

令和 2 年度 修士論文

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

茨城大学大学院理工学研究科理学専攻  
宇宙物理学コース

学籍番号 19NM156N

野上 優人

2021 年 3 月 31 日

## 概要

次世代ガンマ線天文台 Cherenkov Telescope Array (CTA) 計画は、口径の異なる 3 種類の解像型大気チェレンコフ望遠鏡を南北のサイトに合計で約 100 台設置することで 20 GeV から 300 TeV のエネルギーのガンマ線を観測することを目的とする観測計画である。CTA の中で 20 GeV から 3 TeV の低エネルギー側に感度をもつ口径 23 m の大口径望遠鏡 (Large Sized Telescope, LST) は、光電子増倍管 (Photo Multiplier Tube, PMT) と 7 つと波形読み出し回路からなる光検出器 265 本で構成しているカメラを持つ。2018 年 10 月にスペイン領カナリア諸島ラパルマ島にて LST 初号機 (LST1) が完成し、現在は試験観測を行っている。また、LST2-4 号機 (LST2-4) は、LST1 と同じ場所に建設予定であり準備が進められている。

LST カメラに用いている波形読み出し回路には、数 ns に 1 回程度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に GHz での高速波形サンプリングを可能とする波形記録チップ Domino Ring Sampler version 4 (DRS4) を搭載している。DRS4 は、1 入力チャンネルあたり 1024 個のキャパシタアレイを 8 チャンネル有しており、約 1 GHz の周期でスイッチ切り替えを行うことで波形情報をキャパシタに電荷として保存し、波形読み出しは約 30 MHz ほどの低速で行われる。DRS4 では、スイッチを切り替える間隔 (サンプリング時間幅) が不均一であることが知られており、この特性を補正することでピクセル毎の電荷分解能を良くすることができ、LST のエネルギー閾値や宇宙線・ガンマ線イベントの識別などに効果があると考えられる。LST の検出器について、電荷分解能が 1000 光電子以上で 5% 以下という目標性能を設定しているため、較正によって DRS4 のサンプリング時間幅による寄与を 5% 以下に必要がある。LST1 において検出器を取り外すことなくこの特性を補正するために、読み出し回路の前段にあるスローコントロールボード (Slow Control Board, SCB) に搭載されたテストパルス生成回路から読み出し回路に約 100 万発のテストパルスを入力し、DRS4 の各キャパシタでパルスのピークを捉えた回数で重み係数を求めることで補正する方法が考案された。また、これから建設する LST2-4 の波形読み出し回路では、LST1 のものからいくつか変更点があり、そのうちのひとつがサンプリング時間幅の較正用に搭載されたサイン波生成回路である。これにより、前述したテストパルスを用いた較正に比べて精度の良い較正ができるようになった。回路上で生成した周波数 30 MHz のサイン波を測定し得られた波形に対しサイン関数でフィッティングを行い、フィット関数と測定点との時間差を測定することでサンプリング時間幅の揺らぎを較正する。そこで本研究では、LST1 の読み出し回路と LST2-4 の読み出し回路それぞれにおけるサンプリング時間幅の較正をそれぞれ検証した。

1 つ目の検証では、テストパルスを用いた較正方法によって LST1 カメラの全ピクセルでの電荷分解能が改善するかを行った。始めに 2019 年 12 月と 2020 年 6 月に取得された約 200 万イベントを用いて重み係数を求め、DRS4 のサンプリング時間幅が不均一であることを確かめた。前述で用いた 2019 年 12 月の測定データに対して、先ほど得た重み係数を使った補正をする前後でのカメラ平均の電荷分解能を求め比較したところ、高ゲイン系統 (High gain, HG) で  $7.3 \pm 0.7\%$  から  $3.7 \pm 0.5\%$ 、低ゲイン系統 (Low gain, LG) では  $5.4 \pm 0.6\%$  から  $2.9 \pm 0.3\%$  まで改善し、サンプリング時間幅の寄与を 5% 以下まで較正できることを確認した。また、サンプリング時間幅の時間変化を調査するために、2019 年 5 月、同年 12 月、2020 年 6 月に取得した時間幅補正用のデータから重み係数を計算し、それぞれ元のデータに対して補正したときの各電荷分解能のカメラ平均を比較した結果、年間の変動率は HG で約 1.9%、LG で約 4% であることが確かめられた。2019 年 12 月と 2020 年 6 月のデータから計算した重み係数が異なる時期のデータでも用いて問題ないかの検証として、2019 年 5 月から 2020 年 9 月の間に取得されたテストパルスデータに対してサンプリング時間幅補正をしたときの電荷分解能のカメラ平均を比較し、年間の変動率は HG で  $3.52 \pm 0.03\%$ 、LG で  $3.02 \pm 0.02\%$  という結果が得られた。これらの結果より、サンプリング時間幅の時間変化によって約 10 年までは電荷分解能が 5% 以下であることが確認できた。

2 つ目の検証では、LST2 号機以降で用いる読み出し回路でサイン波の生成および測定を行い前述の方法でサンプリング時間幅の較正ができているかを確認するため、始めに 1 万イベント分測定したサイン波の周期の平均を算出

したところ、33.32 ns と理想値である 33.33 ns から 1 ns 短い値となったが標準偏差が 0.04 ns であり数十 ps オーダーの精度で較正できていることを確認した。次に、テストパルスを用いた較正とサイン波を用いた較正の精度を、電荷分解能と時間分解能の観点で比較を行った。その結果、電荷分解能ではテストパルスを用いた較正では HG と LG でそれぞれ 2.4% と 2.7%、サイン波を用いた較正では HG と LG でそれぞれ 1.9% と 2.7% となり、本研究の目標である 5% 以下に電荷分解能が抑えられ、テストパルスを用いた方法よりも較正に用いるイベント数を削減しつつ同等以上の精度で較正できていることを確認した。また時間分解能について、テストパルスを用いた較正では 0.3%、サイン波を用いた較正では 0.07% となり、サイン波を用いた方が較正精度が良いことを確認した。

## Abstract

The Cherenkov Telescope Array (CTA), the next generation gamma-ray observatory, will detect the gamma rays with energies from 20 GeV to 300 TeV to set three kinds of telescopes with different sizes of reflectors on the north and south site. The Large-Sized Telescope (LST), is the largest telescope in the CTA and sensitives to the gamma-ray energies from 20 GeV to 3 TeV, has the camera consisted of 265 photo detector modules that consists 7 photomultiplier tubes (PMTs) and one read-out board. The prototype LST (LST1) built on La Palma Island in Canary Islands in October 2018 and is in the commissioning phase. The second to forth LST (LST2-4) is preparing for the next construction on La Palma Islands.

The readout boards of LSTs has the Domino Ring Sampler version 4 (DRS4), the analog memory can sample the waveforms in GHz, due to remove the incidents of the night sky background in once every several nanoseconds. DRS4 has 8 input channels with 1024 capacitors per channel, and by switching at a cycle of several GHz the waveforms are saved the electric charges and they are read out with low speed at 30 MHz. DRS4 has the non-uniform characteristic of the sampling interval, by correcting this characteristic the electric charge resolution of each camera pixel improves and this correction is effective for lower energy threshold of LST and the identification of cosmic-ray and gamma-ray events. LST is required to reduce the systematic error due to the read-out board to less than 2%. Therefore we should correct the non-uniform of sampling interval of DRS4.



# 目次

第 1 章	序論	1
第 2 章	ガンマ線天文学	3
2.1	ガンマ線天文学	3
2.2	ガンマ線天体	3
第 3 章	Cherenkov Telescope Array (CTA) 計画	9
3.1	空気シャワー現象	9
3.2	解像型大気チェレンコフ望遠鏡	10
3.3	Cherenkov Telescope Array (CTA) 計画	11
3.4	大口径望遠鏡 (LST)	12
3.5	LST 波形読み出し回路	13
第 4 章	大口径望遠鏡初号機 (LST1) における波形サンプリング時間幅の較正	21
4.1	目標	21
4.2	サンプリング時間幅の較正方法	22
4.3	解析環境およびツール	22
4.4	LST 初号機テストデータを用いた時間幅較正の結果	23
4.5	議論	29
第 5 章	大口径望遠鏡 2 から 4 号機 (LST2-4) における波形サンプリング時間幅の較正	31
5.1	目標	31
5.2	サンプリング時間幅の較正方法	31
5.3	測定・解析環境	33
5.4	実験室でのサンプリング時間幅較正の結果	35
5.5	議論	39
第 6 章	結論	43
付録		45
A	テストパルスを使ったサンプリング時間幅較正のスク립ト	45
謝辞		53



## 第 1 章

# 序論

宇宙では GeV ( $= 10^9$  eV) 領域を超えるような超高エネルギーのガンマ線が飛び交い、その一部は地球に飛来する。超高エネルギーのガンマ線は、同等あるいはそれ以上のエネルギーを持った宇宙線陽子やガンマ線バーストのような短時間に膨大なエネルギーを電磁波の放射によって解放する爆発的な現象から生じていると考えられている。しかし、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宇宙で生じている現象は現在も謎に包まれたままである。これらを、解明する鍵の一つが地球に飛来する超高エネルギーガンマ線である。ガンマ線は宇宙空間の磁場によって軌道が曲げられることがないため到来方向の情報が失われにくいことから、どのような天体ないし場所でガンマ線が放射されたかを知ることができる。また、検出したガンマ線のエネルギーが分かればどのような過程で放射されたかを推定することもできる。

超高エネルギーガンマ線は、地球大気に入射すると大気中の原子核との相互作用により空気シャワーを生じる。シャワーで生じた荷電粒子が大気中の光速を超える速度で走ることによってチェレンコフ光と呼ばれる可視光を放射し、ガンマ線の入射軸から約 1 度の広がりを持って地上に降り注ぐ。このシャワーによって生じたチェレンコフ光を集光することで間接的にガンマ線を検出するのが大気チェレンコフ望遠鏡である。超高エネルギーガンマ線は到来頻度が少ないことに加え、地球に飛来する超高エネルギー宇宙線のほとんどを占める陽子もまた大気中の原子核との相互作用により空気シャワーを生じチェレンコフ光を放射する。そのため、ガンマ線観測をするうえでバックグラウンドとなる宇宙線陽子由来のイベントを除去するのが重要となる。宇宙線陽子とガンマ線が形成する空気シャワーの形状は異なることが知られており、この違いを利用することで宇宙線陽子とガンマ線の識別が可能となる。現行の大気チェレンコフ望遠鏡は、焦点面に設置した複数の光検出器から成るカメラでチェレンコフ光を反射鏡で集光し、カメラ面上での分布および光子数を記録しており、このような望遠鏡は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。現行の IACT では、複数の望遠鏡を並べて同時に観測することで検出有効面積を稼ぎつつガンマ線の到来方向を一意に推定できるステレオ観測が主流となっている。

Cherenkov Telescope Array (CTA) 計画は、口径の異なる 3 種類の IACT を南北に合計約 100 台を設置することにより、20 GeV から 300 TeV と広いエネルギー領域のガンマ線を従来の望遠鏡の 10 倍の感度で検出することにより次世代のガンマ線天文台となることを目指す国際計画である。口径 23 m と CTA のなかで最大の IACT である大口径望遠鏡 (Large-Sized Telescope, LST) は、20 GeV から 3 TeV と CTA の低エネルギー領域のガンマ線に感度を持ち、南北に 4 台ずつ設置する予定である。LST に用いているカメラは、光電子増倍管 (Photo Multiplier Tube, PMT) 7 本と波形読み出し回路 1 枚からなる光検出器モジュール 265 組で構成している。LST では、低エネルギー領域のガンマ線に由来するチェレンコフ光を観測するため、数百 MHz の頻度でランダムに混入する星の明かりなどに由来する夜光バックグラウンドの影響が無視できない。この夜光バックグラウンドを除去するには、GHz での高速波形サンプリングが可能な波形読み出し回路が要求される。そこで LST では、1 GHz で波形サンプリングができる Domino Ring Sampler version 4 (DRS4) と呼ばれるアナログメモリチップを搭載した波形読み

出し回路を開発し用いている。2018 年 10 月には、CTA の北サイトであるスペイン カナリア諸島 ラ・パルマ島に LST 初号機 (LST1) の建設が完了し、現在は安定観測中である。また、LST2 から 4 号機に関しては検出器モジュールの組み立てと品質管理試験と同時に建設も進められている。

DRS4 チップは、入力チャンネル 1 つあたり 1024 個のキャパシタをもつチャンネルを 9 つ持っており、複数のインバータをつないだ回路でスイッチ切り替えを行い入力チャンネル内のキャパシタに電荷を随時蓄えることで読み出し回路に入力された波形情報保持し、約 30 MHz の低速で波形読み出すことができる。DRS4 では、複数の入力チャンネルを繋げて 1 つの入力チャンネルとすることができチャンネルカスケードと呼ばれ、LST では、4 つのチャンネルをカスケード接続させることで約  $4\mu\text{s}$  のメモリ深さを確保している。また、入力チャンネル内で読み出すキャパシタの範囲を指定することもできこの範囲を Region of Interest (ROI) と呼ぶ。読み出すキャパシタの数はスライスと表現され、LST では ROI を 40 スライスにしている。DRS4 チップにはいくつかの特性をもっていることが知られており、各キャパシタ毎に固有のペDESTAL 波高値を持っていること (ペDESTAL 特性)、一度電荷を蓄えたキャパシタが放電しきる前に再度電荷を蓄えてしまうことにより本来よりも電荷が大きくなってしまうこと (残留電荷特性)、キャパシタに蓄えた電荷がまわりのキャパシタよりも高くなるものが存在すること (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ばらつきが生じること (サンプリング特性) が挙げられる。いずれの特性も、読み出された信号の電荷量を変化させるため望遠鏡で捉えた光子量の推定を誤ってしまう。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 で波形読み出し回路に用いている MAGIC で研究されてきた。そして、CTA でもペDESTAL 特性、残留電荷特性、スパイク特性の補正方法を確立し LST の解析で実際に行っている。

サンプリング特性については、波形の電荷量だけでなく時間情報も歪めてしまうため望遠鏡への入射光量および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。信号の到来時間情報に関しては、MAGIC での補正方法を参考にして LST の解析にすでに取り入れている。信号波形の電荷量の補正についても、補正方法の確立を目指して波形読み出し回路の開発段階の時から日本チームで検証してきた。具体的には、2011 年の佐々木修論 (茨城大学) では読み出し回路に搭載している Field Programmable Gate Array (FPGA) から出力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 LST1 のスローコントロールボード (Slow Control Board, SCB) で生成できるようになったテストパルスを使った方法、2017 年の野崎修論 (京都大学) 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。

本研究では、既に建設が完了し稼働している LST1 および建設が進められている LST2 以降で、望遠鏡の稼働を止めずに DRS4 のサンプリング特性を補正する方法の検証および実装を目指し、LST1 に関しては谷川修論 (京都大学) で行われた SCB で生成できるテストパルスを使った較正方法を用いて LST1 カメラの全 1855 ピクセルに対して較正を行った。また LST2 号機以降については、野崎修論 (京都大学) で行った 2 号機以降用の読み出し回路で生成できるサイン波を用いた補正および LST1 で検証したテストパルスを用いた補正を行い、電荷分解能と時間分解能の観点で補正精度の比較を行った。これらの検証結果について本論文で述べていく。

## 第 2 章

# ガンマ線天文学

## 2.1 ガンマ線天文学

宇宙からのガンマ線を確実に検出したのは、1967 年に打ち上げられた **Orbiting Solar Observatory 3** 衛星が初めてである [1] [2]。この検出のあと、いくつもの衛星が打ち上げられたことにより次々とガンマ線天体が発見されていった。1973 年には地上核実験の監視のため打ち上げられた **Vela** 衛星の観測データから、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していたことが報告された [3]。ガンマ線天文学が飛躍的に発展していったのは、1989 年に打ち上げられ 100 keV から 1 MeV のガンマ線を観測した **Granat** 衛星や 1991 年に打ち上げられ数十 keV から 10 GeV のガンマ線を観測したコンプトンガンマ線衛星の登場以降のことである。そして 2008 年には、20 MeV から 300 GeV のガンマ線に感度をもつフェルミガンマ線宇宙望遠鏡（フェルミ衛星）が打ち上げられたことにより 5000 以上の天体を検出している [4]。しかし、エネルギーが数十 GeV 以上のガンマ線になると飛来する数が減ってくるため大きな検出面を有する検出器が要求されるが、検出器に伴って人工衛星自体も大きくする必要があり費用の増加などの理由から従来の検出器よりもさらに大きいものを新たに搭載した人工衛星を打ち上げるのは困難である。そこでエネルギーが数十 GeV 以上のガンマ線を観測する際は、宇宙より飛来するガンマ線と地球大気原子核との相互作用によって生成される電子から起こ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地上に配置された大気チェレンコフ望遠鏡が用いられる。宇宙より飛来する陽子を始めとした原子核（宇宙線）もまた大気原子核と相互作用することでガンマ線と同様に空気シャワーを形成する。そのため、宇宙線とガンマ線の識別が大気チェレンコフ望遠鏡においても重要となる。宇宙線とガンマ線それぞれから形成されるシャワー形状の違いを使った識別方法の有用性は、1989 年の **Whipple** 望遠鏡によるかに星雲の観測 [5] によって示された。その後は、ステレオ観測技術や高感度の光検出器、超高速電子回路などを用いている現行の大気チェレンコフ望遠鏡である **H.E.S.S.** [6]、**MAGIC** [7]、**VERITAS** [8] がガンマ線天文学の発展に貢献をしている。

## 2.2 ガンマ線天体

ここでは、ガンマ線天文学で観測対象となる天体の中でガンマ線バーストと活動銀河核について紹介する。

### 2.2.1 ガンマ線バースト

ガンマ線バースト（**Gamma Ray Burst, GRB**）とは、光速の 99.9999% もの速さで放出されるジェット流に伴って数秒から数十秒の間の短時間に  $10^{53}$  erg と非常に大きなエネルギーをガンマ線をはじめとした様々な波長の電磁波の放射によって解放する宇宙で最大の爆発現象である [9]。GRB 放射のイメージを図 2.1 に示す。典型的には地球から数十億光年と宇宙論的距離で GRB が発生することが分かっており、**Swift** 衛星の観測では地球から約 132

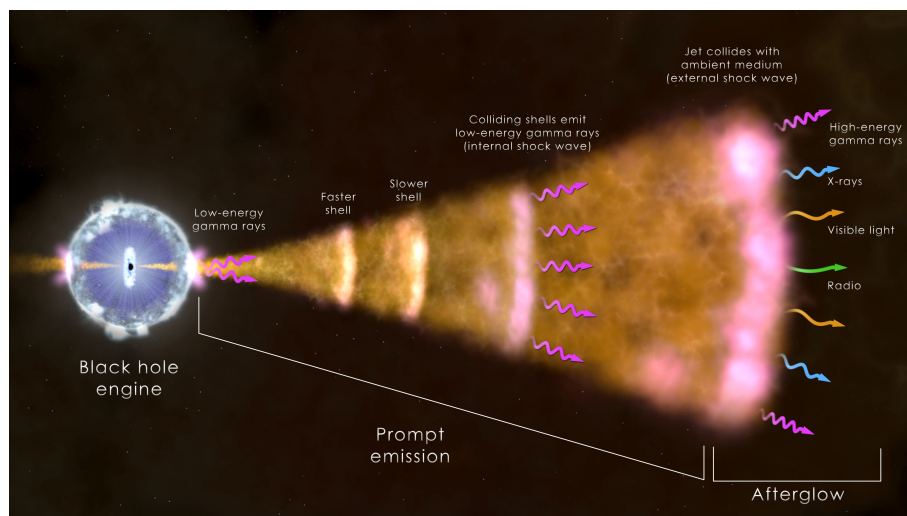


図 2.1 GRB のイメージ図 [13]。即時放射では MeV 領域のガンマ線が放射され、残光放射では電波からガンマ線と様々な波長の電磁波が放射される。

億光年の距離で発生するものが発見されている [10]。GRB には放射継続時間が 2 秒以下の短い種族と 2 秒以上の長い種族が存在しており、放射を起こす中心駆動天体の有力候補として短い種族では中性子星やブラックホールが、長い種族では大質量星の重力崩壊が挙げられているものの正体は未だ不明である [11]。GRB の放射には種族に関わらず、ミリ秒以下の不規則な時間変動で放射する即時放射および数日にわたって徐々に減光していく残光放射がある。即時放射では主に MeV 領域と少し遅れて GeV 領域のガンマ線が、残光放射では電波から GeV 領域のガンマ線までと広い波長域の電磁波が放射される。フェルミ衛星と Swift 衛星で観測された GRB の光度曲線を示す図 2.2 を見ると、数十 keV から 10 MeV のガンマ線に比べて 100 MeV 以上のものは数秒遅れて放射されていることがわかる [12]。GRB の MeV 領域のガンマ線に関しては、人工衛星が主として観測できているが GeV 領域のガンマ線は検出有効面積の制限から困難となるため、広い検出有効面積により数十 GeV 以上のガンマ線への感度を有することが可能な大気チェレンコフ望遠鏡との連携が必要となる。大気チェレンコフ望遠鏡における GRB の観測は、フェルミ衛星や Swift 衛星などからのアラートに応じて追尾観測を行うことが主となる。Cherenkov Telescope Array (CTA) の大口径望遠鏡では、人工衛星からのアラートを受けてから 20 秒で 180 度という速さで回転させることが可能なため、放射継続時間の長い GRB であれば即時放射を途中から観測でき、20 GeV 以上のガンマ線に感度を持つことから GeV 領域の統計数を稼ぐことにより GRB の放射機構などの解明に寄与できると考えられる。

## 2.2.2 活動銀河核

活動銀河核 (Active Galactic Nuclei, AGN) とは、銀河中心にあるブラックホールへガスが降着する際に解放される重力エネルギーによって輝く現象であると考えられている。図 2.3 に現在考えられている AGN の構造を示す。活動銀河核の約 1 割は電波銀河と呼ばれる非熱的放射が優勢の天体であり、相対論的なジェット流が観測されている。特にこのジェット流が観測者に対して正面を向き、相対論的ビーミングにより非常に明るく見える天体はブレイザーと呼ばれ、電波からガンマ線と広い波長域で観測されている。ブレイザーでは、数分から数日の間にガンマ線の放射強度が大きく変動することが知られており、図 2.4 は H.E.S.S が 2006 年 7 月に観測したブレイザー PKS 2155–304 のガンマ線放射強度の時間変動を示している [14]。また、ブレイザーの 1 つである Markarian 421 (Mrk 421) の多波長観測によって得られたエネルギースペクトルを図 2.5 に示す。これまでに観測

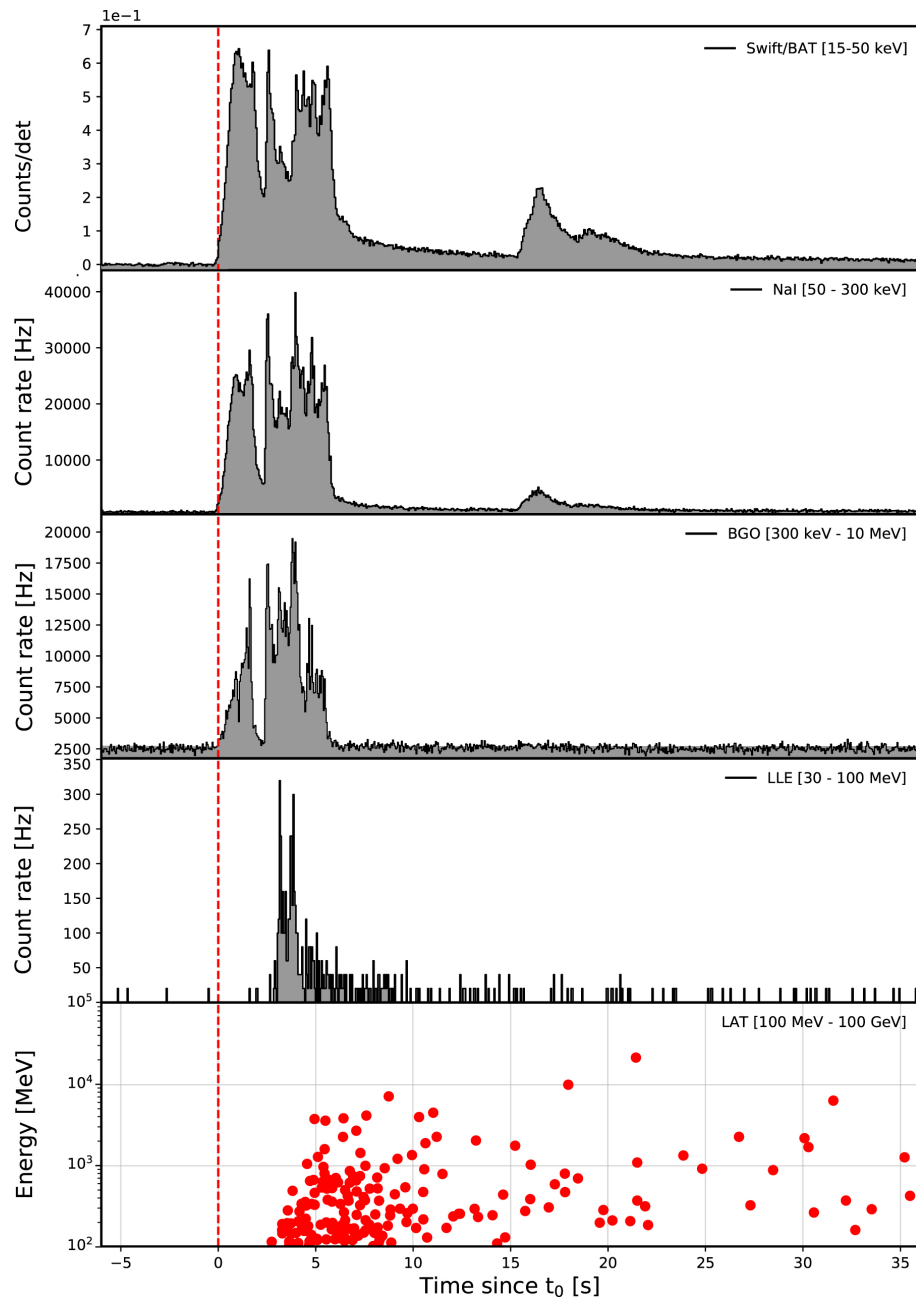


図 2.2 swift 衛星フェルミ衛星で観測した GRB 190114C の光度曲線 [12]。Swift BAT およびフェルミ衛星に搭載している各検出器での光度曲線となっていて、 $t_0$  はフェルミ衛星の GRB Monitor (GBM) が GRB を検出した時間に対応している。

されているブレーザーの多くは図 2.5 のように 2 つのピークを持っており、低エネルギー側のピークは電子のシンクロトロン放射、高エネルギー側のピークは高エネルギーの電子による逆コンプトン散乱に対応すると考えられている [11]。

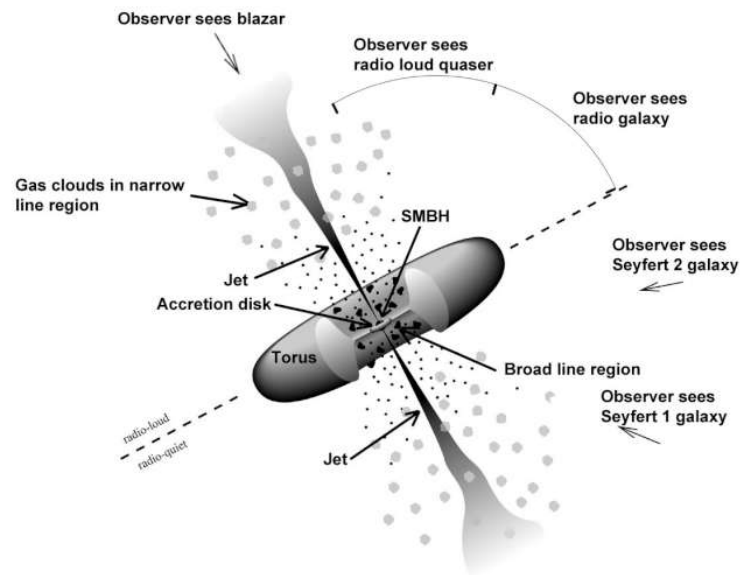


図 2.3 現在推定されている AGN の模式図 [15]。

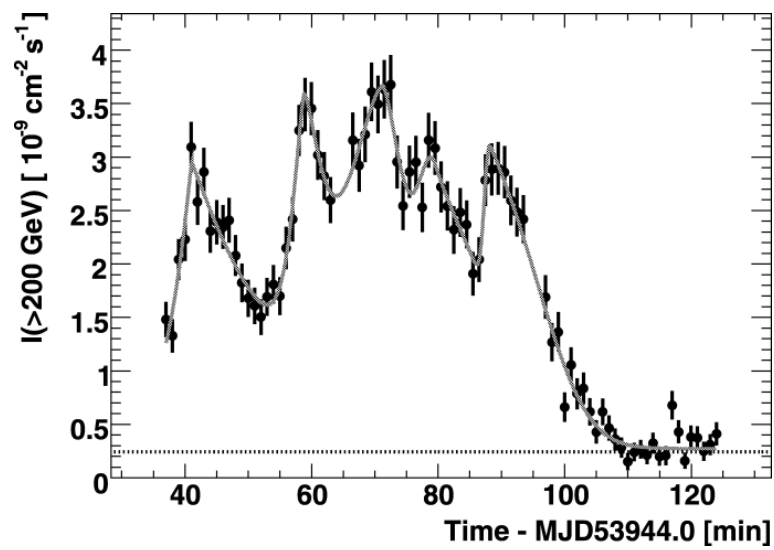


図 2.4 H.E.S.S. で捉えた BL Lac PKS 2155-304 のガンマ線放射強度の時間変動の様子 [14]。



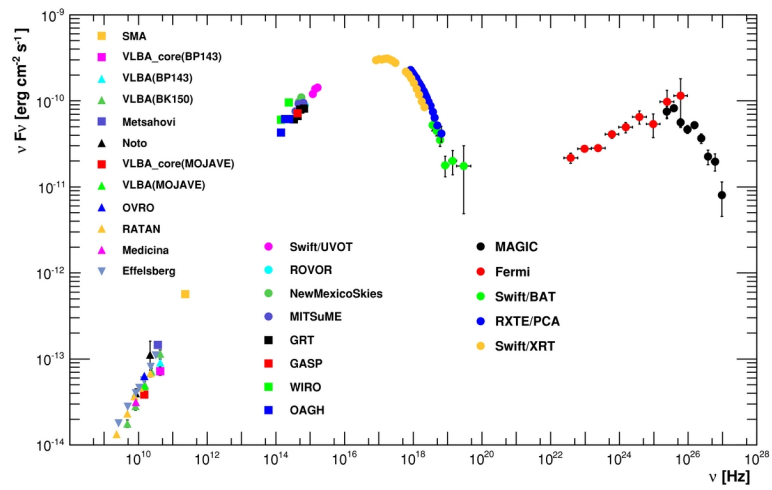


図 2.5 Mrk 421 の多波長観測で得られたエネルギースペクトル [16]。



## 第 3 章

# Cherenkov Telescope Array (CTA) 計画

この章では、はじめに超高エネルギーガンマ線が大気に入射した際に生じる空気シャワー現象および空気シャワーの観測によって間接的にガンマ線の観測を行う解像型大気チェレンコフ望遠鏡 (Imaging Atmospheric Cherenkov Telescope, IACT) について述べる。そして、複数台の IACT を用いる次世代ガンマ線天文台である Cherenkov Telescope Array (CTA) および CTA の中で最大である大口径望遠鏡 (Large-sized Telescope, LST)、LST の焦点面カメラに用いている波形読み出し回路について述べる。

### 3.1 空気シャワー現象

地球大気には宇宙より飛来する高エネルギーの粒子 (1 次粒子) が突入し、大気中の原子核との相互作用から多数の 2 次粒子が生じる。1 次粒子が超高エネルギーのガンマ線である場合は、対生成によって電子と陽電子が生成される [17]。

$$\gamma + \text{原子核} \rightarrow e^- + e^+ + \text{原子核}' \quad (3.1)$$

対生成で生じた電子や陽電子は、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対生成が生じる。このように対生成と制動放射の繰り返しによって荷電粒子の数が増加していく現象を空気シャワー、特に 1 次粒子がガンマ線の時は電磁シャワーという。空気シャワーによって生成された荷電粒子のエネルギーが制動放射の臨界エネルギーである約 84 MeV まで低下すると、制動放射によるガンマ線の放射がなくなることに対生成が生じず、代わりに電離損失が卓越し荷電粒子がエネルギーを失うことで大気に吸収され荷電粒子の数は減少に転じていく [2]。

一次粒子が宇宙線陽子の場合は、大気中の原子核との相互作用によって陽子や中性子、 $\pi^0$ 、 $\pi^\pm$ 、 $k^\pm$  が生成される。

$$p + \text{原子核} \rightarrow p' + \text{原子核}' + \pi^0, \pi^\pm \quad (3.2)$$

$\pi^0$  の寿命は  $0.84 \times 10^{-16}$  秒であり、崩壊するとガンマ線を放射し電磁カスケードシャワーを生じる。

$$\pi^0 \rightarrow \gamma + \gamma \quad (3.3)$$

$\pi^\pm$  の場合は、寿命である  $2.6 \times 10^{-8}$  秒を迎えると  $\mu$  粒子とニュートリノへ崩壊し、生じた  $\mu$  粒子は寿命  $2.2 \times 10^{-6}$  秒で電子とニュートリノに崩壊したのち、電子が制動放射によって放射されるガンマ線より電磁カスケードシャワーが発生する。

$$\pi^+ \rightarrow \mu^+ + \nu_\mu, \pi^- \rightarrow \mu^- + \bar{\nu}_\mu \quad (3.4)$$

$$\mu^+ \rightarrow e^+ + \nu_e + \bar{\nu}_\mu, \mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu \quad (3.5)$$

このように宇宙線陽子由来の空気シャワーの発生過程は、超高エネルギーガンマ線のときと比べて少し異なり、電磁シャワーと区別してハドロンシャワーとも呼ばれる。ハドロンシャワーは、図 3.1 のように電磁シャワーに比べ

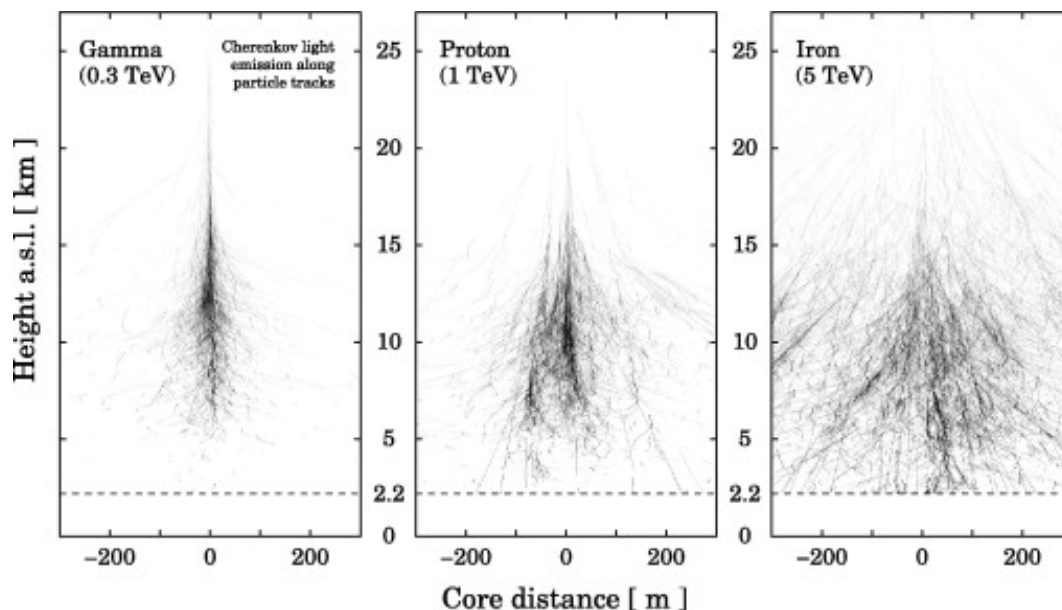


図 3.1 大気に入射する一次粒子を左からガンマ線、陽子、鉄原子核としたときの空気シャワーの発達の様子をシミュレーションした図 [18]。

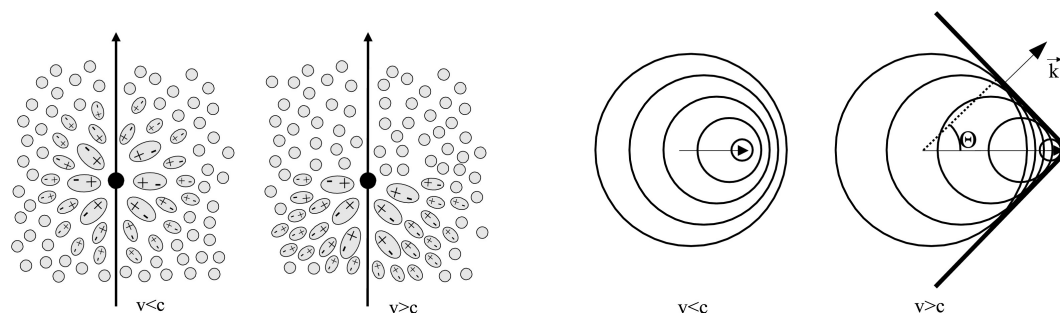


図 3.2 大気中を運動する荷電粒子およびチェレンコフ光の放射のイメージ [19]。左側 2 つの図は大気中を運動する荷電粒子の電場によって周囲の原子分子が分極を起こす様子を示している。右側 2 つは分極で生じた電磁波の伝搬を示している。

て形状が横方向に広がる。空気シャワーで生じた荷電粒子が大気中の光速を超える速度で走ると荷電粒子の後方で原子・分子で分極が生じ、粒子が通過した後に平衡状態へ戻ろうとする際に電磁波が放射される。この電磁波が作る放射面はよく音の衝撃波面に例えられ、放射面で原子・分子の分極で生じた電磁波が強め合うことでチェレンコフ光と呼ばれる可視領域の光子を荷電粒子の進行方向に放射する。このチェレンコフ光の放射角  $\theta_c$  は、大気中での荷電粒子の速度  $v$  と真空中での光速  $c$  及び大気の屈折率  $n$  から  $\cos \theta_c = \frac{c}{vn}$  で与えられる。1 TeV のガンマ線から生じる電磁カスケードシャワーで荷電粒子の数が最大となる高度（最大発達高度）はおよそ 8 km であり [2]、対応する大気の屈折率は地上より小さくなり 1 と近似すると、放射角は  $\theta_c \sim 1$  度となり、地上では半径約 140 m の円の範囲にチェレンコフ光が降り注ぐことになる。

### 3.2 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) は、前節で述べた空気シャワーによって生成された荷電粒子より放射されるチェレンコフ光を反射鏡で焦点面にある検出部へ集光し、検出部

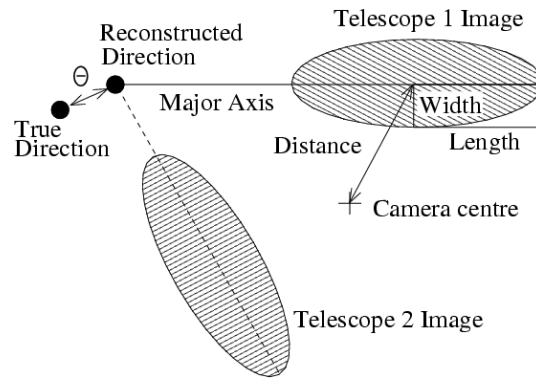


図 3.3 hillas パラメーター及びステレオ観測のイメージ [21]。ガンマ線の到来方向は、各望遠鏡で撮像したシャワーイメージのシャワー軸の交点と推定できる。



図 3.4 CTA の北サイトの完成予想図。[21]

でのイメージを記録することにより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、1 次粒子のほとんどが宇宙線であるためガンマ線観測をする上ではバックグラウンドとなり宇宙線とガンマ線を識別することが必要となる。そこで IACT では、前節で述べたガンマ線と宇宙線が作る空気シャワーの形状の違いを利用しており、検出部で捉えた空気シャワーのイメージは楕円形となるため、シャワー形状を楕円近似することで得られる width や length といった楕円の形状を特徴づけるパラメータを用いて識別をすることが可能であり、この手法は A. M. Hillas によって考案されたため Hillas パラメーターと呼ばれており [20]、図 3.3 に概念図を示す。IACT での観測では、検出部で捉えたシャワーイメージで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それぞれ推定している。また、図 3.3 のように望遠鏡を複数台用いたステレオ観測をすることで、複数のシャワーイメージが取得できそれらの軸の交点からガンマ線の到来方向を推定ができるため、1 台の望遠鏡の時に比べて到来方向の推定を精度良く行うことができる。

### 3.3 Cherenkov Telescope Array (CTA) 計画

Cherenkov Telescope Array (CTA) 計画は、20 GeV から 300 TeV と幅広いエネルギー領域のガンマ線の観測を実現するために口径 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) の 3 種類の異なる大きさの望遠鏡を北半球のスペイン領カナリア諸島ラパルマ島と南半球のチリパラナルに合計で約 100 基を配置することにより現行の望遠鏡の 10 倍の感度で観測することを目指す国際計画である。CTA の北サイトの完成予想図を図 3.4 に、各望遠鏡の仕様は表 3.1 に示す [9]。

表 3.1 CTA の各望遠鏡の仕様。

	LST	MST	SST
口径	23 m	12 m	4.3 m
エネルギー範囲	20 GeV - 3 TeV	80 GeV - 50 TeV	1 TeV - 300 TeV
視野	4.5°	8°	9°
望遠鏡数 (北)	4	15	0
望遠鏡数 (南)	4	25	70



図 3.5 完成した LST1 の写真 [22]。

### 3.4 大口径望遠鏡 (LST)

LST は、口径 23 m と CTA では最大の大きさとなる望遠鏡であることから、低エネルギーのガンマ線より発生する空気シャワーによって放射される僅かなチェレンコフ光を集光することが可能であり、低エネルギーである 20 GeV 以上のガンマ線を検出することが期待される。LST の反射鏡や焦点面検出器の支持構造の主要な部分には軽量で頑丈なカーボンファイバー強化チューブを用いており、望遠鏡全体の質量を 100 トンまで軽量化することにより、20 秒で 180 度の高速回転を行いガンマ線バーストのような突発天体の観測に対応する [9]。口径が 23 m ある LST の主鏡は、チェレンコフ光を焦点面カメラへ集光するために 198 枚の球面分割鏡で構成される放物面鏡を実現している。焦点面カメラには、光検出器として 1855 本の光電子増倍管 (Photo Multiplier Tube, PMT) を用いており、PMT7 本で 1 組とした光検出器モジュール 265 台でカメラを構成している。2018 年 10 月には LST 初号機 (LST1) が北サイトであるラ・パルマ島に完成し、完成した LST1 の様子を図 3.5 に示す。また、LST1 と同じ場所に建設する LST2 号機から 4 号機 (LST2-4) については、現在までに光検出器の組み立てを 800 モジュールの組み立てが完了し、うち 657 モジュールまでは品質管理試験を行い問題なく望遠鏡に用いられると判定したのは 589 モジュールであり、LST2 台分の準備は完了している。2021 年度には LST2-4 の建設が始まる予定である。

## 3.5 LST 波形読み出し回路

IACT が空気シャワー由来のチェレンコフ光を捉える際、光検出器より出力される信号の幅は数 ns と非常に短く、加えて実際の観測で得られるデータには星などに由来する夜光が数百 MHz でランダムに混入するため、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となる。そのため、高速で信号を読み出すことができ数 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を使ったアレイ観測を行い、波形読み出しを行うためのトリガー信号は各望遠鏡間でコインシデンスを取ることで生成されるため、チェレンコフ光を捉えてから数  $\mu$ s 後に読み出し回路へ入力される。そのため、数  $\mu$ s のあいだ波形情報を保持できるメモリ深さが要求される。

### 3.5.1 Dragon Board

LST の波形読み出し回路上での信号の流れを概説し、図 3.6 にブロック図を示す。空気シャワーより生じたチェレンコフ光が焦点面カメラに集光されると、検出器モジュール前面にある PMT によって光信号から電気信号へと変換され、プリアンプ (PreAmplifier for the CTA camera, PACTA) で信号を増幅するとともに、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高ゲイン系統 (High Gain, HG) と低ゲイン系統 (Low Gain, LG) の 2 系統に分けたのち差動信号として出力する。検出器モジュール 1 台は PMT7 本から成るので、HG と LG 合わせて 14 チャンネル分の差動信号がスローコントロールボード (Slow Control Board, SCB) を介して 1 枚の波形読み出し回路へ入力される。読み出し回路に入るとメインアンプで各系統の信号が増幅され、特に HG では信号をさらに二つに分けトリガー生成用の信号も生成される。増幅した信号はアナログメモリ素子 Domino Ring Sampler version 4 (DRS4) に入力され、LST では約 1 GHz でサンプリングされる。そしてトリガーがかかると、サンプリングした信号は Analog Digital Converter (ADC) で A/D 変換され、Field Programmable Gate Array (FPGA) を介してギガビットイーサネット通信によって外部の観測データストレージへ転送される。LST では、夜光各カメラピクセルからの入力信号を隣接する複数のピクセル間で足し合わせたのちに閾値にかけることでトリガーを生成するアナログサム方式を採用している [9]。この方式によって、短時間に集中して観測されるチェレンコフ光信号を大きくし、各ピクセルにランダムに入射する夜光でトリガーが生成されるのを抑えることができる。

### 3.5.2 プリアンプ

PMT より出力される電流信号は、PACTA 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および増幅されたのち差動信号として出力される。PACTA では、増幅率の異なるゲインを用意しており、入力された信号は HG では 24 倍、LG では 1.6 倍に増幅するように設計されている。PACTA で増幅された信号が波形読み出し回路へ入力されるとメインアンプでさらに増幅される。これにより LST の読み出し回路では 2000 光電子までのダイナミックレンジを確保することが可能となっている [24]。

### 3.5.3 Domino Ring Sampler version 4 (DRS4)

#### 概要

まずアナログメモリについて述べる。アナログメモリとは複数のキャパシタを並べて、入出力部分にスイッチの役割をするインバータを多数接続したものであり、図 3.7 にアナログメモリの回路の例を示す。入力部分のスイッチを高速で切り替え、各キャパシタに入力信号の電荷を蓄えることで波形情報を保持することができる。メモリ深さは、アナログメモリに用いるキャパシタの数とサンプリング周波数によって決まるため、例えば 1 GHz で波形サ

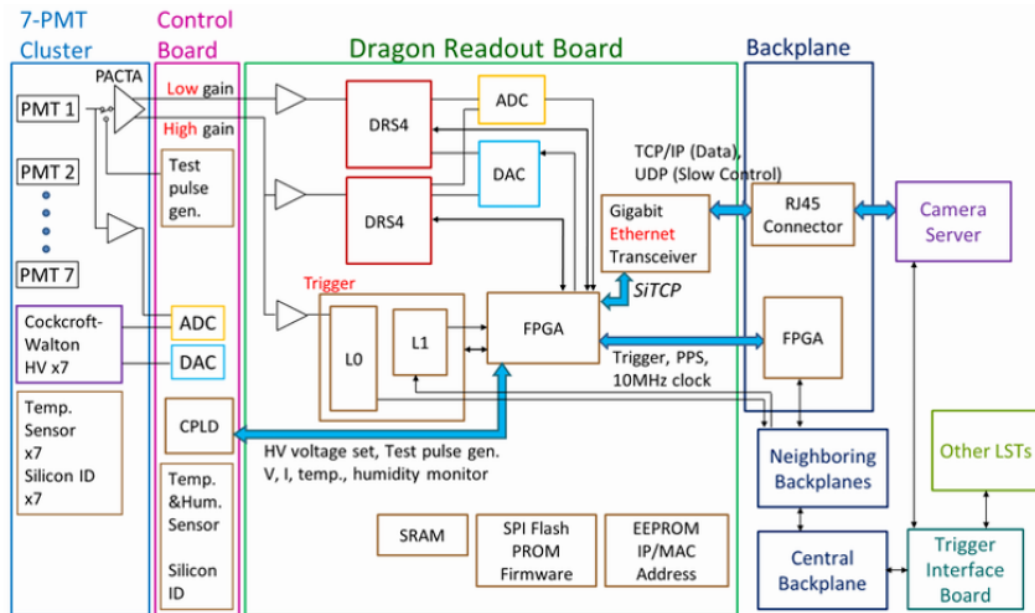


図 3.6 LST1 の波形読み出し回路のブロック図 [23]

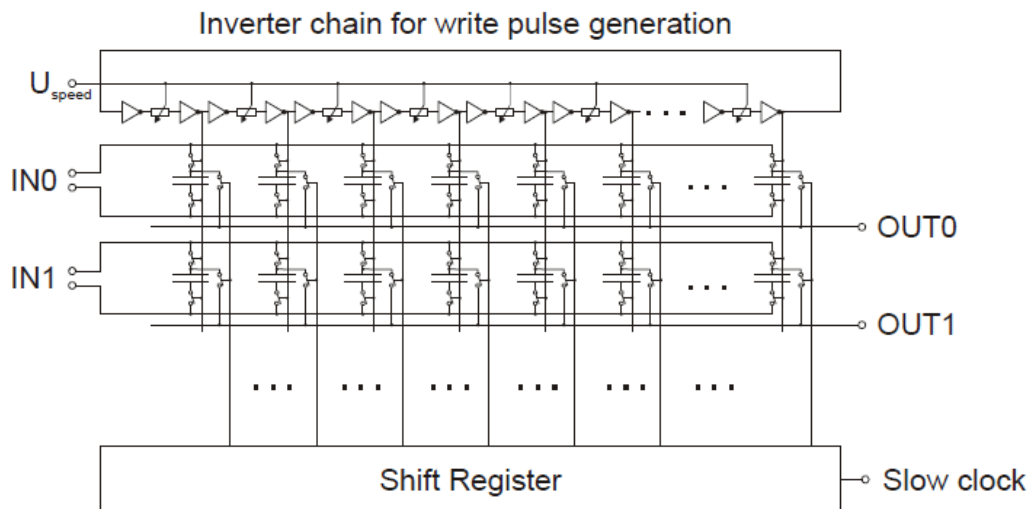


図 3.7 アナログメモリの回路図のイメージ [25]

ンプリングをする場合は数千個のキャパシタを用いることで数  $\mu\text{s}$  のメモリ深さを確保することが可能である。そして、シフトレジスターの切り替えによりキャパシタに蓄えた電荷を読み出すタイミングを制御し、ADC でデジタル信号に変換することで入力信号の波形情報を得ることができる。シフトレジスターは、入力部のスイッチとは独立であるので数十 MHz ほどの低速で動作させることができる。そのため、アナログメモリでは低速での波形読み出しによる低消費電力を実現できる。

LST の波形読み出し回路では、GHz での高速波形サンプリング、数  $\mu\text{s}$  のあいだの波形情報の保持および低消費電力を実現するために、スイスのポール・シェラー研究所がミューオンの電子・光子崩壊の探索を目的として 2008 年より行っている MEG 実験 [26] のために開発されたアナログメモリである DRS4 という ASIC を用いている。DRS4 は現行の IACT の一つである MAGIC 望遠鏡の波形読み出し回路でも用いられている [27]。



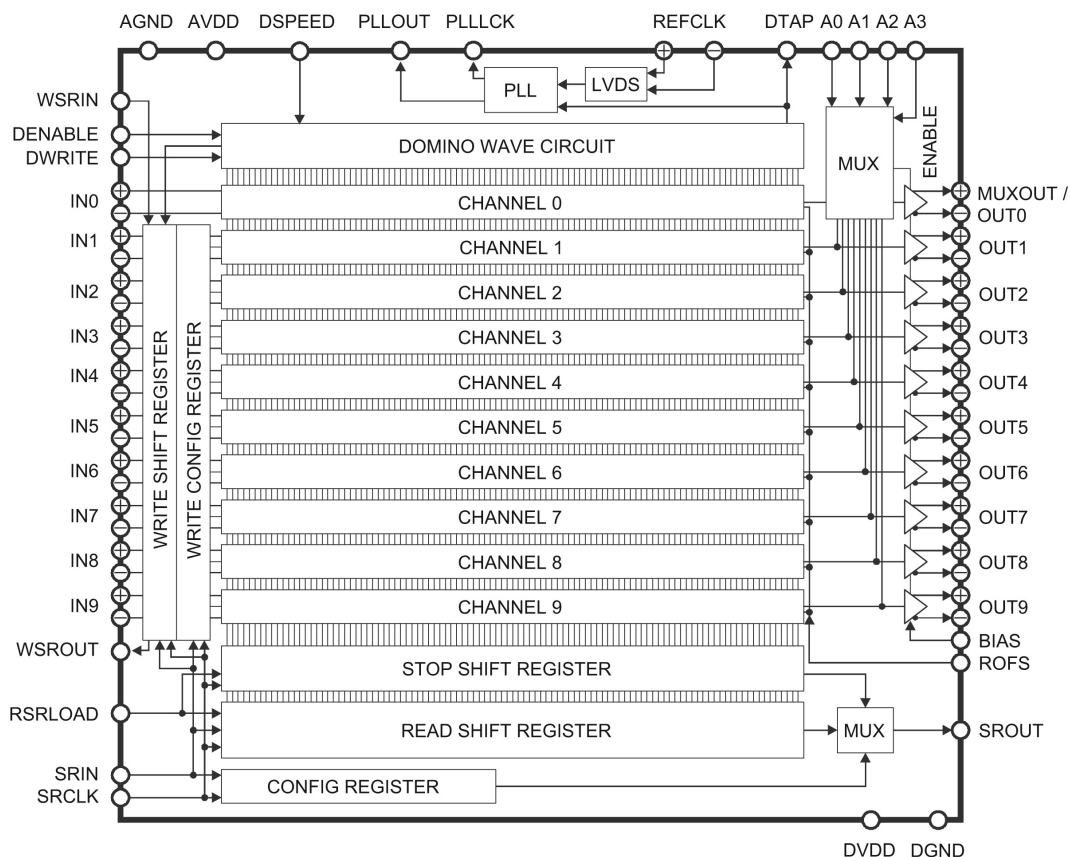


図 3.8 DRS4 チップのブロック図 [29]。

DRS4 は、700 MHz から 5 GHz の範囲でサンプリング周波数を変更することができるアナログメモリであり、9つの差動入力チャンネルを持っている。DRS4 のブロック図は図 3.8 に示す。1 チャンネルあたり 1024 個のキャパシタから成り、8 つのチャンネルは PMT からの入力信号の波形サンプリングに用いられ、残る 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 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る。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) 回路で制御している。DRS4 外部からリファレンスとして周波数  $f_{\text{reference}}$  のクロックを入力すると  $2048 \times f_{\text{reference}}$  Hz で波形のサンプリングが行われる [28]。DRS4 に入力された信号はこうして 1024 個のキャパシタに電荷を保存し、1024 番目のキャパシタまで保存されるとチャンネルの始めのキャパシタの電荷を放電し 1 番目のキャパシタから再び電荷を保存していく。DRS4 の write shift register を書き換えることで、あるチャンネルの 1024 番目のキャパシタまで電荷を保存したあと、同チャンネル内の 1 番目のキャパシタでなく次のチャンネルへ信号を入力・記録させることが可能であり、チャンネルカスケードと呼ばれる。これにより、1 チャンネルあたりのメモリ深さを増やすことができ、LST では各ゲイン系統の PMT1 本あたり 4 チャンネルを用いることで 1 GHz サンプリング時で約  $4\mu\text{s}$  のメモリ深さを確保している。そして、DRS4 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が行われる。波形読み出しを行う際、ファイルへ転送するデータ量を抑えるためにチャンネル内で読み出すキャパシタの範囲を設定することができ、この範囲を Region of Interest (ROI) と呼ばれる。LST では  $\text{ROI} = 40$  で設定しており、これは 1 GHz であれば 40 ns 分の波形情報に相当する。DRS4 が出力する信号には、いくつかの特性があるため観測データを解析する際には補正する必要がある。これらの特性について紹介していく。

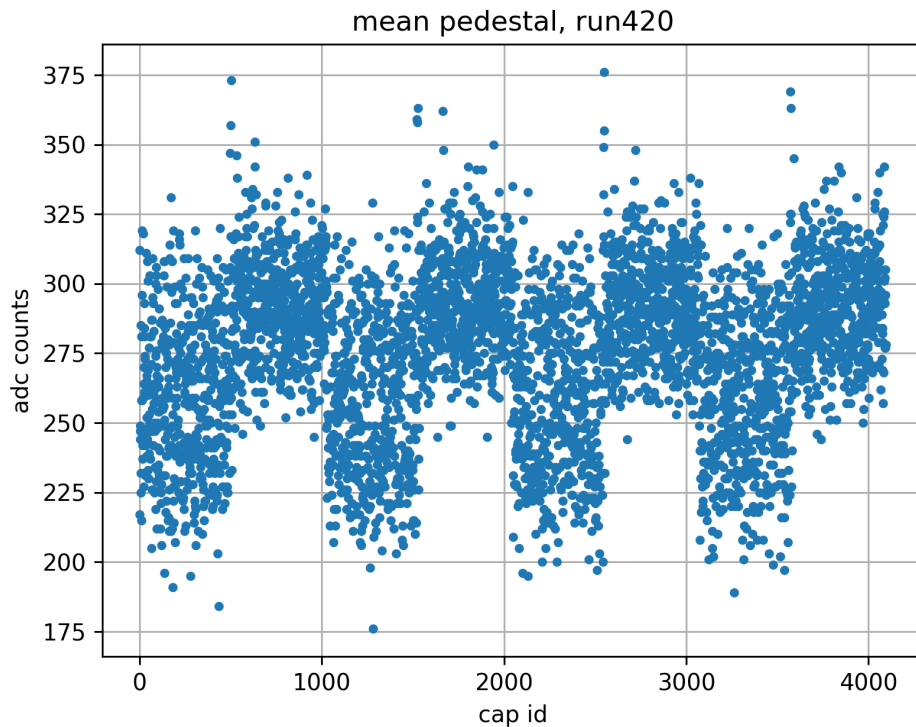


図 3.9 2019 年 5 月に LST1 で測定したペDESTAL (Run420) の平均値の分布。縦軸は ADC 値、横軸は DRS4 のキャパシタ id である。

#### ペDESTAL特性

DRS4 には図 3.9 で示すように、DRS4 の各チャンネルの 511 番目のキャパシタを境にペDESTALの電圧値が異なることが分かっている。この特性は各ゲイン系統の各 PMT チャンネル（ピクセル）毎に、波形読み出し回路にのみ電源をいれた状態にして電気ノイズを測定し（ペDESTAL測定）、キャパシタ毎のノイズ電圧の平均値を解析に用いる観測データに対して差し引くことにより補正が可能である。

#### dt 特性

DRS4 では、入力信号は入力チャンネル内の複数のキャパシタによって電荷として一時的に蓄えることで波形情報を記録している。そしてトリガーがかかると各コンデンサに蓄えていた電荷のうち、ROI の数のキャパシタだけ電荷の読み出しが行われる。そして読み出しが終わると再度各キャパシタに電荷を保存し始める。電荷の読み出しをされたキャパシタが再び読み出されるまでの時間を  $dt$  としたとき、この  $dt$  に応じて読み出される電荷量に変化することが知られている [30]。これは、波形読み出し後にキャパシタで蓄えていた電荷を放電したあとも残留電荷が僅かに残っていることが原因と考えられている。図 3.10 は  $dt$ （横軸）と読み出された電荷量（縦軸）の関係を表しており、 $dt$  が大きくなるにつれて読み出される電荷量が一定になっている。LST 波形読み出し回路では、 $dt$  が 100 ms を超えると残留電荷が放電され読み出される電荷量が一定になることが分かっている [23]。この特性は、 $dt$  に応じた電荷量を差し引くことにより補正することができる。

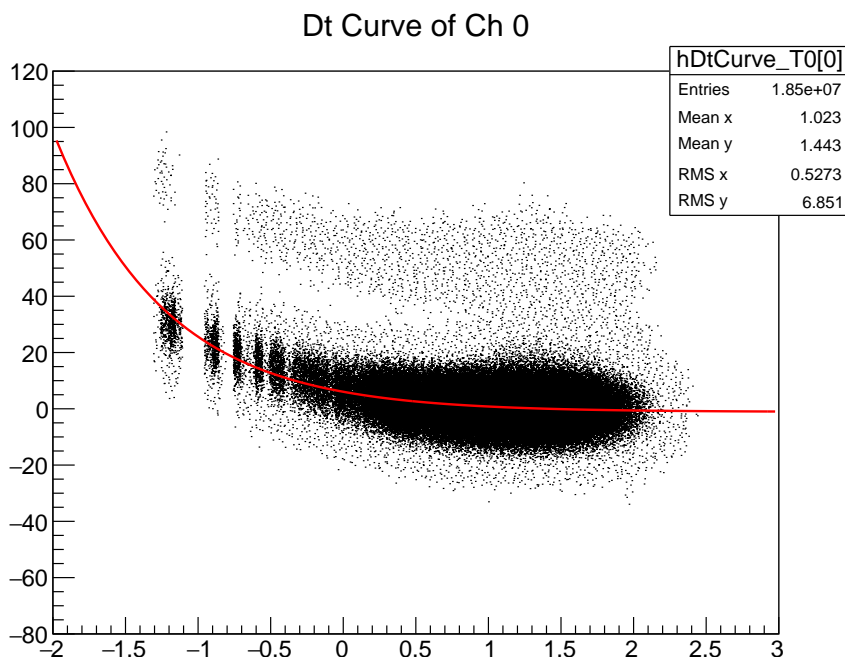


図 3.10 実験室で波形読み出し回路のペDESTALの測定より得られた dt カーブ。横軸はミリ秒をの log スケールを取った値、縦軸は ADC によって変換された電圧値を示している。プロットの上部で見えている構造はスパイク特性によるものであることが分かっている [23]。

### スパイク特性

DRS4 には、キャパシタから読み出した電荷が局所的に高くなることで偽のパルスが 1 つのキャパシタもしくは隣接する二つのキャパシタに現れるスパイク特性があり、図 3.11 に LST1 でのペDESTALの測定データにスパイクが混入した波形を示す。このスパイクには、発生する条件は 1 イベント前に読み出されたキャパシタの id に依存し、さらに発生パターンに対応して一定の電荷量を持つことが知られている [23]。この特性は、スパイクの発生条件を満たすキャパシタ id を見つけ出し、スパイクを記録しているキャパシタの両隣のキャパシタの電荷量で線形補間するもしくはペDESTALの測定ごとに各パターンのスパイクの電荷量の平均を導出し差し引くことによって補正できる。

### サンプリング時間幅の不均一性

DRS4 では、多数のインバータを繋げた Domino wave circuit でスイッチの切り替えを行っているが、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。サンプリング時間間隔を一定としてしまうと、DRS4 によってサンプリングされた信号波形からチェレンコフ光信号の到来時間や入力電荷量を推定する際、元の入力波形から歪んだ波形を再構成してしまうため、入力信号に対する到来時間や電荷量が正しく求められない。図 3.12 にサンプリング時間幅の揺らぎにより入力波形が歪むイメージを示す。LST では、夜光のバックグラウンドが 0.125 p.e./ns の時の入力電荷に対する電荷分解能の要求仕様および最終目標性能を図 3.13 に示すように設定している。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。数百光電子以上になるとポアソン統計による揺らぎの寄与が小さくなりおよそ 5% で頭打ちになることから、LST の目標性能を達成するには統計以外の寄与を 5% 以下にする必要がある。そこで、本研究の目標は、サンプリング時間幅による寄与を 5% 以下に抑

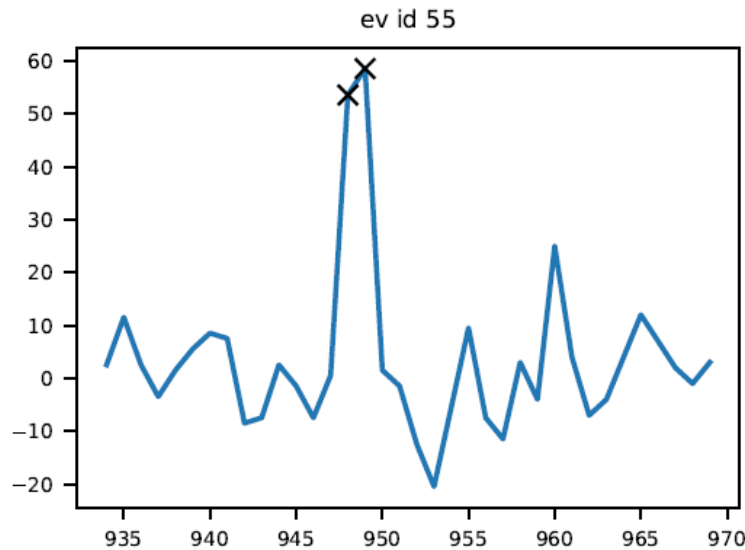


図 3.11 2020 年 6 月に LST1 で測定したペDESTALにスパイクが混入した時の波形。縦軸は電圧値 (ADC counts) で横軸はキャパシタ id である。この波形はペDESTALの差引きおよび残留電荷の校正をした後であり、黒点はスパイク成分を示し周りのキャパシタに比べて 50 ADC counts ほど高くなっていることが分かる。

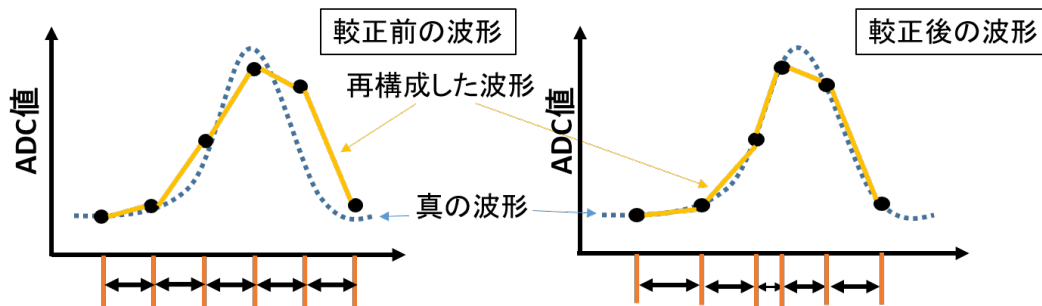


図 3.12 サンプリング時間幅の不均一性のイメージ。横軸は時間、オレンジの線はサンプリング時刻を表している。サンプリング時間幅が均一だと想定して波形を再構成すると左側の絵のように入力信号の波形を歪めてしまう。

えることである。

この特性について、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信号の到来時間に対して補正を行っており [30]、D.Striker-Shaver et al. [31] ではサイン波を用いてサンプリング時間幅のそのものの校正が行われている。また CTA においては、2011 年の佐々木修論 (茨城大学) [32] では矩形波を、2016 年の谷川修論 (京都大学) [33] ではテストパルス、2017 年の野崎修論 (京都大学) ではサイン波をそれぞれ用いたサンプリング時間幅の校正を行ってきた。本研究で行ったテストパルスとサイン波を用いた補正方法については、第 4 章と第 5 章でそれぞれ紹介する。

### 3.5.4 サイン波生成回路

サイン波生成回路は、DRS4 のサンプリング時間幅の不均一性をテストパルスを用いた方法よりも精度良く校正するために、LST2 号機から 4 号機の波形読み出し回路上に新たに搭載された [23]。生成回路上にあるオシレーターから 30 MHz で生成される矩形波を同じく回路上に実装したアナログ回路のローパスフィルターを用いて高

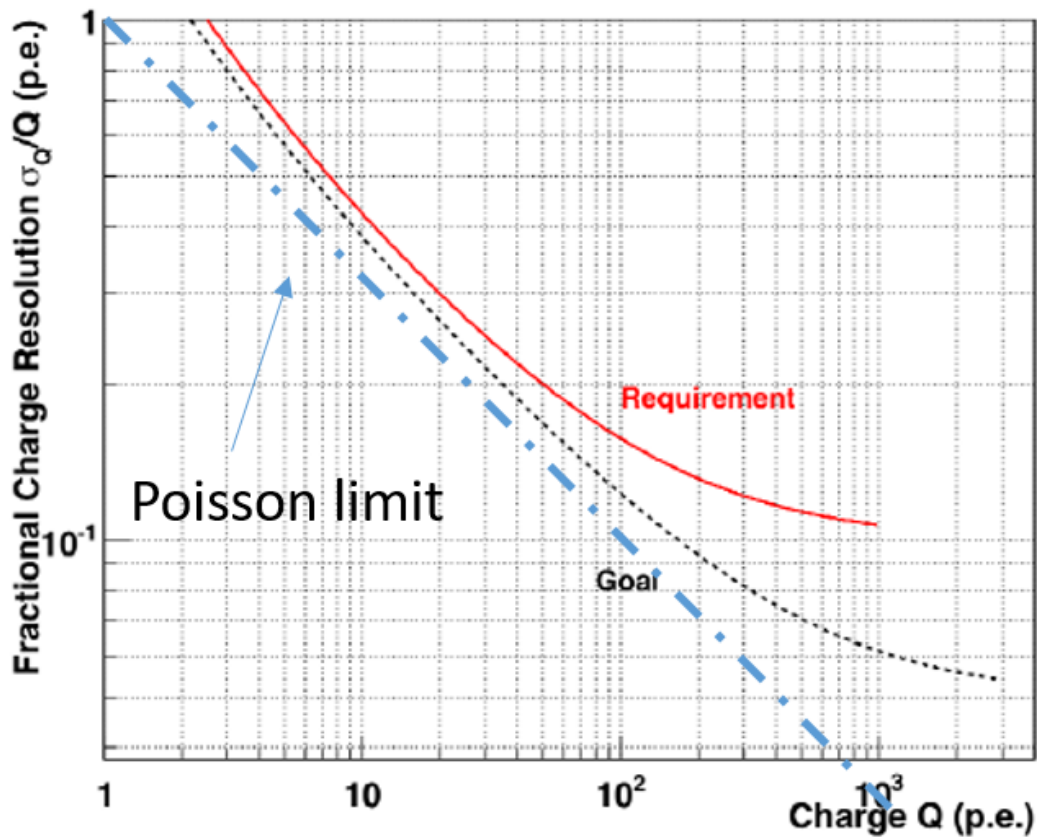


図 3.13 LST1 での電荷分解能の要求仕様のプロット [34] (一部改変している)。

周波成分を除去し、30 MHz のサイン波を取り出している。このサイン波は、読み出し回路上にある 8 つの DRS4 チップのサンプリング時間幅較正用の 9 チャンネル目に入力し、測定で得られた波形情報から較正を行う。

### 3.5.5 周辺回路

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

SCB は、波形読み出し回路と PMT をつなぐ治具としての役割の他にも、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温湿度のモニター、テストパルスの生成などの機能を持っており、図 3.14 に SCB の写真を示す。テストパルスは、読み出し回路上の FPGA から矩形波を SCB へと送り、SCB 上にあるテストパルス生成回路で PMT 信号を模した擬似信号を生成する。この生成された信号は、PMT 信号と同じように PACTA とメインアンプで増幅されたのち DRS4 へ入力される。

#### バックプレーン

バックプレーン (Back Plane, BP) は、波形読み出し回路の後段に接続されており、主な役割としては読み出し回路へ 24 V の電源を供給すること、カメラサーバーとのイーサネット通信、トリガー信号、クロック信号、Busy 信号の伝搬及び分配が挙げられる。また、BP では 10 MHz のクロック信号を生成することができ、本研究ではテストパルス生成回路のトリガー周波数のリファレンスとして用いている。





図 3.14 SCB の写真。この写真では SCB のうえに PACTA が取り付けられている。



図 3.15 バックプレーンの写真

## 第 4 章

# 大口径望遠鏡初号機（LST1）における波形サンプリング時間幅の較正

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 にはサンプリング時間幅に揺らぎがあり、この特性について較正を行わないと電荷分解能が悪くなることを第 3 章で述べた。この章では、現在稼働している LST1 での波形サンプリング時間幅の不均一性に較正について、まず 4.1 で目標について述べる。そして 4.2 節では LST1 でテストパルスを測定したデータを用いた補正係数の導出及び較正方法、4.3 節で解析環境について述べる。4.4 節では、LST1 で測定したテストパルスの電荷を算出する際に 4.2 節で述べた方法で導出した補正係数を適用し、適用の前後で電荷分解能を比較した結果について述べる。

### 4.1 目標

現在稼働している LST1 の読み出し回路では、SCB より生成できるテストパルスを用いてサンプリング時間幅の較正を行う手法が考案されているが、CTA の解析ツールでは確立していなかったため LST1 が完成してからこれまでサンプリング時間幅の較正は行われていなかった。LST の検出器では、電荷分解能について最終目標性能を設定しており、本研究全体としてはサンプリング時間幅による寄与を 5% 以下に抑えることを目標とすることを第 3 章で述べた。よってこの章での目標は、サンプリング時間幅による寄与を 5% 以下に抑えるとともにテストパルスを用いた較正を CTA の解析ツールに確立することである。

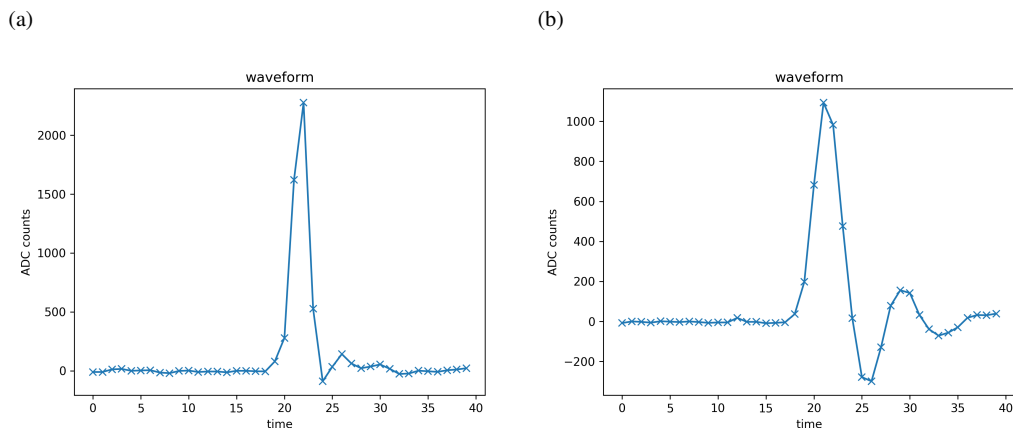


図 4.1 2019 年 12 月に取得したテストパルス波形の例。(a) は高ゲイン系統、(b) は低ゲイン系統の波形である。縦軸は ADC 値、横軸は時間に相当する。

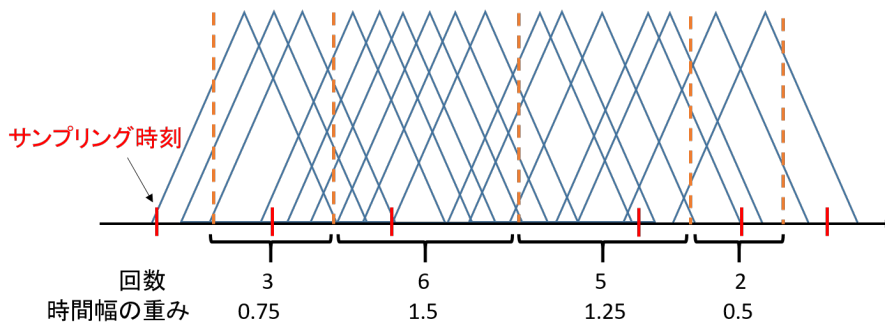


図4.2 テストパルスを用いた校正方法のイメージ図。赤線はサンプリング時刻を表しており DRS4 の各キャパシタと対応する。本研究でのサンプリング時間幅とは、この各赤線間の時間差を指している。橙色の線は隣接するサンプリング時刻の中点を表している。

## 4.2 サンプリング時間幅の校正方法

LST1 では、波形読み出し回路と PMT の間にある SCB 上でテストパルス生成でき、これを用いて校正する方法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）[33] では実際にこの方法を用いた検証を行っている。そして、本研究ではこの校正方法を現在稼働している LST1 に対して適用した。そこで、以降では校正方法について述べる。

校正方法のイメージを図 4.2 に示す。まず、読み出し回路の前段にある SCB でテストパルスを一定の周期で生成し読み出し回路へ 100 万回ほど入力する。テストパルスの生成周期を一定にすることで、波形読み出しが行われるキャパシタが一定間隔で遷移していき各キャパシタにテストパルスが一様に記録させるためである。そして、入力されたテストパルスの最大波高値を記録した回数を DRS4 内のキャパシタ毎に数えていく。入力したテストパルスの総数を  $N_{\text{total}}$ 、DRS4 チップのキャパシタ id ( $= i$ ) でテストパルスの最大波高値を記録した回数を  $N_i$ 、DRS4 の入力チャンネルあたりのキャパシタ数を  $N_{\text{cap}}$  ( $= 1024$ ) とすると、以下の式のように重みづけを行うことで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  $w_i$  が与えられる。

$$w_i = \frac{N_i}{N_{\text{total}}} \times N_{\text{cap}} \quad (4.1)$$

この操作をゲイン系統毎、ピクセル毎に行い  $2 \times 1855 \times 1024$  のテーブルにまとめる。第3章で述べたように、LST では DRS4 の入力チャンネルを 1 ピクセルあたり 4 つ、つまりキャパシタ 4096 個を用いているがキャパシタの切り替えは 1024 個のスイッチで行っているため、求めるべき補正係数は 1 ピクセルあたり 1024 個である。今回の校正ではテストパルスを 100 万イベントほど用いると前述したのは、テストパルスが各キャパシタに記録される確率がポアソン統計に従うとすれば、DRS4 のキャパシタ 1 個あたりおよそ 1000 イベント分の統計があれば約 3% の精度で校正できるためである。また、高ゲイン系統 (High Gain, HG) と低ゲイン系統 (Low Gain, LG) では信号の増幅率が異なるため校正に用いるテストパルスの波高値を各系統で変えている。ちなみに、ここまでで述べたテストパルスの生成に関して、波高値や生成周期は読み出し回路のファームウェアを書き換えることにより設定することができる。最後に、今回の方法を少し簡潔に述べると、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校正している。

## 4.3 解析環境およびツール

ここでは、この章で行った解析の際の環境およびツールについて述べる。



### 4.3.1 解析環境

今回の解析を行った場所は、La Palma 島に配置している LST1 の傍にあるコンテナ内に設置された計算機サーバーであり、2000 CPU core、3.4 PB のディスク容量を有している。望遠鏡の観測データはこのサーバー内に保存されており、サーバー上で観測データの解析を行うことができる。

### 4.3.2 解析ツール

この章での解析には、プログラミング言語の一種である python をベースにした CTA の解析ソフトウェアである ctapipe[35]、同じく python をベースにした LST 用解析ソフトウェアのテストベンチである cta-lstchain (lstchain) [36] を用いており、当時用いていたのは ctapipe でバージョン v.0.7.0、lstchain でバージョン v.0.4.5 であった。また、本章で行ったサンプリング時間幅の補正係数を算出するソースコードは付録に載せた。

## 4.4 LST 初号機テストデータを用いた時間幅校正の結果

ここでは、4.2 節で述べた方法を用いて LST1 で取得したテストパルスデータからサンプリング時間幅の補正係数を導出し、LST1 で測定したテストパルスの電荷量推定にサンプリング時間幅の校正を適用させた時の結果を紹介していく。

### 4.4.1 サンプリング時間幅の選出

ここでは、4.4.2 節で用いるサンプリング時間幅の補正係数テーブルを異なる 2 つの時期の測定データから作成した経緯を述べる。

本章で用いたテストパルスの測定データおよびベデスタルの測定データの全てを表 4.1 に示す。このうち本節および次節で用いたデータセットは Run01732 から Run02066 の計 6 つである。4.2 節で述べた方法でサンプリング時間幅を算出するにあたり、LST1 で測定したテストパルスが全 1855 ピクセルの ROI 内に入力されているかを確認するために 2019 年 12 月に測定した HG 用の校正データである Run01732 および LG 用の校正データである Run01734 について全イベントの積算電荷を確認した。電荷の算出には ctapipe に実装された電荷算出方法の一つで、ROI 内の入力信号波形の最大波高値となるキャパシタ位置（ピーク位置）を探し出し、その位置を含めた任意の範囲を波高値  $q_i$  ( $i$  はキャパシタ id) の時間積分によって電荷量  $Q$  を推定する LocalPeakWindowSum を用いた。DRS4 のキャパシタ id  $i$  のサンプリング時間幅を  $\Delta t_i$  とすると、電荷量  $Q$  は以下式で表せる。

$$Q = \sum_{i=0}^n \frac{q_i + q_{i+1}}{2} \times \Delta t_i \quad (4.2)$$

この時の設定としては、ピーク位置の 3 つ前のキャパシタ（セル）から 7 つ先のキャパシタ（セル）までを積分範囲とプログラム上で設定した。図 4.3 に Run01732 および Run01734 の全イベントに対する積算電荷を示す。図 4.3 のカラーバーは LST1 の各カメラピクセルの積算電荷量を示しており、図 4.3(a) で暗くなっている部分は該当の検出器モジュール（7 ピクセル分）の BP 上での遅延時間が他のモジュールとずれていたためである。この問題は、ずれてしまっているモジュールの BP の遅延時間の設定をファームウェア上で変更することによって 2020 年 2 月以降の測定では解消している\*<sup>1</sup>。この結果から、HG のデータセットでは全ピクセル分のサンプリング時間幅の較

\*<sup>1</sup> 詳 し く は 、<http://www.lst1.iac.es/elog/LST+commissioning/713>、<http://www.lst1.iac.es/elog/LST+commissioning/768>、<http://www.lst1.iac.es/elog/LST+commissioning/771>

表 4.1 サンプリング時間幅の校正に用いたデータセット

data set	取得日	subrun 数	イベント数	波形
Run00418	2019 年 5 月 22 日	19	$10^6$	HG 用のテストパルス
Run00419	2019 年 5 月 22 日	22	$10^6$	LG 用のテストパルス
Run00420	2019 年 5 月 22 日	3	$1.5 \times 10^5$	ベデスタル
Run00971	2019 年 7 月 12 日	3	$1.5 \times 10^5$	ベデスタル
Run00971	2019 年 7 月 12 日	20	$10^6$	LG 用のテストパルス
Run00973	2019 年 7 月 12 日	20	$10^6$	HG 用のテストパルス
Run01732	2019 年 12 月 17 日	38	$2 \times 10^6$	HG 用のテストパルス
Run01734	2019 年 12 月 17 日	38	$2 \times 10^6$	LG 用のテストパルス
Run01735	2019 年 12 月 17 日	5	$2.5 \times 10^5$	ベデスタル
Run02064	2020 年 6 月 1 日	42	$2 \times 10^6$	HG 用のテストパルス
Run02065	2020 年 6 月 1 日	41	$2 \times 10^6$	LG 用のテストパルス
Run02066	2020 年 6 月 1 日	4	$2 \times 10^5$	ベデスタル
Run02172	2020 年 7 月 3 日	1	$5 \times 10^4$	HG 用のテストパルス
Run02173	2020 年 7 月 3 日	2	$5 \times 10^4$	LG 用のテストパルス
Run02231	2020 年 7 月 14 日	2	$10^5$	ベデスタル
Run02561	2020 年 9 月 6 日	5	$2.5 \times 10^5$	ベデスタル
Run02562	2020 年 9 月 6 日	38	$2 \times 10^6$	HG 用のテストパルス
Run02567	2020 年 9 月 6 日	38	$2 \times 10^6$	LG 用のテストパルス

正ができないことになる。そこで、新たに 2020 年 6 月にテストパルスを測定したデータセットである Run02064 (HG 用) および Run02065 (LG 用) についても先ほどと同様にして全イベントに対する積算電荷を算出し、カメラマップに描画した結果を図 4.4 に示す。2020 年 6 月のデータセットでは、HG と LG とともに同じ検出器モジュールでテストパルスが ROI 内に入っていないが、これはテストパルス生成のリファレンスクロックとして用いている 10 MHz クロックのカウント数をリセットする際に用いている Pulse Per Second (PPS) クロック信号の位相がカメラ中心とカメラ端のモジュールで 100 ns ずれていることに起因している。この問題はバックプレーンのファームウェアを書き換えることで解消できる。

前述の 픽セル以外に、テストパルスが DRS4 の各セルにまんべんなく入力できなかったことにより 4.2 節の校正がうまくできない 픽セルが複数存在していた。5 万 3 千イベント毎 (subrun<sup>\*2</sup>) に DRS4 の各セルでテストパルスの最大波高値を記録した回数のヒストグラムを Run1734 および Run2065 での 픽セル id 1487 について 200 万イベント分プロットしたものを図 4.5 に示す。図 4.5 の縦軸は DRS4 のセル id、横軸は subrun id、カラーバーは各セルでテストパルスのピークを記録した回数である。縦軸にとったセル id に対するテストパルスを記録した回数の分布が subrun id に依らず一様であることが期待されるが、Run1734 (LG) の pixel id 1487 の結果である図 4.5(a) では subrun id 2 から 6 の間をはじめとしてプロット上では大きく 3 箇所テストパルスの記録回수에偏りが見えている。一方、別の時期に同じように時間幅校正用にテストパルスを測定した Run2065 (LG) の pixel id 1487 の結果である図 4.5(b) では subrun に対するテストパルスの記録回数の偏りが見えておらず校正に用いることができる。この現象は Run1734 だけでなく Run2065、HG 用のデータである Run1732、Run2064 でも確認されており、原因は調査中である。

<sup>\*2</sup> LST の観測毎のデータは Run、Run データを 5 万 3 千イベント毎に区切ってファイルを subrun と呼ぶ。

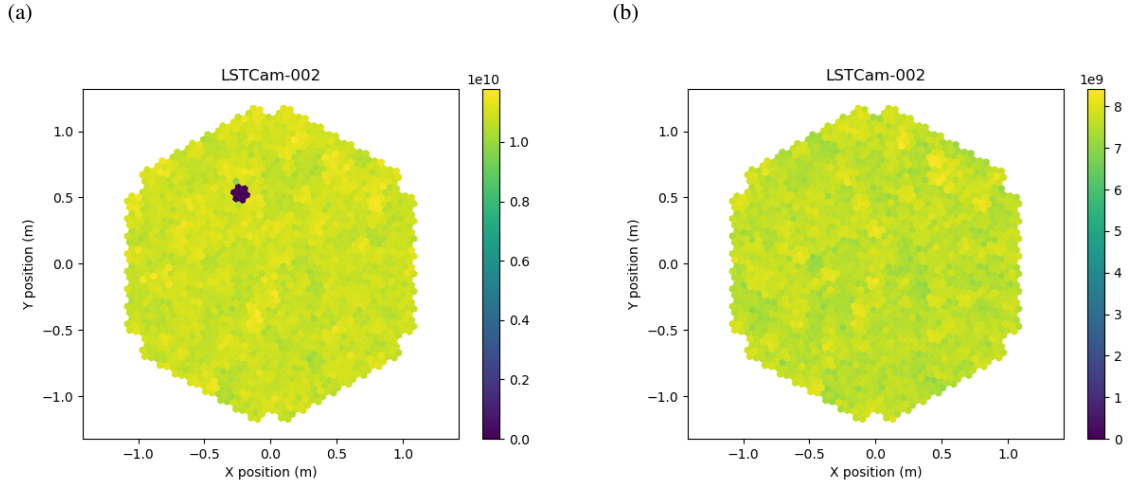


図 4.3 2019 年 12 月に測定したテストパルスデータである Run01732 および Run01734 の全イベントに対する積算電荷を LST のカメラマップ上にプロットした図。(a) は Run01732 の HG、(b) は Run01734 の LG に対応している。カラーバーが積算電荷量を示しており、Run01732 で暗くなっている部分（7 ピクセル分）は ROI 内にテストパルスが入っていないために積算値が 0 になっている。

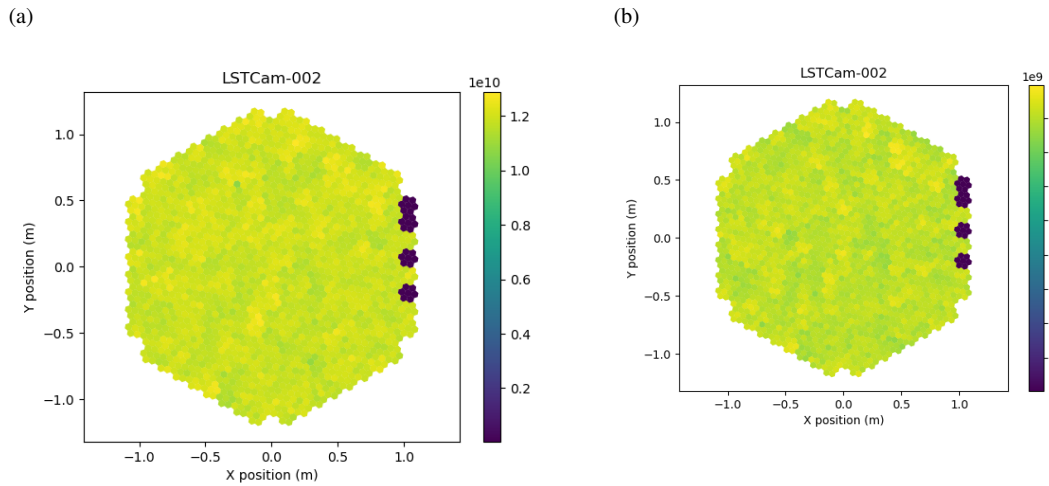


図 4.4 2020 年 6 月に測定したテストパルスデータである Run02064 および Run02065 の全イベントに対する積算電荷を LST のカメラマップ上にプロットした図。(a) は Run02064 の HG、(b) は Run02065 の LG に対応している。カラーバーは積算電荷量を示しており、各ゲイン系統で暗くなっている部分（28 ピクセル分）は ROI 内にテストパルスが入っていないために積算値が 0 になっている。

ここまでの結果から、各時期のデータセットのみではサンプリング時間幅の較正ができない場合も生じうることがあることが分かった。そこで、今回は 2019 年 12 月（Run1732 と Run1734）と 2020 年 6 月（Run2064 と 2065）の異なる時期のデータセットを用いて較正を行うことにした。手順としては、まず各データセットからサンプリング時間幅の補正係数のテーブルを 4.2 節で述べたように作成する。次に、各データセットから作成した補正係数のテーブルをそれぞれの作成元のデータセットに対して適用してテストパルスの電荷量を算出する。サンプリング時刻  $t_i$  および  $t_{i+1}$  とサンプリング時間幅  $\Delta t_i$  には  $\Delta t_i = t_{i+1} - t_i$  の関係があることを用いて、式 ?? を以下の

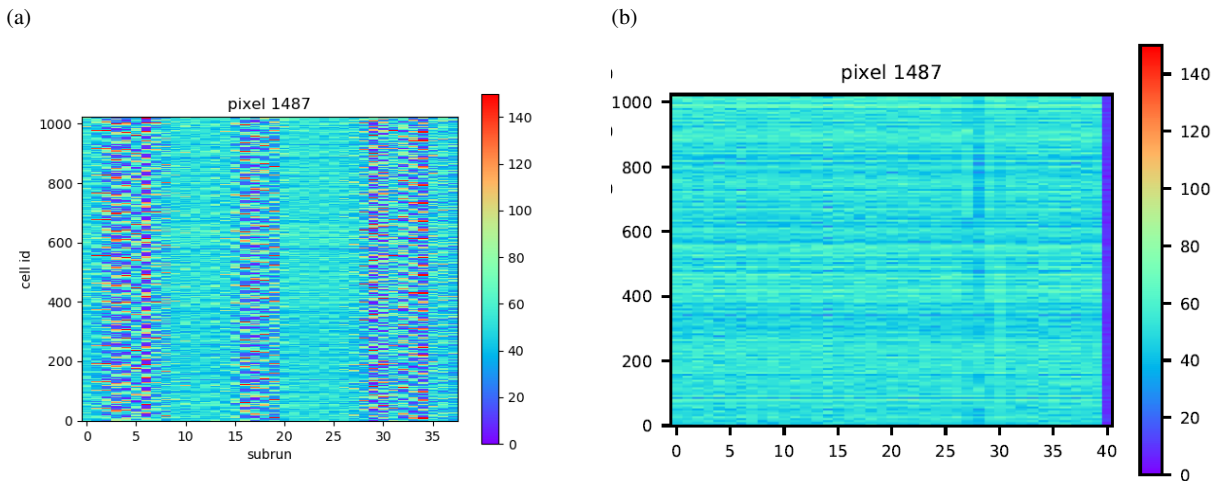


図 4.5 Run1734 (2019 年 12 月に取得) と Run2065 (2020 年 6 月に取得) の pixel id 1487 についてテストパルスの最大波高値を DRS4 の各セルを記録した回数をヒストグラムにしたものを 5 万イベント毎に、200 万イベント分をプロットした図である。(a) は Run1734 の LG、(b) は Run2065 の LG について示している。縦軸は DRS4 のセル id、横軸は 5 万イベント毎に区切った id (subrun id、LST での観測 Run を 5 万 3 千イベント毎に区切ったファイルの単位に対応)、カラーバーはテストパルスを記録した回数を示している。Run2065 のように各 subrun での分布が似た傾向を示しているのに対し、Run1734 では一部の subrun でテストパルスを記録する回数に偏りが出ている傾向が見えている。(b) の subrun id 40 は測定データファイルのイベント数が元々少ないため、他の subrun に比べて記録回数が少なくなっている。

ように変形できる。

$$Q = \frac{q_0 + q_1}{2} \times \Delta t_0 + \frac{q_1 + q_2}{2} \times \Delta t_1 + \cdots + \frac{q_{n-1} + q_n}{2} \times \Delta t_{n-1} + \frac{q_n + q_{n+1}}{2} \times \Delta t_n \quad (4.3)$$

$$= \frac{\Delta t_0 + \Delta t_1}{2} \times q_1 + \frac{\Delta t_1 + \Delta t_2}{2} \times q_2 + \cdots + \frac{\Delta t_{n-1} + \Delta t_n}{2} \times q_{n-1} + \frac{\Delta t_0 \times q_0 + \Delta t_n \times q_{n+1}}{2} \quad (4.4)$$

$$= \sum_{i=1}^{n-1} \frac{\Delta t_i + \Delta t_{i-1}}{2} \times q_i + \frac{\Delta t_0 \times q_0 + \Delta t_n \times q_{n+1}}{2} \quad (4.5)$$

式 4.5 の最後の項の  $\frac{\Delta t_i + \Delta t_{i+1}}{2}$  が本章で算出している補正係数にあたる。このようにして補正係数を適用した場合の電荷量を算出する。そして、電荷分解能がより良くなっているものをピクセル毎に選び出すことで最終的なテーブルを作成する。以後の結果では、Run1732 と Run2064、Run1734 と Run2065 に対してはこの節で述べた方法で作成したテーブルを用い、そのほかのデータセットについては各 Run のみを用いてテーブルを作成した。

#### 4.4.2 テストパルスに対する時間幅較正の検証

前節で説明した方法を実証するために、サンプリング時間幅較正用のデータを取得し LST1 カメラの全ピクセルの補正係数の導出を行った。サンプリング時間幅が較正できているかの検証として、今回は補正係数の導出に用いた較正用データから電荷を補正係数の適用前後で算出し電荷分解能を比較した。今回の較正用データは、2019 年の 12 月および 2020 年の 6 月に LST1 の各検出器モジュールの SCB からテストパルスを平均 1 kHz でそれぞれ約 200 万発を読み出し回路に入力したときのものを用い、高ゲイン系統と低ゲイン系統それぞれで読み出し回路のペダスタルと区別できるように、PMT から 50 光電子、500 光電子それぞれの信号が入力されたときと同等の波高値に調整した。通常は各系統に対する測定データが 1 セットずつあれば十分であるが、バックプレーンでの信号の遅延時間の調整ができていなかったことやカメラ端のピクセルでテストパルス生成に用いている Pulse Per Second

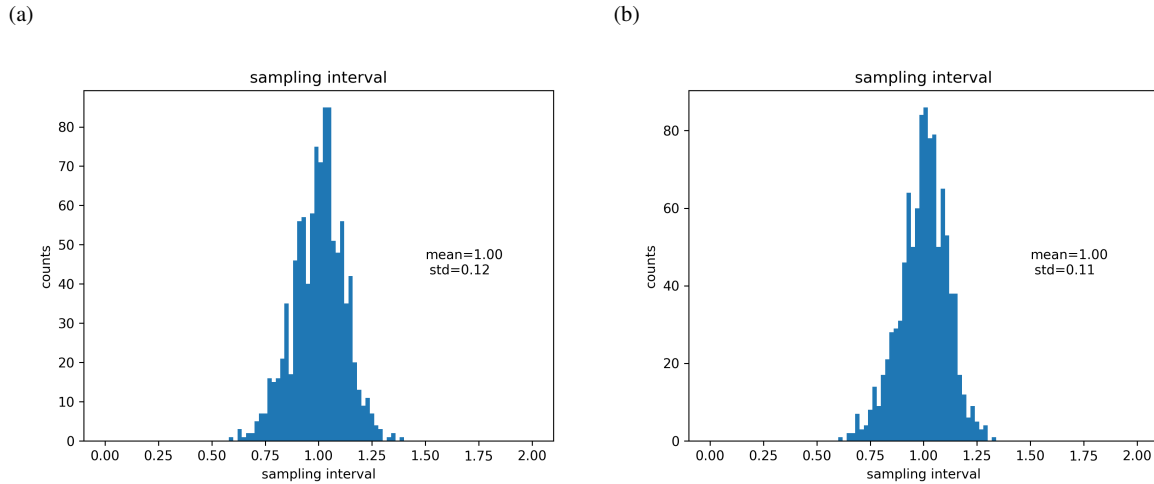


図 4.6 2019 年 12 月に取得した校正用データから導出したある 1 ピクセルの補正係数の分布。(a) は高ゲイン系統、(b) は低ゲイン系統の結果である。

(PPS) クロック信号の位相がずれてしまったことによって読み出し範囲内にテストパルスが入らなかったピクセルが各データセットで存在しており片方のデータでは全ピクセル分の補正係数を取得できないため今回は 4.4.1 節で述べたように 2 つのデータセットを用いている。補正係数の算出および電荷の計算をする際にはベデスタル補正をした上で行った。また、波形の読み出し範囲である 40 スライス内で最大の波高値をもつキャパシタを中心とした 7 スライス分の波高の積分値を電荷量とした。まず、補正係数は各データで 200 万イベントを用いて前節の式 4.1 より算出をし、各データに算出した補正係数を用いて電荷を計算して電荷分解能が良い補正係数をピクセル毎に選ぶことにより全ピクセル分の補正係数を用意した。ここでは、テストパルスのピークの 3 スライス前から 7 スライス分の波高値を足し合わせた値を電荷量とし、ヒストグラムに詰めてガウス分布でフィッティングして得られたパラメータ  $\mu$  と  $\sigma$  の比を電荷分解能とする。

$$f = C * e^{\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.6)$$

これにより得られた 1 ピクセルの補正係数の分布が図 4.6 である。

次にサンプリング時間幅の校正ができているかの検証として、2019 年 12 月の校正用データから補正係数を使った補正前後の電荷分解能の比較を行い、その結果を図 4.7 で示している。電荷及び電荷分解能の計算は前述と同様である。ただし、高ゲイン系統のデータでは先ほど述べたテストパルスが読み出し範囲内に入らなかったピクセルがいたためプロットからは除いた。高ゲイン系統では、ほとんどのピクセルでサンプリング時間幅の校正によって電荷分解能が改善し、平均と標準偏差は  $7.3 \pm 0.7\%$  から  $3.7 \pm 0.5\%$  まで改善することが確認できた。低ゲイン系統では校正によって全ピクセルで改善し、平均と標準偏差は  $5.4 \pm 0.6\%$  から  $2.9 \pm 0.3\%$  まで改善し、本研究の目標であるサンプリング時間幅による寄与を 5% 以下に抑えることができた。

#### 4.4.3 サンプリング時間幅の時間変化の調査

前節では 4.2 節で述べた方法でサンプリング時間幅の補正係数を導出し、校正前後の電荷分解能を比較することで校正ができていることを確認した。この節では、解析によって導出した補正係数が時間変化するのかを調査した。LST1 で異なる時期に取得されたテストパルスが入力された校正用データから補正係数を導出し、時期的に独立な測定データに適用した時の電荷分解能の年間の変動率を指標として評価を行った。この調査では、補正係数の導出には前節で用いた 2019 年 12 月と 2020 年 6 月の校正用データに加えて 2019 年 5 月に取得した同様の校正用デー

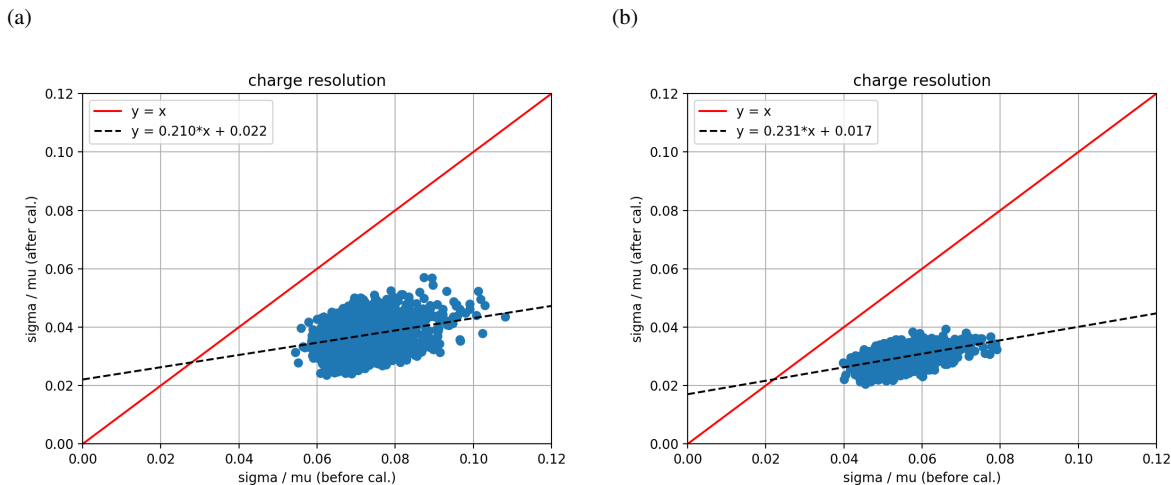


図 4.7 2019 年 12 月に取得した較正用データに対して、2019 年 12 月、2020 年 6 月に取得した各較正用データから導出した補正係数の適用前後での電荷分解能の比較した結果。(a) は高ゲイン系統、(b) は低ゲイン系統の結果で、縦軸は補正前の電荷分解能、横軸は補正後の電荷分解能を示している。

表 4.2 サンプリング時間幅の時間変化の調査で用いたデータセット

取得日	イベント数	用途
2019 年 5 月	$10^6$	補正係数の導出
2019 年 7 月	$10^6$	電荷の計算
2019 年 12 月	$2 \times 10^6$	補正係数の導出
2020 年 6 月	$2 \times 10^6$	補正係数の導出

タを、また電荷分解能の算出には 2019 年 7 月に取得したテストパルスの測定データを用い、使用したデータをまとめたものを表 4.2 に示す。ただし、2019 年 10 月に行った 4 つの検出器モジュールの交換によって読み出し回路の DRS4 チップが変わりサンプリング時間幅も変わってしまっているため交換したモジュール及び前節で述べた理由で読み出し範囲内にテストパルスが入っていないモジュールを除いたうえで評価を行った。この調査で得られた結果のプロットは図 4.8 に示す。補正係数の導出および電荷の計算には、各データセットの全イベントを用いた。プロットした 2 点（2019 年 12 月および 2020 年 6 月の値）に対して直線のフィッティングを行うことで得られた傾きおよび切片は、高ゲイン系統では  $-6.35 \times 10^{-5} \pm 3.12 \times 10^{-5}$  と  $4.09 \times 10^{-2} \pm 2.67 \times 10^{-4}$ 、低ゲイン系統では  $-1.07 \times 10^{-4} \pm 2.49 \times 10^{-5}$  と  $3.25 \times 10^{-2} \pm 2.25 \times 10^{-4}$  となった。ここでは最小二乗法によるフィッティングの誤差を各パラメータの誤差としている。年間の変動率  $f_{\text{year}}$  は、フィッティングを行うことで得られる傾き  $a$  と切片  $b$  から式 4.7 のように見積もった。図 4.8 での傾きは月間での値なので年間の値に直すために 12 倍する。

$$f_{\text{year}} = \frac{a}{b} \times 12 \quad (4.7)$$

LST1 のカメラの平均の電荷分解能の年間の変動率とフィッティングで得た傾きと切片の各誤差を考慮した誤差は、高ゲイン系統では約  $1.9 \pm 0.1\%$ 、低ゲイン系統では約  $4 \pm 0.1\%$  であることが確かめられた。今回の検証では、サンプリング時間幅の較正に用いるデータを変え、電荷分解能の算出に用いたデータを共通にしていることから、前述の変動率はサンプリング時間幅の変動率を示していると考えられる。ここで、サンプリング時間幅の較正によって得られた電荷分解能を  $\sigma_Q$ 、LST の目標性能である電荷分解能を  $\sigma_{\text{goal}}$  ( $=5\%$ ) とすると、 $\sigma_{\text{goal}}$  に至るまでの時間は、簡易的には次の式で与えられる。

$$\sigma_Q + \sigma_Q \times f_{\text{year}} \times t_{\text{year}} = \sigma_{\text{goal}} \quad (4.8)$$

4.4.2 項での結果では、 $\sigma_Q$  は高ゲイン系統で 3.7% であり低ゲイン系統で 2.9% であった。誤差を大きく見積



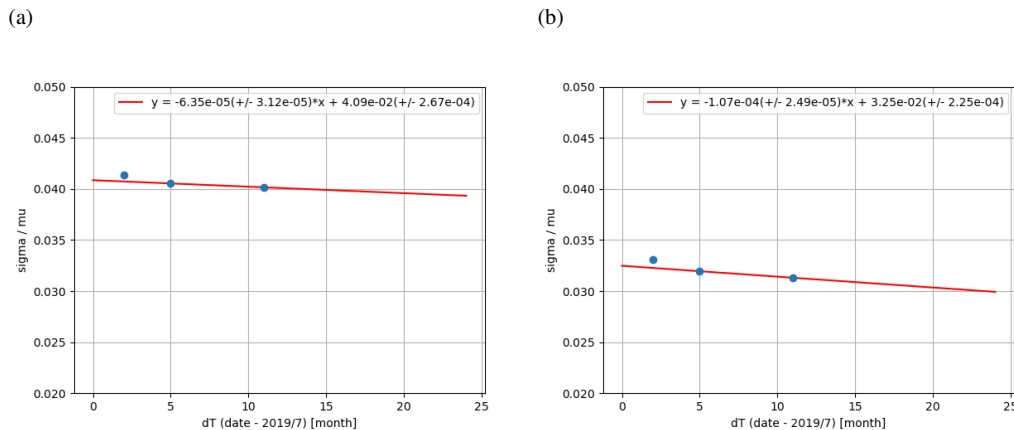


図 4.8 2019 年 7 月に取得したテストパルスの測定データに対して、2019 年 5 月、12 月および 2020 年 6 月に取得した各校正用データから導出した補正係数を適用したときの平均の電荷分解能の時間変化。(a) は高ゲイン系統、(b) は低ゲイン系統の結果で、縦軸は電荷分解能、横軸は 2019 年 7 月からの時間差の絶対値である。プロットでは見えていないが、カメラ全体の振舞いを知るために電荷分解能の標準誤差を誤差棒としている。

もったとき、上式より高ゲイン系統では変動率が 1.91% となり  $t_{\text{year}}$  は約 18 年、低ゲイン系統では変動率が 4.01% となり  $t_{\text{year}}$  は約 18 年とそれぞれ求まり、サンプリング時間幅は 10 年間は問題なく校正に用いることができることを確認した。

#### 4.4.4 サンプリング時間幅の汎用性の調査

DRS4 のサンプリング時間幅はベDESTAL のように測定ごとの変動がほとんどないことが知られているが、補正係数を導出してからどのくらいの時間までは問題なく校正に用いることができるかは知っておく必要がある。この節では、4.4.2 項において 2019 年 12 月及び 2020 年 6 月のテストパルスの測定データから作成した補正係数が、異なる時期の測定データに対してもサンプリング時間幅の校正ができるかを調査した。今回の調査では、表 4.3 に示すように 2019 年 5 月、7 月、12 月、2020 年 6 月、7 月、9 月の 6 つの異なる時期のテストパルスを測定したデータに対して補正係数を用いて校正をしたうえで電荷分解能を算出し、補正係数の導出に用いたデータの取得月からの変動率を求めた。電荷分解能の算出には、2020 年 7 月のデータのみ 5 万イベント、その他のデータでは 100 万イベント用い、各データにおける全ピクセルの電荷分解能の平均をプロットしたものが図 4.9 である。このプロットに対しても直線でフィッティングを行い得られた傾きおよび切片は、高ゲイン系統で  $1.09 \times 10^{-4} \pm 1.05 \times 10^{-5}$  と  $3.72 \times 10^{-2} \pm 8.01 \times 10^{-5}$ 、低ゲイン系統で  $7.40 \times 10^{-5} \pm 6.03 \times 10^{-6}$  と  $2.94 \times 10^{-2} \pm 4.31 \times 10^{-5}$  のようになった。ここでの誤差も、最小二乗法によるフィッティングパラメータの誤差を用いた。年間の変動率は 4.4.3 項と同様に式 4.7 で算出した。この調査から、高ゲイン系統では年間での変動率が  $3.52 \pm 0.03\%$ 、低ゲイン系統では  $3.02 \pm 0.02\%$  という結果が得られた。また、4.4.2 項で得た LST1 の平均の電荷分解能、誤差を大きく見積もったときの変動率および式 4.8 を用いて、本研究の目標である電荷分解能が 5% に至るまでの時間を求めると、高ゲイン系統では変動率を 3.55% としたときで  $t_{\text{year}}$  は約 10 年、低ゲイン系統では変動率を 3.04% としたときで  $t_{\text{year}}$  は約 24 年となり、各系統ともに 10 年間は問題なく校正に用いることができることを確認した。

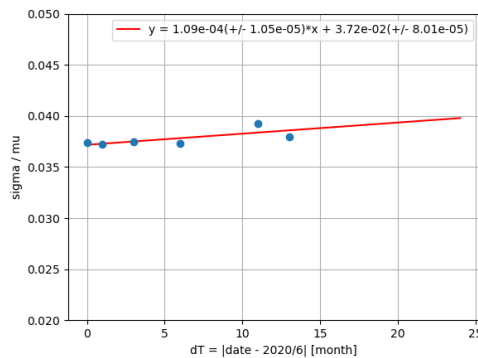
## 4.5 議論

LST1 の読み出し回路では、DRS4 チップのサンプリング時間幅の揺らぎを約 100 万イベントのテストパルスを用いて校正を行うことができ、4.4.2 項では LST1 のカメラ全体の電荷分解能が高ゲイン系統で 3.7%、低ゲイン系

表 4.3 サンプリング時間幅の汎用性の調査で用いたデータセット

取得日	イベント数	用途
2019 年 5 月	$10^6$	電荷の計算
2019 年 7 月	$10^6$	電荷の計算
2019 年 12 月	$2 * 10^6$	補正係数の導出及び電荷の計算
2020 年 6 月	$2 * 10^6$	補正係数の導出及び電荷の計算
2020 年 7 月	$5 * 10^4$	電荷の計算
2020 年 9 月	$2 * 10^6$	電荷の計算

(a)



(b)

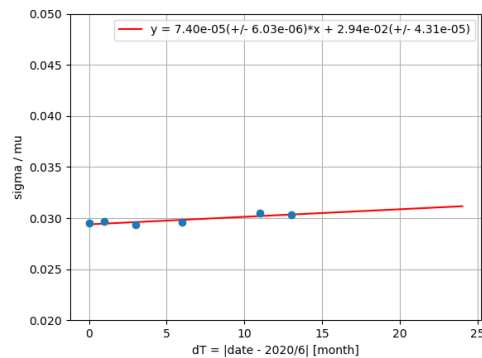


図 4.9 2019 年 5 月から 2020 年 9 月の中の異なる 6 つの時期に取得したテストパルスの測定データに対して、2019 年 12 月および 2020 年 6 月に取得した較正用データから導出した補正係数を適用したときの平均の電荷分解能をプロット。(a) は高ゲイン系統、(b) は低ゲイン系統の結果で、縦軸は電荷分解能、横軸は 2019 年 7 月からの時間差の絶対値である。プロットでは見えていないが、カメラ全体の振舞いを知るために電荷分解能の標準誤差を誤差棒としている。

統で 2.9% となり、LST の目標性能である 5% 以下を満たしていることが確認できた。しかし、LST の電荷分解能の目標性能は 0.125 p.e./ns の夜光がバックグラウンドを加味した場合であるので、今後は LST1 のカメラに実際に光を入射させた観測データから LST1 のカメラの電荷分解能について評価する必要がある。4.4.3 項および 4.4.4 項で得た年間の変動率から、カメラモジュールの交換が行われない限りサンプリング時間幅のテーブルは約 10 年間は問題ないと考ええる。



## 第 5 章

# 大口径望遠鏡 2 から 4 号機（LST2-4）における波形サンプリング時間幅の較正

LST2 号機以降用の波形読み出し回路は、LST1 で用いているものから改良されている。主な変更点としては、読み出し回路上の各要素に対する温度依存性や入力電圧依存性をモニターするための温湿度計・電圧モニター回路および DRS4 チップのサンプリング時間間隔較正用サイン波生成回路の導入である [23]。サイン波生成回路を導入したことにより、LST1 で行っているテストパルスを用いた方法に比べて精度良くサンプリング時間幅の較正ができ、さらに較正に用いるイベント数の削減によって計算時間が短縮され、測定したあとすぐに較正結果を確認できるようになった。5.1 節ではこの章での目標、5.2 節でサイン波生成回路を用いた DRS4 のサンプリング特性の較正方法について、5.3 節では実験室で LST2 号機以降用の読み出し回路を用いた較正用データの測定方法などについて、5.4 節ではサイン波生成回路を使った較正の検証結果および第 4 章で紹介した SCB で生成できるテストパルスを用いた方法との較正精度の比較した結果について述べる。

### 5.1 目標

第 3 章で述べたように、LST では検出器の電荷分解能について最終目標性能を設定しており、本研究の共通の目標としては LST の電荷分解能についてサンプリング時間幅に依る寄与を 5% 以下に抑えることである。この章で扱う LST2 号機以降の読み出し回路では、テストパルスを用いた方法よりも精度良く較正を行うためにサイン波生成回路を搭載している。よってこの章での目標は、サンプリング時間幅の寄与を 5% 以下に抑えるとともにサイン波を用いた較正によってテストパルスのときよりも精度良く較正し、較正にかかる時間を抑えることである。

### 5.2 サンプリング時間幅の較正方法

LST2 号機以降の波形読み出し回路では、回路上に新たに搭載したサイン波生成回路で生成したサイン波を DRS4 チップのサンプリング時間間隔較正用である 9 つ目のチャンネルに入力している。生成回路では周波数 30 MHz のサイン波を生成し、ROI は 100 個のキャパシタ（100 スライス）に設定して 1 万イベント分の測定を行った。今回はサンプリング周波数を 1.034 GHz に設定しているためサンプリング時間間隔は等間隔の場合であれば 0.967 ns である。また、サイン波 1 周期の時間は 33.33 ns である。サイン波を使った較正には、D. Striker-Shaver et. al の較正手法 [31] を元を開発した野崎修論の較正手法 [23] を用いており、隣り合う 2 つのキャパシタ間で補正するローカルキャリブレーションを行い、そのあとに入力サイン波 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。

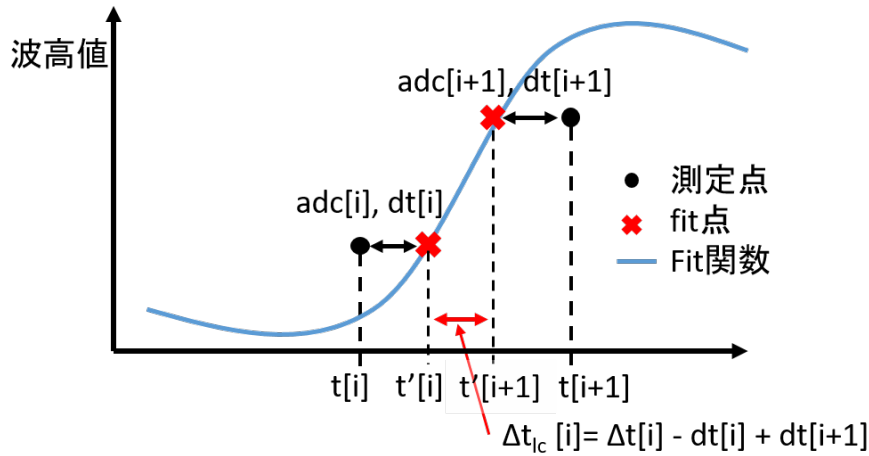


図 5.1 ローカルキャリブレーションの模式図。

### 5.2.1 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 5.1 に示すように隣り合う 2 つの各データ点とフィット関数との時間方向の差からサンプリング時間間隔を較正する方法である。まず、測定したサイン波について以下のようなサイン関数でフィッティングを ROI 全体に対して行う。

$$p[0] \times \sin(2\pi p[1]t + p[2]) + p[3] \quad (5.1)$$

このフィット関数のパラメータのうち入力周波数に対応する  $|p[1]|$  は、入力周波数が既知であるため 30 MHz で固定している。較正には、フィッティングで得た波高値のパラメータ  $|p[0]|(1 \pm 0.75)$  のサイン波の立ち上がり・立ち下りの部分に含まれるデータ点を用いる。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  $\Delta t[i] = t[i+1] - t[i]$  となる。サンプリングされた波形に対するフィット関数が各キャパシタの波高値と同じであるときの時刻をそれぞれ  $t'[i], t'[i+1]$  とする。この際ペDESTAL特性や残留電荷の補正によって各キャパシタの波高値は正しく揺らぎがないとしている。このとき、キャパシタ  $i, i+1$  でのサンプリング時刻とフィット関数上でのサンプリング時刻の差は  $dt[i] = t'[i] - t[i]$ 、 $dt[i+1] = t'[i+1] - t[i+1]$  とすれば、予想されるキャパシタ  $i, i+1$  のサンプリング時間幅  $\Delta t_{ic}[i]$  は以下の式で表せる。

$$\Delta t_{ic}[i] = t'[i+1] - t'[i] \quad (5.2)$$

$$= \Delta t[i] - dt[i] + dt[i+1] \quad (5.3)$$

式 5.3 を各イベントのキャパシタ毎に行い、全イベント  $N_{event}$  で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していく。

$$\overline{\Delta t_{ic}} = \frac{\sum \Delta t_{ic}[i]}{N_{event}} \quad (5.4)$$

今回はサンプリング周波数が約 1 GHz、ROI が 100 スライス、入力サイン波の周波数が 30 MHz であるのでサイン波 1 周期は 33.33 ns となり、ROI 内には 3 周期分のサイン波が入ることになる。

### 5.2.2 グローバルキャリブレーション

隣り合う 2 点間のデータ点から局所的な補正をするのがローカルキャリブレーションであったが、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たサンプリング時間幅を用いて図 5.2 に示すようにサ

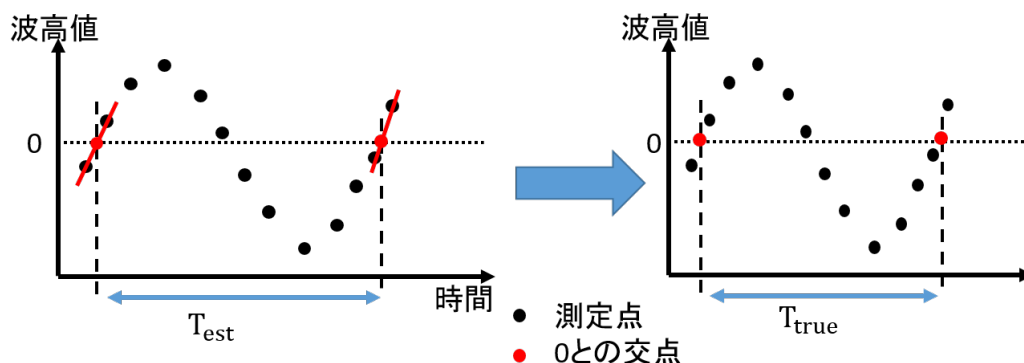


図 5.2 グローバルキャリブレーションの模式図。

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っていく。流れとしては、初めに ROI 内のサイン波 1 周期分を見つけ出すために測定波形が波高値 0 の直線との交点（crossing point）を探す。この crossing point は、波高値が 0 になる前後のキャパシタの波高値を用いて線形補間することにより決定する。2 点の crossing point からサイン波の周期を推定できこの時間を  $T_{\text{est}}$ 、入力したサイン波の理想的な周期を  $T_{\text{true}} (= 33.33 \text{ ns})$  とする。着目したサイン波 1 周期に含まれたキャパシタのサンプリング時間幅  $\Delta t_{\text{lc}}[i]$  を測定から得た周期と理想値との比を用いて以下のように補正する。

$$\Delta t_{\text{gc}}[i] = \Delta t_{\text{lc}}[i] \times \frac{T_{\text{true}}}{T_{\text{est}}} \quad (5.5)$$

式 5.5 を各イベント・各サイン波周期内の各キャパシタ毎に行い、全イベント  $N_{\text{event}}$  で平均したものを最終的なサンプリング時間幅とする。

$$\overline{\Delta t_{\text{gc}}} = \frac{\sum \Delta t_{\text{gc}}[i]}{N_{\text{event}}} \quad (5.6)$$

## 5.3 測定・解析環境

ここでは、実験室で LST2 号機以降用の波形読み出し回路を用いた測定をする際の環境や、実際の測定や解析に用いたツールについて触れる。

### 5.3.1 測定環境

図 5.3 に実際の測定をする際のセットアップを示す。測定には、実際に LST2 号機以降の焦点面カメラに用いるものと同じ波形読み出し回路、テストパルスの生成をするためのスローコントロールボード（Slow Control Board, SCB）および信号増幅をするプリアンプ（Preamplifier for CTA, PACTA）、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担っているバックプレーン（Back Plane, BP）、読み出し回路への電源の供給には 24 V の電圧と 1 A の電流が流せる直流安定化電源<sup>\*1</sup>、読み出し回路の制御や測定データの解析に用いるデスクトップ PC<sup>\*2</sup>を用いている。波形読み出し回路、SCB および PACTA、BP の概要については第 3 章の通りである。波形読み出し回路にはの直流電圧が必要であるため、電源装置の正出力端子と GND 端子に BP からの銅線を繋ぎ、出力電圧を +24 V に設定している<sup>\*3</sup>。

<sup>\*1</sup> 菊水電子工業株式会社製の PMM24-1QU

<sup>\*2</sup> OS は CentOS7.9、HDD は 1 TB、メモリは 4 GB である。

<sup>\*3</sup> 実際に 24 V の電圧をかけた時の電流は 0.7 A 程度であり、テストパルスやサイン波の生成を行うと 0.8 A まで増加した。

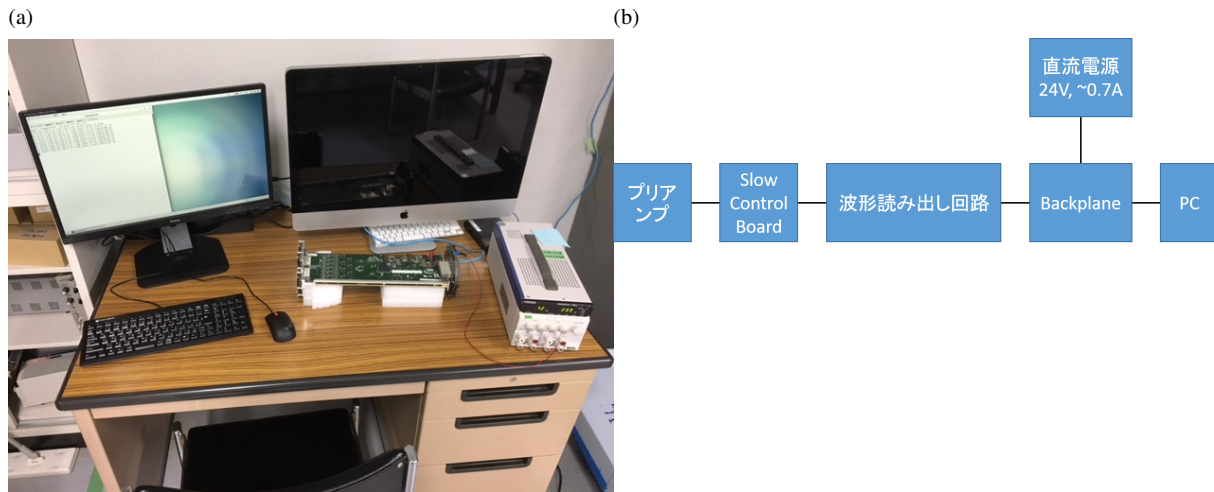


図 5.3 測定のセットアップ。5.3(a) 実際の写真であり、5.3(b) はブロック図で簡易的に示したものである。直流安定化電源から 24 V の電圧をバックプレーンを介して波形読み出し回路へ供給している。また、読み出し回路でサンプリングし A/D 変換された信号波形の情報や FPGA へのコマンド送信などはバックプレーンを介して実験室のデスクトップ PC-読み出し回路間で通信を行う。

表 5.1 実験室での測定で用いた解析ツール等

名称	バージョン (時期)	用途
dragonqc	(2020 年 7 月)	読み出し回路の制御および解析
ROOT	v.5.34.34	dragonqc 上での解析
ConverToZfits	(2018 年 8 月)	ファイル形式の変換
ctapipe	v.0.8.0	解析
cta-lstchain	v.0.6.3	解析

### 5.3.2 解析ツール

この章での解析では大きく分けて 2 種類の解析ツールを用いた。1 つは、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`<sup>\*4</sup>で、読み出し回路上の操作や測定したサイン波の解析に用いた。もう一つは、CTA の python 言語ベースの解析ソフトウェアである `ctapipe`[35] 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` (`lstchain`) [36] を第 4 章と同様にテストパルスの解析に用いている。また、実験室での測定データはバイナリ形式、LST1 の観測データファイルは `fits.fz` 形式とファイル形式が異なり、`ctapipe` および `lstchain` でのファイル読み出しは実験室データで取得したバイナリ形式を想定していないため、実験室データファイルを LST1 のファイル形式に変換するソフトウェア `ConverToZfits` も用いている。表 5.1 に解析に用いたツールを示す。

### 5.3.3 解析に用いた波形データ

今回の検証では、各測定データの較正に用いる波形読み出し回路のペデスタル、DRS4 サンプリング時間幅の較正に用いる読み出し回路上で生成したサイン波および SCB で生成したテストパルスの 3 種類の信号を測定し解析を行った。テストパルスおよびテストパルスの補正用ペデスタルの測定時は、LST1 と同様に ROI は 40 スライス

<sup>\*4</sup> このソフトウェアは宇宙線研究所助教藤隆之氏が LST の読み出し回路の品質管理を行うためにまとめたものであり公開はされていない。

とした。また、テストパルスの測定では、トリガー周波数を 2 kHz で固定することにより、DRS4 の読み出し始めの位置を約 7 スライスの間隔でずらし、各キャパシタにテストパルスのピークが一様に記録されるようにしている。実験室で測定したテストパルスの波形を図 5.4 に示す。また、テストパルスの測波高値はデスクトップ PC から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により dB 単位で設定することができ、LST1 での設定と同じく高ゲイン系統では 40 dB、低ゲイン系統では 60 dB と設定することによりそれぞれ 50 光電子と 500 光電子に対応する波高値を実現している。光電子数と電圧値の対応は、高ゲイン系統を基準にするとペDESTAL、残留電荷、スパイクの校正後でおよそ 40 ADC counts/光電子である<sup>\*5</sup>。テストパルスに対するペDESTALの校正用のデータは、LST1 でのペDESTAL測定を参考に 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て 20 万イベント分を測定した。ランダムなトリガーの生成は読み出し回路のファームウェア上で実装した線形帰還シフトレジスタによって行っている。

次に、読み出し回路で生成できるサイン波について述べる。図 5.5 にペDESTAL補正後のサイン波の測定波形を示す。サイン波の測定では、通常の観測やテストパルスの測定のときに比べて DRS4 にかかるオフセット電圧をデスクトップ PC からコマンドを送り読み出し回路のファームウェアを書き換えることで高く設定している。これは、生成サイン波の波高値が ± 450 mV であるのに対し DRS4 に入力できる電圧の範囲である −50 mV から +950 mV を超えてしまうためである。また、サイン波の測定時は、LST2 号機以降の読み出し回路に対して行った品質管理測定を参考に ROI を 100 スライス、トリガー周波数を 300 Hz に設定している。サイン波用のペDESTALの測定でもオフセット電圧およびトリガー周波数はサイン波の測定時と同じであるが、ROI は 1024 スライスに設定している。また、DRS4 のサンプリング周波数は DRS4 外部からのリファレンスクロック信号の周波数  $f_{\text{ref}}$  およびクロック数  $N_{\text{clock}}$  を用いて以下の式 5.7 を用いて表される。

$$f_{\text{sampling}} = \frac{f_{\text{ref}}[\text{Hz}] \times 1024[\text{cell}]}{N_{\text{clock}}} \quad (5.7)$$

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て使い、クロック数を  $N_{\text{clock}} = 66$  とすることでサンプリング周波数を 1.034 GHz に設定した。

## 5.4 実験室でのサンプリング時間幅校正の結果

前節で説明したサイン波の測定データを用いた DRS4 のサンプリング時間幅の校正を行った結果を 5.4.1 項で示す。また、第 4 章で行ってきたテストパルスの測定データを用いたサンプリング時間幅の校正を 2 号機以降の波形読み出し回路でも行い、サイン波データを用いた方法との校正精度を電荷分解能と時間分解能の 2 点での比較をした結果を 5.4.2 項で示す。

### 5.4.1 サイン波を用いたサンプリング時間幅校正の検証

ここでは、LST2 号機以降の波形読み出し回路で生成したサイン波を測定したデータを用いてサンプリング時間幅の校正ができていないかの検証を行った。校正に用いた測定データについては表 5.2 に示す。ペDESTALやサイン波の測定後の解析の流れとしては、まず測定したペDESTALデータからオフセットの平均の波高値を DRS4 チップ毎、キャパシタ毎に算出し  $7[\text{pixels}] \times 4096[\text{capacitors}]$  テーブルにまとめる。次に測定サイン波に対して先ほど算出したペDESTALテーブルの値を差し引き、5.2 節で述べた方法でサンプリング時間幅の校正を読み出し回路上の DRS4 チップ毎に行う。表 5.2 に示した data2 のサイン波を用いてローカルキャリブレーションおよびグローバルキャリブレーションによって得られた読み出し回路上の 1 チップの全キャパシタのサンプリング時間幅の分布を

<sup>\*5</sup> 2 系統間で 16 倍の差があるため低ゲイン系統を考える場合は 16 分の 1 する点に注意。

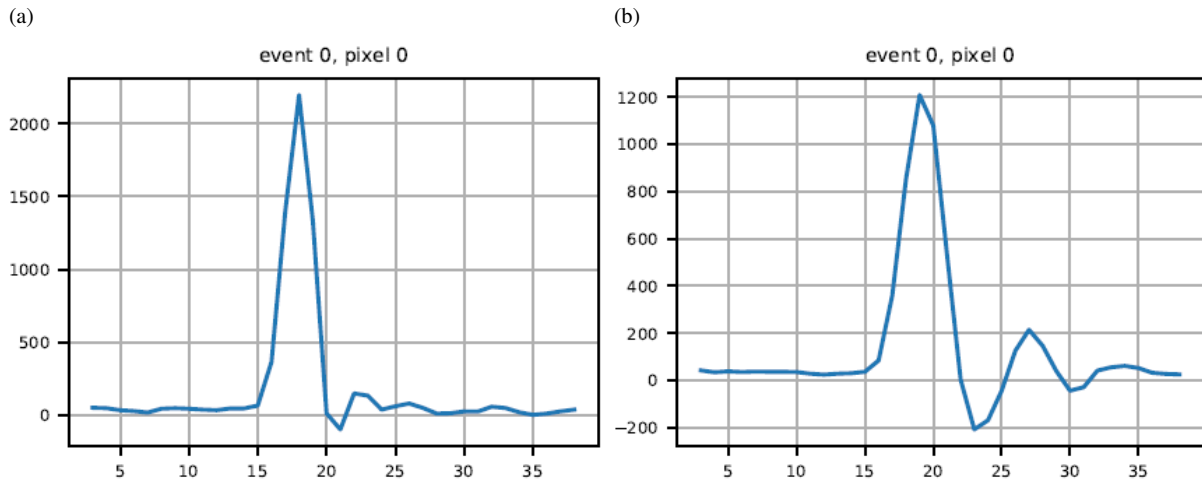


図 5.4 LST2 号機以降用の波形読み出し回路で測定したテストパルスのベデスタル補正後の波形。縦軸は波高値、横軸はスライス数である。(a) は高ゲイン系統で 50 光電子に相当する波高値をもった波形、(b) は低ゲイン系統で 500 光電子に相当する波高値をもった波形をプロットしている。

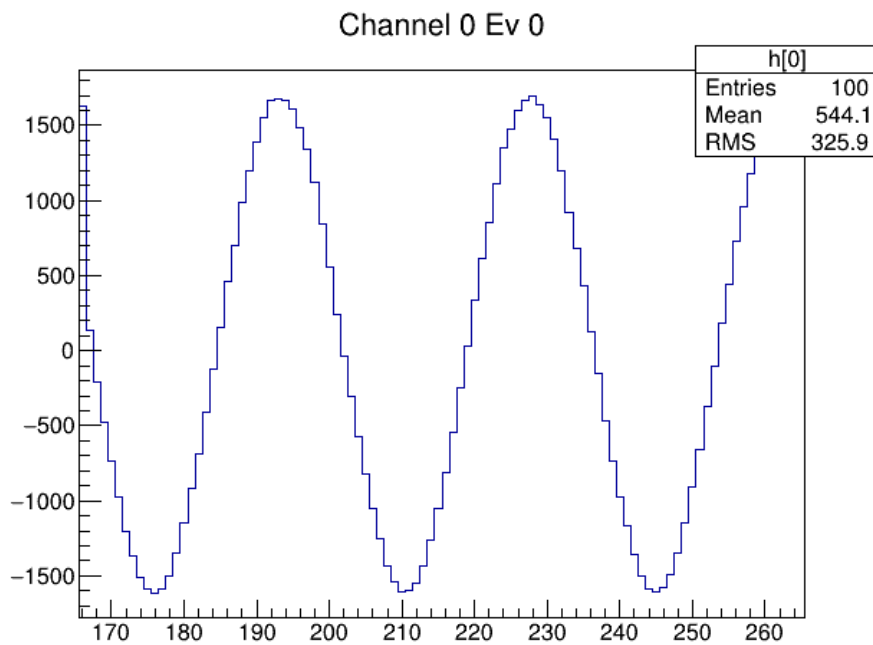


図 5.5 波形読み出し回路上で生成したサイン波の波形をプロットした図。縦軸は ADC 値、横軸はキャパシタ id である。約 1 GHz で波形サンプリングをしており、入力サイン波の周波数が 30 MHz なのでおよそキャパシタ 33 個でサイン波 1 周期分を記録している。

表 5.2

	波形	取得日	イベント数	用途
data1	ベデスタル	2020 年 7 月 16 日	$10^3$	ベデスタル補正
data2	サイン波	2020 年 7 月 16 日	$10^4$	サンプリング時間幅の較正
data3	ベデスタル	2021 年 1 月 14 日	$10^3$	ベデスタル補正
data4	サイン波	2021 年 1 月 14 日	$10^4$	period time の算出

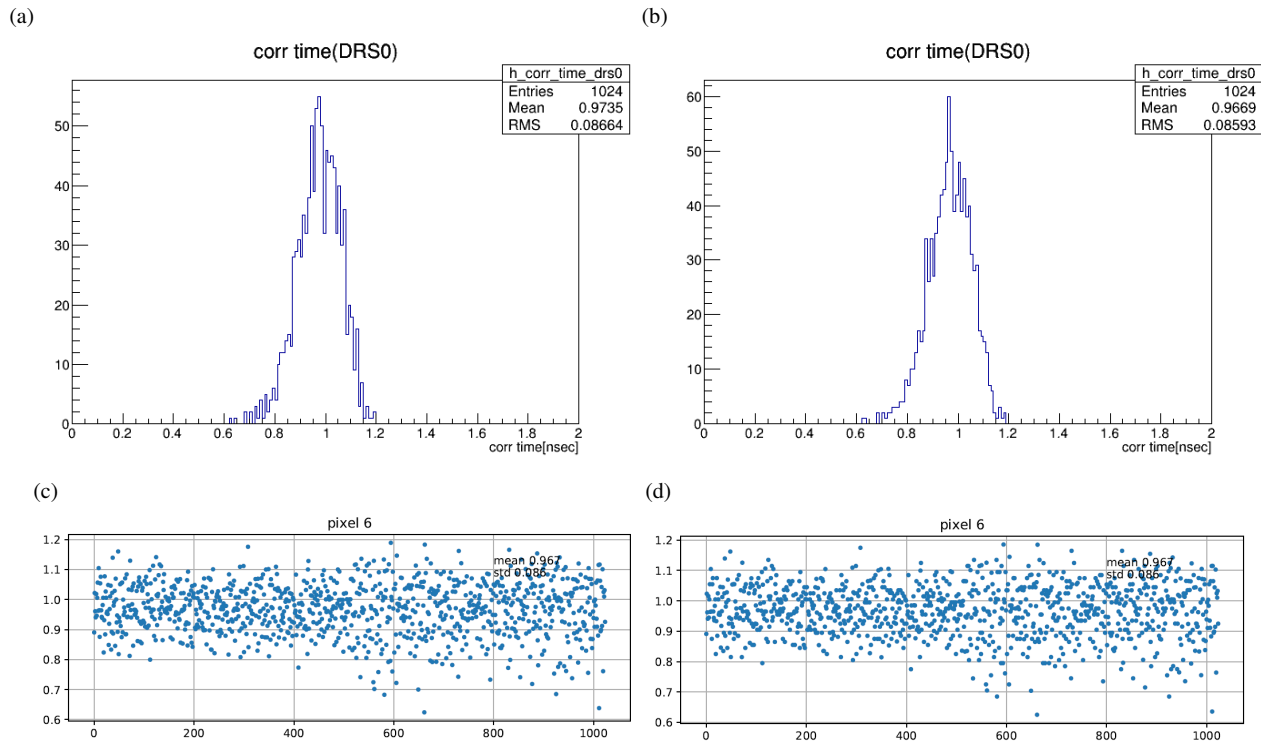


図 5.6 サイン波を使った較正によって得られたサンプリング時間幅の分布の図。(a) と (c) はローカルキャリブレーション後、(b) と (d) はローカルキャリブレーションしたのちにグローバルキャリブレーションを行ったときの結果を示している。また、(a)、(b) の横軸はサンプリング時間幅を ns 単位で示している。そして、(c)、(d) の縦軸と横軸はそれぞれサンプリング時間幅と DRS4 のキャパシタ id である。図中では DRS 0 と pixel 6 と表記しているが、これらは同じ DRS4 チップ、ピクセル id である。

図 5.6 に示す。今回の測定では、サンプリング周波数  $f_{\text{sampling}}$  は 1.034 GHz と設定しているため理想的なサンプリング時間幅は 0.967 ns となる。図 5.6 の平均値を改めて見ると、ローカルキャリブレーションのみだと平均のサンプリング時間幅が 0.974 ns と理想値に比べて約 0.007 ns 大きくなっているがグローバルキャリブレーションまで行くと平均は理想値と一致しサンプリング時間幅の較正ができていることが確認できた。また、グローバルキャリブレーションまで行った時の標準偏差が 0.086 ns であることからサンプリング時間幅にはおよそ 9% の幅で各キャパシタ固有のサンプリング時間幅を持っていることが分かる。

次に、先ほど述べたサンプリング時間幅の較正が正しくできているかを確かめるために D. Stricker-Shaver et al. や 2017 年の野崎修論（京都大学）で行っている period time test を行った [31] [23]。period time test は、入力したサイン波に対して crossing point（波高値が 0 になる点）を探すことでサイン波 1 周期に含まれるキャパシタを求め（図 5.2 の左側）、これらのキャパシタに対応したサンプリング時間幅の足し上げによって周期 T を、ROI 内のサイン波周期毎、イベント毎に算出し、ヒストグラムから平均の周期を求める。今回は、先ほど data2 で較正したサンプリング時間幅を用いて、data2 とは別の時期に読み出し回路上で生成したサイン波を測定した data4 に



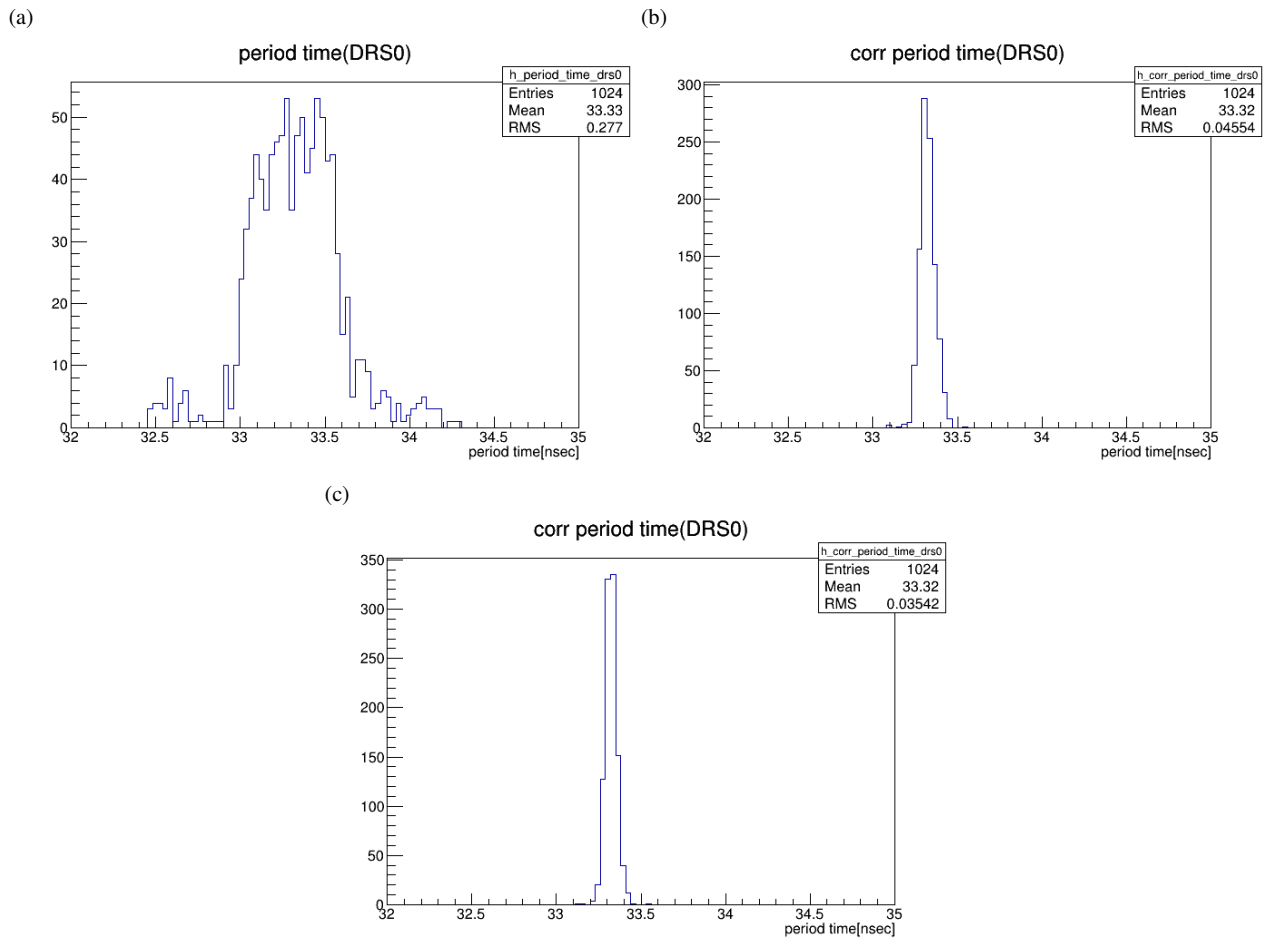


図 5.7 サンプルング時間幅の校正の有無でのサイン波周期の比較した図。各キャパシタごとに詰められたサイン波の周期の平均値の分布を示している。(a) は校正前、(b) はローカルキャリブレーション後、(c) はローカルキャリブレーションおよびグローバルキャリブレーションをした後の結果をそれぞれ示している。

対して **period time test** を行い、サンプルング時間幅の校正前、ローカルキャリブレーション後およびグローバルキャリブレーション後の結果を図 5.7 に示す。入力したサイン波の周波数は **30 MHz** であるので理想的な周期は **33.33 ns** であり、まず校正前 (図 5.11(a)) に関しては平均値は理想値と一致するが標準偏差が **0.28 ns** と揺らぎが大きいことが分かる。ローカルキャリブレーション後 (図 5.7(b)) に関しては、算出した周期の平均が理想値に比べて **0.01 ns** ほど小さくなっているが標準偏差は校正前に比べて  $\frac{1}{6}$  の **0.05 ns** まで抑えられている。グローバルキャリブレーション後 (図 5.7(c)) でも周期の平均は理想値より **0.01 ns** 小さくなっているが、標準偏差は **0.04 ns** とさらに改善し、数十 ps オーダーとなっていることは 2017 年の野崎修論 (京都大学) の結果とも整合していることが確認できた。

#### 5.4.2 テストパルスでの校正との比較

第 4 章では、LST1 カメラの全ピクセルに対して SCB で生成できるテストパルスを用いたサンプルング時間幅の校正によって電荷分解能が改善することを述べた。そして、前項までは LST2 号機以降の読み出し回路で生成できるサイン波を用いてサンプルング時間幅を算出し、**period time test** の結果から校正ができていることを確認した。そこで、今度はテストパルスを用いた校正も行い、サイン波を用いた方法との校正精度を比較し、5.1 項で述べた



表 5.3 較正精度の違いの検証に用いた測定データ

	波形	取得日	イベント数	用途
data1	サイン波	2021 年 1 月 14 日	$10^4$	period time の算出
data2	サイン波	2021 年 1 月 21 日	$10^4$	サンプリング時間幅の較正
data3	テストパルス	2021 年 1 月 26 日	$2 \times 10^6$	サンプリング時間幅の較正（高ゲイン系統）
data4	テストパルス	2021 年 1 月 26 日	$2 \times 10^6$	サンプリング時間幅の較正（低ゲイン系統）

ようにサイン波を用いた較正によって電荷分解能が 5% 以下に抑えることおよびテストパルスを用いた方法よりも較正精度が良いかを検証した。この検証で用いた測定データについては表 5.3 にまとめた。

テストパルスを用いた較正については、第 4 章で述べたように DRS4 の各キャパシタに一樣にテストパルスが入るようにおよそ 100 万イベント分を読み出し回路に入力し、各キャパシタでテストパルスを記録した回数で重みづけすることで各キャパシタのサンプリング時間幅に対応した補正係数を算出する。今回は、表 5.3 の data3 および data4 から補正係数の算出を行い、各系統の 1 ピクセルでの補正係数の分布を図 5.8 に示す。また、サイン波を用いた較正については 5.2 節で述べたようにローカルキャリブレーションの後にグローバルキャリブレーションを施すことで得られるサンプリング時間幅を算出する。表 5.3 の data2 を用いて算出した。

初めに、各方法で求めたサンプリング時間幅および補正係数を用いてテストパルスの電荷の算出を行い電荷分解能の比較を行った。電荷の算出には、補正係数の算出にも用いた表 5.3 の data3 と data4 を用いており、電荷は第 4 章と同じようにキャパシタ id に対する波高値を  $adc_i$ 、補正係数を  $w_i$  としたとき  $\sum adc_i \times w_i$  のように計算した。また、サイン波から求めたサンプリング時間幅  $\Delta t_i$  は、テストパルスから求めた補正係数  $w_i$  と以下の関係があるため補正係数に換算したうえで電荷計算に用いた。

$$w_i = \frac{\Delta t_i + \Delta t_{i+1}}{2} \quad (5.8)$$

高ゲイン系統について、サンプリング時間幅の較正前および各方法で較正した後の電荷を 1000 イベント分のヒストグラムを図 5.9 に示す。各ヒストグラムに対してガウス分布（式（4.6））でフィッティングを行い、得られた平均値と標準偏差のパラメータの比を電荷分解能とすると、較正前は 3.3%、テストパルスを用いた補正後で 2.4%、サイン波を用いた較正後で 2% となった。また、低ゲイン系統でも同様に 1000 イベント分の電荷を詰めたヒストグラムを図 5.10 に示す。低ゲイン系統での電荷分解能は、較正前で 3.1%、テストパルスおよびサイン波を用いた各較正ではどちらも 2.7% であった。この結果から、サイン波を用いたサンプリング時間幅の較正によって今回の検証の目標である電荷分解能を 5% 以下に抑え、さらにテストパルスを用いた方法と同等以上の精度で較正でき、使用するイベント数が 100 分の 1 まで削減できることにより測定した後すぐにサンプリング時間幅の較正ができることが確かめられた。

次に、各較正方法によって求めたサンプリング時間幅および補正係数を用いて 5.4 節で行った period time test によって時間の観点での比較も行った。この検証で用いたのは読み出し回路上で生成し測定したサイン波データである表 5.3 の data1 であり、period time を 1 万イベント分算出しヒストグラムにしたものを図 5.11 に示す。ここで、period time のヒストグラムの平均値と標準偏差の比を時間分解能とすると、較正前は 0.8%、テストパルスを用いた較正では 0.3%、サイン波を用いた較正では 0.07% となりサイン波を用いた較正が最も精度良く較正ができることが確かめられた。

## 5.5 議論

LST2 号機以降の波形読み出し回路に搭載された DRS4 のサンプリング時間幅較正用サイン波生成回路で生成したサイン波を用いてサンプリング時間幅の較正を行った結果、5.4.1 項で述べたように DRS4 の各キャパシタのサンプリング時間幅は、設定したサンプリング周波数に対する理想的な時間幅を中心に 9% 程度のばらつきをもつこ

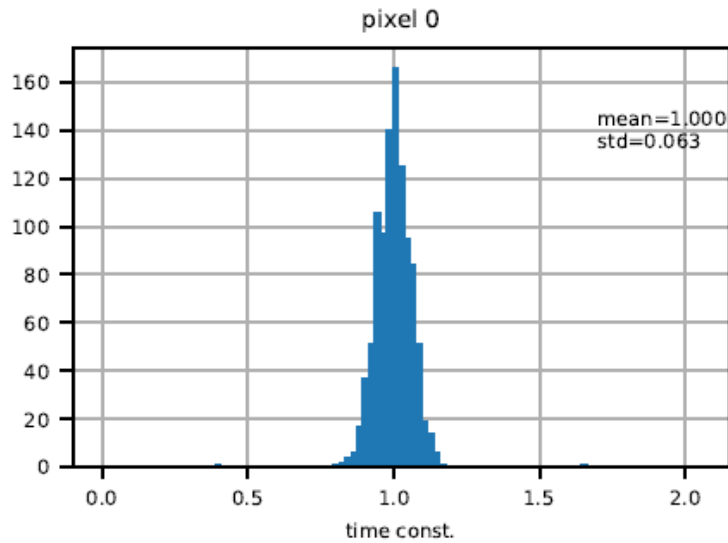


図 5.8 テストパルスを用いて算出したサンプリング時間幅の補正係数の分布図。

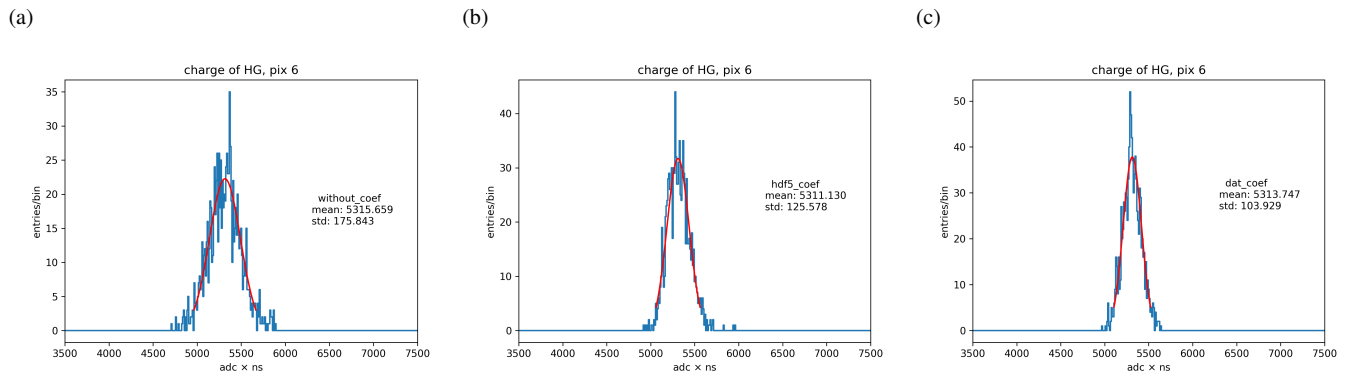


図 5.9 サンプリング時間幅の較正前およびテストパルス、サイン波を使った較正後の高ゲイン系統 1 ピクセルでの積分電荷のヒストグラム。横軸は電荷量を示している。(a) はサンプリング時間幅の較正前、(b) はテストパルスを用いた較正後、(c) はサイン波を用いた較正後の積分電荷を示している。図中の平均値および標準偏差はガウス分布でのフィッティングで得られたパラメータで、赤曲線はフィット関数である。

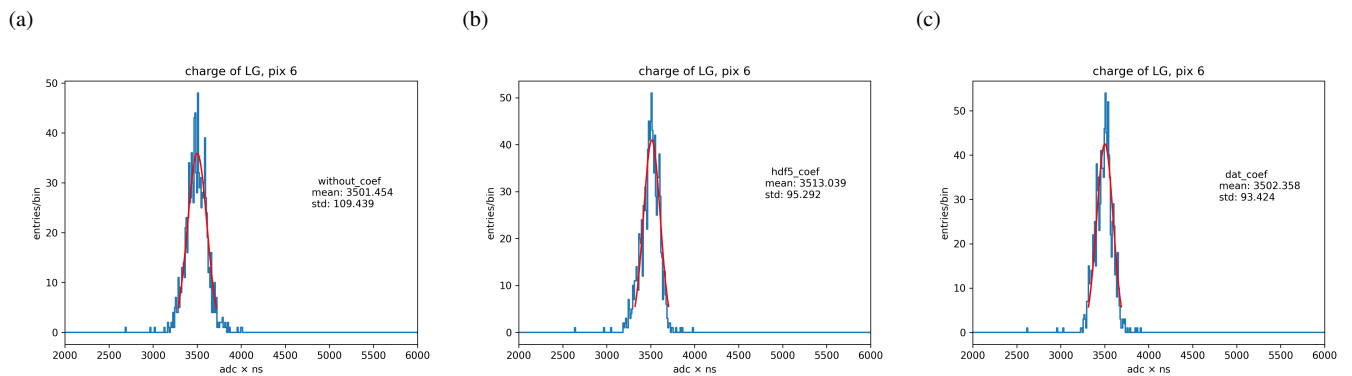


図 5.10 サンプリング時間幅の較正前およびテストパルス、サイン波を使った較正後の低ゲイン系統 1 ピクセルでの積分電荷のヒストグラム。横軸は電荷量を示している。(a) はサンプリング時間幅の較正前、(b) はテストパルスを用いた較正後、(c) はサイン波を用いた較正後の積分電荷を示している。図中の平均値および標準偏差はガウス分布でのフィッティングで得られたパラメータで、赤曲線はフィット関数である。

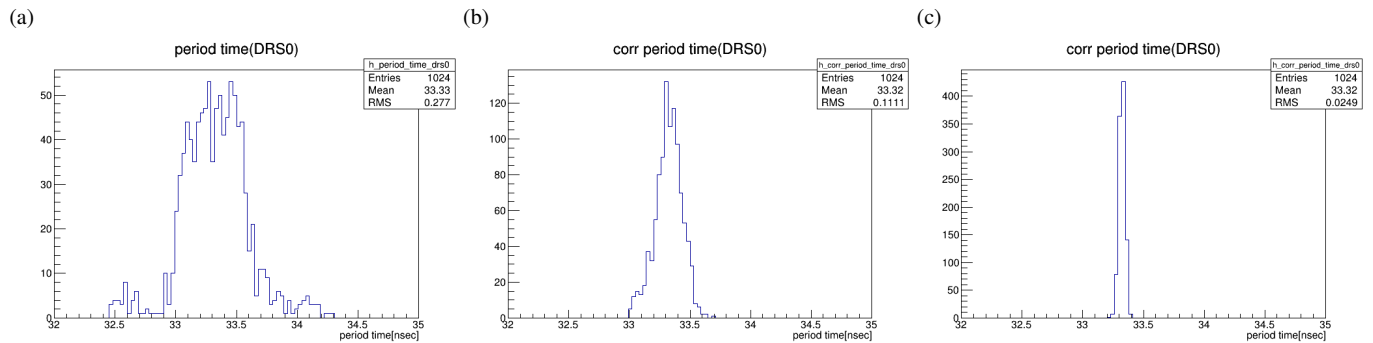


図 5.11 サンプル時間幅較正前およびテストパルス、サイン波を使った較正後のサイン波の周期の図。横軸はサイン波の周期 (period time) である、(a) は較正する前、(b) はテストパルスを用いて較正した後、(c) はサイン波を用いて較正した後のそれぞれの各キャパシタ毎の平均値のヒストグラムである。

とを確認できた。

テストパルスを用いた較正との比較では、サイン波を用いた方法によって各系統の電荷分解能を 5% 以下に抑え、さらにテストパルスを用いた方法よりも用いるイベント数を削減しつつ同等以上の精度で較正するという本研究の目標を達成できた。しかし、今回行ったサイン波を用いたサンプル時間幅の較正方法はまだ CTA の解析ツールに実装していないこと、ROI を LST1 と同じ 40 セル分のサイン波データで較正ができるかの検証ができていないこと、読み出し回路の電荷分解能が LST の性能に及ぼす影響についての調査ができていないことから、今後の課題と考える。



## 第 6 章

# 結論

本修士論文では、LST の波形読み出し回路に搭載したアナログメモリ DRS4 チップの特性の一つであるサンプリング時間幅の揺らぎの較正について述べた。まず、LST1 では SCB で生成できるテストパルスを用いて試験観測中の LST1 カメラの全 1855 ピクセルについてサンプリング時間幅の較正を行い、同様に測定したテストパルスの電荷を較正前と較正後とで電荷分解能の比較を行ったところ、高ゲイン系統ではカメラ平均で 7.3% から 3.7%、低ゲイン系統では 5.4% から 2.9% まで改善し、サンプリング時間幅の寄与を 5% 以下に抑えることができた。また、サンプリング時間幅の時間変化の調査から、電荷分解能が 5% を超えるまでには約 10 年かかることが分かった。

LST2 号機以降の読み出し回路では、新たに搭載したサイン波生成回路を用いて、測定したサイン波に対してフィッティングを行い、フィット関数と測定点との時間差から較正するローカルキャリブレーションおよびサイン波の周期ごとに波形サンプリングによる時間の伸縮を補正するグローバルキャリブレーションによってサンプリング時間幅の較正を行った。その後、period time test を行い入力サイン波の周期を数十 ps のオーダーで再現する結果が得られ較正ができていたことが確認できた。また、同じ 2 号機以降の読み出し回路で、LST1 で検証したテストパルスを用いた較正と前述のサイン波を用いた較正の較正精度を電荷分解能と時間分解能の 2 つの観点で比較した。その結果、電荷分解能ではテストパルスを用いた場合は高ゲイン系統で 2.4%、低ゲイン系統では 2.7%、サイン波を用いた場合は高ゲイン系統で 1.9%、低ゲイン系統で 2.7% となり目標性能である 5% 以下を満たし、テストパルスを用いた方法と同等以上の精度で較正できることさらには、較正に用いるイベント数を 100 分の 1 まで抑えられたことで測定後すぐに較正ができることが確認できた。また、時間分解能に関しては、テストパルスを用いた場合は 0.3% でサイン波を用いた場合では 0.07% となった。これらの結果から、サイン波を用いた方がサンプリング時間幅の較正の精度が良いことが確認できた。

今後の課題として、読み出し回路の電荷分解能が LST の性能にどのように影響するかの調査をしていく必要がある。また、LST2-4 用の読み出し回路に新たに搭載したサイン波生成回路を用いたサンプリング時間幅の較正方法については、LST1 と同じように ROI を 40 セルにした場合でも同等の較正精度が得られるかの検証および CTA の解析ツールへの実装も課題として挙げられる。



# 付録

## A テストパルスを使ったサンプリング時間幅較正のスク립ト

テストパルスを使ったサンプリング時間幅較正に用いた `python` スクリプトを以下に記す。なお、`ctapipe` および `cta-lstchain` に依存したスクリプトであり、現在も各解析ツールの開発が進められているため仕様変更が行われた際には修正が必要となるので注意が必要である。

CTA の解析ツールである `ctapipe` および `cta-lstchain` は `GitHub` 上で開発が行われており、以下のスクリプトは、2021 年 3 月時点で `cta-lstchain` へ `pull request` をしているところである<sup>\*1</sup>。現状としては、サンプリング時間幅のテーブル作成に 1 日以上かかってしまうため、計算時間が改善することが必要である。すると、4.4.1 項で述べたような問題があるのかを測定後すぐに確認でき、測定当日に再測定を行うことができる。

ソースコード 6.1 `lstchain/script/sampling_interval_creation.py`

```

"""
This script creates the sampling intervals for each DRS4 cell of all pixels
only high gain or low gain, and the result outputs as fits file.
Note:
1) The interval table creation is needed ~10^6 events (or more) of test pulse data.
2) This script subtracts pedestal and corrects the spikes & time-lapse before timing-calibration.
3) The sampling interval table should be re-created if the LST camera modules are changed.

This script was validated with ctapipe(v0.8.0), cta-lstchain(v0.6.3)
and ctapipe_io_lst(v0.5.3) in 2020/12/30.

Authors: Yuto Nogami (Ibaraki Univ.), Hideaki Katagri (Ibaraki Univ.)
Last update: January 18, 2021
"""

import os
import time
import h5py
import argparse
import numpy as np
import scipy.optimize as so
from ctapipe.io import event_source
from ctapipe.image import LocalPeakWindowSum
from ctapipe.version import get_version as ctapipe_ver
from lstchain import get_version as lstchain_ver
from lstchain.calib.camera_r0 import LSTR0Corrections
from traitlets.config.loader import Config

parser = argparse.ArgumentParser(
    usage='>python sampling_interval_creation.py --input_file_.../LST.1.1.Run02064.0000.
    fits.fz --output_file_sampling_interval_20201222.hdf5 --pedestal_file_
    pedestal_Run2065.fits --coefficient_file_sampling_interval_20201222.hdf5 --
    gain_selection_0 --num_events_1000000, --num_charges_1000',

```

<sup>\*1</sup> pull request の状況は <https://github.com/cta-observatory/cta-lstchain/pull/255>

```

description='This script creates DRS4 sampling interval for each capacitor of all pixels.
            We need ~10^6 test pulse events to correct the sampling interval.',
add_help=True,
)
# required arguments
parser.add_argument('-i', '--input_file', help='Input event file', action='store',
                    type=str, dest='path_to_data', required=True)
parser.add_argument('-o', '--output_file', help='Output file, file format is .fits',
                    action='store', type=str, dest='output_file', required=True)
parser.add_argument('-ped', '--pedestal_file', help='Pedestal file for the subtraction',
                    action='store', type=str, dest='pedestal_file', required=True)
parser.add_argument('-gain', '--gain_selection', help='Select gain to create. High-gain is 0,
            low-gain is 1',
                    action='store', type=int, choices=[0,1], dest='gain', required=True)
# optional arguments
parser.add_argument('-coef', '--coefficient_file', help='please add if the coefficient file was
            created and error pixels were existed.', action='store',
                    type=str, dest='coefficient_file')
parser.add_argument('-ne', '--num_events', help='Set the number of events you want to use. The
            default is "None", i.e. all events are used.',
                    action='store', type=str, default=None, dest='n_events')
parser.add_argument('-nc', '--num_charges', help='Set the number of events for charge
            calculation. Default: 1000',
                    action='store', type=int, default=1000, dest='n_charges')

args = parser.parse_args()

# set the number of events to calculate the sampling interval.
if args.n_events != None:
    args.n_events = int(args.n_events)
    print('{} events will be used to create new sampling interval.'.format(args.n_events))
else:
    print('All events will be used to create new sampling interval.')

# serach the event file for the sampling interval calculation.
# Example filename was as LST-1.1.Run00001.0000.fits.fz.
path, name = os.path.split(os.path.abspath(args.path_to_data))
if 'Run' in name:
    # ex) divide to "LST-1.1." and "0001.0000.fits.fz"
    stream, run = name.split('Run', 1)
    # ex) divide to "0001" and "0000.fits.fz"
    run_num, run_sub = run.split('.', 1)
    run_name = 'Run' + run_num
    run_sub = stream + run_name

# for log
command = 'python_sampling_interval_creation.py\' \
    + '_i_' + args.path_to_data \
    + '_o_' + args.output_file \
    + '_ped_' + args.pedestal_file \
    + '_gain_{ }'.format(args.gain) \
    + '_ne_' + str(args.n_events) \
    + '_nc_{ }'.format(args.n_charges)

if args.coefficient_file != None:
    command = command + '_coef_' + args.coefficient_file

method_name = 'counting_method'
date = time.strftime('%Y-%m-%d')

```



```

ls = os.listdir(path)
file_list = []
for file_name in ls:
    if run_num in file_name:
        full_name = os.path.join(path, file_name)
        file_list.append(full_name)

file_list = sorted(file_list)

# setting for reading and analyzing events
n_gain = 2 # number of gain channels
n_cell = 1024 # number of DRS4 capacitors per 1 domino ring.
n_pixels = 7 # number of pixels per 1 module.
n_module = 265 # number of modules.
roi = 40 # region of interest
num_pixels = n_pixels*n_module # number of all pixels for LST.
offset_adc = 400 # offset adc counts of LST1.
# fill the counts of test pulse peaks for each drs4 cell, each pixel.
peak_counts = np.zeros((num_pixels, n_cell), dtype=np.float32)
sampling_interval = np.zeros((n_gain, num_pixels, n_cell), dtype=np.float32)
# high gain
if args.gain == 0:
    # for drs chip id
    chan = 0
# low gain
elif args.gain == 1:
    # for drs chip id
    chan = 4

config = Config({
    "LSTR0Corrections": {
        "pedestal_path": args.pedestal_file,
        "offset": 0,
        "tel_id": 1,
        "r1_sample_start": None,
        "r1_sample_end": None
    }
})
lst_r0 = LSTR0Corrections(config=config)

# read events and counting test pulse peak for sampling interval creation.
print("-----sampling_interval_calculation_is_started-----")
# this flag controls the number of events to readout.
flag = False
# control the number of events to create the interval.
num_used_events = 0
start_time = time.time()
for filename in file_list:
    print(filename)
    with event_source(filename) as source:
        for event in source:
            tel_id = event.lst.tels_with_data[0]
            # convert first capacitor ids from 0-4095 to 0-1023
            fc_all = event.lst.tel[tel_id].evt.first_capacitor_id % 1024
            pid = event.lst.tel[tel_id].svc.pixel_ids
            # DRS4 correction
            lst_r0.calibrate(event)
            # The counts of first 3 cells & last 1 cell become strange.
            event.r1.tel[tel_id].waveform[args.gain, :, :3] = 0
            event.r1.tel[tel_id].waveform[args.gain, :, 39] = 0
            for n_mod in range(n_module):

```

```

    for n_pix in range(n_pixels):
        drs_chip_id = 8*n_mod+n_pix//2 + chan
        fc = fc_all[drs_chip_id]
        pix_id = pid[n_mod*7+n_pix]
        adc = event.r1.tel[tel_id].waveform[args.gain, pix_id]
        max_cell = adc.argmax() + fc
        true_max = max_cell
        # for the cell ids of pulse peak > 1023.
        if true_max >= n_cell:
            true_max = max_cell - n_cell
        peak_counts[pix_id, true_max] += 1

    num_used_events += 1
    # when event_id reach to number of events that user set, the counting process finishes.
    if args.n_events != None:
        if num_used_events == args.n_events:
            flag = True
            break

    if flag == True:
        break

# convert from the counts to the sampling intervals
    sampling_interval[args.gain] = (peak_counts / num_used_events) * n_cell

    end_time_1 = time.time()
    print('{}_events_are_used_to_obtain_the_sampling_interval.'.format(num_used_events))
    print('-----sampling_interval_calculation_was_finished-----')
    time1 = (end_time_1 - start_time) / 3600
    print('this_process_took{}_hours_to_calculate.'.format(time1))

    # validate the sampling interval coefficients
    print('-----Charge_calculation_is_started-----')
    num_loop = 0
    # number of events we calculate the charge.
    num_events = args.n_charges
    print('{}_events_were_used_to_obtain_the_charge_resolution'.format(num_events))
    # histogram setting
    num_bins = 1000
    pix_arr = np.arange(num_pixels)
    x_min = 0
    x_max = num_pixels
    y_min = 0
    y_max = 10000

    # charge histogram without the coefficients
    hist_without_coef = np.zeros((num_pixels, num_bins), dtype=np.int16)
    # with previous coefficients
    hist_old_coef = np.zeros((num_pixels, num_bins), dtype=np.int16)
    # with new coefficients
    hist_new_coef = np.zeros((num_pixels, num_bins), dtype=np.int16)

    # read old sampling interval constant
    if args.coefficient_file == None:
        old_coef = np.ones((num_pixels, n_cell), dtype=np.float32)
    else:
        with h5py.File(args.coefficient_file, 'r') as f:
            old_coef = f['sampling_interval_coefficient'][args.gain]

    print(old_coef)
    # charge calculation process

```

```

flag = False
for filename in file_list:
    print(filename)
    with event_source(filename) as source:
        integrator = LocalPeakWindowSum(source.subarray)
        for event in source:
            tel_id = event.lst.tels_with_data[0]
            # convert ids 0-4095 to 0-1023
            fc_all = event.lst.tel[tel_id].evt.first_capacitor_id%1024
            pid = event.lst.tel[tel_id].svc.pixel_ids
            # DRS4's properties correction process
            lst_r0.calibrate(event)
            # The counts of first 3 cells & last 1 cell become strange.
            event.r1.tel[tel_id].waveform[args.gain, :, :3] = 0
            event.r1.tel[tel_id].waveform[args.gain, :, 39] = 0
            raw_r1 = event.r1.tel[tel_id].waveform[args.gain]
            # fill the charge to histogram
            raw_charge = integrator(raw_r1, tel_id, args.gain)[0]
            count, xedge, yedge = np.histogram2d(pix_arr, raw_charge, bins=(num_pixels, num_bins), range=((
                x_min, num_pixels), (y_min, y_max)))
            hist_without_coef += np.int16(count)
            old_coef_r1 = event.r1.tel[tel_id].waveform[args.gain]
            # generate the independent array
            new_coef_r1 = event.r1.tel[tel_id].waveform[args.gain].copy()
            for n_mod in range(n_module):
                for n_pix in range(n_pixels):
                    drs_chip_id = 8*n_mod+n_pix//2 + chan
                    fc = fc_all[drs_chip_id]
                    pix_id = pid[n_mod*7+n_pix]
                    residual = n_cell - fc
                    # apply DRS4 sampling interval coefficients
                    # treatments for some capacitor ids of ROI over capacitor id=1023.
                    if residual < roi:
                        old_coef_r1[pix_id, :residual] = old_coef_r1[pix_id, :residual]*old_coef[pix_id, fc:]
                        old_coef_r1[pix_id, residual:] = old_coef_r1[pix_id, residual:]*old_coef[pix_id, :roi -
                            residual]

                        new_coef_r1[pix_id, :residual] = new_coef_r1[pix_id, :residual]*sampling_interval[args.
                            gain, pix_id, fc:]
                        new_coef_r1[pix_id, residual:] = new_coef_r1[pix_id, residual:]*sampling_interval[args.
                            gain, pix_id, :roi - residual]
                    else:
                        old_coef_r1[pix_id, :] = old_coef_r1[pix_id, :]*old_coef[pix_id, fc:fc+roi]
                        new_coef_r1[pix_id, :] = new_coef_r1[pix_id, :]*sampling_interval[args.gain, pix_id, fc:fc
                            +roi]

            # fill the charge to histogram
            old_coef_charge = integrator(old_coef_r1, tel_id, args.gain)[0]
            count, xedge, yedge = np.histogram2d(pix_arr, old_coef_charge, bins=(num_pixels, num_bins), range=((
                x_min, num_pixels), (y_min, y_max)))
            hist_old_coef += np.int16(count)

            new_coef_charge = integrator(new_coef_r1, tel_id, args.gain)[0]
            count, xedge, yedge = np.histogram2d(pix_arr, new_coef_charge, bins=(num_pixels, num_bins), range=((
                x_min, num_pixels), (y_min, y_max)))
            hist_new_coef += np.int16(count)

            num_loop += 1
            # when event_id reach to number of events that user set, the counting process finishes.
            if num_loop == num_events:
                flag = True

```

```

        break

    if flag == True:
        break

end_time_2 = time.time()
time2 = (end_time_2 - end_time_1) / 60
print(yedge)
print('{} events were used'.format(num_loop))
print('-----Charge calculation was finished-----')
print('this process took {} minutes'.format(time2))

print('-----sampling interval selection is started-----')

def gaus_fit(x, amp, mu, sigma):
    return amp * np.exp(-(x - mu)**2 / (2 * sigma**2))

# number of pixels considered to have the best coefficients of the three cases.
pixel_ratio = {'without_coef': 0, 'old_coef': 0, 'new_coef': 0}
# charge resolution
reso_without_coef = np.zeros(num_pixels)
reso_old_coef = np.zeros(num_pixels)
reso_new_coef = np.zeros(num_pixels)

y_axis = np.array([sum(yedge[i:i+2])/2 for i in range(num_bins)])
for pix in range(num_pixels):
    # estimate the max amplitude of histogram
    peak_index = np.argmax(hist_new_coef[pix])
    estimate_amp = hist_new_coef[pix, peak_index]
    # convert the amplitude of histogram to charge values array
    charge_arr = []
    for i, n_events in enumerate(hist_new_coef[pix]):
        for j in range(n_events):
            charge_arr.append(y_axis[i])

    estimate_mu = np.mean(charge_arr)
    estimate_sigma = np.std(charge_arr)
    # initial parameters for fitting
    param_init = np.array([estimate_amp, estimate_mu, estimate_sigma])
    # the case of pedestal event.
    if estimate_mu < offset_adc:
        print('pixel {} had pedestal events (lower charge than offset_adc(={}))'.format(pix,
            offset_adc))
        reso_without_coef[pix] = 1
        reso_old_coef[pix] = 1
        reso_new_coef[pix] = 1
        pixel_ratio['old_coef'] += 1
        sampling_interval[args.gain, pix, :] = old_coef[pix]
    # the normal case.
    else:
        try:
            popt, pcov = so.curve_fit(gaus_fit, y_axis, hist_without_coef[pix], p0=param_init)
            reso_without_coef[pix] = popt[2] / popt[1]
            popt, pcov = so.curve_fit(gaus_fit, y_axis, hist_old_coef[pix], p0=param_init)
            reso_old_coef[pix] = popt[2] / popt[1]
            popt, pcov = so.curve_fit(gaus_fit, y_axis, hist_new_coef[pix], p0=param_init)
            reso_new_coef[pix] = popt[2] / popt[1]

            # for comparison of charge resolution
            tmp_reso = {'without_coef': reso_without_coef[pix]}
            tmp_reso['old_coef'] = reso_old_coef[pix]

```

```

tmp_reso['new_coef'] = reso_new_coef[pix]

except (so.OptimizeWarning, RuntimeError, ZeroDivisionError) as err:
    print('pixel_{:}: '.format(pix), err)
    reso_without_coef[pix] = 1
    reso_old_coef[pix] = 1
    reso_new_coef[pix] = 1

    # for comparison of charge resolution
    tmp_reso = {'without_coef': reso_without_coef[pix]}
    tmp_reso['old_coef'] = reso_old_coef[pix]
    tmp_reso['new_coef'] = reso_new_coef[pix]

    # the coefficient selection using charge resolutions
    min_key = min(tmp_reso, key=tmp_reso.get)
    if min_key == 'new_coef':
        pixel_ratio[min_key] += 1
    elif min_key == 'old_coef':
        sampling_interval[args.gain, pix] = old_coef[pix]
        pixel_ratio[min_key] += 1
    elif min_key == 'without_coef':
        sampling_interval[args.gain, pix, :] = 1
        pixel_ratio[min_key] += 1
    print('pixel_{id:} was bad! '.format(pix))

end_time_3 = time.time()
time3 = (end_time_3 - end_time_2) / 60
print('this process took_{:}min. '.format(time3))
print('-----sampling_interval_selection_was_finished-----')

# set the key, high gain or low gain
if args.gain == 0:
    gain_key = 'hg'
elif args.gain == 1:
    gain_key = 'lg'
# output as hdf5 format.
with h5py.File(args.output_file, 'a') as f:
    # the case if the constants file was already created.
    if 'sampling_interval_coefficient' in f:
        dataset = f['sampling_interval_coefficient']
        dataset[args.gain] = sampling_interval[args.gain]
        # meta data
        dataset.attrs['creation_date'] = np.append(dataset.attrs['creation_date'], date)
        # add to existing attribute
        if gain_key + '_dataset' in dataset.attrs:
            dataset.attrs[gain_key + '_dataset'] = np.append(dataset.attrs[gain_key + '_dataset'], run_name)
        # create the new attribute for dataset
        else:
            dataset.attrs[gain_key + '_dataset'] = np.array([run_name], dtype=object)

    dataset.attrs['shell'] = np.append(dataset.attrs['shell'], command)
    dataset.attrs['num_events'] = np.append(dataset.attrs['num_events'], num_used_events)
    dataset.attrs[gain_key + '_num_pixels_wo_coef'] = pixel_ratio['without_coef']
    dataset.attrs[gain_key + '_num_pixels_old_coef'] = pixel_ratio['old_coef']
    dataset.attrs[gain_key + '_num_pixels_new_coef'] = pixel_ratio['new_coef']
    dataset.attrs[gain_key + '_charge_resolution_wo_coef'] = reso_without_coef
    dataset.attrs[gain_key + '_charge_resolution_old_coef'] = reso_old_coef
    dataset.attrs[gain_key + '_charge_resolution_new_coef'] = reso_new_coef

# for the first time

```

```

else:
    dataset = f.create_dataset('sampling_interval_coefficient', data=sampling_interval)
    # meta data
    dataset.attrs['creation_date'] = np.array([date], dtype=object)
    dataset.attrs[gain_key + '_dataset'] = np.array([run_name], dtype=object)
    dataset.attrs['shell'] = np.array([command], dtype=object)
    dataset.attrs['method'] = method_name
    # number of events to create sampling interval coefficients.
    dataset.attrs['num_events'] = np.array([num_used_events], dtype=np.int)
    # number of pixels selected as best coefficients of the three.
    dataset.attrs[gain_key + '_num_pixels_wo_coef'] = pixel_ratio['without_coef']
    dataset.attrs[gain_key + '_num_pixels_old_coef'] = pixel_ratio['old_coef']
    dataset.attrs[gain_key + '_num_pixels_new_coef'] = pixel_ratio['new_coef']
    dataset.attrs[gain_key + '_charge_resolution_wo_coef'] = reso_without_coef
    dataset.attrs[gain_key + '_charge_resolution_old_coef'] = reso_old_coef
    dataset.attrs[gain_key + '_charge_resolution_new_coef'] = reso_new_coef
    dataset.attrs['ctapipe_version'] = ctapipe_ver()
    dataset.attrs['lstchain_version'] = lstchain_ver()

total_time = (end_time_3 - start_time) / 3600
print('Total_time_is_{:}hours'.format(total_time))
print('Finished!!')
%c\end{verbatim}

```

---

## 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秀明准教授には学部4年次の研究室配属からの3年間、研究をする姿勢や研究内容の相談・議論、ミーティングでの報告の作法など多方面で手厚いご指導を賜りました。研究の相談や議論時には、私の理解力・説明力の至らなさに長時間付き合わせてしまうこともありましたが、納得するまでお付き合い頂いたこと感謝しております。また、学会発表やテネリフェへの出張の打診があったときに積極的になれなず中々決断できない私に、良い経験になるのではと後押しをしてくださったこと感謝しております。また、自分の至らなさに数えきれないほど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。それでも私のことを指導してくださったことには感謝してもしきれません、本当にありがとうございました。最後に、昼食に毎回誘っていただきありがとうございました。吉田龍生教授には、学内でのミーティングで進捗報告やゼミで多くの助言を頂いたこと、本当に感謝しております。また修論執筆時には心配していただいていたことを小原君から伝え聞きました、ご心配をおかけしました。柳田昭平名誉教授には、ゼミで多くの助言やご指導をいただきましたこと感謝申し上げます。

CTA-japan グループの皆様、特に AS/ODA グループおよび FPI/CAL/ELEC グループの皆様には、各定期ミーティングでの進捗報告に対するアドバイスや研究を進める上でのサポートなど多くのことをしていただき心から感謝しております。FPI/CAL、ELEC のメンバーの皆様、特に京都大学の窪秀利教授には研究の進捗報告をする場を与えてくださったこと、研究を進めるうえで数多くの助言を頂きましたこと、感謝申し上げます。AS/ODA グループの皆様、特に宇宙線研究所の吉越貴紀准教授、大石理子助教には、まだ正式に CTA メンバーでなかった学部4年次のから研究の進捗報告の場とさせて頂いたこと、報告内容に対して質問やご指摘を数多くしていただいたこと、CTA の解析等に関する情報共有および学外の CTA-japan メンバーとの交流する場として F2F meeting を開催してくださったことなど多岐にわたってお世話になりました、感謝しております。宇宙線研究所の齋藤隆之 助教には、2018 年度の学部4年次の方からお世話になりました。本研究をする上では欠かせない LST の読み出し回路周辺について多くのご指導を賜り、定期ミーティングでは進捗報告のたびに質問や助言、議論などを欠かさずして頂いたことには、心から感謝しております。修士論文に向けた研究内容や学会発表、テネリフェ出張の打診をしてくださったことにも感謝しております。また、ヨーロッパの方々とのミーティングでは、相手から質問やコメント等をいただいた際に英会話能力が乏しい私に代わって伝えていただくなどサポートいただきましたこと、感謝しております。京都大学の野崎誠也さんには、LST の読み出し回路周辺のことや LST の実データ解析の仕方、解析結果などを CTA-japan の各定期ミーティングやメール等を通じて数々の助言や議論、サポートなど多岐にわたってお世話になりました、本当に感謝しております。また、対面する機会が 2019 年度の F2F meeting での1度切りであったことがとても残念であり、コミュニケーション能力が低いですがそれでも対面でお話する機会があったらとミーティングやメールの文面から野崎さんの優しいような雰囲気を垣間見るたびに思っておりました。宇宙線研究所の小林志鳳さんには、LST の実データ解析の方法に関しての情報共有や定期ミーティングでは多くの助言を頂きました、本当にありがとうございました。宇宙線研究所およびテネリフェで行った LST2 号機以降の光検出器モジュールの組立て・品質管理では、宇宙線研究所の高橋光成 特任研究員、学術支援専門職員の猪目祐介さん、櫻井駿介さん、埼玉大学の砂田 裕志さん、京都大学の岡知彦さん、甲南大学の川村孔明君、塚本友祐君にはお世話に

なりました。特にテネリフェに関しては、初の海外出張ということもあり出張が決まったころは不安でいっぱいでしたが、高橋さんには日本出立時から一緒に行動をしてくださったので空港の乗り継ぎ等も無事できたと思っています。現地の生活では、高橋さん、櫻井さん、砂田さん、岡さん、川村君、塚本君、マックスプランク物理学研究所の技術職員である Antonios Dettlaff さんと多くの方に支えられました、皆さんのおかげで海外でも安心して日々を過ごすことができました、本当に感謝しております。

茨城大学では、高エネルギー宇宙物理グループの先輩方、同期および後輩の皆さんには大変お世話になりました。特に、同期で共に CTA に携わってきた小原光太郎君には日々の生活では元気を、研究では刺激を沢山受けました。また、小原君はとても活発で、研究室のみんなとの交流する場を沢山作ってくれたおかげで、人見知りな僕でも研究室のみんなと楽しく日々を送ることができました。君がいてくれたからこそ、僕はここまで研究を続けることができたと思っています。本当にありがとうございます、感謝しています。

両親には、大学院前期博士課程まで通わせて頂いたことや日々の衣食住を支えて下さったことなど、本当に感謝しています。今度は徐々に僕が支えられるように社会人として頑張っていきます。

最後に、改めて本研究を進めるにあたりお世話になりました皆様に心より感謝申し上げます。



## 引用文献

- [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] T. C. Weekes, et al., “Observation of TeV Gamma Rays from the Crab Nebula Using the Atmospheric Cerenkov Imaging Technique”, *The Astrophysical Journal*, **342**, 379 - 395, 1989.
- [6] <https://www.mpi-hd.mpg.de/hfm/HESS/>.
- [7] <https://magic.mpp.mpg.de/>.
- [8] <https://veritas.sao.arizona.edu/>.
- [9] 「Cherenkov, Telescope, Array, 計画書」, 2014.
- [10] A. Cucchiara, et al., “A PHOTOMETRIC REDSHIFT OF  $z \sim 9.4$  FOR GRB 090429B”, *The Astrophysical Journal*, **736**, 7, 2011.
- [11] 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.
- [12] M. Ajello, et al., “Fermi and Swift Observations of GRB 190114C: Tracing the Evolution of High-energy Emission from Prompt to Afterglow”, *The Astrophysical Journal*, **890**, 2020.
- [13] NASA website, <https://www.nasa.gov/feature/goddard/~nasas-swift-spots-its-thousandth-gamma-ray-burst/>.
- [14] F. Aharonian, et al., “An Exceptional Very High Energy Gamma-Ray Flare of PKS 2155-304”, *The Astrophysical Journal*, **664**, L71-L74, 2007.
- [15] Fermi gamma-ray space telescope website, <https://fermi.gsfc.nasa.gov/science/etev/agn/>.
- [16] A. A. Abdo, et al., *The Astrophysical Journal*, **736**, 131, 2011.
- [17] C. Grupen, 『宇宙素粒子物理学』, 丸善出版, 2012, (小早川恵三訳).
- [18] K. Bernlöhr, “Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim\_telarray”, *Astroparticle Physics*, **30**, 149 - 158, 2008.
- [19] M. de Naurois and D. Mazin, “Ground-based detectors in very-high-energy gamma-ray astronomy”, *Comptes Rendus Physique*, **16**, 610 - 627, 2015.
- [20] 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.
- [21] F. Aharonian, et al., “Observations of the Crab nebula with HESS”, *A&A*, **457**, 899-915, 2006.
- [22] cta-japan, 公式 HP, <http://www.cta-observatory.jp/overview.html>.

- [23] 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.
- [24] S. Masuda, et al., “Development of the photomultiplier tube readout system for the first Large-Sized Telescope of the Cherenkov Telescope Array”, in *Proceedings of The 34th International Cosmic Ray Conference — PoS(ICRC2015)* **236**, 1003, 2016.
- [25] S. Ritt, “Development of high speed waveform sampling ASICs”, in *National Symposium on Nuclear Instrumentation*, 2010.
- [26] J. Adam, et al., “The MEG detector for  $\mu^+ \rightarrow e^+ \gamma$  decay search”, *The European Physical Journal C*, **73**, 2365, 2013.
- [27] J. Aleksić, et al., “The major upgrade of the MAGIC telescopes, Part I: The hardware improvements and the commissioning of the system”, *Astroparticle Physics*, **72**, 61 - 75, 2016.
- [28] “DRS4 data sheet”, [https://www.psi.ch/sites/default/files/2020-08/DRS4\\_rev09\\_2.pdf](https://www.psi.ch/sites/default/files/2020-08/DRS4_rev09_2.pdf).
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 「Level B:CTA Sub-System Performance Requirements」, 2012.
- [35] “ctapipe”, CTA Consortium, <https://github.com/cta-observatory/ctapipe>.
- [36] “cta-lstchain”, CTA Consortium, <https://github.com/cta-observatory/cta-lstchain>.