション後の時間幅較正後の電荷分解能を調べた。その結果が図 5.6.2 である。ROI100 と比べて ROI40 の方が電荷分解能が 0.5% ほど低くなっていることが分かる。電荷分解能の結果にこのような差が生まれた原因として前述のフィット関数とのズレと考え ROI=40 でのサイン波フィッティングを改善する方法を考える。

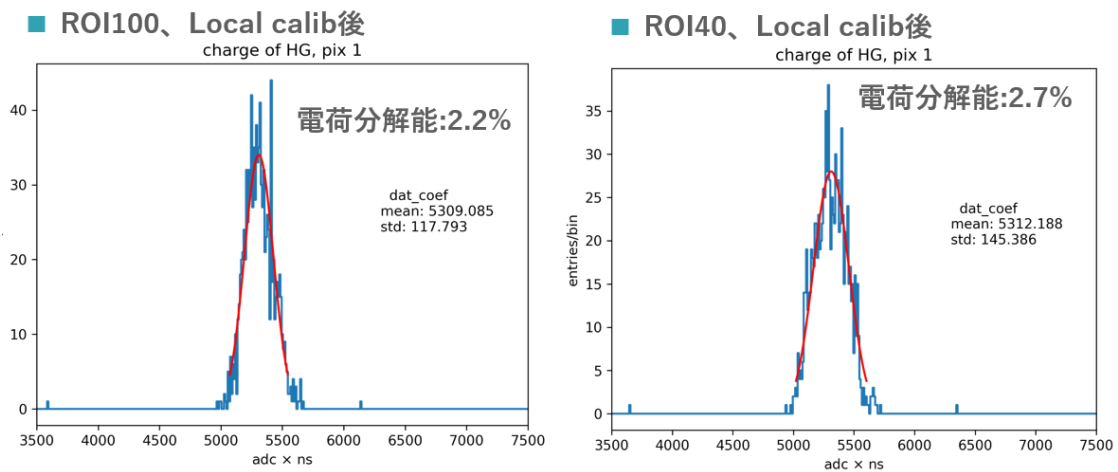


図 5.6.2: ROI=100 及び ROI=40 でのローカルキャリブレーション後の電荷分解能

高速フーリエ変換

LST2-4 号機用の波形読み出し回路に搭載されている 3 サイン波生成回路で生成される 30MHz のサイン波では、30MHz のサイン波に対して 0.56 %ほど 60MHz、0.19 %ほど 90MHz の成分が含まれていることが分かっている [18]。これらの成分が 30MHz のサイン波に混入することで 30MHz のサイン波フィッティングとズレが現れるのではないかと考えた。そこで、今回は FFT(高速フーリエ変換) と呼ばれる時系列データから直接フーリエ変換することでパワースペクトルを得る方法を用いて、生成され DRS4 で記録されたサイン波の高周波成分を調べる。まず ROI=40 でサイン波測定を行い、パワースペクトル解析を行う。図 5.6.3 にその結果を示す。

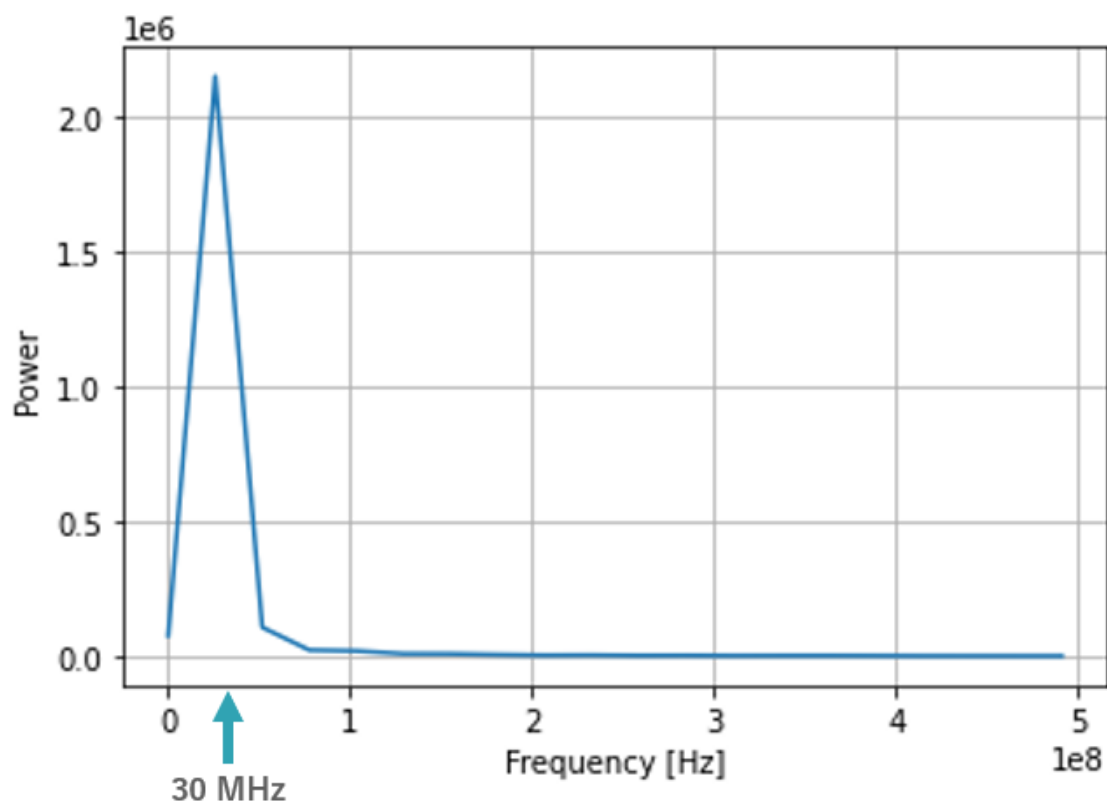


図 5.6.3: ROI=40 でのパワースペクトル解析

30MHz にピークが見られており、サイン波が 30MHz であることがわかる。ただし、横軸の分解能が悪い
ため 30MHz 以外の成分の有無を判別できない。そこで、F F Tでのデータ数 (ROI) と横軸の分解能が比例
関係であることから ROI を 1024 としてサイン波測定を行う。以下の図 5.6.4 にその結果を示した。

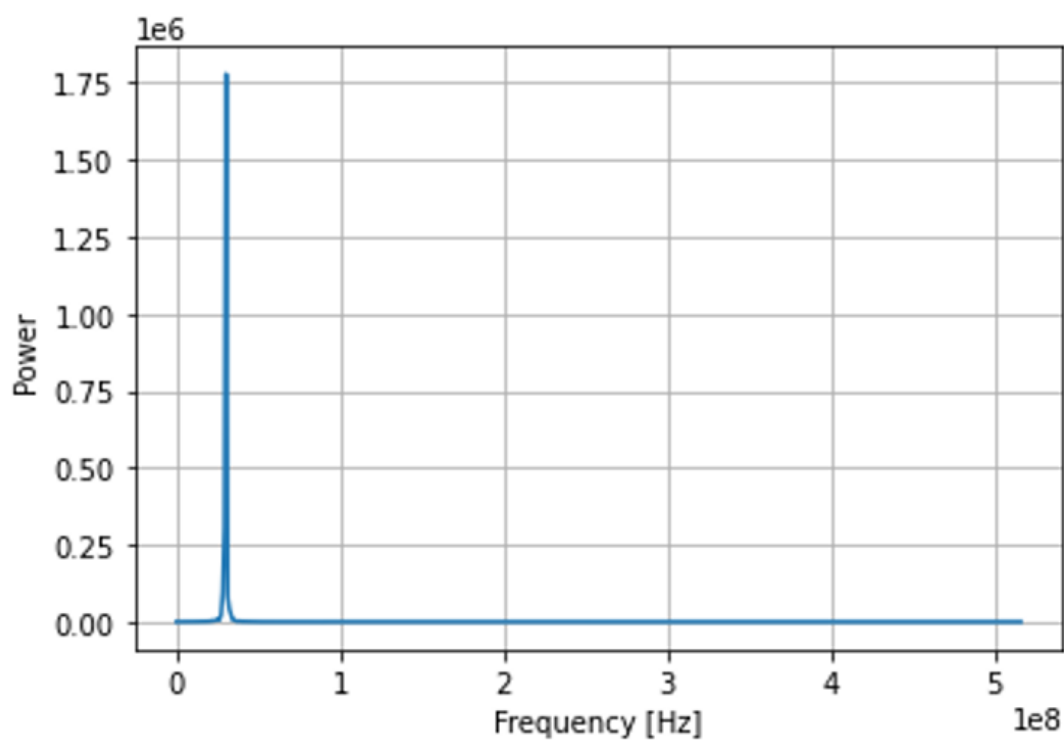


図 5.6.4: ROI=1024 でのパワースペクトル解析

横軸の分解能が改善したが、30MHz 以外にピークは確認できなかった。このことから、DRS4 で読み出されたサイン波測定データへの 60MHz 以降の高周波成分の寄与は十分に小さくフィッティングを悪くする影響はないと考えた。

DRS4 の波形読み出しでは ROI のうち最初の 2 セルと最後の 1 セルは揺らぎやすいことが知られており、実際に図 5.6.1 で赤丸で示すように最初 2 セルはフィット関数から大きくずれていることが分かる。そこで今回サイン波フィッティングに使用するキャパシタを ROI 全体から最初 2 セル、最後 1 セルを除いた残りでサイン波フィッティングを行った。図 5.6.5 にそのフィッティング結果を示した。

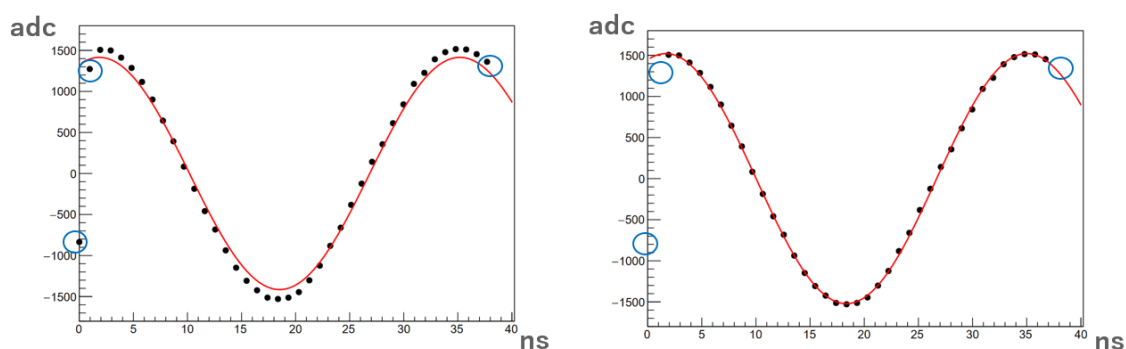
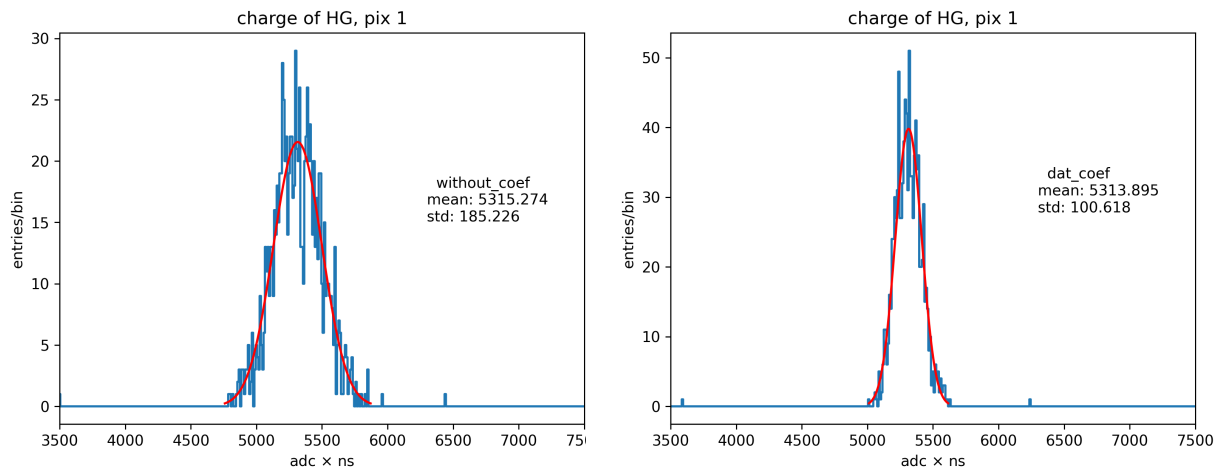


図 5.6.5: ROI=40 でのローカルキャリブレーション

左図が ROI=40 のサイン波測定データに関して全キャパシタ 40 セルを使用してフィッティングしたサイン波である。右図が最初 2 セル最後 1 セルを除いた 37 セルでサイン波フィッティングを行った結果である。グラフより青丸で示した最初 2 セル最後 1 セルを除いてフィッティングを行ったことでフィッティングのズレが大きく解消したことが分かる。このフィッティング方法を用いて時間幅較正をし、その較正前後の電荷分解能を調べる。その電荷分解能を調べる。図 5.6.6 にその結果を示す。左図が較正前の電荷分解能、右図が ROI=40 でのローカルキャリブレーションから時間幅較正を行って求めた電荷分解能である。較正前の電荷分解能が $3.48 \pm 0.03\%$ で較正後には電荷分解能が $1.89 \pm 0.02\%$ と較正によって 1.6% 程統計的に有意に電荷分解能が改善した。よって、今回フィッティング方法を最初 2 セル最後 1 セルを取り除くと変更したことで電荷分解能が改善した。

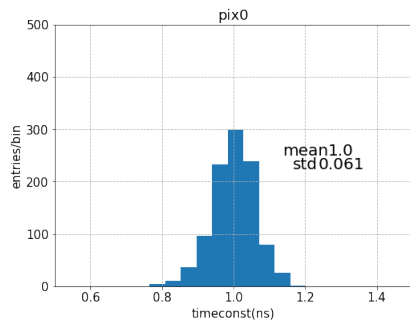


(a) 時間幅較正無し

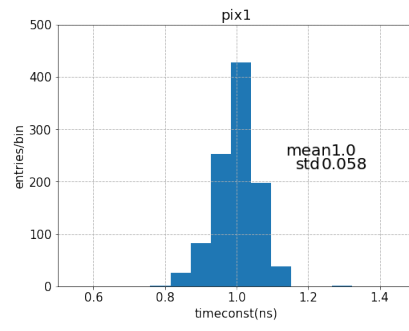
(b) ローカルキャリブレーション後

図 5.6.6: ROI=40 でのローカルキャリブレーション。左図が較正前の電荷量のヒストグラム。電荷分解能は $3.48 \pm 0.03\%$ 、右図が ROI=40 でのローカルキャリブレーションから時間幅較正を行って求めた電荷量のヒストグラムである。電荷分解能は $1.89 \pm 0.02\%$

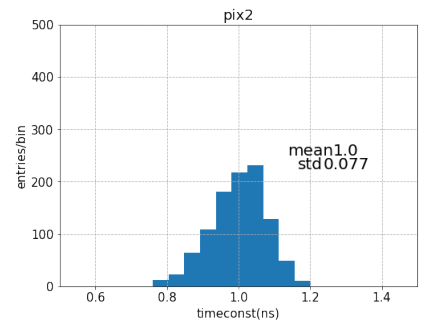
図 5.6.7 にローカルキャリブレーション + グローバルキャリブレーション後の時間幅のヒストグラムを、図 5.6.8 に pix=0 でのキャパシタ iD 毎の時間幅の散布図をそれぞれ示す。



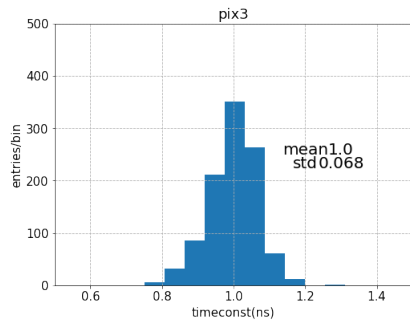
(a) Pix=0



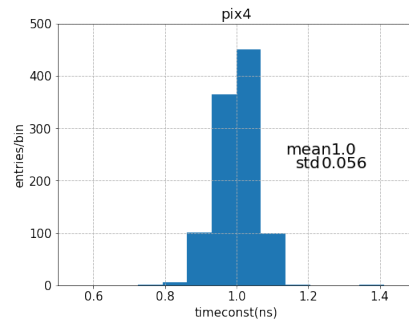
(b) Pix=1



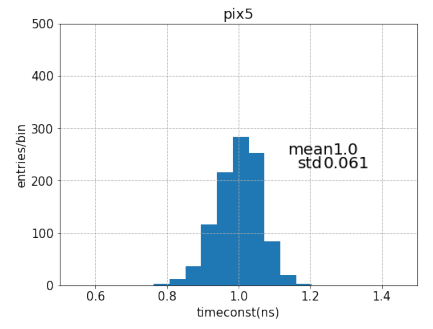
(c) Pix=2



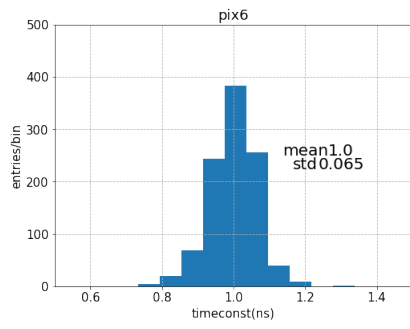
(d) Pix=3



(e) Pix=4



(f) Pix=5



(g) Pix=6

図 5.6.7: 各ピクセル毎の時間幅のヒストグラム

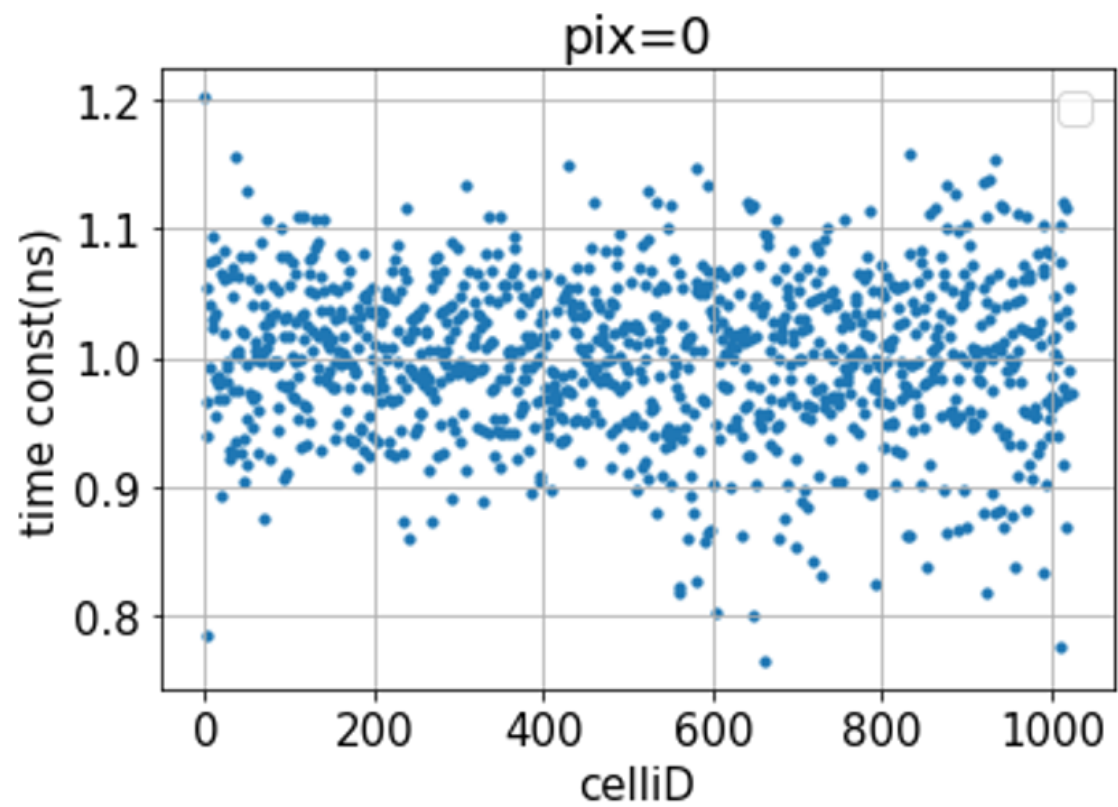


図 5.6.8: pix=0 でのキャパシタ iD に対する較正後の時間幅

図 5.6.7 より、較正後の時間幅は平均 1.000ns であり、時間幅の不均一性はおよそ 0.064 程度 ns であることがわかる。また、図 5.6.8 で各キャパシタ毎に固有の時間幅を持つと分かる。

次に、求めた較正後の時間幅を用いて電荷計算する。

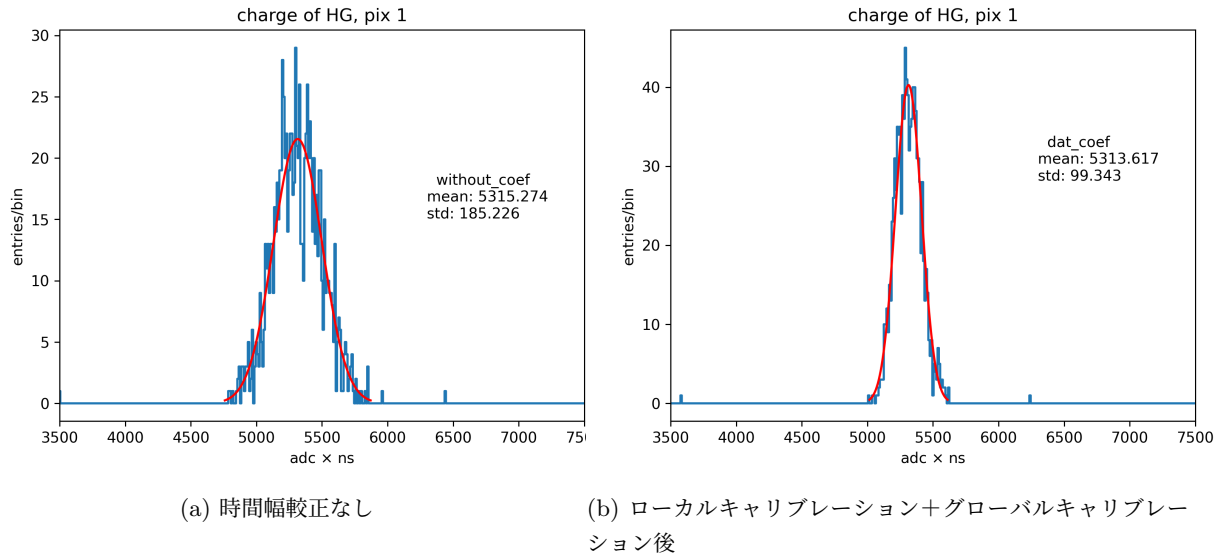


図 5.6.9: ROI=40 でのローカルキャリブレーション+グローバルキャリブレーション。左図が較正前の電荷量のヒストグラム。電荷分解能は $3.48 \pm 0.03\%$ 、右図が ROI=40 でのローカルキャリブレーションから時間幅較正を行って求めた電荷量のヒストグラムである。電荷分解能は $1.87 \pm 0.012\%$ で 1.6 ポイントほど電荷分解能が改善した。

図 5.6.9 で時間幅較正前とローカルキャリブレーション+グローバルキャリブレーション後の電荷分解能を比較した。時間幅較正後に電荷分解能が 1.6 ポイントほど改善し $1.87 \pm 0.012\%$ まで抑えることができた。

図 5.6.10 の左図は ROI=40 でのローカルキャリブレーション後の時間幅を用いて求めた電荷量のヒストグラムである。電荷分解能は $1.89 \pm 0.02\%$ 、右図は ROI=40 でのローカルキャリブレーション後にグローバルキャリブレーションを行って求めた時間幅を用いて求めた電荷量のヒストグラムである。電荷分解能は $1.86 \pm 0.02\%$

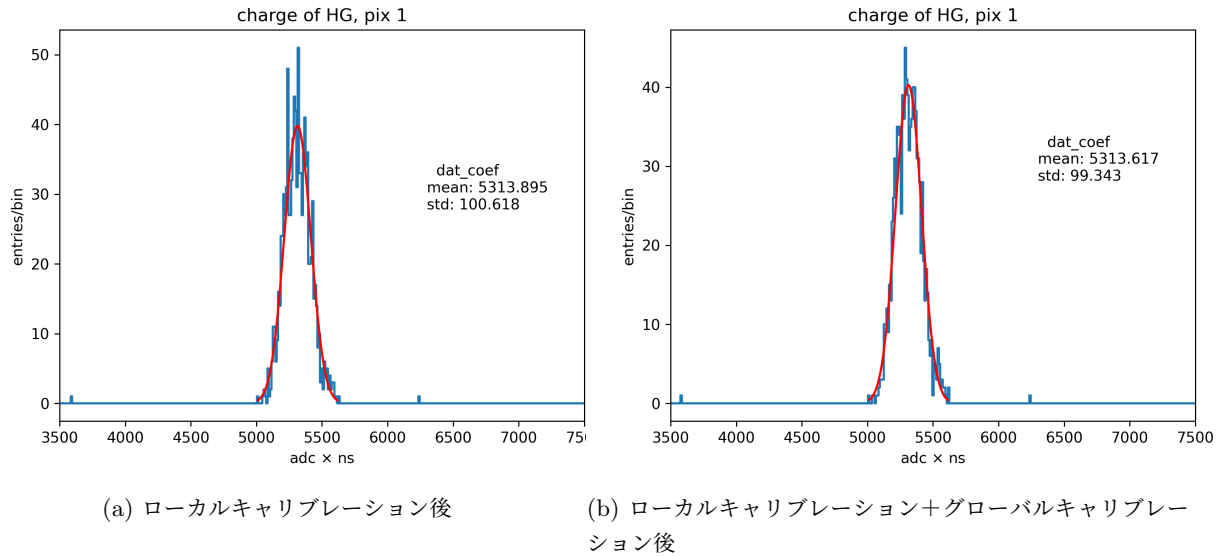


図 5.6.10: ROI=40 でのローカルキャリブレーション。左図がローカルキャリブレーション後の電荷量のヒストグラム。電荷分解能は $3.48 \pm 0.03\%$ 、右図が ROI=40 でのローカルキャリブレーション+グローバルキャリブレーション後の電荷量のヒストグラムである。電荷分解能は $1.87 \pm 0.02\%$

グローバルキャリブレーション前の結果より、統計的に有意に電荷分解能が改善しているとは言えない。そこで図 5.6.11 のように波形読み出しボード A と B の二枚を用意し合計 14 ピクセルについてサイン波測定を行い時間幅較正を行う。その後、その時間幅を用いて電荷分解能を計算し各ピクセル毎に平均の電荷分解能を求めプロットした。エラーバーは標準誤差とする。ここで今まで使用していた波形読み出し回路が A のボードである。

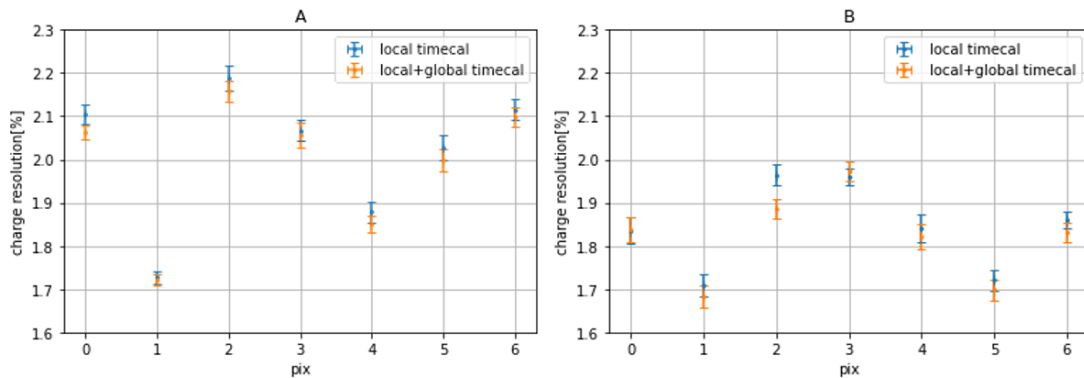


図 5.6.11: 二枚の波形読み出し回路 AB それぞれについて各 7 ピクセルでのサイン波による時間幅較正後の電荷分解能の平均値と誤差。それぞれ 10 回測定し 10 回分の電荷分解能の平均値をプロットしており、エラーバーは標準誤差としている。

図 5.6.11 から全体としてはグローバルキャリブレーション後に約 0.02% 程電荷分解能が改善しているが、各点のエラーバーがおおよそ 0.02 でありグローバルキャリブレーションの効果は薄いと考えられる。これは

元々ローカルキャリブレーションが不完全な補正であったため、より大局的な補正を行うグローバルキャリブレーションが必要であったが、今回ローカルキャリブレーションが改善したことでグローバルキャリブレーションの効果が小さくなったことが要因であると考えられる。

次に ROI=100 及び ROI=40 それぞれ最初 2 セル最後 1 セルを取り除くローカルキャリブレーション後にサイン波半周期でのグローバルキャリブレーションを行った。図 5.6.12 がその結果である。

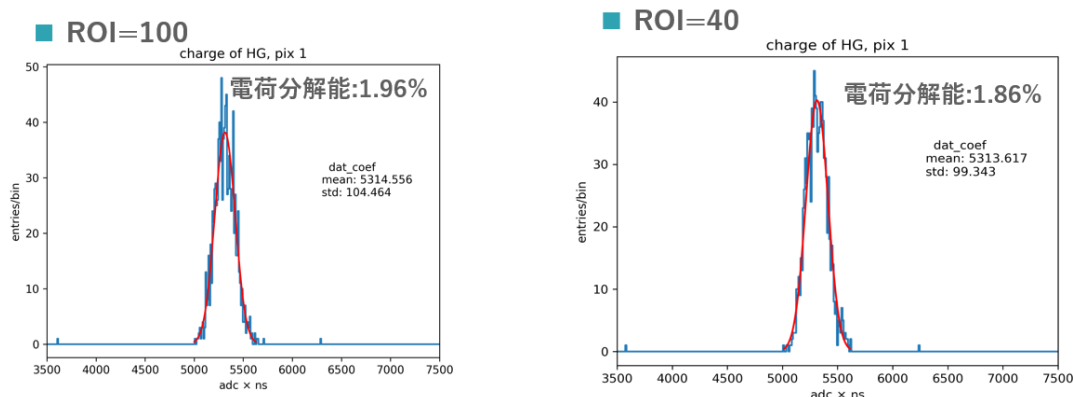


図 5.6.12: 左図が ROI=100 のサイン波での時間幅較正後の電荷量のヒストグラム、右図が ROI=40 のサイン波での時間幅較正後の電荷量のヒストグラム

その結果、ROI=100 では電荷分解能が 1.96%、ROI=40 で電荷分解能が 1.86% であった。この結果から今回のローカルキャリブレーション及びグローバルキャリブレーションでの較正方法の変更により、ROI=40 でのサンプリング時間幅較正によって従来の ROI=100 とほぼ同精度で較正が可能であることが分かった。

次に今回の結果が使用している Dragon board 固有の結果ではないことを確かめるために別の 2-4 号機用の Dragon board を使用する。これをボード B とし、これまで使用していた Dragon board をボード A とする。同様に ROI=40 でローカルキャリブレーション及びグローバルキャリブレーションを行い、サンプリング時間幅較正をする。今回は二枚の 2-4 号機用の Dragon board それぞれ 10 回ずつサイン波測定及び電荷分解能を調べるテストパルス測定を行い、各ボード、各ピクセル毎に平均の電荷分解能を求めた。図 5.6.13 および図 5.6.14 で示した。ここでエラーバーは各ボード各ピクセル毎の 10 測定分から求めた電荷分解能の標準誤差とした。

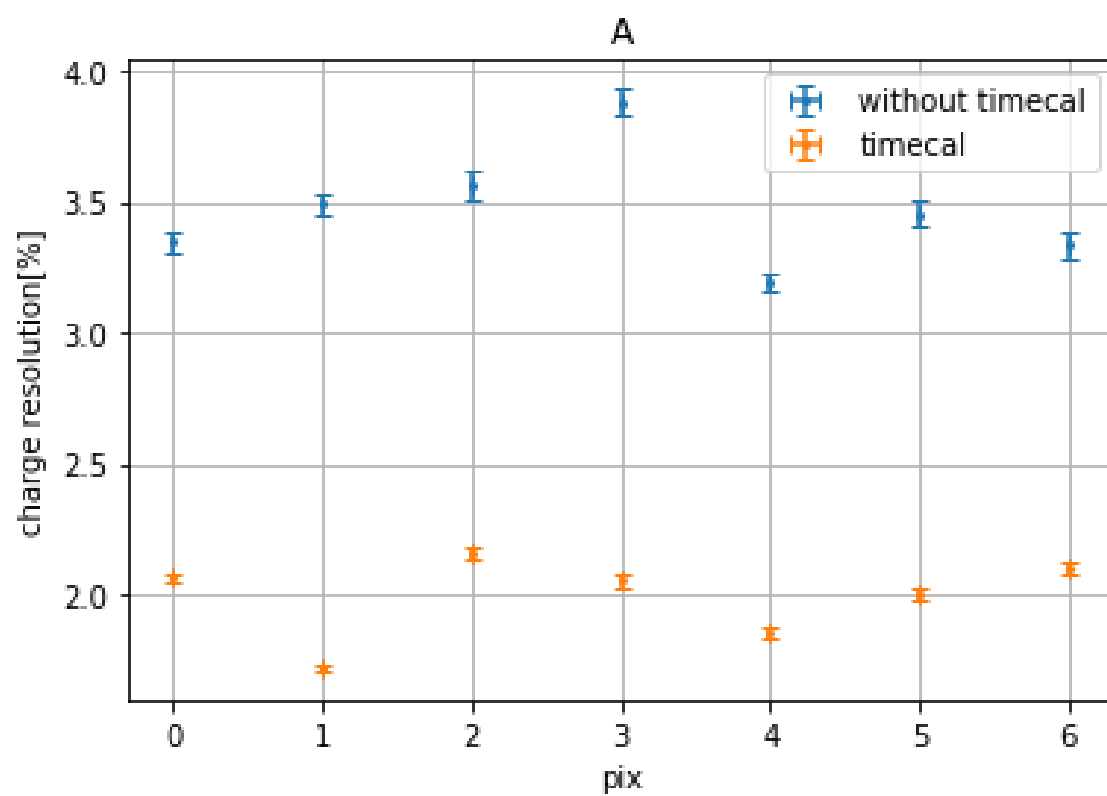


図 5.6.13

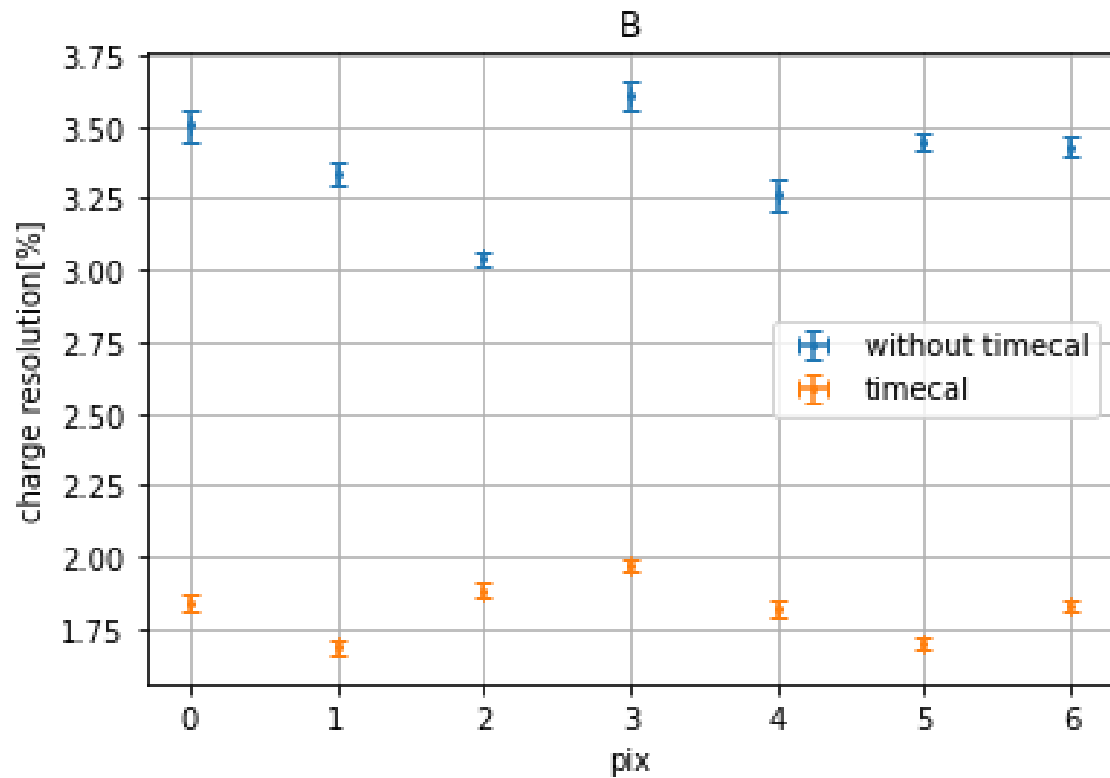


図 5.6.14: 横軸がピクセル番号、縦軸が電荷分解能。青線が時間幅較正前であり、黄色線が時間幅較正後の結果である。各データ点は 10 回測定分の平均値でありエラーバーは標準誤差を用いている。

各ピクセル全てにおいてサイン波によるサンプリング時間幅較正後に約 1.5 ポイント軒並み改善していることが分かる。次に各ボードそれぞれ 7 ピクセル分の電荷分解能の平均値を求めプロットした。エラーバーは同様に標準誤差とする。今回使用していたボード A では較正前が電荷分解能が $3.466 \pm 0.029\%$ から較正後には $1.992 \pm 0.018\%$ 、ボード B では電荷分解能が $3.375 \pm 0.026\%$ から較正後に $1.818 \pm 0.014\%$ と各ボードそれぞれで電荷分解能が較正後に約 1.5 ポイント程度改善していることが分かる。よって、本研究の目標値としていた 3% 以下に電荷分解能を抑えることが出来た。

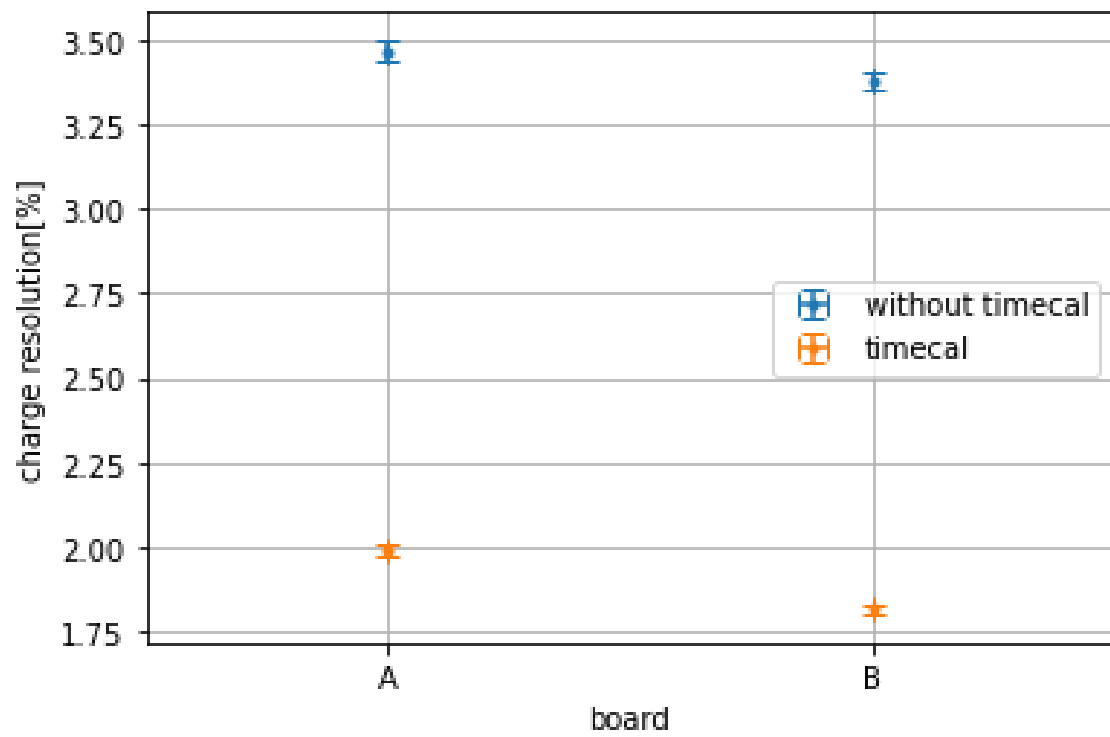


図 5.6.15: 各ボードごとのサイン波によるサンプリング時間幅較正前後の電荷分解能の変化

第 6 章

まとめと今後の展望

LST1 号機及び 2-4 号機用の波形読み出し回路へ波形を入力しない状態での電気ノイズについて各 DRS4 特性の補正を行いペデスタルノイズを評価し、パルスを入力した際の電気ノイズの影響を評価した。LST1 用のボードではペデスタル補正後で 7.5adc 値ほどの揺らぎがみられ dt 補正スパイク補正を行うことで揺らぎが 5.8adc 値ほどまで小さくなる。LST2-4 号機用のボードではペデスタル補正後で 6.5adc 値ほどの揺らぎが見られ、dt 補正スパイク補正を行うことで揺らぎが 6.0adc 値まで小さくなる。また、ペデスタルノイズはペデスタル補正後でも $\pm 6.5\text{adc}$ 値とサイン波やテストパルスと比べて非常に小さいため実際の較正ではそれらの影響は無視できると考えられる。また、単一のキャパシタに現れるスパイクのようなイベントはキャパシタ iD が 0 で発生しているためキャパシタ iD が 0 を含むイベントを除くことで無視することが出来る。また、単純には $\text{pix}=1,2,4,6$ ではそのようなイベントが発生していないので $\text{pix}=1,2,4,6$ での測定データを使用すればよい。5 章では LST2-4 号機用の波形読み出し回路を用いて $\text{ROI}=40$ でのテストパルス及び $\text{ROI}=100$ 及び 40 でのサイン波を用いたサンプリング時間幅特性の較正を行い電荷分機能を比較した。そうして補正された各サンプリング時間幅を用いてテストパルスに関して電荷計算をし、それぞれの補正精度を電荷分解能の観点で比較を行った。 $\text{ROI}=40$ でのサイン波でのサンプリング時間幅較正によって従来の $\text{ROI}=100$ とほぼ同精度で較正が可能であることが分かった。また今回用いた二枚の波形読み出し回路ボード A ボード B について、ボード A では較正前が電荷分解能が $3.466 \pm 0.029\%$ から較正後には $1.992 \pm 0.018\%$ 、ボード B では電荷分解能が $3.375 \pm 0.026\%$ から較正後に $1.818 \pm 0.014\%$ と各ボードそれぞれで電荷分解能が較正後に約 1.5 ポイント改善した。よって、実際の望遠鏡の ROI でのサイン波較正により本研究の目標値とした電荷分解能を 3% 以下に抑えることを達成し、LST2-4 号機でのサイン波によるサンプリング時間幅較正の較正手法を確立することが出来た。今後だが、本研究で使用したサイン波でのサンプリング時間幅較正をするスクリプトを CTA の解析フレームワークに入れ、LST2-4 号機での解析で使用する予定である。

付録

6.1 ROI=40 でのサイン波によるサンプリング時間幅を計算するスクリプト

サイン波を使ったサンプリング時間幅較正を行い、サンプリング時間幅を計算する C++ の簡易的なプログラム構成 6.1.1 及びスクリプトを以下に記す。このスクリプトの内紫文字で示した箇所が本研究で主な加筆修正した箇所である。本スクリプトの元となったスクリプトは dragonqc 用の解析プログラム群で実装されている AnalyzeTimeCal.C 用いている。ただし、以下のスクリプトは dragonqc に依存したスクリプトであるのでそれらの仕様変更等がある場合は使用する際に修正が必要になるので注意が必要である。

プログラム構成

ローカルキャリブレーション

```
DRS loop(ピクセル数){  
  イベントloop(イベント数){  
    データ点読み取り  
    サイン波フィッティング  
    crossing pointを探す  
    校正後の時間幅算出  
  }  
}  
校正後の時間幅平均計算  
ローカルキャリブレーション後の時間幅テーブル出力
```



グローバルキャリブレーション

```
ローカルキャリブレーション後の時間幅読み取り  
DRS loop(ピクセル数){  
  イベントloop(イベント数){  
    データ点読み取り  
    校正済みの時間幅を用いてサイン波の半周期を調べる  
    半周期の理想値と測定値の比を使い校正  
  }  
}  
校正後の時間幅の平均値計算  
ローカル+グローバルキャリブレーション後の時間幅テーブル出力
```

図 6.1.1: ROI=40 でのサイン波を用いた校正を行うスクリプトの簡易的なプログラム構成

```
1 #include <TR00T.h>
2 #include <TApplication.h>
3 #include <TProfile.h>
4 #include <TFile.h>
5 #include <TTree.h>
6 #include <TH1.h>
7 #include <TH2.h>
8 #include <TTimer.h>
9 #include <TCanvas.h>
10 #include <TF1.h>
11 #include <Getline.h>
12 #include <TH1F.h>
13 #include <TPaveText.h>
14 #include <iostream>
15 #include <fstream>
16 #include <sstream>
17 #include <TGraph.h>
18 #include <TGraphErrors.h>
19 #include <TMath.h>
20 #include "DRSSlices.h"
21 #include <vector>
22 #include "TMath.h"
23 // #include "TError.h"
24 #include <cmath>
25 #include <cstring>
26 #include <algorithm>
27 #include <iostream>
28 // #include "TString.h"
29 #include <Math/SpecFuncMathCore.h>
30 #include <Math/PdfFuncMathCore.h>
```

```

31 #include <Math/ProbFuncMathCore.h>
32 #include <cassert>
33 #include <iomanip>
34 using namespace std;
35
36 void ErrMsg(){
37
38
39     cout << " ##### " << endl;
40     cout << " ##### " << endl;
41     cout << " ##### " << endl;
42 }
43 void AnalyzeTimeCalLocal(TString infilename, const int refclock) {
44
45     //ここでローカルキャリブレーションを行う
46     const int depth = 40; //の設定。ROIROIの場合はここを書き換え
        る=100
47     const int totaldepth = 4096;
48
49     TCanvas *c1 = new TCanvas();
50     int flag=0;
51
52     c1->Print("Sinwave_1208fitplot.pdf"); //start
53
54     cout << infilename << " " << endl;
55
56     TFile *f = new TFile(infilename);
57     TTree *trEv = (TTree *) f->Get("Events");
58     DRSSlices *DRSSLcs[16];
59     for(int i=0;i<16;i++){
60         DRSSLcs[i] = new DRSSlices();

```

```

61     trEv->SetBranchAddress(Form("Event_ch%d",i), &(DRSSlcs[i]));
62 }
63 int Nev = trEv->GetEntries();//イベント数の設定。デフォルトでは全イベント
64 //int Nev = 5000;
65 cout << "Number of Events: " << Nev << endl;
66 double time1ns;
67 if(refclock==0){
68     time1ns = (65+1)/(66.666e6 * 1024) * 1e9;
69     //when register X1094=65 & internal clock
70 }else{
71     time1ns = (9+1)/(10.e6 * 1024) * 1e9; //when external
72     10MHz clock is used
73 }
74 cout<<"average time constant : "<<time1ns<<" ns"<<endl;
75
76 double adc, time;
77 double adcmax=0.;
78 vector<double> adc_vc(0);
79 int stopcell, ich;
80 int count_cell=0;
81 double v_amp, v_slope;
82 double dx_old, x, y, dx;
83 int cap_cor;
84 TGraphErrors *g_sin;
85 TGraphErrors *g_sin_2;
86 //TGraph *g_sin;
87 TGraphErrors *g_sin_corr;
88 //TGraph *g_sin_corr;
89 TF1 *f_sin;
90 //TGraph->DrawFrame(1,-1800,39,1800)

```

```

91  int index=0;
92
93  TH2D *h2_corr[8];
94  TH1D *h1_sin_amp[8];
95  TH1D *h1_sin_freq[8];
96  TH1D *h1_sin_base[8];
97  for(int idrs=0;idrs<8;idrs++){
98      h2_corr[idrs] = new TH2D(Form("h2_corr_drs%d", idrs),
99      Form("DRS %d;cell ID;time constant", idrs), 1024, -0.5
100      ,1023.5, 200, 0, 2);
101      h1_sin_amp[idrs] = new TH1D(Form("h1_sin_amp_drs%d", idrs),
102      Form("Sine wave amplitude (DRS %d); amplitude", idrs)
103      , 100, 1000,2000);
104      h1_sin_freq[idrs] = new TH1D(Form("h1_sin_freq_drs%d", idrs),
105      Form("Sine wave freq (DRS %d); amplitude", idrs),
106      100, 0.02,0.04);
107      h1_sin_base[idrs] = new TH1D(Form("h1_sin_base_drs%d", idrs),
108      Form("Sine wave baseline (DRS %d); amplitude", idrs)
109      , 100, -20,20);
110  }
111
112  for(int iev = 10; iev< Nev; iev++){
113      // skip first 10 events, 2015_08_10
114      if(iev%1000 == 0)
115          cout << "Event " << iev << endl;
116
117      trEv->GetEvent(iev);
118
119      for(int idrs=0; idrs<8; idrs++){
120          ich=idrs*2;
121          stopcell = DRSSlcs[ich]->GetStopCell();

```

```

122     time = 0.;
123     //
124     vector<double> adc_vc(0);
125     double adcmax=0.;
126     //g_sin = new TGraph();
127     g_sin = new TGraphErrors();
128     //g_sin_2 = new TGraphErrors();
129     //g_sin_corr = new TGraph();
130     g_sin_corr = new TGraphErrors();
131
132     //
133     for(int icell=1;icell<depth-1;icell++){
134         adc = DRSSLcs[ich]->GetSlice(icell);
135
136         adc_vc.insert(adc_vc.begin()+icell-1,abs(adc));
137
138
139     }
140     adcmax= *max_element(adc_vc.begin(),adc_vc.end());
141
142
143     // データ点を読み取る
144     for(int icell=0;icell<depth-1;icell++){
145         adc = DRSSLcs[ich]->GetSlice(icell);
146
147         if(icell!=0 && icell!=1 && icell!=depth-1){
148
149
150             g_sin->SetPoint(count_cell, time, adc);
151             g_sin->SetPointError(count_cell,0 , 6.5);
152

```

```

153             count_cell+=1;
154         }
155         time += time1ns;
156     }
157     //
158     int count_i =0;
159     int cellid_arr[count_cell];
160     for(int icell=0;icell<depth-1;icell++){
161         adc = DRSSlcs[ich]->GetSlice(icell);
162         if(icell!=0 && icell!=1 && icell!=depth-1){
163
164             cellid_arr[count_i]=(stopcell+icell)%1024;
165
166             count_i+=1;
167         }
168     }
169     count_i=0;
170
171
172     // サイン波でフィッティングを行う。
173     f_sin = new TF1("f_sin","[0]*sin(TMath::Pi()*2*[1]*x
174     +[2])+[3]", 3 , depth);
175     f_sin->SetParameters(1500,0.03,0,0);
176     //f_sin->SetParLimits(1,0.029,0.031);
177     f_sin->FixParameter(1, 0.03);
178     f_sin->FixParameter(3, 0.);
179     g_sin->Fit("f_sin", "Q");
180
181     v_amp= abs(f_sin->GetParameter(0));
182
183     h1_sin_amp[idrs]->Fill(v_amp);

```

```

184     h1_sin_freq[idrs]->Fill(f_sin->GetParameter(1));
185     h1_sin_base[idrs]->Fill(f_sin->GetParameter(2));
186
187     if(v_amp < 1000) continue; // bad fitting
188
189     v_slope = v_amp * 0.75;
190     dx_old = 99999;
191
192
193     for(int icell=0;icell<count_cell;icell++){
194         x = g_sin->GetX()[icell];
195         y = g_sin->GetY()[icell];
196         if(abs(y) > 0){ //only slope
197
198             dx =( ( f_sin->GetX(y, x-1., x+1., 1e-10,
199 100, false) ) - x) ;
200             //がフィット関数とのズレdx
201             g_sin_corr->SetPoint(icell, x+dx, y);
202
203             if(dx_old != 99999){
204                 //cap_cor = (stopcell+icell)%1024;
205                 cap_cor = cellid_arr[icell];
206
207                 h2_corr[idrs]->Fill(cap_cor, time1ns
208 - dx_old + dx);
209             }
210             dx_old = dx;
211         }else{
212             dx_old = 99999;
213
214         }

```

```

215     }
216
217
218
219     if(iev<300){
220         if(idrs==1) {//pixのサイン波のデータ点とフィット関数と較
正後のフィット点を描く=1
221             //g_sin->Print();
222             g_sin->SetMarkerStyle(20);
223             g_sin->SetMarkerColor(1);
224             g_sin->SetLineColor(1);
225             g_sin->GetXaxis()->SetRangeUser(0,depth);
226             g_sin->GetYaxis()->SetRangeUser(-1800,1800);
227             g_sin->Draw("AP");
228             //duble chi2 = f_sin->GetChisquare();//chi-square
229             //double Ndof = f_sin->GetNDF();
230             //g_sin_corr->Print();
231             g_sin_corr->GetXaxis()->SetRangeUser(0,depth);
232             g_sin->GetYaxis()->SetRangeUser(-1800,1800);
233             g_sin_corr->SetMarkerColor(2);
234             g_sin_corr->SetMarkerStyle(20);
235             g_sin_corr->Draw("PSAME");
236
237             //}
238             //f_sin->SetMarkerColor(1);
239
240             c1->Update();
241             //f_sin->Draw("LSAME");
242             c1->Print("Sinwave_1208fitplot.pdf");
243
244

```



```

245         }
246     }
247     //
248
249     delete g_sin;
250     delete g_sin_corr;
251     delete f_sin;
252     count_cell=0;
253 } // DRS loop
254
255 } //event loop
256 //1106
257 c1->Print("Sinwave_1208fitplot.pdf"); //end
258 //
259
260
261
262 //c1->SetLogy();
263 //c1->Print("Sinwave_chi-square1208fit.png");
264 /*+
265 TCanvas *c1 = new TCanvas();
266 g_sin->Draw("AP");
267 g_sin->SetMarkerStyle(20);
268 g_sin_corr->Draw("PSAME");
269 g_sin_corr->SetMarkerColor(2);
270 g_sin_corr->SetMarkerStyle(20);
271 f_sin->Draw("LSAME");
272 */
273
274 TH1D *h_cal[8];
275 TGraph *g_cal[8];

```

```

276 TProfile *pf_cal[8];
277 for(int idrs=0;idrs<8;idrs++){
278     h_cal[idrs] = new TH1D(Form("h_time_drs%d",idrs),Form("time
279     constant(DRS %d);time constant[ns]",idrs),200,0,2);
280     g_cal[idrs] = new TGraph();
281     g_cal[idrs]->SetName(Form("g_time_drs%d",idrs));
282     g_cal[idrs]->SetTitle(Form("time constant(DRS %d)
283     ;time constant[ns]",idrs));
284     pf_cal[idrs] = h2_corr[idrs]->ProfileX();
285 }
286
287 TString outtablename = infilename;
288 outtablename.Replace(outtablename.Last('.'), 5,
289 "_TimeConstant_Local.dat");
290 ofstream ofs(outtablename);
291
292 double time_sum[8]={};
293 for(int icell=0;icell<1024;icell++){
294     for(int idrs=0;idrs<8;idrs++){
295         time_sum[idrs] += pf_cal[idrs]->GetBinContent(icell+1);
296     }
297 }
298
299 double time_constant;
300 for(int icell=0;icell<1024;icell++){
301     for(int idrs=0;idrs<8;idrs++){
302         time_constant = pf_cal[idrs]->GetBinContent(icell+1)
303         /time_sum[idrs]*1024.;
304         h_cal[idrs]->Fill(time_constant);
305         g_cal[idrs]->SetPoint(icell, icell, time_constant);
306

```

```

307         cout<<time_constant<<" ";
308         ofs<<time_constant<<" ";
309     }
310     cout<<endl;
311     ofs<<endl;
312 }
313
314 TString outfilename = infilename;
315 outfilename.Replace(outfilename.Last('.'), 5,
316 "_TimeCalLocalAna.root");
317 TFile *outfile = new TFile(outfilename, "RECREATE");
318 for(int idrs=0;idrs<8;idrs++){
319     h_cal[idrs]->Write();
320     g_cal[idrs]->Write();
321     h2_corr[idrs]->Write();
322     pf_cal[idrs]->Write();
323     h1_sin_amp[idrs]->Write();
324     h1_sin_freq[idrs]->Write();
325     h1_sin_base[idrs]->Write();
326
327 }
328
329 }
330
331
332 void AnalyzeTimeCalGlobal(TString infilename, const int refclock) {
333     //ここでグローバルキャリブレーションを行う
334     const int depth = 40 ;//の設定。ROIROIの場合はここを書き換
    える=100
335     const int totaldepth = 4096;
336

```

```

337 // TEMPORALY READ CAL DATA
338 //double t[][4096];
339 TString tablename = infilename;
340 tablename.Replace(tablename.Last('.'), 5,
341 "_TimeConstant_Local.dat");
342 double t_corr[8][1024];
343
344
345 int cellbuf;
346 double buf;
347 ifstream ifs(tablename, ios::in);
348
349 double time1ns;
350 if(refclock==0){
351     time1ns = (65+1)/(66.666e6 * 1024) * 1e9;
352     //when register X1094=65 & internal clock
353 }else{
354     time1ns = (9+1)/(10.e6 * 1024) * 1e9;
355     //when external 10MHz clock is used
356 }
357 cout<<"average time constant : "<<time1ns<<" ns"<<endl;
358
359 int itemp=0;
360 while(ifs>>buf){
361     t_corr[itemp%8][itemp/8] = buf * time1ns;
362     itemp++;
363     //cout <<"buf" << buf <<"t_corr[itemp%8][itemp/8]" << " " <<
364     t_corr[itemp%8][itemp/8] << endl;
365 }
366
367 // input data

```

```

368 TString outfile = infile;
369 outfile.Replace(outfile.Last('.'), 5,
370 "_TimeCalGlobalAna.root");
371 cout << infile << " " << outfile << endl;
372
373 TFile *f = new TFile(infile);
374 TTree *trEv = (TTree *) f->Get("Events");
375 DRSSlices *DRSSlcs[16];
376
377 for(int i=0;i<16;i++){
378     DRSSlcs[i] = new DRSSlices();
379     trEv->SetBranchAddress(Form("Event_ch%d",i), &(DRSSlcs[i]));
380 }
381
382 int Nev = trEv->GetEntries();
383 cout << "Number of Events: " << Nev << endl;
384
385 // definition
386 int cellnum, cellnum_tmp;
387 double adc, adc_old;
388 const double adc_th=0; //threshold
389 double sign;
390 double t_real, t_real_corr, t_start, t_finish, t_frac;
391 double t_ideal = 1/(30.e6) * 1e9; //ns
392 double frac_start, frac_finish;
393
394 int stopcell;
395 int ch;
396
397 TH2D *h2_cell_period[8];
398 TH2D *h2_cell_period_corr[8];

```

```

399  TH2D *h2_cell_time_corr[8];
400
401  for(int idrs=0;idrs<8;idrs++){
402      h2_cell_period[idrs]  =new TH2D(Form("h2_cell_period_drs%d",
403      idrs),Form("Period Time Test(DRS%d);cell_id;period[nsec]",
404      idrs),1024,0,1024,600,30,36);
405      h2_cell_period_corr[idrs]  =new
406      TH2D(Form("h2_cell_period_corr_drs%d", idrs),
407      Form("Period Time Test(DRS%d);cell_id;period[nsec]",
408      idrs),1024,0,1024,600,30,36);
409
410      h2_cell_time_corr[idrs]  =new
411      TH2D(Form("h2_cell_time_corr_drs%d", idrs),Form("corr
412      time(DRS%d);cell_id;corr time[nsec]",
413      idrs),1024,0,1024,200,0,2);
414  }
415
416  bool first_point_flag;
417  int first_point;
418  int last_point;
419
420  for(int i = 10; i< Nev; i++){
421      // skip first 10 events, 2015_08_10
422
423      if(i%1000 == 0)
424          cout << "Event " << i << endl;
425
426      trEv->GetEvent(i);
427
428      for(int ndrs=0; ndrs<8; ndrs++){
429          ch=ndrs*2;

```

```

430     stopcell = DRSSLcs[ch]->GetStopCell();
431
432     // Set data point
433     first_point_flag = true;
434     t_real = 0.;
435     t_real_corr = 0.;
436
437     for(int j = 0; j < depth; j++){
438         adc = DRSSLcs[ch]->GetSlice(j);
439         cellnum = (stopcell + j)%4096;
440
441         if(j<3){
442             adc_old = adc;
443             continue;
444         }
445
446         // crossing を探すpoint
447         if((adc_old - adc_th)*(adc - adc_th)<0){
448
449             if(first_point_flag){
450                 first_point_flag = false;
451                 sign = adc - adc_old;
452                 frac_start = (adc - adc_th)/(adc - adc_old);
453
454                 // first bin
455                 first_point = j;
456                 t_real = time1ns * frac_start;
457                 t_real_corr = t_corr[ndrs][ (4096+cellnum)%1024]
458                 * frac_start;
459
460

```

```

461     }else{
462         if( (adc - adc_old)*sign < 0){
463             //time interval from the last bin to threshold
464             frac_finish = (adc_old - adc_th)/(adc_old - adc);
465
466             last_point = j;
467             t_real += time1ns * frac_finish;
468             t_real_corr += t_corr[ndrs][(4096+cellnum)%1024]
469 * frac_finish;
470
471             // Fill
472             h2_cell_period[ndrs]->Fill(cellnum%1024, t_real);
473             h2_cell_period_corr[ndrs]->Fill(cellnum%1024,
474 t_real_corr);
475
476             // correction
477             t_frac = (t_ideal*0.5)/t_real_corr;
478 //半周期の場合なのでt_idealとしている*0.5
479             //t_frac = t_ideal/t_real_corr;
480             for(int icorr=first_point;icorr<last_point; icorr++){
481                 cellnum_tmp = (stopcell + icorr)%4096;
482                 h2_cell_time_corr[ndrs]
483 ->Fill(cellnum_tmp%1024,
484 t_corr[ndrs][cellnum_tmp%1024] * t_frac);
485             }
486
487             // first bin
488             frac_start = (adc - adc_th)/(adc - adc_old);
489
490             first_point = j;
491             t_real = time1ns * frac_start;

```



```

492         t_real_corr = t_corr[ndrs][(4096+cellnum)%1024]
493     * frac_start;
494     //cout << "first_point " << first_point << endl;
495     }else{
496         t_real+= time1ns;
497         t_real_corr+=t_corr[ndrs][(4096+cellnum)%1024];
498     }
499     }// search threshold point
500
501     }else{
502         // from the first bin to the last bin
503         t_real+= time1ns;
504         t_real_corr+=t_corr[ndrs][(4096+cellnum)%1024];
505     }
506
507     adc_old = adc;
508 } // cell loop
509 } // DRS loop
510
511 } //event loop
512
513 TFile *outfile = new TFile(outfilename, "RECREATE");
514
515 TProfile *pf_corr[8];
516
517 for(int idrs=0;idrs<8;idrs++){
518     h2_cell_period[idrs]->Write();
519     h2_cell_period_corr[idrs]->Write();
520     h2_cell_time_corr[idrs]->Write();
521
522     pf_corr[idrs] = h2_cell_time_corr[idrs]->ProfileX();

```

```

523     pf_corr[idrs]->Write();
524 }
525
526 TString outtablename = infilename;
527 outtablename.Replace(outtablename.Last('.'), 5,
528 "_TimeConstant_Final.dat");
529
530 ofstream ofs(outtablename);
531
532 double time_sum[8]={};
533 for(int icell=0;icell<1024;icell++){
534     for(int idrs=0;idrs<8;idrs++){
535         time_sum[idrs] += pf_corr[idrs]->GetBinContent(icell+1);
536     }
537 }
538
539 for(int i=0;i<1024;i++){
540     for(int idrs=0;idrs<8;idrs++){
541         ofs<< pf_corr[idrs]->GetBinContent(i+1)
542         /time_sum[idrs]*1024.<<" ";
543     }
544     ofs<<endl;
545 }
546
547
548 }
549
550
551 /*-----*/
552 int main(int argc, char* argv[]) {
553

```

```
554     if(argc !=3){
555         printf("Usage: %s filename reference_clock(0:local, 1:external)
556         \n", argv[0]);
557         return -1;
558     }
559
560     TString infilename = (TString) argv[1];
561     const int refclock = atoi(argv[2]);
562
563     cout << infilename << endl;
564
565     TApplication app("app",&argc, argv);
566
567
568     AnalyzeTimeCalLocal(infilename, refclock);
569
570     AnalyzeTimeCalGlobal(infilename, refclock);
571
572     return 0;
573 }
```

6.2 サイン波でのサンプリング時間幅較正の実験及び解析マニュアル

6.2.1 実験

準備

- Dragon board(波形読み出し回路)
- Slow control board
- PACTA(プリアンプ) × 7
- Back plane(BP)
- 直流安定化電源 (1A,24V)
- Linux PC

(ROOT v.6.16.00, Dragonqc 用のプログラム群, ConverToZfits.C(CTA 内部のプログラム), ctapipe(v0.11.0), lstchain(v0.8.4))

注意点

- Dragon に電源を繋ぐ際は +24V で出力する前に電源の設定電圧を確認すること。
- 体内の静電気をよく除去しておくこと

セットアップ

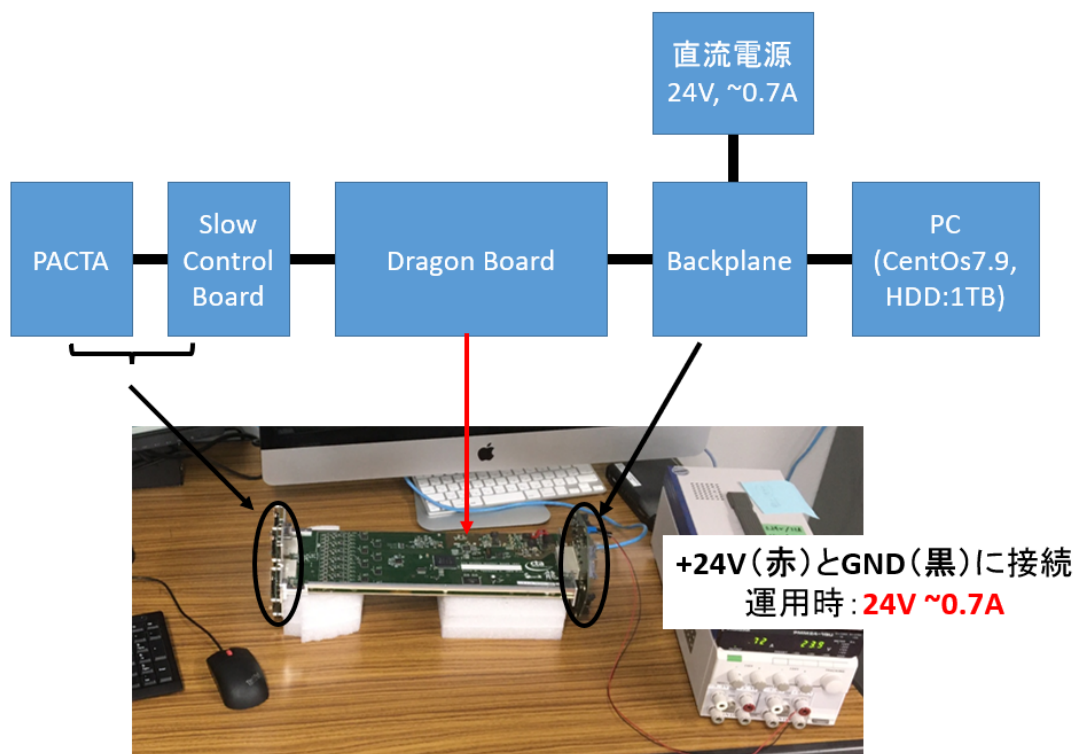


図 6.2.1

図及び写真のように Dragon Board を PC 及び直流電源に接続する。この時以下の写真のようにコネクタの形状が L 字になっているので向きに気を付ける。

Dragonとの接続

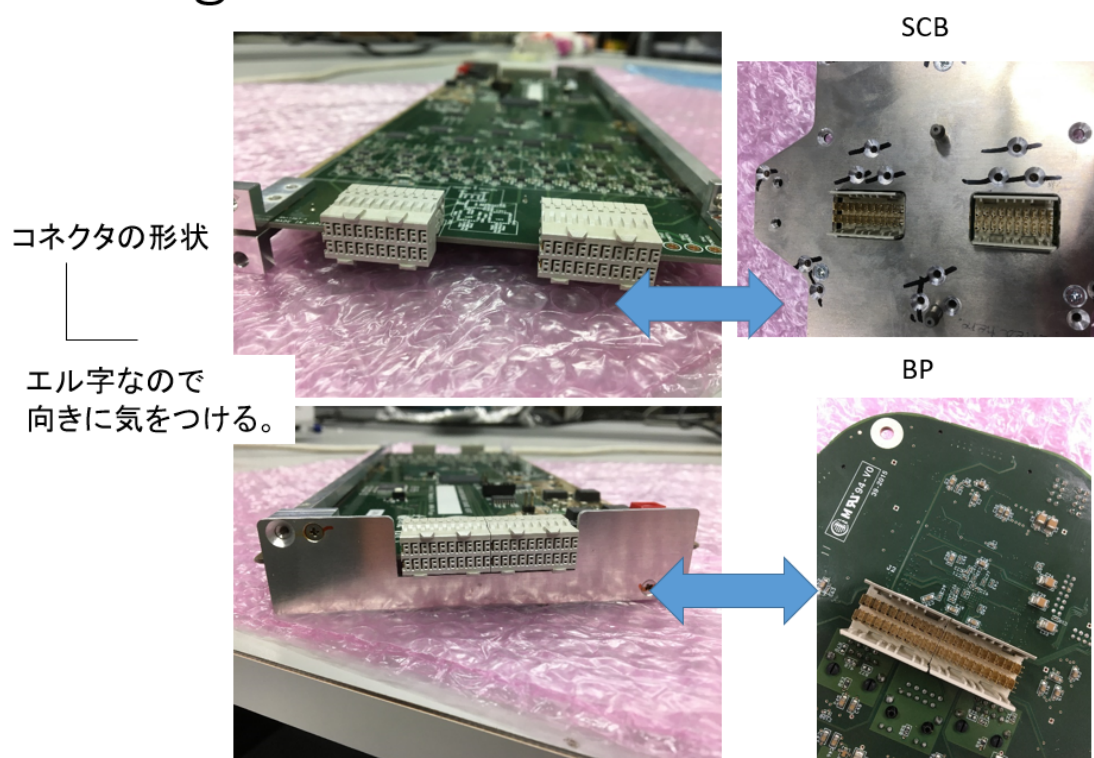


図 6.2.2

テストパルス測定

テストパルスでの時間幅較正のほか電荷分解能を計算する際に使用する波形データの測定。測定では Dragonqc 用のプログラム群を用いる。

①SetRandomPedestalRunInt.C

ペDESTAL較正用のデータ取得するスクリプト。引数は Dragon の IP、トリガーレート=2000 Hz、seed=5 (5 でなくても良い)

②SetBPClock.C

10MHz クロックの分配開始するスクリプト。引数は Dragon の IP

③SetLinearityRunExtRef.C

テストパルスの生成開始するスクリプト。引数：IP、チャンネル (255 で全ピクセル)、Delay= 526 ns (ROI = 40 のとき)、クロック数= 1 万 (Freq=10000 ($1/(100 * \text{Freq})=1000$ Hz), offset=10

④SetTpGain.C

テストパルスの波高値設定するスクリプト。引数：IP、ゲイン= 40 dB (50p.e. 相当の場合、60 dB で 500p.e. 相当)

- ⑤Eth IP ROI イベント数 filename
測定開始するスクリプト

サイン波測定

TakeTimingCal.sh

サイン波測定及びペDESTALの差し引きに使用するペDESTAL測定を行うシェルスクリプト。引数:IP ROI。
中身は TakePedestalRunForTimecal.sh と SetTimingCal.C を実行

- ①TakePedestalRunForTimecal.sh

SetPedestalRun.C と SetBaselineLevel.C(ベースラインを 2000adc 値上げる。尚、ペDESTALの差し引きを行う際にキャンセルされる) を実行するシェルスクリプトでペDESTAL測定を行う。

- ②SetTimingCal.C サイン波測定を行うスクリプト。出力はサイン波形データの dat ファイル。

6.2.2 解析

サイン波測定データの解析では Dragonqc 用のプログラム群を用いる。

サイン波を用いたサンプリング時間幅較正

- ①Rootify sine.dat pedfile ROI

サイン波測定で出力されたサイン波形データ (dat ファイル) について同じく出力されるペDESTALファイル、Rootify を用いてサイン波形データの root ファイルを作成する。

- ②AnalyzeTimeCalLST2-4.C sine.root 0(DRS4 のリファレンスクロックの選択)

ROI=40 のサイン波での時間幅較正を行うスクリプト AnalyzeTimeCal.C を用いて測定したサイン波測定データをもとに時間幅較正を行う。出力はローカルキャリブレーション後の時間幅テーブルと、さらにグローバルキャリブレーションも行った時間幅テーブルのテキストファイルが出力される。

テストパルスを用いたサンプリング時間幅較正

テストパルスを用いたサンプリング時間幅較正は lstchain と ctapipe を使用して較正を行う。

- ①ConverToZfits.C testpulse.dat イベント数

測定で得られたテストパルス波形データ (dat ファイル) を lstchain で読めるように ConverToZfits.C で zfits(～fits.fz) ファイルに変換する。

- ②Sampling_interval_creation.py

テストパルスデータ (zfits ファイル) を用いてテストパルスでのサンプリング時間幅較正を行う。引数は以下の通りである。

- 1.zfits ファイル (較正用テストパルスデータ)
2. 出力ファイル (時間幅テーブル、hdf5 形式)
3. ペDESTALテーブル (fits 形式)
4. ゲインの選択 (0 なら HG、1 なら LG)
5. テーブル作成に用いるイベント数
7. 電荷計算するイベント数

謝辞

本修士論文を作成するにあたり、指導教員である片桐先生と副指導教員である吉田先生には学部4年生から3年間日々の進捗報告やゼミなどで研究内容だけでなくミーティングの報告の作法や見やすく伝わりやすいパワーポイントの作成の仕方など多方面で手厚いご指導を頂きました。また、この研究生生活を通して予定管理の不足などで沢山のご迷惑をおかけしたこと大変申し訳なく思っております。自分の理解が足らずゼミや研究などで長時間にも関わらずご指導いただいたこと大変感謝しております。新型コロナウイルスなどの影響もありオンラインでの研究をしなければならない期間もありましたがオンラインでの進捗報告や個別報告など研究を続けるためのサポートをしていただきました。おかげさまで大学に入れない期間も不都合少なく研究を続けることが出来ました。ありがとうございました。また、本研究を通して物事に取り組む際の計画の立案とその実行について今後も重要となるだろう考えや取り組む姿勢を沢山学ばせていただきました。まだまだ至らないところばかりですが頂いた教えを胸によりこれからも精進致します。

CTA-japan グループの皆様、特にFPI/CAL/ELEC グループの皆さまには研究に躓いた際に沢山サポートしていただきました。定期ミーティングでの進捗報告では研究で躓いた点や至らない点を丁寧にご教授いただき円滑に研究を進めることが出来ました。また特にマックスプランク物理学研究所の野崎誠也さんや宇宙線研究所の齋藤隆之助教、京都大学の岡智彦さんにはミーティング外でもslackを通してより具体的な問題解決にご助力頂きより有意義な研究活動を行うことが出来ました。大変感謝しております。本研究で重要な時間幅較正前のDRS4の諸特性の較正に関して、当時データ解析が不慣れでlstchainなどを上手く扱えなかった私に実際のctapeやlstchainの扱い方、DRS4の諸特性の較正方法や関連する過去の資料などを懇切丁寧に教えていただきました。おかげさまでlstchainなどの解析ソフトを自分で使用してDRS4の諸特性の較正を行うことができました。その他にも研究に関する疑問や躓きを懇切丁寧にご教授していただき誠に感謝しております。

また、6階7階の宇宙系の研究室の皆さまからは研究に関する相談に限らずプライベートな雑談など大変お世話になりました。皆様が研究のモチベーションをこの三年間保つことの大きな要因であったことか間違いありません。私自身1人家で勉強することが苦手なのもあり、皆と共に研究できる環境はとても居心地がよく研究を進めるうえで自分に合った環境でした。また、互いに分からない所を相談しあったり、研究室で共に切磋琢磨することができたこと大変感謝しております。

最後に、大学浪人で沢山の苦勞をかけてしまったにも拘わらず修士まで進学させてくれた両親に深くお礼申し上げます。この三年間の経験を生かし、これからも頑張っていります。皆様、本当にありがとうございました。

参考文献

- [1] G. W. Clark, G. P. Garmire, and W. L. Kraushaar, “Observation of High-Energy Cosmic Gamma Rays” , The Astrophysical Journal, 153, L203, 1968.
- [2] 井上一・他『シリーズ現代の天文学第 17 巻宇宙の観測 III-高エネルギー天文学』, 日本評論社, 2008.
- [3] R. W. Klebesadel, I. B. Strong, and R. A. Olson, “Observations of Gamma-Ray Bursts of Cosmic Origin” , The Astrophysical Journal, 182, L85, 1973.
- [4] S. Abdollahi, et al., 247, 33, 2020, URL: <https://doi.org/10.3847/1538-4365/ab6bcb>, DOI: <http://dx.doi.org/10.3847/1538-4365/ab6bcb>.
- [5] F. Acero and et al. Fermi Large Area Telescope Third Source Catalog. The Astrophysical Journal Supplement Series, 218:23, June 2015.
- [6] <https://www.mpi-hd.mpg.de/hfm/HESS/>.
- [7] <https://magic.mpp.mpg.de/>.
- [8] <https://veritas.sao.arizona.edu/>.
- [9] cta-japan, 公式 HP, <http://www.cta-observatory.jp/overview.html>
- [10] P. Kumar and B. Zhang. The physics of gamma-ray bursts relativistic jets. Physics Reports, 561:1–109, February 2015.
- [11] A. Cucchiara, et al., “A PHOTOMETRIC REDSHIFT OF $z \approx 9.4$ FOR GRB 090429B” , The Astrophysical Journal, 736, 7, 2011.
- [12] 小山勝二・他『シリーズ現代の天文学第 8 巻ブラックホールと高エネルギー現象-高エネルギー天文学』, 日本評論社, 2007.
- [13] <http://gammaray.msfc.nasa.gov/batse/>
- [14] M. Ackermann et al., Science 343, 42 (2014)
- [15] 野田浩司, 高橋光成, 深見哲志「ついに捉えた宇宙ガンマ線バーストからの TeV ガンマ線放射」日本物理学会誌 Vol.76, No.5, 2021
- [16] A. M. Hillas, “Cerenkov Light Images of EAS Produced by Primary Gamma Rays and by Nuclei” , in 19th International Cosmic Ray Conference (ICRC19) 3 of International Cosmic Ray Conference, 445, 1985.
- [17] F. Aharonian, et al., “Observations of the Crab nebula with HESS” , A&A, 457, 899-915, 2006.
- [18] 野崎誠也「次世代ガンマ線天文台 CTA, 大口径望遠鏡用 GHz, 波形サンプリング回路の性能評価」, 修士論文, 京都大学大学院理学研究科物理学・宇宙物理学専攻, 2017.
- [19] A. Okumura, T. V. Dang, S. Ono, S. Tanaka, M. Hayashida, J. Hinton, H. Katagiri, K. Noda, M. Teshima, T. Yamamoto, and T. Yoshida. Prototyping hexagonal light concentrators using high-reflectance specular lms for the Large-Sized Telescopes of the Cherenkov Telescope Array. Journal of

- Instrumentation, Vol. 12, No. 12, 2017.
- [20] 浜松ホトニクス株式会社編集委員会. 光電子増倍管 -その基礎と応用-. 浜松ホトニクス株式会社, 第 4 版, 2017.
 - [21] S. Ritt, “Development of high speed waveform sampling ASICs”, in National Symposium on Nuclear Instrumentation, 2010.
 - [22] S. Ritt, R. Dinapoli, and U. Hartmann, “Application of the DRS chip for fast waveform digitizing”, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 623, 486 - 488, 2010
 - [23] J. Sitarek, et al., “Analysis techniques and performance of the Domino Ring Sampler version 4 based readout for the MAGIC telescopes”, Nuclear Instruments and Methods in Physics Research A, 723, 109 - 120, 2013
 - [24] D. Stricker-Shaver, S. Ritt, and B. J. Pichler, “Novel Calibration Method for Switched Capacitor Arrays Enables Time Measurements With Sub-Picosecond Resolution”, IEEE Transactions on Nuclear Science, 61, 3607-3617, 2014.
 - [25] “DRS4 data sheet”, https://www.psi.ch/sites/default/files/2020-08/DRS4_rev09_2.pdf.
 - [26] 佐々木美佳「次世代ガンマ線天文台 CTA のカメラに用いるアナログメモリ DRS4, チップの特性評価」, 修士論文, 茨城大学大学院理工学研究科理学専攻, 2011.
 - [27] 谷川俊介「次世代ガンマ線天文台 CTA, 大口径望遠鏡初号機搭載用 GHz, 波形サンプリング回路の性能評価」, 修士論文, 京都大学大学院理学研究科物理学・宇宙物理学専攻, 2016
 - [28] 小林志鳳「CTA 大口径望遠鏡初号機の高精度カメラ較正」 修士論文, 東京大学大学院理学系研究科物理学専攻, 2019 年
 - [29] 小林志鳳, et al. 「Camera Calibration of the CTA-LST protot」, 2021
 - [30] 野上優人「次世代ガンマ線天文台 CTA 大口径望遠鏡の波形記録チップ DRS4 におけるサンプリング時間幅の較正」 修士論文, 茨城大学大学院理工学研究科理学専攻, 2021 年