

修士論文

ATLAS 実験における高速飛跡トリガーシステムの
環境依存性と処理時間の評価

早稲田大学 先進理工学研究科
物理学及応用物理学専攻
寄田研究室

千葉 英誉

2012 年 2 月 7 日

概要

スイス・ジュネーブ近郊にある LHC 加速器の中の 1 つである ATLAS 実験では多くの背景事象 (QCD 事象が主過程) の中から物理的に意味のある事象を取得するために 3 段階のトリガーシステムが実装されているが、今後の瞬間ルミノシティの増加によりシステムの処理能力の向上が必要となっている。その一つの手段として開発が行われているのが高速飛跡トリガーシステム (FTK) である。

FTK はシリコン内部飛跡検出器の全情報を使用して運動量が $1\text{GeV}/c$ 以上の飛跡をリアルタイムで算出する新規システムである。全領域の飛跡を再構成するため、後段トリガーの判断材料に良質な情報を渡すことができると期待されリアルタイムで現状の飛跡再構成オフラインアルゴリズムと同等の力を発揮すると考えられている。

本研究では、様々な環境において FTK の飛跡再構成能力をシミュレーションにて評価・検討を行い、システムの高い質と安定動作が可能か確認する。本研究の第一段階としてシミュレーションソフトの改良を行い動作確認を行った上で、現在の衝突エネルギーと瞬間ルミノシティにおけるモンテカルロデータを用いて FTK が再構成する飛跡情報とオフラインソフトウェアで再構成された飛跡情報との比較検討を行った。現在のデザインで事象再構成率はオフラインとの比で 80 % 以上であることが判明した。また、インパクトパラメータの分解能は $150\ \mu\text{m}$ 程度と b ジェットの同定に十分な値であり、オフライン並の分解能で測定可能である。一方、運動量の分解能は低運動量用域では FTK による分解能は 3 % 程度であり、ほぼオフライン並の分解能であるのに対し、高運動量領域では分解能がオフラインより 3 倍悪化することが判明した。

更に、実装後の 2020 年の最大瞬間ルミノシティを超えた高輝度下において、飛跡の再構成率が、ルミノシティの上昇につれて $69.6\pm0.1\%$ から $27.9\pm0.3\%$ に悪化しており、現状のシステムでは高ルミノシティ下において飛跡再構成率に問題があり改善の必要があるということが分かった。一方、処理時間の評価では高ルミノシティ下で得たい物理事象、ランダムな QCD 事象においても常に約 $40\mu\text{sec}$ 以下で遅延することなく動作可能であることが判明した。

目次

1	序論	6
2	LHC-ATLAS 実験	8
2.1	LHC	8
2.2	ATLAS 実験	10
2.2.1	内部飛跡検出器	10
2.2.2	カロリメータ	12
2.2.3	マグネットシステム	12
2.2.4	ミューオン検出器	13
2.2.5	ATLAS トリガーシステム	14
2.2.6	2011 年のトリガーメニューと蓄積データ量	16
3	高速飛跡トリガーシステム	20
3.1	序論	20
3.2	データフロー	22
3.3	システム概要	23
3.4	第 3 章の纏め	30
4	性能評価	31
4.1	シミュレーションソフトウェア	31
4.2	低ルミノシティにおける Track の評価	37
4.2.1	データセット	37
4.2.2	評価方法	38
4.2.3	結果と考察	39
4.2.4	低ルミノシティ評価の纏め	46
4.3	高ルミノシティにおける Track の評価	47
4.3.1	結果と考察	48
4.3.2	高ルミノシティ評価の纏め	52
4.4	処理時間の計測	53
4.4.1	計算方法	53
4.4.2	高ルミノシティ下における処理時間	55
5	今後の展望	59
5.1	性能評価, FTK 実装前の展望	59
5.2	FTK 実装後の展望	62

図目次	2
-----	---

6	纏め	64
付録 A	FTK シミュレーションの使用方法	65
付録 B	Simulation Code	72
謝辞		75

図目次

2.1	Large Hadron Collider 概略図	9
2.2	ATLAS 検出器概観	10
2.3	内部飛跡検出器概観図	11
2.4	Pixel 検出器外観図	11
2.5	カロリメータ全体図	12
2.6	電磁カロリメータのバレル部	12
2.7	マグネットシステムの全体像	13
2.8	ソレノイド磁石による積分磁場強度の η 分布	13
2.9	ミューオン検出器の全体像	13
2.10	トリガーシステムとデータフローの全体図	14
2.11	LVL1 トリガーのブロックダイアグラム	15
2.12	2011 年 4 月 30 日から 5 月 1 日において取得したイベントレート	18
2.13	Minimum Bias Trigger Scintillator	18
2.14	2011 年における積分ルミノシティ	19
2.15	2011 年度の瞬間ルミノシティの変化	19
2.16	2011 年のトリガーレートの推移	19
3.1	LHC における生成断面積	21
3.2	FTK を導入した時のトリガーシステムにおけるデータフローチャート	21
3.3	Dual Output 可能な HOLA(High-speed Optical Link for Atlas)	22
3.4	FTK のシステム全体像概略図	23
3.5	Processor Unit	23
3.6	FTK で使用する内部飛跡検出器のアライメント	24
3.7	SCT モジュール	24
3.8	Region 分けの模式図	25
3.9	FTK VME Crate	25
3.10	Input Mezzanine	25
3.11	Input Mezzanine 内の FPGA における機能	25
3.12	Input Mezzanine における SCT の Hit 情報のクラスター化	26

3.13	Data Formatter Board	26
3.14	Data Organizer の模式図	27
3.15	Associative Memory Board の全体図	28
3.16	1 つの Local Associative Memory Banks の図	28
3.17	SS サイズが大きい時の記憶パターンと Track の再構成効率	29
3.18	SS サイズが小さい時の記憶パターンと Track の再構成効率	29
4.1	ATLAS ソフトウェアチェーン	32
4.2	Standalone FTK ソフトウェアチェーン	32
4.3	ATLAS ソフトウェアと FTK シミュレーションソフトウェアの関連性	33
4.4	Athena での FTK ソフトウェアチェーン	33
4.5	FTK シミュレーションソフトのディレクトリ構造	33
4.6	Standalone でのパターン ID	34
4.7	Athena でのパターン ID	34
4.8	パターン ID の差分	35
4.9	Standalone での Track ID	35
4.10	Athena での Track ID	35
4.11	Track ID の差分	35
4.12	Athena でのインパクトパラメータ d_0	36
4.13	インパクトパラメータ d_0 の差分	36
4.14	VBF $H \rightarrow \tau\tau \rightarrow lh$ のダイアグラム	37
4.15	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における ΔR 分布	39
4.16	MinimumBias 事象における ΔR 分布	39
4.17	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインの再構成率	39
4.18	MinimumBias 事象における FTK と オフラインの再構成率	39
4.19	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における再構 成率のオフラインと FTK の比	40
4.20	MinimumBias 事象における再構成率 のオフラインと FTK の比	40
4.21	各横運動量における FTK Track ごとのミスヒットの層数	40
4.22	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインのインパクトパラメータ	42
4.23	MinimumBias 事象における FTK とオフラインのインパクトパラメータ	42
4.24	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインの横運動量分解能	43
4.25	MinimumBias 事象における FTK と オフラインの横運動量分解能	43

4.26	オフラインにおける各粒子ごとの横運動量分解能	44
4.27	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインの横運動量分解能	45
4.28	VBF 事象における横運動量 $6\text{GeV}/c$ 以下の FTK とオフラインの横運動量分解能	45
4.29	VBF 事象における横運動量 $6\text{GeV}/c$ 以上の FTK とオフラインの横運動量分解能	45
4.30	$3 \times 10^{34}[\text{cm}^{-2}\text{sec}^{-1}]$ における VBF 事象での $\Delta R < 0.02$ カットのみの再構成率	48
4.31	$3 \times 10^{34}[\text{cm}^{-2}\text{sec}^{-1}]$ における角運動量 ごとに ΔR でマッチした再構成 Track の平均数	48
4.32	$3 \times 10^{34}[\text{cm}^{-2}\text{sec}^{-1}]$ における VBF 事象の ΔR と横運動量カットでの再構成率	49
4.33	$1 \times 10^{34}[\text{cm}^{-2}\text{sec}^{-1}]$ における VBF 事象の ΔR と横運動量カットでの再構成率	49
4.34	低ルミノシティにおける VBF 事象の ΔR と横運動量カットでの再構成率	50
4.35	横運動量 $2\text{GeV}/c$ 以下での運動量分解能	50
4.36	横運動量 $2 \sim 6\text{GeV}/c$ での運動量分解能	50
4.37	横運動量 $6\text{GeV}/c$ 以上での運動量分解能	50
4.38	ルミノシティの違いによる FTK Track の違い	51
4.39	ヒットのクラスター化の概念図	52
4.40	データの入出力の概念図	54
4.41	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象の処理時間	57
4.42	DiJet 事象の処理時間	57
4.43	VBF $H \rightarrow \tau\tau \rightarrow lh$ における各事象ごとの各機能での処理時間	58
4.44	DiJet における各事象ごとの各機能での処理時間	58
5.1	Vertical Slice Test に使用する Baby FTK	59
5.2	EDRO(Event Dispatch and Read-Out) Board	59
5.3	Run183347 MinimumBias Trigger における FTK とオフライン Track 数	60
5.4	1Road に対する荷電粒子が信号を残した層の数	61
5.5	2011 年度におけるビームの x 方向の分布	61
5.6	2011 年度におけるビームの y 方向の分布	61
5.7	LVL2 における 4 ジェットトリガー とトリガーレート	62
5.8	LVL2, EF トリガーにおける飛跡再構 成率の立ち上がり	62
5.9	2.3fb^{-1} における 95 % Confidence Level の Cross Section Limit	63
5.10	湯川結合定数とフェルミオンの質量との関係	63

表目次

2.1	LHC における主なデザインパラメータ	9
2.2	2011 年 10 月におけるトリガーマニューとトリガーレート	17
4.1	7TeV の RDO モンテカルロデータセット	38
4.2	Super Strip サイズ	38
4.3	各事象におけるインパクトパラメータ分解能の二乗平均平方根	43
4.4	VBF 事象における横運動量分解能の二乗平均平方根	44
4.5	高ルミノシティのモンテカルロデータセット	47
4.6	各瞬間ルミノシティにおける再構成率	48
4.7	各瞬間ルミノシティにおける横運動量の二乗平均平方根	49
4.8	各機能における 1 データ辺りの処理能力	54
4.9	データセット	56
4.10	処理時間の平均値	56

1 序論

自然界には4つの基本となる力が存在するとされている。その4つの力とは、重力、強い力、弱い力、電磁力のことである。このうち重力以外の3つの力を標準模型と呼ばれる理論的枠組みで記述できることが素粒子物理学では知られている。標準模型では物質を構成する粒子としてクォークとレプトン、3つの力を媒介する粒子としてゲージボゾン、そして質量を与えるとされている粒子として Higgs 粒子が提唱されている。しかし、標準理論では力の統一におけるエネルギースケール的大幅な違いを説明することができない。そこで標準理論を拡張した理論として超対称性理論やテクニカラー模型など様々な統一理論が存在する。

しかし、標準理論で予測されている Higgs 粒子については未発見であり、その他の理論の検証もできていない状況である。素粒子物理学の世界では、実験による標準理論などの理論の検証が重要課題である。

これらの理論を検証するために建設されたのがスイス・ジュネーブ近郊の欧州原子核研究機構 (CERN) にある大型陽子衝突加速器 LHC (Large Hadron Collider) である。LHC は重心系衝突エネルギーが 14 TeV の世界最高エネルギーの加速器であり、LHC では衝突回数を 40 MHz, 1 秒辺りのデータ量である瞬間ルミノシティを $1 \times 10^{34} [\text{cm}^{-2} \text{sec}^{-1}]$ とデザインされている。

ATLAS 実験は LHC を用いた実験の1つであり、最高エネルギーの衝突事象を検出し様々な粒子の探索が可能である。高い瞬間ルミノシティ下では膨大な観測データを取得することになるが、観測量の多くが QCD 事象である。そこで ATLAS 実験では QCD 事象の中から物理的に意味のある b クォークや τ 粒子などの重要なデータをより効率良く取得する必要がある。

しかし、現在の事象選択トリガーシステムでは高い瞬間ルミノシティ下において処理能力の限界に到達することが予想される。この問題を抜本的に解決するのが 2014 年に実装予定である高速飛跡トリガーシステム, FastTracker (FTK) である。

FTK は ATCA や 9U VME Crate などハードウェアを用いて飛跡を再構成する新規システムとして現在開発が進んでいる。ハードウェアを用いることにより、PC ファームで計算を行う現状のトリガーシステムより短時間で処理することによりシリコン内部飛跡検出器の全情報を使用して、独自に飛跡を算出することが可能である。全領域の飛跡を再構成するため、後段トリガーの判断材料に良質な情報を渡すことができると期待され、リアルタイムで現状の飛跡再構成オフラインアルゴリズムと同等の力を発揮すると考えられている。

本論文では、環境や事象に依らず、現在の FTK がオフラインと比べどの程度の再構成能力をもつのか評価・検討することがシステムの更なる改善を行う上で重要であることに着目し、モンテカルロデータを用いて算出される飛跡を比較評価する。第3章で FTK の概要、動作について、4.1 節でシミュレーションソフトの概要と動作確認を行う。

このシミュレーションソフトを使用し、4.2 節で 2012 年度に FTK の一部を ATLAS 検出器に実装し、動作テストを行う予定であることから、現在の衝突エネルギー 7TeV、瞬間ルミノシティ $10^{32} \sim 10^{33} [\text{cm}^{-2} \text{sec}^{-1}]$ のモンテカルロデータを用いて FTK が再構成する飛跡の評価・検討について述べる。

更に FTK をシステムに導入することで、トリガーシステムにおいて τ 粒子を同定することで高ルミノシティ下の爆発的な量の QCD 事象から我々が求める観測データを取捨選択することが可能になると考え、本論文では Vector Boson Fusion (VBF) 過程による $H \rightarrow \tau\tau \rightarrow lh$ 事象の飛跡の評価を行う。これらの詳細な結果と考察を 4.3 節で述べる。

また、FTK の処理が飛跡を多く生む事象でも安定した動作が必要であることから、現状のデザインでの処理時間の見積もりを行う。この詳細は 4.4 節にて述べる。

最後に第5章では今後の展望として実装前の性能評価の必要事項と実装後の研究展望について述べ、第6章に纏めを記載する。

2 LHC-ATLAS 実験

2.1 LHC

LHC(Large Hadron Collider) は, CERN(欧州原子核研究機構) にある世界最大のエネルギーをもつ大型陽子陽子衝突型加速器である。LHC は地下 100m にある, 主リングの直径は 9km, 周長は約 27km にわたる超伝導磁石を用いたハドロンコライダーである。レプトンコライダーの欠点であるシンクロトロン放射によるエネルギー損失が少なく, 重心系衝突エネルギーは 7TeV+7TeV 衝突の 14TeV まで出せる。2010 年 3 月から現在まで重心系衝突エネルギー 7TeV で稼働している。

陽子ビームは 1.1×10^{11} 個ずつバンチ化され, 40MHz で衝突を行う。このビームの値とデータ量を繋ぐパラメータとして瞬間ルミノシティがある。LHC での瞬間ルミノシティはビームのパラメータのみで書くことができ,

$$L = \frac{N_b^2 n_b f_{rev} \gamma_r}{4\pi \epsilon_n \beta^*} F \quad (2.1)$$

と与えられる。ここで, N_b はビームバンチ内の陽子の数, n_b はバンチ数, f_r は回転周波数, γ_r はローレンツ因子, ϵ_b はビーム軸に垂直な平面でのビームの広がり (transverse emittance), β^* は衝突点での振幅の大きさ, F は衝突角度による幾何学的な減衰係数である。これらの値を代入することにより, デザインされている最大ルミノシティは約 $10^{34} [cm^{-2} \cdot sec^{-1}]$ となる。表 2.1 に LHC における主なパラメータを示す。また, 図 2.1 に LHC の概略図を示す。

LHC には 4 箇所の衝突点が存在し, それぞれに異なる検出器が設置されている。それぞれ研究目的が異なり, 高エネルギー陽子陽子衝突現象の研究として ATLAS(A Toroidal Lhc ApparatuS), CMS(the Compact Muon Solenoid), 重イオン衝突現象の研究として ALICE(A Large Ion Collider Experiment), B 中間子の物理に特化した LHCb がある。

表 2.1: LHC における主なデザインパラメータ

パラメータ	値
重心系エネルギー	7.0TeV
最高ルミノシティ	$3 \times 10^{34} cm^{-2} sec^{-1}$
バンチ間隔	25nsec
バンチ当りの陽子数	1.15×10^4 個
バンチの長さ	75mm
バンチ数	2808 個
バンチ衝突当りの陽子衝突数	23 回
衝突点のビーム半径	$16\mu m$
衝突角度	$\pm 142.5\mu m$

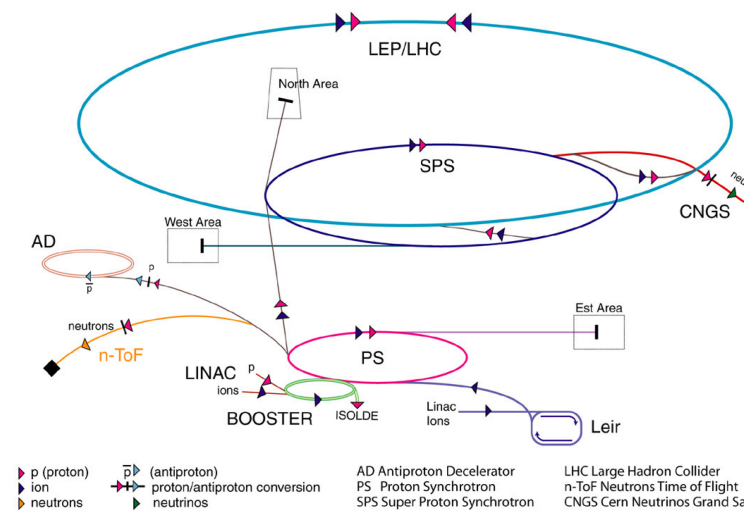
Accelerator chain of CERN (operating or approved project)

図 2.1: Large Hadron Collider 概略図

2.2 ATLAS 実験

ATLAS 実験は LHC の衝突点の 1 つに置かれた最も大きな複合型汎用検出器を用いて行われる実験であり、標準理論から要請される Higgs 粒子のみならず、様々な物理現象を観測するための実験である。図 2.2 に ATLAS 検出器の概観を示す。検出器は図 2.2 にあるように円筒形で、長さ 44m、高さ 22m、総重量は 7000 トンに及ぶ。検出器の内側から内部飛跡検出器、ソレノイド磁石、電磁カロリメータ、ハドロンカロリメータ、トロイダル磁石、ミューオン検出器で構成されている。以下の項でそれぞれの検出器を纏める。

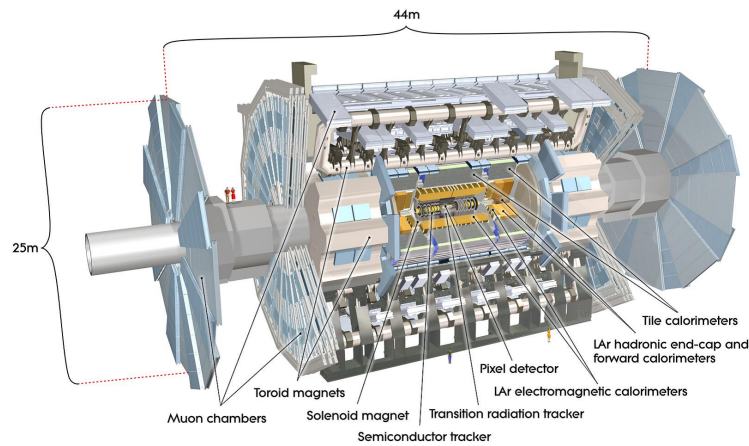


図 2.2: ATLAS 検出器概観

2.2.1 内部飛跡検出器

ATLAS 検出器の最内部にあり、粒子の飛跡を検出するのが内部飛跡検出器 (Inner Detector) である。内部飛跡検出器は直径 2.3m、高さ 5.6m の円筒形をしており、内側から順に、Pixel 検出器、SCT(半導体飛跡検出器)、TRT(遷移輻射飛跡検出器) の 3 つで構成されている。覆い被さる領域は $|\eta| = |-\log(\tan \frac{\theta}{2})| < 2.5$ であり、ソレノイド磁石によってつくられる磁場を用いて飛跡の曲率から粒子の運動量や電荷の符号、衝突点の位置の測定など荷電粒子の飛跡を再構成する。図 2.3 に内部飛跡検出器全体図を示す。

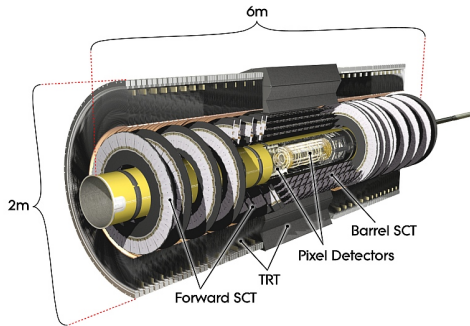


図 2.3: 内部飛跡検出器概観図

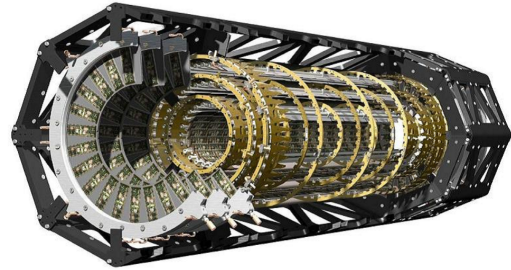


図 2.4: Pixel 検出器外観図

●Pixel 検出器

Pixel 検出器はビーム軸から最も近く、軸から 45.5mm ～ 242mm に 3 層ある半導体検出器である。図 2.4 がその全体図である。1 つの Pixel モジュールは $50\mu\text{m} \times 400\mu\text{m}$ または $50\mu\text{m} \times 600\mu\text{m}$ を 1 ピクセルとして 47278 ピクセルから構成されている。それぞれ 2 次元読み出しが可能となっており、3 層での全読み出し数は 8040 万チャンネルである。位置分解能は $R - \phi$ 方向で $10\mu\text{m}$, z 方向で $115\mu\text{m}$ という高い精度を持っており、衝突点の位置測定に使用される。

●SCT(Semi Conductor Tracker)

SCT はビーム軸から 255mm ～ 549mm の位置に Pixel に次いで内側に設置されている。SCT のモジュールは 2 枚のシリコンウエハを互いに 40mrad 傾けた構造をしている。この両面を荷電粒子が通過することで、シリコンウエハ上に $80\mu\text{m}$ 間隔で 768 本走っているストリップ電極に電気信号が流れるため、ストリップからの電気信号の情報によって粒子の通過位置を求めることが出来る。衝突点付近のバレル部には SCT のモジュールを円筒に沿って 4 層、前後方部ではモジュールを円盤状に並べたものが 9 層ずつ配置されており、これらから得る情報で軌跡を再構成することが出来る。SCT の位置分解能は $R - \phi$ 方向で $17\mu\text{m}$, z 方向で $580\mu\text{m}$ となっている。

●TRT(Transition Radiation Tracker)

TRT はビーム軸から 550mm ～ 1080mm 程度で、SCT の外側にある検出器である。この検出器では荷電粒子が通過する際の遷移放射で放出される X 線から荷電粒子の通過位置を読み取る。TRT はストロー状のチューブを重ね、その隙間をポリプロピレンファイバーで埋めた構成となっており、このファイバーで遷移放射が起きる。位置分解能は $R - \phi$ 方向のみで $130\mu\text{m}$ と Pixel や SCT より低い、TRT では層を多く配置しているため、連続的な飛跡検出から運動量の測定に特化している。

2.2.2 カロリメータ

ATLAS 検出器ではソレノイドマグネットの外側に電磁カロリメータとハドロンカロリメータの2つが設置されている。電磁カロリメータでは電子や γ 線、ハドロンカロリメータではクォークやグルーオンによるハドロン集合体であるジェットなどのエネルギーと角度の測定を行う。電磁カロリメータでは鉛の吸収体と放射線耐性に優れた液体アルゴンからなり、ATLAS のバレル、エンドキャップ領域を $|\eta| < 3.2$ まで担当し、電子と光子の同定に用いられる。一方、ハドロンカロリメータはバレル部が鉄の吸収体とシンチレータ、エンドキャップ部は銅の吸収体と液体アルゴン、放射線強度が高い前後方は銅とタングステンの吸収体と液体アルゴンで構成されており、ハドロン同定、エネルギー測定、ジェットの再構成などを行う。図 2.5 にカロリメータの全体図を、図 2.6 に電磁カロリメータのバレル部を示す。

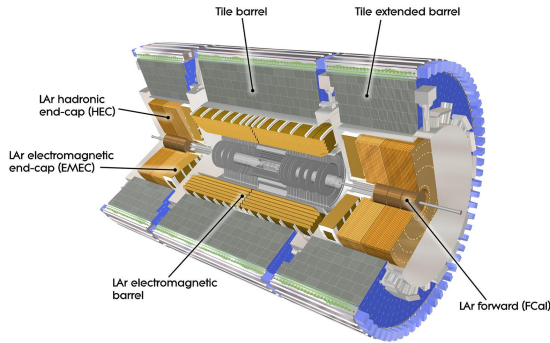


図 2.5: カロリメータ全体図

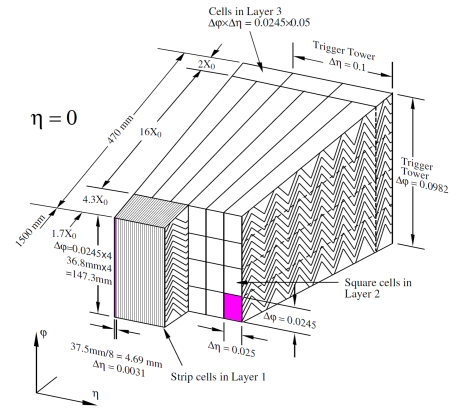


図 2.6: 電磁カロリメータのバレル部

2.2.3 マグネットシステム

ATLAS では内部飛跡検出器と電磁カロリメータの間にソレノイド磁石、カロリメータの外側にトロイド磁石という超伝導磁石がそれぞれ設置してある (図 2.7)。ソレノイド磁石はビーム軸方向に電流が流れるため、内部飛跡検出器に ϕ 方向の磁場を作る。一方、トロイド磁石はビーム軸対称に8つのコイルが設置されており、 ϕ 方向に最大で約 8T の磁場を作ることで飛跡を曲げることで、ミューオンの運動量測定に用いられる。図 2.8 はソレノイド磁石によってつくられる積分磁場強度の η 分布である。バレル部では平均で 0.5T、エンドキャップ部では 1.0T 程度となっており、 $1.4 < |\eta| < 1.6$ ではバレル部の磁石とエンドキャップ部の磁場が打消しあって弱くなっている。

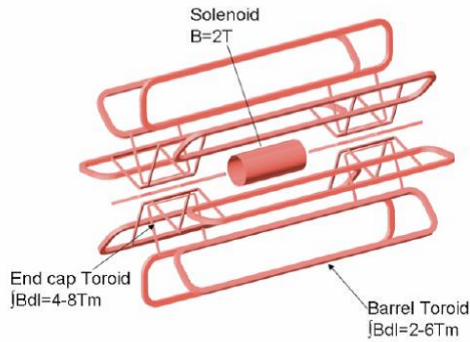
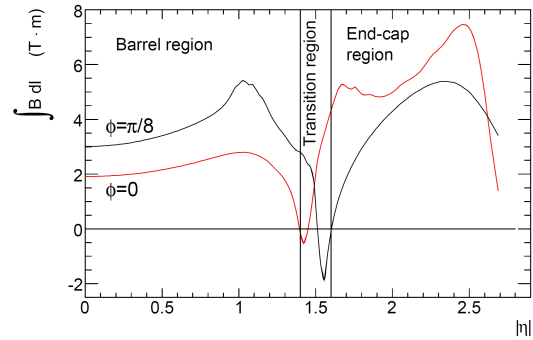


図 2.7: マグネットシステムの全体像

図 2.8: ソレノイド磁石による積分磁場強度の η 分布

2.2.4 ミューオン検出器

図 2.9 がミューオン検出器の全体像である。ミューオンは質量が電子の約 200 倍と大きいので、制動輻射を起こしにくい上に強い相互作用を起こさないため、物質の透過力が強い。このためミューオン検出器は一番外側に配置される。ミューオン検出器の構成は運動量、位置測定用の MDT(Monitored Drift Tube), CSC(Cathode Strip Chamber), トリガー用の RPC(Rasistive Plate Chamber), TGC(Thin Gap Chamber) の 4 種類から構成されている。

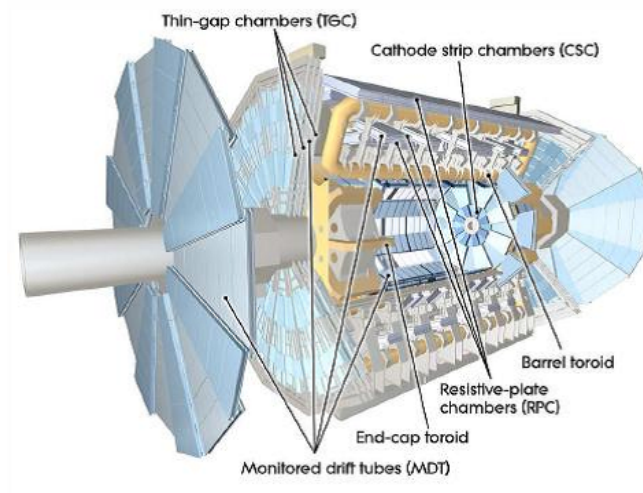


図 2.9: ミューオン検出器の全体像

2.2.5 ATLAS トリガーシステム

ATLAS では 40MHz の頻度でビーム衝突が起き、1 衝突辺りに約 23 個の事象が発生するので、事象生成数は約 1GHz になる。1 事象辺りのデータ量は約 1.5MByte と見積もられているため、1 秒間では 1.5PByte にもなり容量が非常に多い。従って全ての情報を記憶するのは現在の技術では困難である。そこで ATLAS ではリアルタイムで生成された事象を計算・解析を行うトリガーシステムを用いることで興味のある事象を選別し保存している。

ATLAS においてトリガーとデータ収集 (DAQ) システムは選別効率を上げるために LVL1 トリガーと LVL2 トリガー、及び EF(Event Filter) の順で 3 段階に分かれて行う。各段階ごとに前段階より解析精度の高い情報を加えることで精密な選別を行い、LVL1 トリガーでは 75kHz～100kHz に、LVL2 トリガーでは 3.5kHz まで情報量を削減する。最終段階の EF のアウトプットは 200Hz となり、データは CERN の計算機センターのストレージに保存される。図 2.10 にトリガーシステムとデータフローの全体像を示す。

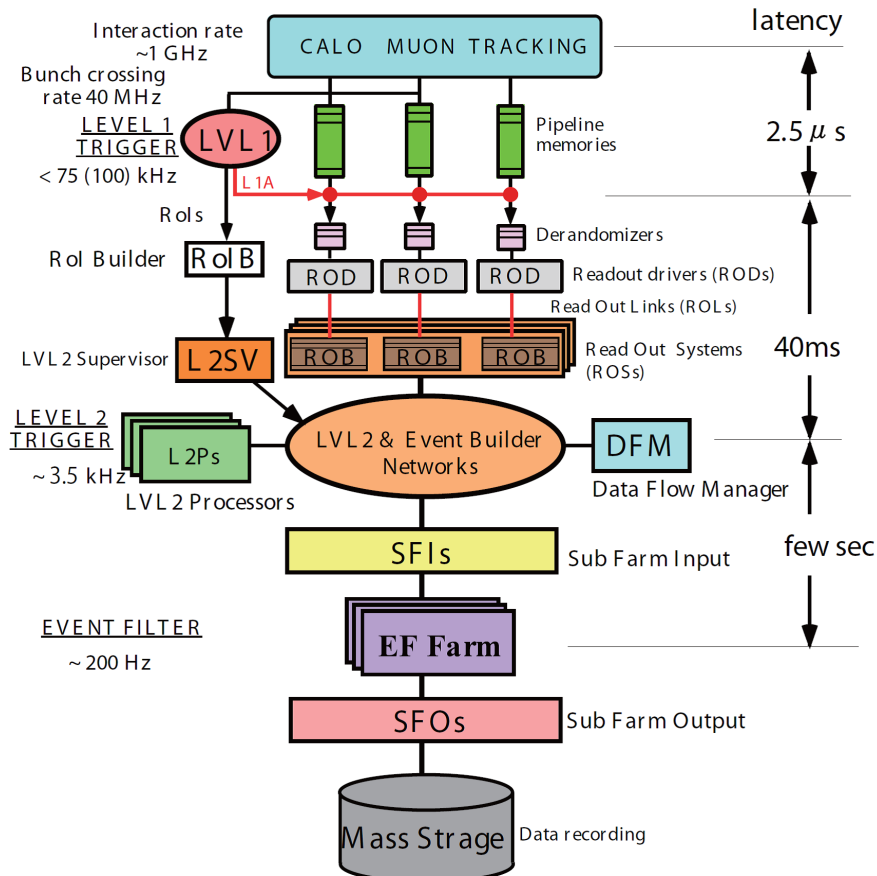


図 2.10: トリガーシステムとデータフローの全体図

●LVL1 トリガー

LVL1 トリガーではイベントレートを 40MHz から 100kHz まで落とす。一番始めのトリガーシステムある LVL1 トリガーは、カロリメータ検出器で得られる e , γ , ジェット等のシグナルと TGC, RPC で検出されるミューオン検出器からの情報を用いてバックグラウンドを排除していくものである。

図 2.11 が LVL1 トリガーにおけるデータフローである。まず最初に各検出器から得た全ての情報は、CTP(Central Trigger Processor) に送られる。この CTP で最終的な判定を行い、L1A(Level 1 Accept) と呼ばれるトリガー信号が TTC(Timing Trigger and Control system) を通じて ATLAS の各検出器のフロントエンドに運ばれる。L1A 信号は $2.5 \mu s$ 以内に発行されることが決まっているため、各検出器には L1A の信号を受け取るまでの間、全データを一時保持するパイプラインメモリーが実装されている。パイプラインメモリーは 128 段構造になっていて $25 ns \times 128 = 3.2 \mu s$ の間は全ての情報を保持できるようになっている。L1A 信号は $2.5 \mu s$ という非常に短い時間の間に発行されなければならないため、コンピュータのソフトウェアによる計算では間に合わないため、LVL1 トリガーは簡易な処理を全てハードウェア上の電気回路のみで行う。各検出器はこの L1A を受け取るとデータを後段のリードアウトラインに流し、ROD(Read Out Driver) で ATLAS 共通のフォーマットに纏め、ROS(Read Out System) へ送る。

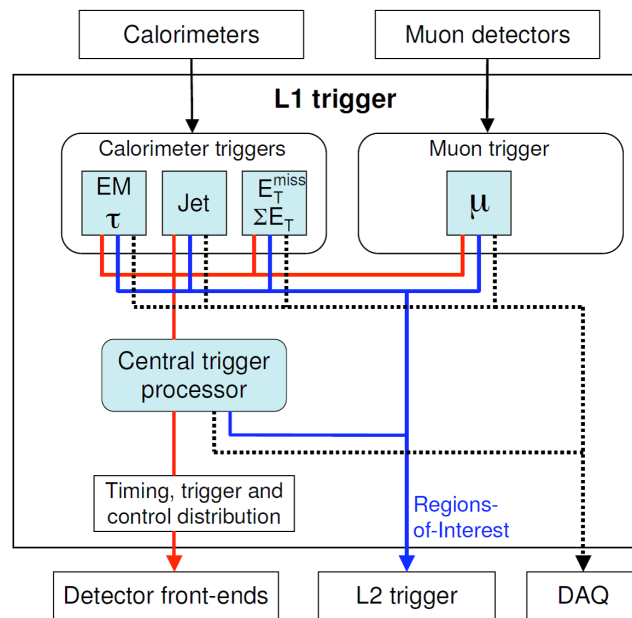


図 2.11: LVL1 トリガーのブロックダイアグラム

●LVL2 Trigger

LVL2 トリガーでは内部飛跡検出器とミュオン検出器の情報から荷電粒子の軌跡情報と LVL1 の情報に加えてカロリメータの詳細な情報を基に、ソフトウェアを用いてコンピュータ計算によって精度の高い、より物理的な情報でトリガー判断をする。但し、トリガーの判断に許容されている時間は 40ms となっているため、検出器の全領域をコンピュータ計算することができない。そこで、ATLAS では LVL1 トリガーで判定を通過した粒子が検出された領域およびその付近、ROI(Region Of Interest) の領域のみを LVL2 では判断する。

この ROI 情報は LVL1 トリガーの情報と共に L2SV(Level 2 SuperVisor) に移した後、L2P(Level 2 Processor) へ渡される。L2P は必要な検出器の情報を ROS(ReadOut System) から受け取り、この情報と LVL1 の情報を使用して判定を行う。これを通過した場合は L2A (Level 2 Accept) 信号を発行する。L2A 信号は L2SV を経由して DFM(Data Flow Manager) に渡され、DFM はデータを SFI (Sub Farm Input buffer) に移される。SFI は各 ROS から全検出器のデータを受信し、ATLAS フォーマットに従ってデータをまとめ、イベントを構築する。構築されたイベントは SFI のバッファに保持され EF の判断を待つ。

これらの作業により LVL2 トリガーはイベントレートを 100kHz から 3.5 kHz に落とすことができる。どのような時にトリガーするかどうかは、トリガーメニューとして決めることができる。

●Event Filter

EF では各検出器の全領域の完全な情報、トリガー条件を基に最終的なトリガー判断が行われる。LVL2 と同じくソフトウェアで再構成するため、LVL2 と合わせて HLT(High Level Trigger) と ATLAS では呼ばれる。この EF での段階では千数百台の PC を用いて、ほぼオフライン解析のような計算が行われる。選別されたイベント情報はオフライン解析のためにデータストレージに書き込まれる。この最終段階で貯蓄されるトリガーレートは 200Hz 程度まで絞り込まれる。すなわち 1 事象辺り 1.5MByte であるから 1 秒辺りでは約 300MByte の情報が記録されてゆくことになる。

2.2.6 2011 年のトリガーメニューと蓄積データ量

2011 年の実験では電子のような粒子が観測される事象を取得する Electron トリガー、ある運動量閾値を超えたジェット、 τ 粒子、あるいは消失横運動量と定義される Missing Et を観測した場合に取得する JetTauEtMis トリガー、ミュオンのような粒子を観測した場合に取得する Muon トリガーなどが使用された。表 2.2 に 2011 年 10 月に使用されたトリガーメニューを記載する。また、図 2.12 に、2011 年 4 月 30 日から 5 月 1 日にかけて行われた実験におけるイベントレートを記載する。表 2.2、図 2.12 から大部分は上記のトリガーでデータを取得していることが分かる。

また、物理的に意味のある事象の取捨選択を目標とするこれらのトリガーとは別に、衝突のタイミングのみ同期したイベントを取得するのが Minimum Bias トリガーである。Minimum Bias トリガーは、 $2.09 < |\eta| < 3.84$, $|Z| = 3560\text{mm}$ に存在する 2 枚のシンチレーションカウンタである MBTS(Minimum Bias Trigger Scintillator)(図 2.13) が少なくとも 1 つ動作した時にデータを取得するトリガーである。衝突したイベントをランダムに取得するため、取得する主な事象は QCD 事象である。本論文では、この特徴に着目して Minimum Bias トリガーで取得したデータセットを性能評価に使用した。詳しくは第 4 章で述べる。

表 2.2: 2011 年 10 月におけるトリガーメニューとトリガーレート

トリガーメニュー		
トリガー名	L1 トリガー [GeV]	EF トリガー [GeV]
Single Muon	11	18
Single Electron	16	22
2 Muon	11,11	15,10
2electron	10,10	12,12
2 taus	15,11	29,20
2 Photons	12,12	20
Single Jet+Met	50,35	75,55
Met	50	70
Multi-jets	10,10,10,10,10	30,30,30,30,30
トリガーレート		
トリガー名	L1 トリガーレート [kHz]	EF トリガーレート [Hz]
Single Muon	8	100
Single Electron	9	55
2 Muon	8	4
2 electron	2	1.3
2 taus	7.5	15
2 Photons	3.5	5
Single Jet+Met	0.8	18
Met	0.6	5
Multi-jets	0.2	9

トリガーメニューは 1 秒辺りのイベント数、すなわち瞬間ルミノシティによって異なってくる。図 2.14 は取得したデータ量、図 2.15 は 2011 年の瞬間ルミノシティの推移である。図から明らかに瞬間ルミノシティが増加しており、取得するデータ量が増加していることが分かる。これに応じ

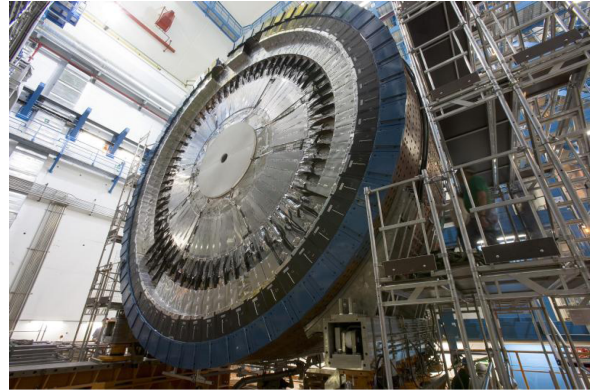
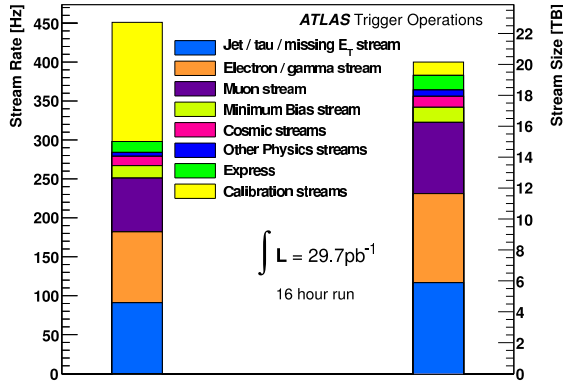


図 2.12: 2011 年 4 月 30 日から 5 月 1 日において取得したイベントレート

図 2.13: Minimum Bias Trigger Scintillator

て図 2.16 のように、先に述べたトリガーにおける取得量も 2011 年 7 月まで増加している。7 月以降はトリガーレートが一定値となるが、これはデータ保存の許容量に達したためである。しかし、7 月以降もルミノシティは増加しているため、あるトリガーにはイベントを数回に一度しか取得しない PreScale をかけることで取得するデータ量を制御している。例えば、2011 年 3 月では LVL1 で 1Jet の P_T が 75GeV/c 以上かつ LVL2 で 1Jet の P_T が 95GeV/c 以上かつ EF で 1Jet の P_T が 180GeV/c 以上の 1Jet トリガーには PreScale はないが、2011 年 10 月には同じトリガーマニューに PreScale がかけられている。

このように、情報量が多くなると事象の取捨選択をせざるを得なくなる。今後は瞬間ルミノシティが更に高まるのでこの効果は顕著に表れる。つまり、我々が解析できる新たな物理を含むデータがトリガーの時点で消去されている可能性も拭えない。そこで、高い瞬間ルミノシティ下におけるトリガー問題への対策として開発中であるのが次章で述べる本論文の主題である高速飛跡トリガーシステムである。

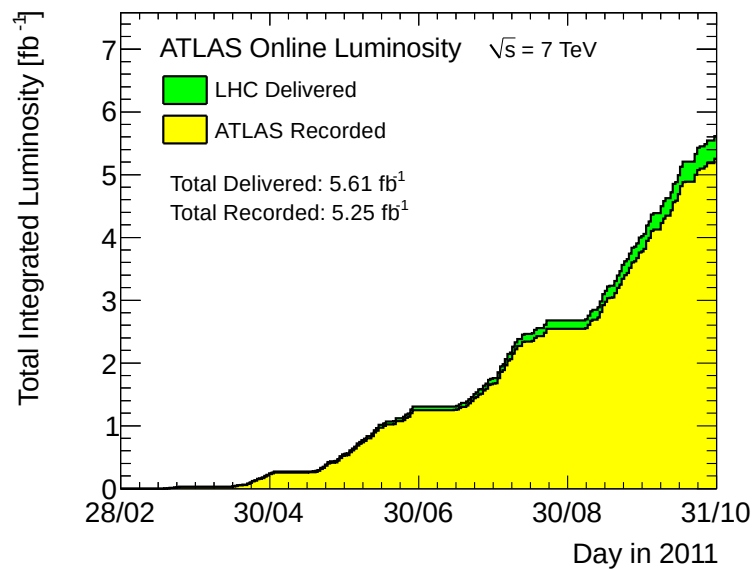


図 2.14: 2011 年における積分ルミノシティ

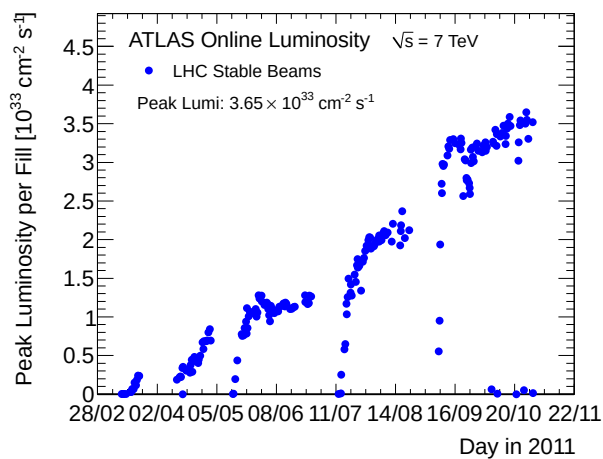


図 2.15: 2011 年度の瞬間ルミノシティの変化

現在最高ルミノシティは $3.65 \times 10^{33} [\text{cm}^{-2} \text{s}^{-1}]$

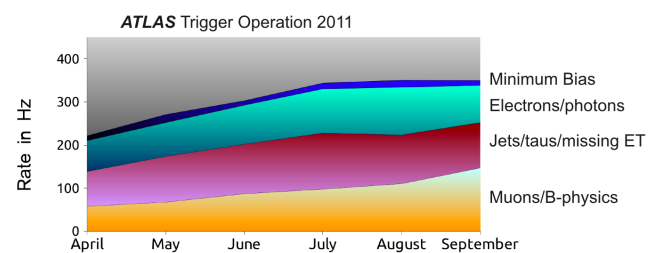


図 2.16: 2011 年のトリガーレートの推移

7 月からレートが頭打ちになっている

3 高速飛跡トリガーシステム

3.1 序論

この章では高速飛跡トリガーシステムの全体像について述べる。LHC で生じる事象は図 3.1 にあるように生成断面積が数十 mb^{-1} 程度である QCD 事象がほとんどであり、LHC が狙う物理目標の一つである Higgs 粒子などの新物理事象の生成断面積は数十 pb^{-1} 以下と 10^9 倍以上もの違いがある。そのため効率良く物理的に意味のある事象を取得する必要がある、トリガーシステムの充実化が重要である。

現在の ATLAS のトリガーシステムには3段階存在し、飛跡（今後は全て Track と表記する）は LVL2 のトリガーで判定が行われる。しかし、LVL2 で許容されている時間は 40ms と非常に時間が短いため、Track を再構成するのは LVL1 で通過した粒子が存在する領域、RoI の領域のみである。現在はこの領域のみを探索し、膨大な観測データを取捨選択をしている。しかし、瞬間ルミノシティの増大により RoI 領域が増え、LVL2 における飛跡再構成が困難となり LVL2 の事象取捨に支障をきたす恐れがある。

そこでシステム改善の一手段として内部飛跡検出器全体の情報を用いて、LVL2 で許容されている時間内で 1GeV/c 以上の全 Track 情報を計算・算出し、LVL2 トリガーにより良い物理情報を渡す役割を担うのが現在開発中である高速飛跡トリガーシステム、FTK(Fast TracKer) の導入が予定されている。

FTK では許容時間が限られているので、PC のようなソフトウェアではなくハードウェア処理として 9U VME Crate や ATCA などを用いて並列で電気信号処理をする FPGA(Field-Programmable Gate Array) 回路で組まれている。

ハードウェア処理を並行して行うことで、FTK 導入によって高ルミノシティ下において再構成時の処理時間の増大によりトリガー判断が不可能となる現象 (Dead Time という) をつくることなく、より一層安定したトリガー動作を行うことができるようになる。また全 Track 情報を用いるため、リアルタイムでの事象再構成の質が増加し、LVL2 トリガーにおいてより一層高度な物理情報を用いて事象選択をすることが可能になると期待される。

以下では FTK の詳細なハードウェアや動作について述べる。

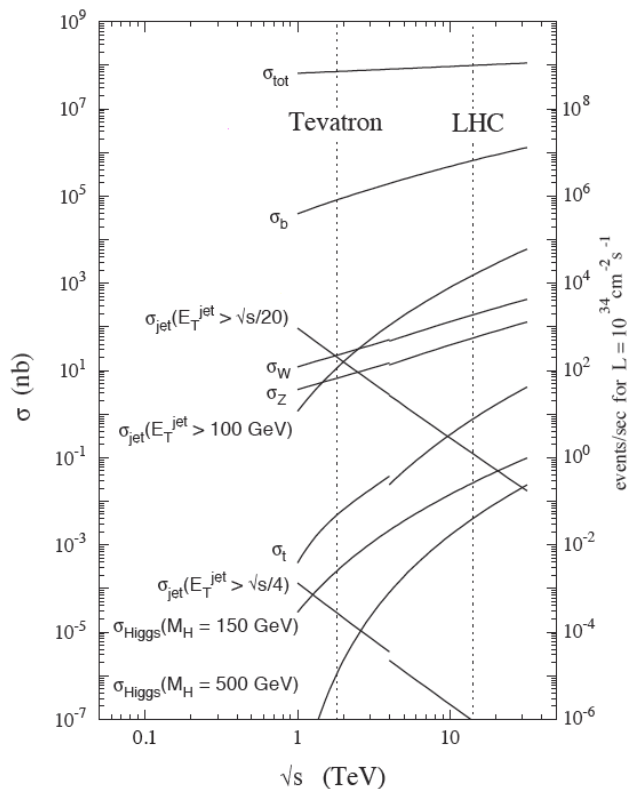


図 3.1: LHC における生成断面積

QCD 事象が 10^7 nb に対し, Higgs 粒子は 10^{-2} nb レベルなので
解析をするためにも事象の取捨選択時点から重要となる

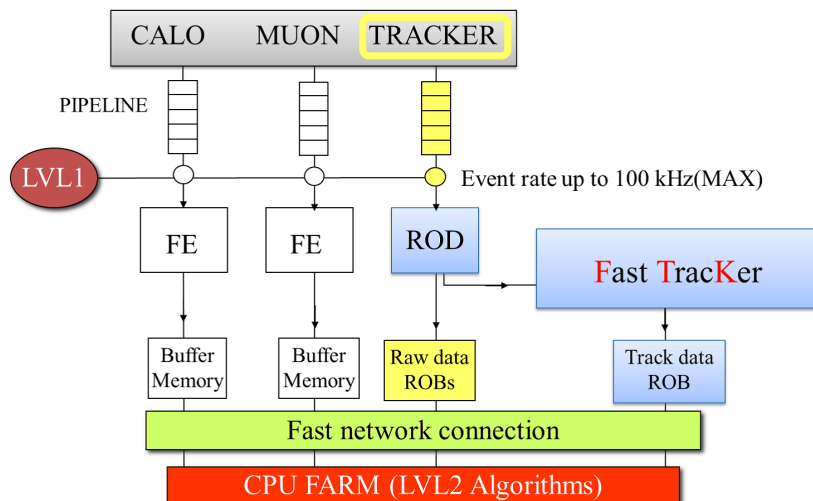


図 3.2: FTK を導入した時のトリガーシステムにおけるデータフローチャート
LVL1~LVL2 の間にシステムを導入予定

3.2 データフロー

図 3.2 に FTK が検出器から受け取る情報の場所と情報提供先を示すフローチャートを記載する。FTK はまず最初に Level1 Accept された後に内部飛跡検出器の全 Hit 情報を ROS から受け取る。この際、現行で使用しているデータの流を遮らないように、図 3.3 に示す ATLAS で使用されている HOLA(High-speed Optical Link for Atlas) を用いてコピーしたデータを FTK は受信する。HOLA は従来1つのアウトプットであったものを2つに改良することで最大 160Mbps のスピードでコピーデータを光転送することができ、LVL1 を通過した 100kHz の頻度で受信する情報を FTK に提供する。

Hit 情報を受信した FTK は内部飛跡検出器が担当する $|\eta| < 2.5$ に存在する xy 方向の運動量 1GeV/c 以上の Track を再構成し、それぞれの Track に対し、Track のパラメータを算出する。算出する Track のパラメータは、飛跡方向である ϕ , η , xy 方向の運動量 P_T , Primary Vertex である z_0 , Track の Impact Parameter である d_0 の5つである。

最後にこれらの情報を算出した後、1 事象辺り約 2.5kByte の大きさに LVL2 トリガーの PC ファームへ送られる。万一、FTK が算出できなかった場合は FTK はエラーとして処理されるが、本流のデータフローは変えていないので現行通り LVL2 トリガーで判断できる。

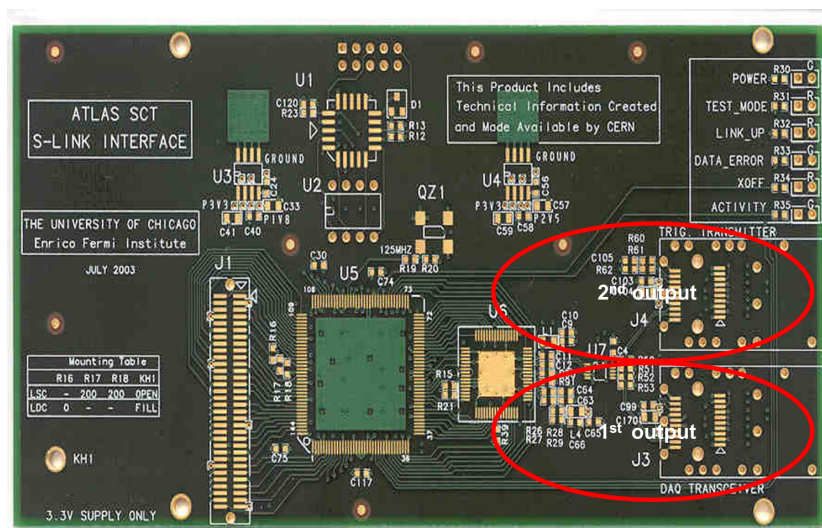


図 3.3: Dual Output 可能な HOLA(High-speed Optical Link for Atlas)
内部飛跡検出器の ROD に実装しコピーデータを FTK に送信する

3.3 システム概要

図 3.4 に FTK のシステム全体図を示す。FTK の動作は大きく分けて 2 段階存在する。1 段階目では、内部飛跡検出器のストリップ情報を幾つか統合し、図 3.5 にあるような AM(Addosiative Memory) とよばれる FPGA 回路に予め記憶させた Hit パターンと照らし合わせる。これにより粗い Track 情報, Road を算出する。2 段階目では、算出された Road 1 つ 1 つに対し、内部飛跡検出器のもつ分解能で Road 内の Hit 情報と予め算出した真の位置情報からの線形一次近似で Track のパラメータを算出する。

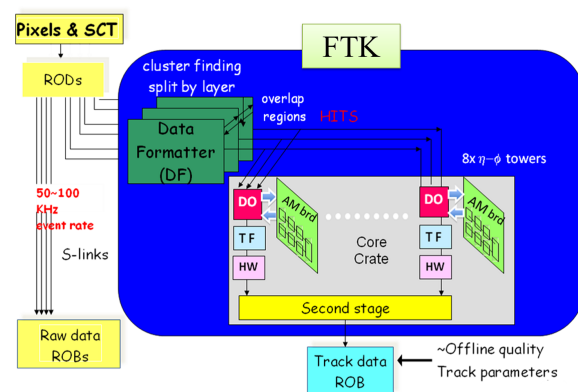


図 3.4: FTK のシステム全体像概略図

並列処理で高速化を目指す

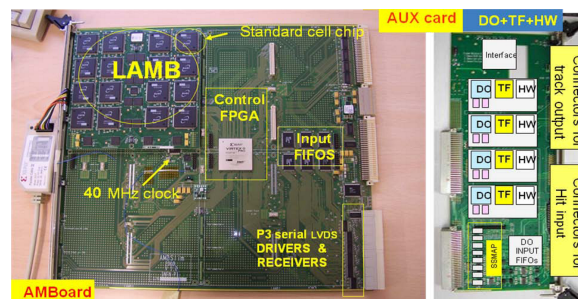


図 3.5: Processor Unit

Associative Memory と他の機能を備えた FPGA 回路が一つになっている

内部飛跡検出器は図 3.6 に示すように Pixel 検出器はバレル部で 3 層, SCT はバレル部で 4 層で構成されている。但し, SCT は図 3.7 のように 1 モジュールに 2 枚のシリコンがあるため, FTK では 8 層と数える。それぞれのモジュール数は, 全 Pixel モジュール数は 1744 個, 全 SCT モジュール数は 4088 個である。

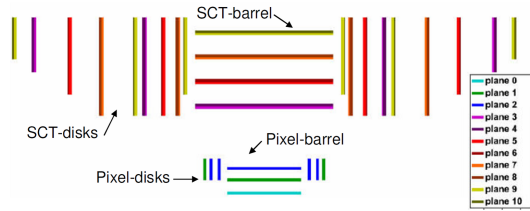


図 3.6: FTK で使用する内部飛跡検出器のアライメント

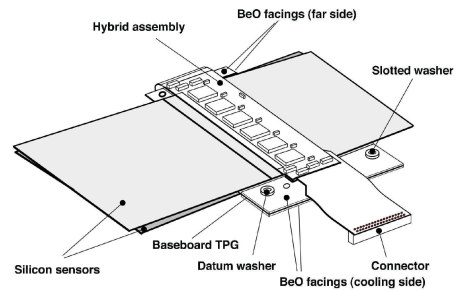


図 3.7: SCT モジュール

現在の科学技術では, これらの Hit 情報を図 3.5 の 1 つの Board で高速処理することは不可能である。そこで FTK では η 方向を 4 分割, ϕ 方向を 16 分割, つまり η - ϕ を 64 分割して, それぞれの領域 (SubRegion と呼び, η 方向を分割せず, ϕ 方向のみ 8 分割に集約したものを Region と呼ぶ) を 1 つずつ Board で並行処理することになっている。但し, ϕ 方向では $1\text{GeV}/c$ の粒子が約 10 度曲がることから, 図 3.8 にあるように 10 度分だけ領域を跨ぐのを許可し, $22.5 + 10$ 度を 1SubRegion として処理する。

また, 全 11 層の Hit 情報を使用すると処理時間の増加に繋がるため, 過去の研究から Track のパラメータが変化しない Pixel3 層, SCT4 層で Road, Track を算出する。その後, 残りの層を使用して Track のパラメータの質をチェックする。

以上のことから, FTK はいくつかの Board で並行処理を行うため, 図 3.9 にあるような VME Crate に挿入して使用される予定である。以下では, それぞれの信号を処理する機能について述べる。

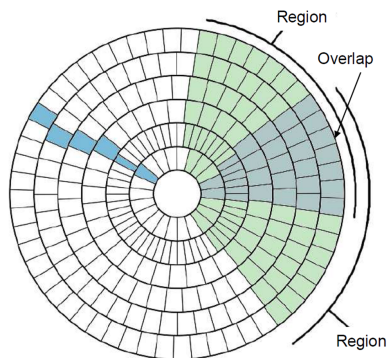


図 3.8: Region 分けの模式図
overlap 領域は 10 度分

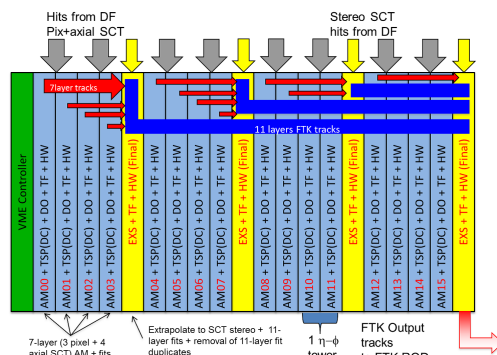


図 3.9: FTK VME Crate

●Data Formatter

DataFormatter では内部飛跡検出器から得た Hit 情報をクラスター化し下流に情報を分配する。ROS から情報を受信しクラスター化を行うのが Input Mezzanine と呼ばれる FPGA 回路を用いた装置である。図 3.10 に Input Mezzanine を、図 3.11 に Input Mezzanine 内の FPGA に記憶されている機能を示す。Input Mezzanine は S-Link と呼ばれる光通信を使用して、1Gbps の速度で受信をすることができる。次に Hit 情報を受信した後、FPGA 回路にてクラスター化される。クラスター化では図 3.12 にあるように、SCT では Hit のあるストリップ番号の中心値の 2 つの値と Hit したストリップ数 (クラスターサイズと呼ぶ) を出力する。ストリップ番号中心値の 2 つの値を出力するのは図 3.12 の右図のように、偶数個のストリップが Hit した場合に正しく中心値を判定するためである。

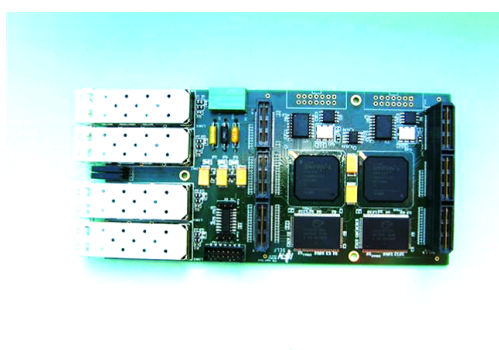


図 3.10: Input Mezzanine
Hit 情報を受信する

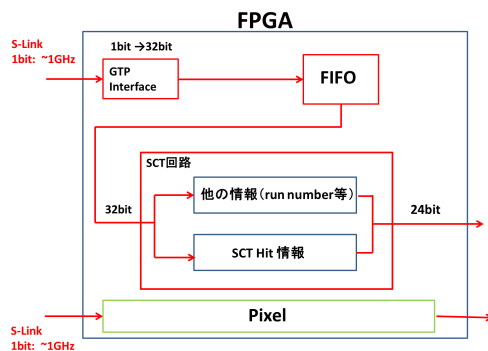


図 3.11: Input Mezzanine 内の FPGA における機能

FPGA で Hit 情報のクラスター化を行う

クラスター化した後, Data Formatter Board(図 3.13) に情報が流れる。Data Formatter Board では P3 コネクターと呼ばれる通信手段を用いて他の Board とやり取りし, クラスター化された情報のうち自身の η - ϕ SubRegion と近いクラスターが存在するか確認をする。これは ROD から得た η - ϕ モジュール情報と FTK が独自に分割して番号付している η - ϕ モジュール情報が異なるため, ROD の幾つかの η - ϕ モジュール情報を確認し, それぞれを FTK 側で使用する SubRegion 情報に直す必要があるからである。これらを確認をしたのち, 下流に SubRegion ごとに Hit 情報が送られる。

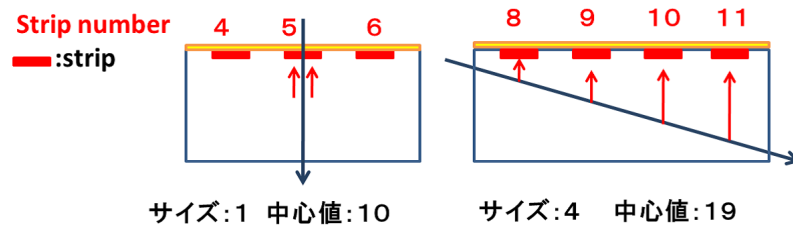


図 3.12: Input Mezzanine における SCT の Hit 情報のクラスター化

奇数では中心値の 2 倍値, 偶数時は 2 つの番号を出力する

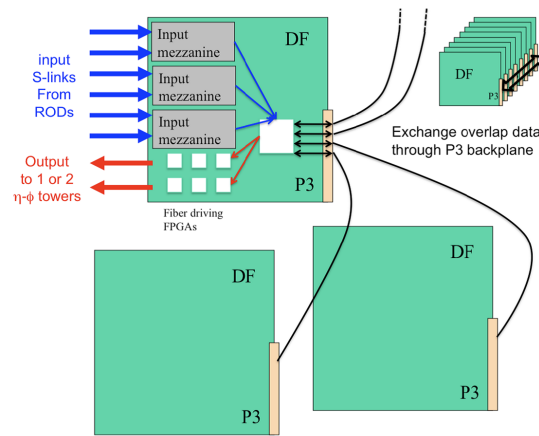


図 3.13: Data Formatter Board

●Data Organizer

Data Formatter から受信した Hit 情報は DO(Data Organizer) で束ねられる。これを FTK では Super Strip(SS) と呼び, 独自に番号が割り振られている。現在検証中の Super Strip サイズは, Pixel が ϕ 方向 24 ピクセル, z 方向 36 ピクセル, SCT のストリップ数 20 本である。この Super Strip のサイズが小さいほど, 記憶する必要がある Road のパターン数が多くなり処理時間が遅くなる。逆に Super Strip サイズを大きくすると, 1 つの Road 辺りに存在する Hit 数が増加し, Track の算出に時間がかかることや線形一次近似の適応領域が拡大するために Track の分解能が悪くなることが懸念される。従って, Super Strip のサイズは今後もシミュレーションや実機の

テストを用いて最適化しなければならない値である。

DO は Super Strip に束ねた後、下流の AM に Super Strip を送信し、Road が算出されるまで Hit 情報をバッファに一時記憶しておく。図 3.14 が DO の模式図で、現在では Pixel の最内層を 2 つの DO に、残り Pixel 2 層と SCT 4 層を 1 つの DO が並列で 100MHz というスピードで処理をする予定である。

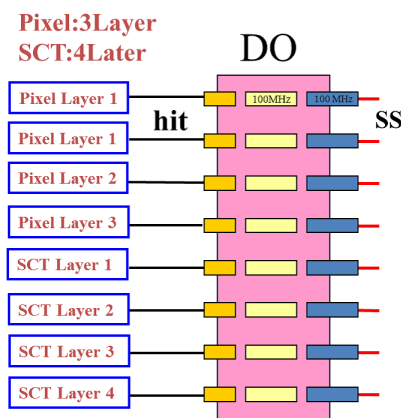


図 3.14: Data Organizer の模式図

Hit 情報を Super Strip 情報に変換する

●Associative Memory

AM(Associative Memory) は DO から受信した SS 情報とあらかじめ記憶されているパターンと照らし合わせ、照合したものを粗い Track, Road として算出する。図 3.15 に AM Board の全体図を示す。

1 つの AM Board につき図 3.16 にある LAMB(Local Associative Memory Banks) が 4 つ装着されている。LAMB には片面 16 個、両面に 32 個の FPGA 回路の AM チップが埋め込まれており、この回路の中に Hit パターンが記憶されている。

データの流れを述べると、DO から得た情報は AMBoard の中心にある FPGA 回路に集約された後、4 つの LAMB に同じデータを FIFO(Fast In Fast Out) で止めることなく送信する。LAMB では GIUE と呼ばれるチップが情報を受信し、それぞれパターンが記憶されている FPGA と SS の番号を参照する。同じものを発見した場合即座に AMBoard の中心にある FPGA に情報を送り、そのまま FIFO で再び DO に Road の情報が送られる。

LAMB は 1 つにつき 200MHz の処理能力をもつため、1 つの AM Board は 800MHz の処理能力をもつ。これは 100kHz で情報を取得するため、1 つの Board に対し 8000Road まで算出できることを意味する。これより多くの Road を算出する必要がある場合には、この FPGA 回路では処理仕切ることができないので、更に処理能力の高い FPGA 回路にする必要がある。

したがって、シミュレーションにおいて、あるパターンを記憶させた時に Road の算出数が処理時間を見積もる上で重要となる。

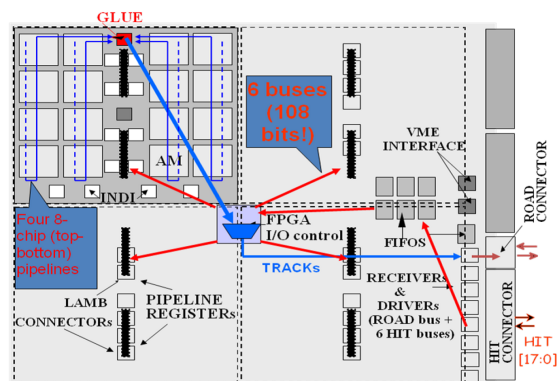


図 3.15: Associative Memory Board の全体図

4つの LAMB が敷き詰められ、それぞれ並列で処理する

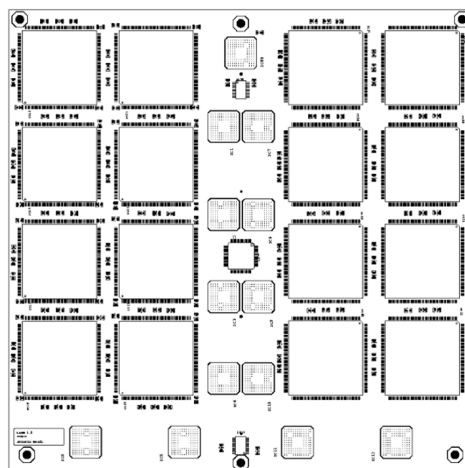


図 3.16: 1つの Local Associative Memory Banks の図

次に AM に記憶されるパターンについて述べる。記憶されているパターンは予め Track から算出しておく。現在では1つのイベントに1つのミューオンが飛跡するシングルミューオンのモンテカルロサンプルを使用してパターンを作成される。但し、実機を ATLAS に挿入する最終段階では、取得したリアルデータからパターンを作成する必要がある。

1つの AM に記憶できるパターンは現状では 5000 パターンである。すなわち、1Region では $5000 \text{ pattern} \times 32 \text{ AM Chips} \times 4 \text{ LAMB} \times 8 \text{ Processor Unit} = \text{約 } 500 \text{ 万パターン}$ を記憶することができる。将来的には AM チップを新しい FPGA 回路に改良し、現在の 16 倍の 80000 パターンまで記憶可能となる予定である。

一方、記憶されるべきパターン数は DO で述べた SS のサイズと Track の再構成効率に依存する。図 3.17 では SS のサイズを Pixel の ϕ 方向 22 ピクセル, z 方向 36 ピクセル, SCT のストリップ数 16 本, 図 3.18 は Pixel の ϕ 方向 11 ピクセル, z 方向 36 ピクセル, SCT のストリップ数 8 本にした場合に横軸は記憶させたパターン数, 縦軸が Track の再構成効率である。

まず第一に FTK は高い Track 再構成効率を持つ必要がある。そこで 90 % の点を仮定する。この点において、記憶させるべきパターン数は前者は 128 万パターン、後者は 1280 万パターンとなり後者は記憶容量の限界に達する。すなわち、SS のサイズを小さくするほどパターン数は増加し、AM で記憶できるパターン数を超えることになる。逆に SS のサイズを大きくすると、記憶パターン数は減るが、デメリットとして Track の再構成時に多くの Hit 情報を処理する必要があることや、領域の広さから一次近似の分解能が悪化することが挙げられる。したがって以上のことから、SS のサイズの大きさはパターン数, Track の再構成効率, Track パラメータの質, 処理時間などを考慮して最適化することが重要である。

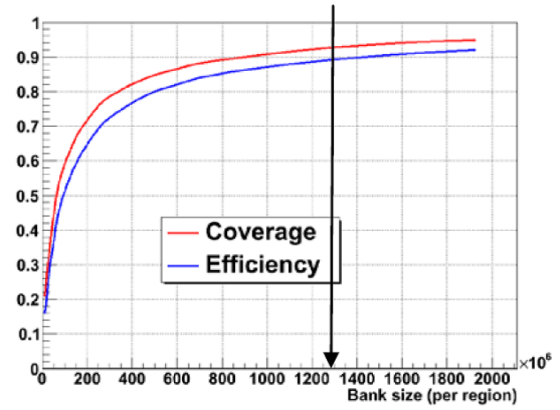
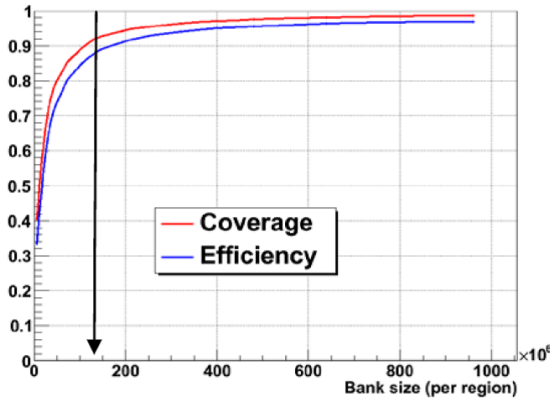


図 3.17: SS サイズが大きい時の記憶パターンと Track の再構成効率

図 3.18: SS サイズが小さい時の記憶パターンと Track の再構成効率

再構成率 90 % を要請すると 128 万パターン必要

再構成率 90 % を要請すると 1280 万パターン必要

●Track Fitter

DO から Road 情報と、その Road 部分に相当する Hit 情報を TF(Track Fitter) は FIFO で受け取る。TF では 1 つの Road 内にある Hit 情報を内部飛跡検出器の分解能で参照し、予め算出し記憶されている Track のパラメータから線形 1 次近似法 (Linear Approximation という) で算出する。

各層の観測した Hit 座標を x_j (j は層の番号), 観測する Track のパラメータを p_i (i はパラメータの数で FTK では 5 個), あらかじめモンテカルロデータから算出した Hit 情報を x_{0j} , パラメータ算出時の 1 次近似の係数を C_{ij} とすれば

$$p_i \simeq \sum_{j=1}^{10} C_{ij}(x_j - x_{0j}) = \sum_{j=1}^{10} (C_{ij}x_j - q_j) \quad (3.1)$$

と1次近似で表すことができる。この時の定数項 C_{ij} と q_j はモンテカルロの真の情報から共分散行列で算出することができるため、予め算出し TF に記憶させることでリアルタイムで Track のパラメータを瞬時に算出することが可能となる。

また、近似が粗悪なものは Track とすべきではないので、Track の質の評価として、検定量である χ^2 を次式で定義しリアルタイムで評価に使用する。

$$\chi^2 = \sum_{i=1}^5 \left(\sum_{j=1}^{10} C_{ij} x_j + q_j \right)^2 \quad (3.2)$$

Track パラメータを5個算出するので各変数の検定量を5つの成分まで和を取っている。この上記の式を用いて FTK の TF では検定量が20以下のものを Track として算出する。また、Track の再構成効率を上昇させるために Hit が1層ない場合でも、記憶されているパラメータから逆算することで位置情報を計算して、同様に χ^2 を算出することができる。したがって、7層中、最低でも6層の Hit 情報が存在すれば Track は構成できることになる。

また、SS サイズが大きすぎると1 Road 辺りの track が増加するため、処理できなくなる可能性があることに注意しなければならない。処理できる数は、1Board 辺り 4GHz なので1Road 辺り 40000Track まで算出することが可能である。

以上のように Track のパラメータを算出した後、Road 内で χ^2 が良いもののみを下流へ転送する。

●Hit Warrior

HW(Hit Warrior) では TF から受信した Track の Hit 情報を受け取り、TF で算出した Track のうち同じものはないかチェックし、同一のものは削除する。

この段階を終えたあとは、残りの SCT を追加し、再び Track の χ^2 のみを算出することで Track の質のチェックを行い FTK の動作は終了する。

3.4 第3章の纏め

以上のことから FTK の性能で重要なことは Track の再構成効率、Track のパラメータの質、ハードウェアとしての実行処理時間である。これらの値がルミノシティや事象に依らず安定して高い質を保つことが FTK の満たすべき条件である。またこれらの値は主に、FTK の SS サイズ、記憶するパターン数に依存する。したがって SS サイズとパターン数は常に最適化を行っていかなければならないことに注意する。今回は SS サイズとパターン数を固定し、第4章では FTK の動作をシミュレーションし、上記の値を検討・評価する。

4 性能評価

FTK の性能で重要なパラメータは Track の再構成効率, Track のパラメータの質, ハードウェアの実行処理時間である。これらを最適化した状態で, FTK はいかなる物理現象, 環境下でも一定以上のパフォーマンスを発揮しなければならない。これらを背景とし, 第4章では様々な環境下における FTK の動作をシミュレーションにて確認し, 性能を評価・検討する。

まず最初に今回性能評価に使用する FTK シミュレーションソフトウェアの改良を行った。詳細には 4.1 節にて述べる。次に現在の現在のルミノシティや将来到達することが予想される最高瞬間ルミノシティのモンテカルロデータを用いて, FTK が再構成する Track についてシミュレーションで詳しく調査した。詳しくは 4.2 節, 4.3 節で述べる。更に, 4.4 節において高ルミノシティ下における FTK の処理時間の計測を記載する。以下詳細に述べてゆく。

4.1 シミュレーションソフトウェア

ATLAS では保存された観測データ (バイトストリームデータ と呼ぶ) のビット信号から物理量に変換, 物理解析を Athena というオフラインソフトウェアを使用して行う。Athena は一般的なプログラム機能やコンポーネント間のやり取り, 更にはソースコードの共有するフレームワークである。プログラミング言語として C++, スクリプトとして Python が使われており, 各検出器や HLT などのグループごとにシステムやフレームワークで使われるソフトウェア (アプリケーションやライブラリなど) を開発している。このフレームワークにより各検出器で開発されたソフトウェアを相互に使用することが可能となり, 検出器のデータから物理解析用のデータを生成することが可能となる。

実際に Athena を用いたシミュレーションから解析までの流れを示したものが図 4.1 である。まず始めに粒子を生成する Generation の過程でモンテカルロ情報 (ATLAS では Truth 情報と呼ぶ) を生成する。その後, Geant4 により検出器動作のシミュレーションを行い, 検出器の情報を得る。このモンテカルロ情報は RDO(Raw Data Object) と呼ばれるファイルに詰められる。更に Reconstruction では, シミュレーション時は RDO ファイルから, 実データの観測では BS(バイトストリームデータ) から検出器の情報を基に飛跡の再構成を行い, ESD(Event Summary Data) や AOD(Analysis Object Data) を生成する。最後にこれらのデータオブジェクトから物理解析を行うのに適した情報のみを詰めた Ntuple を生成し, このファイル形式で解析を行う。



図 4.1: ATLAS ソフトウェアチェーン

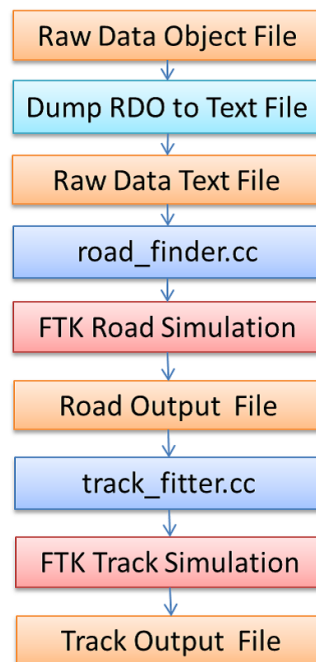


図 4.2: Standalone FTK ソフトウェアチェーン

FTK グループでは他のグループがアルゴリズムを開発するのと同様に動作シミュレーション用のアルゴリズムの開発が行われていた。しかし、今までは独自のシミュレーションパッケージ (Standalone と呼ぶ) として開発されていたため ATLAS の他の検出器情報と同時に使用することができなかった。図 4.2 が Standalone バージョンの FTK シミュレーションの一連の流れである。改良以前は独自のシミュレーションソフトのため、一旦 RDO ファイルから FTK に必要な内部飛跡検出器の信号情報をテキスト化していた。そのテキスト情報を基に AM が算出する Road 情報をシミュレーションし Ntuple 化、またこの Ntuple を用いて TrackFitter のシミュレーション結果を Ntuple に詰めていた。

しかし、FTK は 2014 年に実装予定であるためソフトウェアも ATLAS のフォーマットに合わせる必要があった。そこで図 4.3 のように Athena のソフトウェアチェーン上で FTK シミュレーションが動作できるよう改良開発を行った。Athena に組み込むことで、図 4.4 のようにテキスト化していた所を簡略化し、Reconstruction と同時にシミュレーションすることが可能となった。現在ではアウトプットとして既存の ESD ファイルと同時に FTK のファイルが生成されるようになっている。

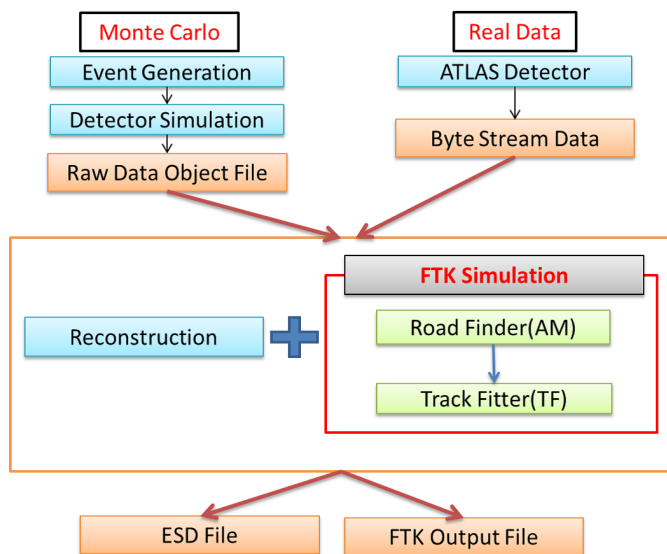


図 4.3: ATLAS ソフトウェアと FTK シミュレーションソフトウェアの関連性

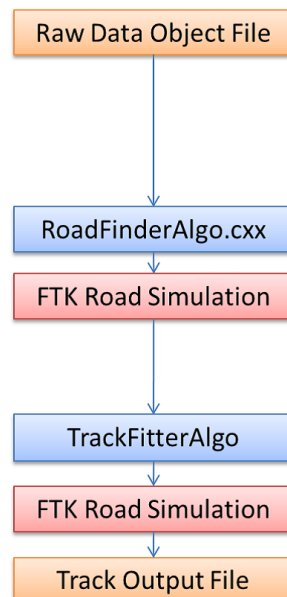


図 4.4: Athena での FTK ソフトウェアチェーン

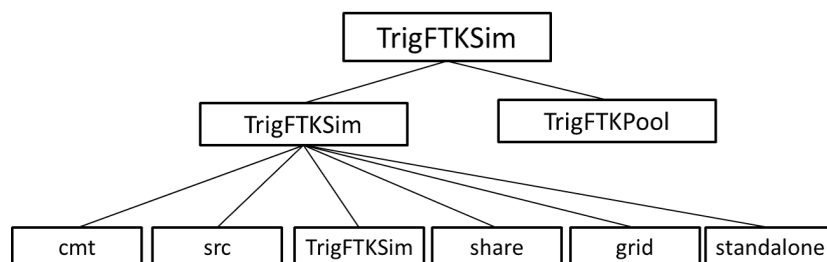


図 4.5: FTK シミュレーションソフトのディレクトリ構造

TrigFTKSim がシミュレーション本体

現在は開発段階であるためアウトプットファイルが2つに分割されているが、将来的には既存の ESD ファイルに組み込み必要な FTK の情報を選定して Ntuple 化するところまでアルゴリズムを開発する予定である。

今回改良を行った FTK シミュレーションソフトのディレクトリ構造を図 4.5 に示す。TrigFTK-Sim は FTK シミュレーションの核を成すパッケージである。それぞれのディレクトリの中身は以下となっている。

FTKSim パッケージ内のディレクトリ

- **cmt** : compile 用コード。requirement で他のソフトウェアを共有することができる。
- **src** : ソースコード。主なものは RoadFinder.cc と TrackFitter.cc
- **TrigFTKSim** : ヘッダーファイル用。
- **share** : Athena を使用して動かすための joboption と呼ばれるスクリプトが存在している。
- **grid** : Standalone のシミュレーションを Grid 上で行うためのスクリプト。
- **standalone** : Standalone 用のコード。

一方, TrigFTKPool は, FTK シミュレーションで得た情報ファイルを他の Athena 下のソフトウェアと共有する Athena 特有の情報ファイル形式に変更を行うアルゴリズムパッケージである。ディレクトリ構造は TrigFTKSim と同様に cmt, src, TrigFTKPool, share となっており他の Athena ソフトウェアと同じ形を取っている。

上記のように FTK シミュレーションを Athena に組み込んだことで, 今までと同等のシミュレーション結果が得られるか動作テストを行った。図 4.6~4.8 は AM のシミュレーションで算出された Road のパターン ID 番号を Standalone でのシミュレーションと Athena 上でのシミュレーションで求めたものと, その差分を取ったものである。同様に図 4.9~4.11 は TF のシミュレーションで算出された Track の ID 番号, 図 4.12~4.13 は Track のインパクトパラメータ d_0 を示したものである。

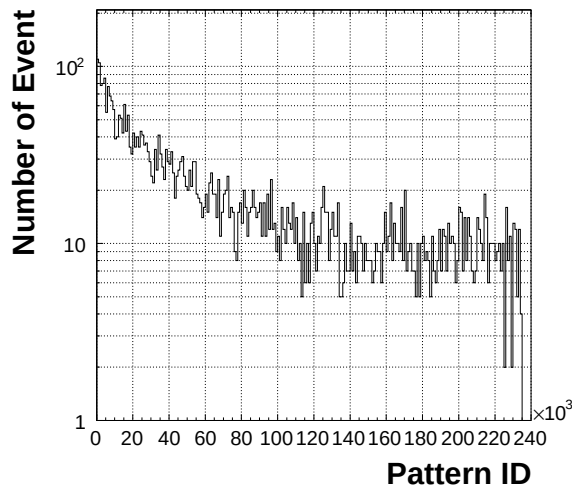


図 4.6: Standalone でのパターン ID

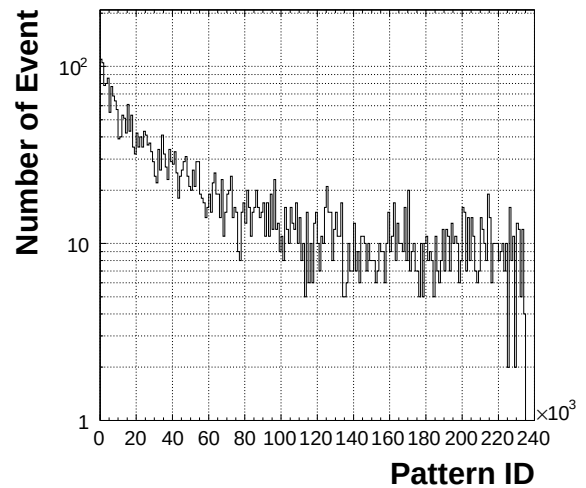


図 4.7: Athena でのパターン ID

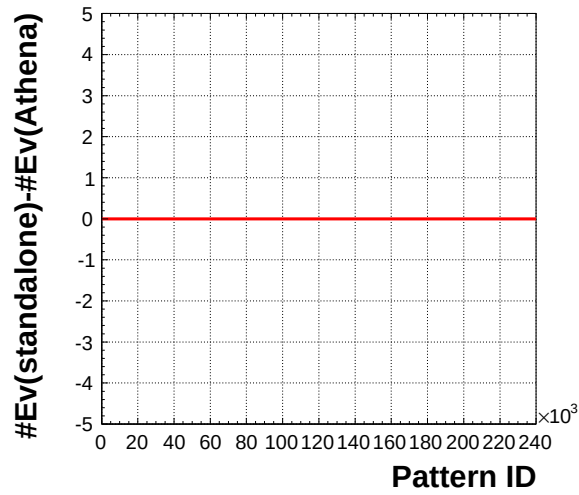


図 4.8: パターン ID の差分
差分が 0 なので完全に一致している

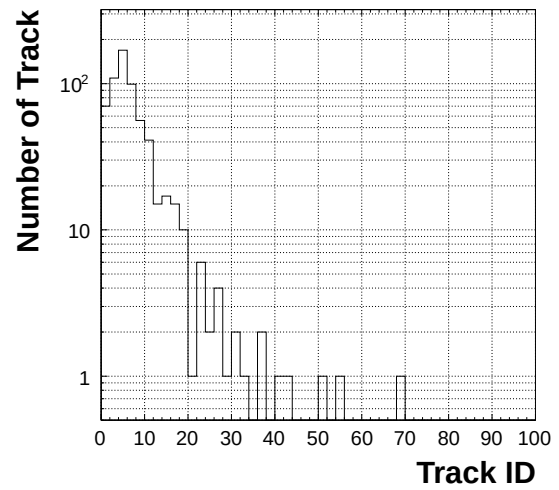


図 4.9: Standalone での Track ID

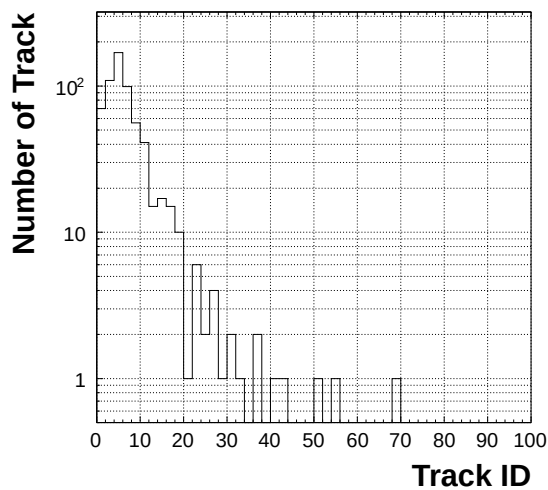


図 4.10: Athena での Track ID

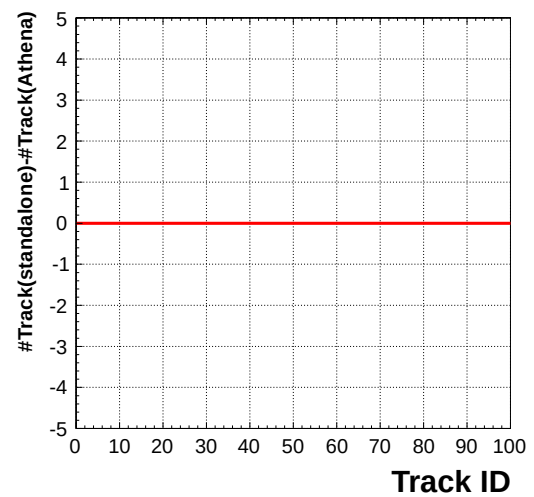
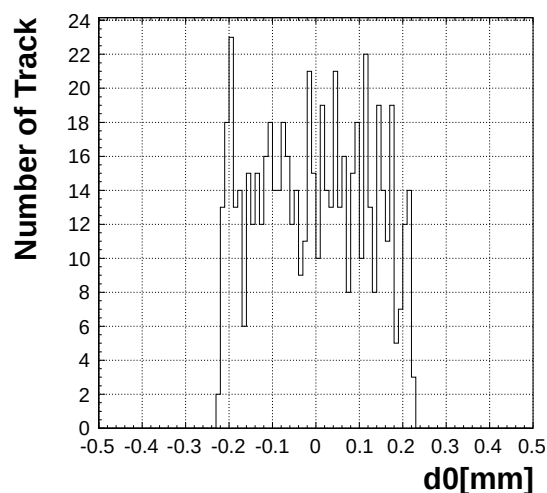
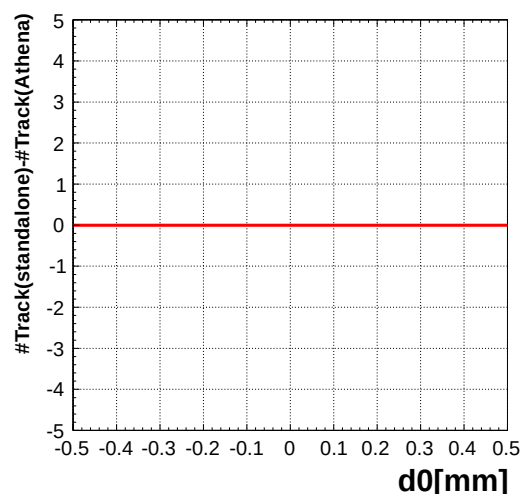


図 4.11: Track ID の差分
差分が 0 なので完全に一致している

図 4.12: Athena でのインパクトパラメータ d_0 図 4.13: インパクトパラメータ d_0 の差分
差分が 0 なので完全に一致している

これらの図から差分の値が完全に一致しているため、既存の Standalone でのシミュレーションと Athena 上でのシミュレーションは AM も TF も同等に動作していることが確認された。

今後の開発では、主に 2 つの課題が残っている。1 つ目は Grid で使用できるスクリプトの開発と動作確認である。Grid で使用できれば FTK シミュレーションを含みながら Reconstruction できるようになる。2 つ目の課題はアウトプットファイルにどの値を詰めるかの選定である。開発者用の Ntuple ではなく、物理解析に汎用的な Ntuple にする必要がある。また、Track の値を入れる時にはユーザーが分かりやすい値にする必要がある。これらを含めて実機導入の 2014 年までに完成する必要がある。

このシミュレーションを用いて次節以降では飛跡再構成率や算出した Track パラメータの質、処理時間について評価・検討を行う。

4.2 低ルミノシティにおける Track の評価

環境に依らず FTK が再構成する Track がオフラインに近い能力であることが FTK システムの安定性の面において重要な事柄である。また、2012 年度には FTK の一部を ATLAS 検出器に実装し、動作テストを行う予定であることから、まず初めに、現在の衝突エネルギー 7TeV、瞬間ルミノシティ $10^{32} \sim 10^{33} [\text{cm}^{-2}\text{sec}^{-1}]$ のモンテカルロデータを用いて FTK が再構成する Track 情報とデータ取得後にオフラインソフトウェアで再構成された Track 情報との比較検討を行う。

4.2.1 データセット

表 4.1 に使用する RDO データセットを示す。今回のシミュレーションでは MinimumBias トリガーの事象、Vector Boson Fusion(VBF) 過程による $H \rightarrow \tau\tau \rightarrow lh$ 事象 (図 4.2.1) を使用した。これらの事象を用いる理由として、MinimumBias 事象は 2.2.6 節で述べたように衝突のタイミングのみ同期したランダムな事象であることに注目し、常に一定以上のパフォーマンスを求められる FTK にとって重要であるため使用した。最後に VBF $H \rightarrow \tau\tau \rightarrow lh$ は FTK が特に得たい物理事象の一つであるため評価に用いた。

また、今回のシミュレーションでは現在実装段階で使用予定の SS サイズと内部飛跡検出器の Barrel ($|\eta| < 1.2$) のみの記憶パターンで固定した。値は表 4.2 に記載する。

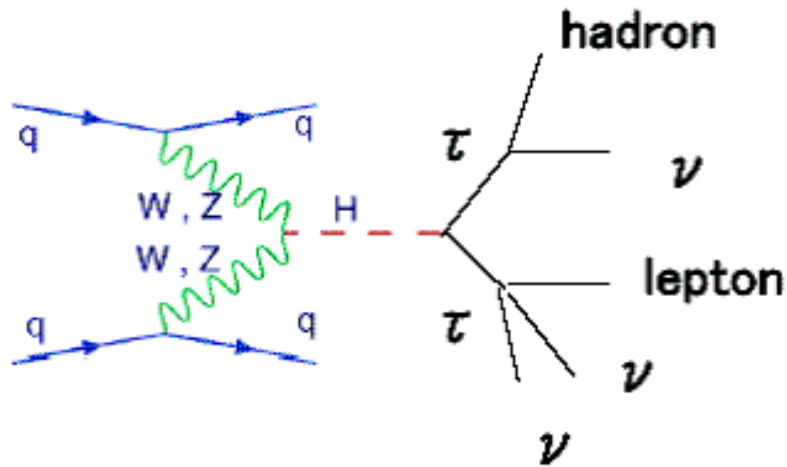


図 4.14: VBF $H \rightarrow \tau\tau \rightarrow lh$ のダイアグラム

表 4.1: 7TeV の RDO モンテカルロデータセット

データ名	イベント数
mc10_7TeV.105001.pythia_minbias.digit.RDO.e574_s932_s946_d495	30k
mc10_7TeV.105334.HerwigVBFH120tautaulh.digit.RDO.e598_s933_s946_d426	13.5k

表 4.2: Super Strip サイズ

	Pixel ϕ 方向	Pixel z 方向	SCT Strip
Length[mm]	1.2(24 ピクセル)	14.4(36 ピクセル)	1.6(20 ストリップ)

4.2.2 評価方法

FTK によって再構成された Track の評価方法として、同じデータセットからオフラインソフトウェアで再構成された Track 情報との比較を行う。そこで事象生成時に得られる Track の Truth 情報と再構成された Track 情報それぞれの位置情報 η と ϕ を用いて

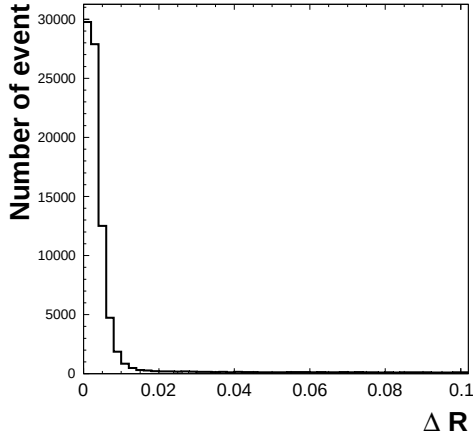
$$\Delta R = \sqrt{(\eta_{\text{reconstruction}} - \eta_{\text{truth}})^2 - (\phi_{\text{reconstruction}} - \phi_{\text{truth}})^2} \quad (4.1)$$

を定義し、再構成された Track 1 つに対し最小の ΔR を算出する。但し本論文では Barrel 部のみのパターンを使用しているため、Truth 情報の Track は Barrel ($|\eta| < 1.2$) 内部かつ FTK が再構成できる下限横運動量 1GeV/c を超えたものに限ることに注意する。次に、求めた ΔR のうち、ある値の範囲内にある Track を Truth 情報と同一したものとしてカウントし、

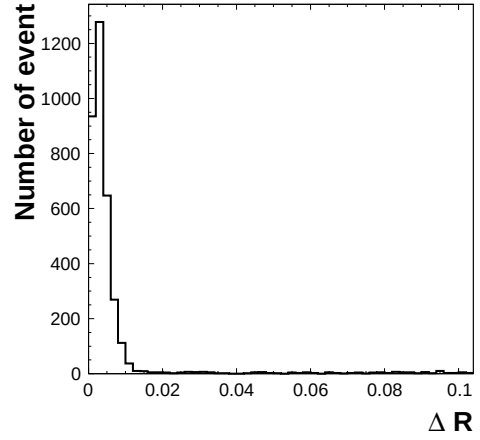
$$\text{Efficiency} = \frac{\text{Number of Matched Reconstruction Track}}{\text{Number of Truth Track}} \quad (4.2)$$

と定義し、事象再構成率を FTK の場合とオフラインの場合でそれぞれ算出する。但し同じ TruthTrack に対し 2 回マッチした場合は分母分子両方ともカウントする。

ΔR の取り方を決めるために、VBF, MinimumBias の両サンプルにおいて全 ΔR 分布を表したものが図 4.15 と図 4.16 である。この分布から ΔR の値を 0.02 以下とし、マッチングが取れた Track での再構成率と分解能を算出し比較検討を行った。

図 4.15: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における ΔR 分布

ほぼ 0.02 以内から 0.02 をコーンサイズとする

図 4.16: MinimumBias 事象における ΔR 分布

ほぼ 0.02 以内から 0.02 をコーンサイズとする

4.2.3 結果と考察

●再構成率

VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象と MinimumBias 事象における FTK とオフラインの再構成率を図 4.17 と図 4.18 に示す。2つのグラフから、FTK が再構成する Track とオフラインアルゴリズムで再構成した Track との再構成率の比を算出すると VBF $H \rightarrow \tau\tau \rightarrow lh$, MinimumBias, 共に 80 % 以上という結果 (図 4.19, 図 4.20) が得られた。これは現在の SS Size や記憶パターンにおいてリアルタイムでオフラインの 80 % 以上を再構成可能であると結論づけられる。

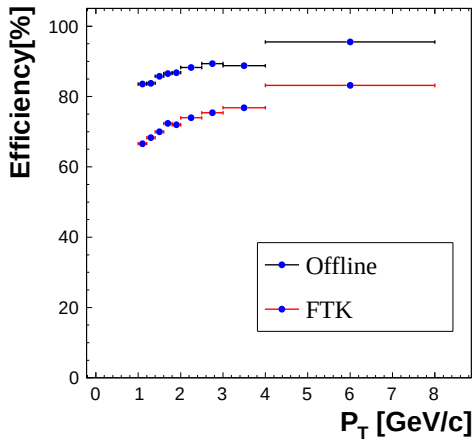
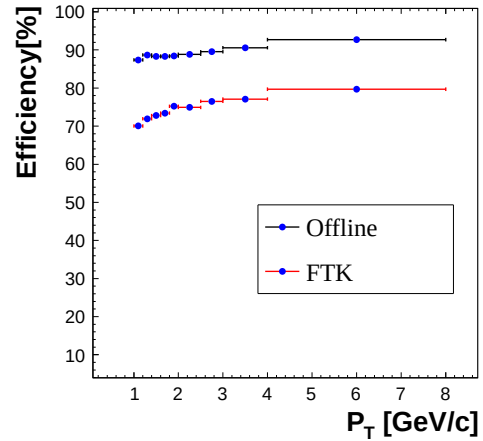
図 4.17: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインの再構成率

図 4.18: MinimumBias 事象における FTK とオフラインの再構成率

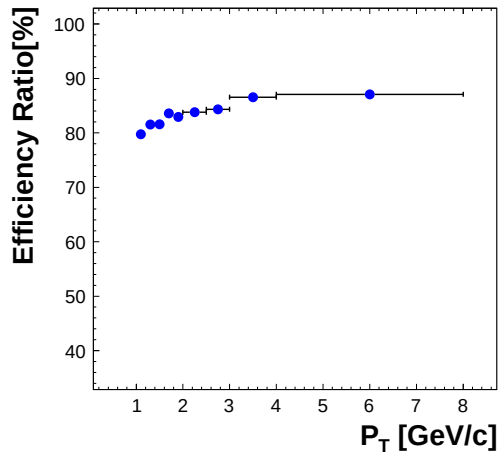


図 4.19: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における再構成率のオフラインと FTK の比
オフラインに対し FTK は 80 % 以上再構成可能

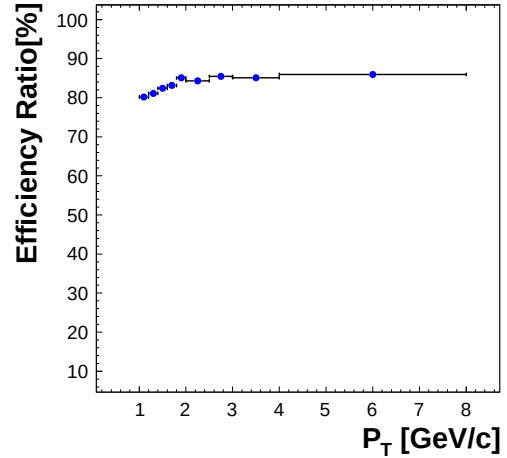


図 4.20: MinimumBias 事象における再構成率のオフラインと FTK の比
オフラインに対し FTK は 80 % 以上再構成可能

詳細に各運動量ごとに再構成率を算出すると、運動量が増加するにつれて再構成率が上昇が見られた。これは Track の運動量が低くなる程、曲率が大きくなるため運動量分解能は上昇するが、ノイズ信号や PileUp による信号の影響を受けやすくなり Track として認識しづらくなる影響だと考えられる。図 4.21 に各横運動量における FTK Track ごとのミスヒットの層数を示す。この図から FTK が再構成した Track のうち低運動量ほどヒットが存在しない層が増加することが示される。したがって低運動量領域の再構成率が下がる現象は一般的に表れる。

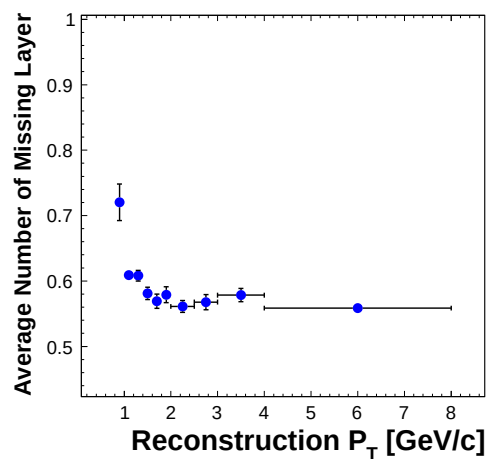


図 4.21: 各横運動量における FTK Track ごとのミスヒットの層数
低運動量で曲がる程ミスヒットが増加する

以上から低ルミノシティでかつ、現在のエネルギーではオフラインのうち 80 % 以上を FTK は再構成していることが結論付けられる。今後の研究課題として、FTK 側ではこの比率を上昇させる新しいパターンの生成が考えられる。また FTK との連携として、他のトリガーアルゴリズムと FTK の情報との最適な組み合わせを検討することで更なる再構成率上昇を目指すことも重要である。

• インパクトパラメータ

次に粒子のインパクトパラメータの分解能を比較する。VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象のインパクトパラメータ分解能を図 4.22 に、MinimumBias 事象の結果を図 4.23 に示す。また、表 4.3 に各事象の二乗平均平方根を示す。二乗平均平方根 (Root Mean Square : RMS) は N 個の bin のうち i 番目の bin に対する変数を x_i として以下の式で表される。

$$\text{Root Mean Square} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (4.3)$$

また、FTK の分布から算出した分解能の広がり、 $\text{RMS}(\sigma_{\text{FTK}}$ とする) は検出器が持つ分解能の寄与を内包しているため、パターンや SS サイズなどから影響を受ける FTK に由来する分解能の広がり (σ_{PureFTK}) はオフラインの RMS を σ_{Off} とすると

$$\sigma_{\text{FTK}} = \sigma_{\text{Off}} \oplus \sigma_{\text{PureFTK}} \quad (4.4)$$

$$\sigma_{\text{PureFTK}} = \sqrt{\sigma_{\text{FTK}}^2 - \sigma_{\text{Off}}^2} \quad (4.5)$$

と表される。以下では

$$\text{Percent} = \frac{\sigma_{\text{FTK}}}{\sigma_{\text{Off}}} \quad (4.6)$$

から得られる増減のパーセントと $\sigma_{\text{Off}} \oplus \sigma_{\text{PureFTK}}$ の値を用いて検討を行う。

表 4.3 から MinimumBias 事象では分解能の二乗平均平方根で 3 % 以内で一致しており、FTK の影響による分解能の広がり σ_{PureFTK} は 0.032mm であった。一方、VBF 事象では 0.079mm(Offline) $\oplus$ 0.096mm(FTK) となり、FTK が固有にもつ分解能が数十 μm 程度悪くなっている。しかしながら、今回使用したパターン、SS サイズでは FTK に由来する分解能は 100 μm 程度であり、その他の検出器の影響等を含めても FTK で再構成した Track は 150 μm 程度の分解能を持っていると言える。

この分解能は、 τ の飛跡である $c\tau$ が 87 μm であり 0.5 σ 程度の分解能であるため、 τ の同定に FTK のインパクトパラメータのみでは不十分である。したがって FTK の他のパラメータやカロリメータの情報と合わせて使用して効率の良い τ の同定方法を確立することが今後の課題である。

一方 b ジェットは $c\tau$ が約 500 μm であり、平均で数 mm 単位で 2 次崩壊点まで飛来するため、今回の FTK の分解能であれば十分に同定できることが分かる。

以上からインパクトパラメータの分解能は現状のルミノシティ・エネルギー下において FTK はリアルタイムの飛跡再構成ながらオフラインとほぼ同等の再構成能力を持っており、b ジェットを同定するには十分な分解能を備えていると言える。

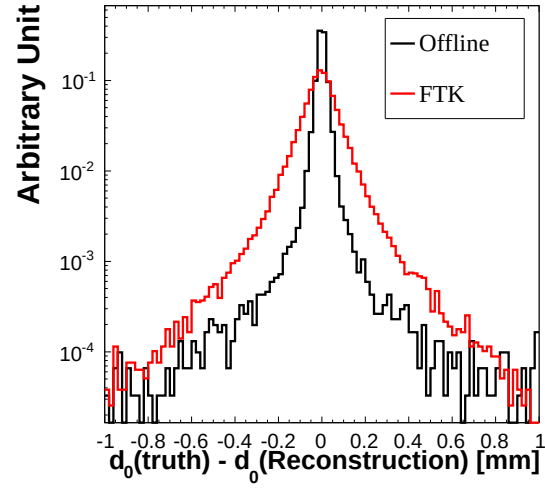


図 4.22: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインのインパクトパラメータ

分解能は $0.079\text{mm}(\text{Offline}) \oplus 0.096\text{mm}(\text{FTK})$ をもつ

拡がりは生じているが b ジェットの 2 次崩壊点を測定するには十分な分解能

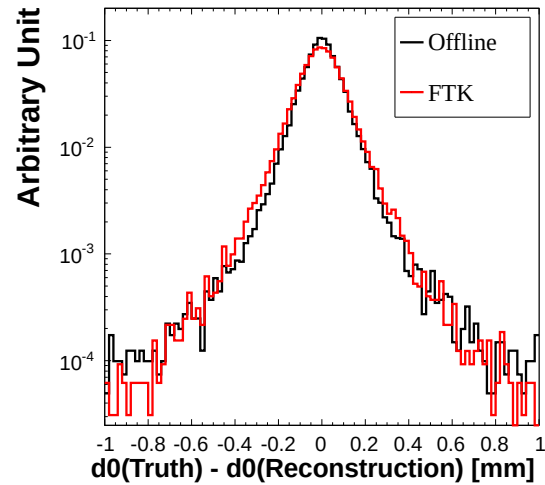


図 4.23: MinimumBias 事象における FTK とオフラインのインパクトパラメータ

FTK がオフラインとほぼ同等の能力を発揮

表 4.3: 各事象におけるインパクトパラメータ分解能の二乗平均平方根

	VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象	MinimumBias 事象
二乗平均平方根 (オフライン)	0.079mm	0.129mm
二乗平均平方根 (FTK)	0.124mm	0.133mm

● 横運動量 P_T

最後に横運動量の分解能を比較する。VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象の横運動量分解能を図 4.24 に、MinimumBias 事象の横運動量分解能を図 4.25 に示す。

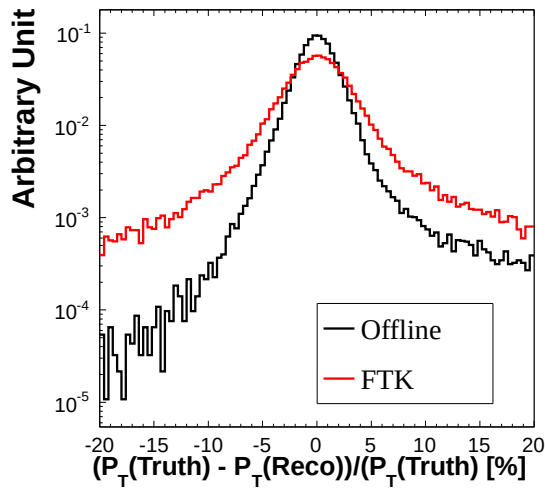


図 4.24: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインの横運動量分解能

オフライン Track の横運動量が Truth Track より低いいため正方向に広がっている

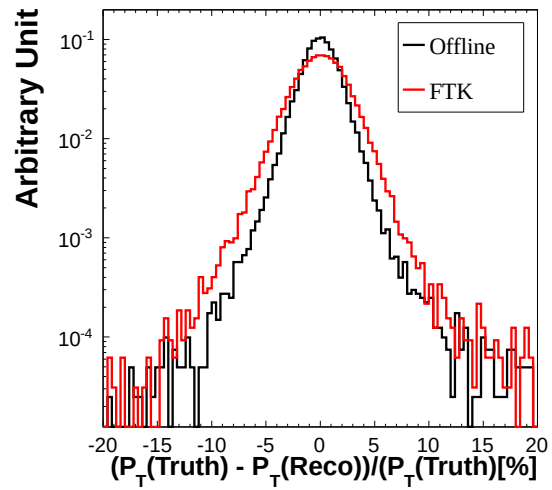


図 4.25: MinimumBias 事象における FTK とオフラインの横運動量分解能

分解能は 2.4 % (Offline) $\oplus$ 1.8 % (FTK) となりオフライン並の分解能である

結果から、MinBias 事象の分解能の二乗平均平方根は FTK が 3.0 %、オフラインは 2.4 %、FTK 本来がもつ分解能は 1.8 %となっており、オフライン並の分解能となっている。

一方、VBF 事象ではオフラインの分布において正方向に偏りが生じている。この偏りを調査するために、粒子ごとの横運動量分解能を算出した。結果を図 4.26 に掲載する。

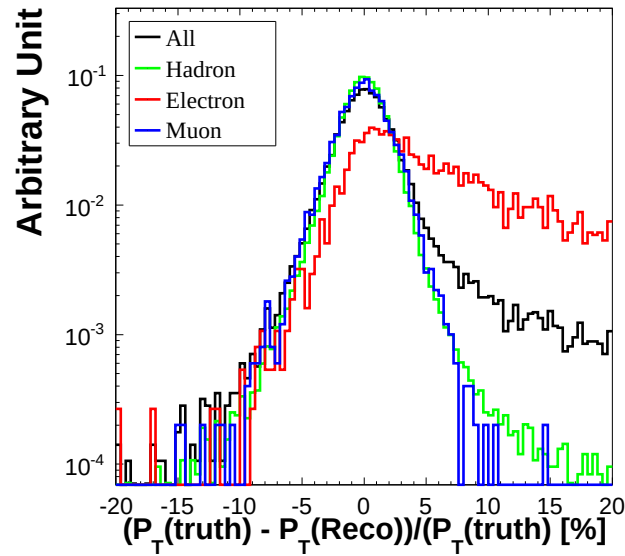


図 4.26: オフラインにおける各粒子ごとの横運動量分解能

電子はフォトンを放出するため、観測した運動量が減少する

このためオフラインの運動量にズレが生じていた

グラフから Electron のみ再構成した横運動量が小さく正方向にズレが生じている。これは Electron が制動放射によりフォトンを放出するため実測の横運動量が真の値より小さくなるためである。一方 FTK ではヒットの場所から予め記憶されている横運動量の一次近似で構成するため、この影響は受けない。従って、本論文で横運動量の分解能をオフラインと FTK で比較するために、VBF 事象では Electron の Track 情報は使用しないことにした。その時の横運動量の分解能を図 4.27 に示す。この時の二乗平均平方根は FTK が 6.1 %，オフラインは 2.6 %となり、約 2 倍以上の拡がりが生じていることが分かった。

そこで、運動量ごとに分解能を確認した。低運動量領域 ($6\text{GeV}/c$ 以下) と高運動量領域 ($6\text{GeV}/c$) での運動量分解能を図 4.28, 図 4.29 に示し、二乗平均平方根を表 4.4 に記載する。

表 4.4: VBF 事象における横運動量分解能の二乗平均平方根

	FTK Track	オフライン Track
二乗平均平方根 ($6\text{GeV}/c$ 以下)	3.5 %	2.6 %
二乗平均平方根 ($6\text{GeV}/c$ 以上)	12.1 %	3.9 %

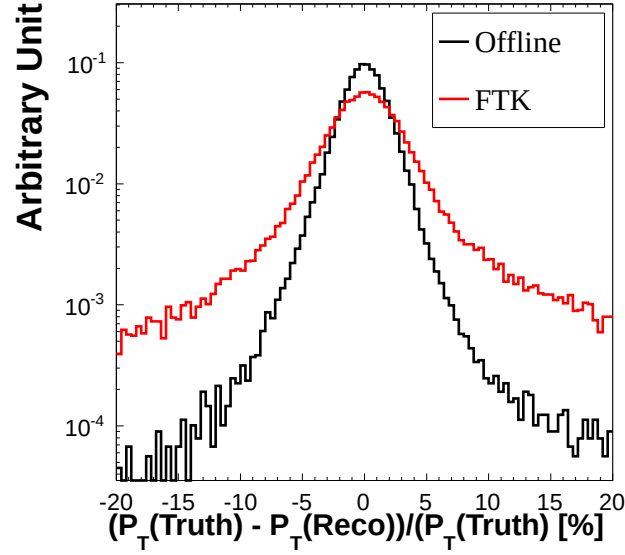


図 4.27: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象における FTK とオフラインの横運動量分解能
FTK が 2 倍以上の広がりを持っている

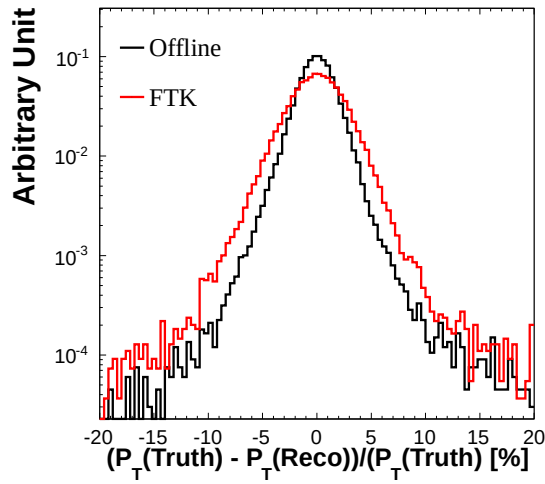


図 4.28: VBF 事象における横運動量 $6\text{GeV}/c$
以下の FTK とオフラインの横運動量分解能
分解能は $2.6\% (\text{Offline}) \oplus 2.3\% (\text{FTK})$
オフライン並の能力を発揮

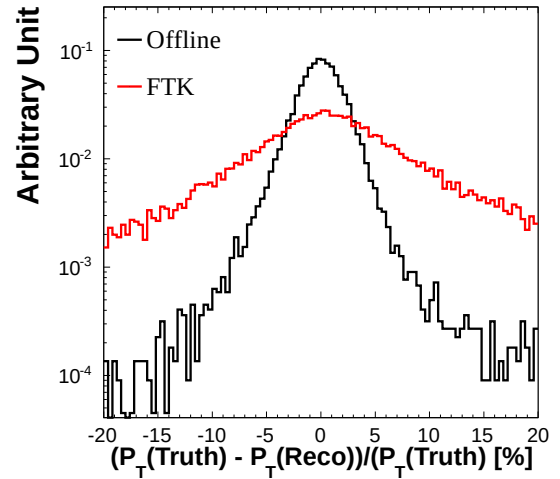


図 4.29: VBF 事象における横運動量 $6\text{GeV}/c$
以上の FTK とオフラインの横運動量分解能
分解能は $3.9\% (\text{Offline}) \oplus 11.5\% (\text{FTK})$
高運動量における FTK 本来がもつ分解能が
悪いことが示唆された

表 4.4 から高運動量になるほどオフライン、FTK の両者共に分解能が悪くなっていることが分かる。この高運動量の分解能が悪化する現象は下記のような一般式で示すことができる。荷電粒子が通過する時の運動方程式

$$\frac{mv^2}{r} = q(v \times B) \quad (4.7)$$

を解き、単位を横運動量 P_T [GeV/c], 磁場 B [T], 半径 r [m] とすれば

$$P_T = 0.3Br \quad (4.8)$$

となる。ここで荷電粒子が描く Track の円弦を L [m] とすれば、分解能は一般に

$$\frac{\sigma(P_T)}{P_T} \simeq \frac{8\sqrt{\frac{3}{2}}\sigma(x)P_T}{0.3BL^2} \quad (4.9)$$

と表すことができる。従って横運動量に分解能は比例することが一般に言える。ゆえにオフラインと FTK が共に悪化する現象は理解できる。

しかし、本論文で使用したパターンでは高運動量に対してオフラインより分解能が約 3 倍ほど劣化していることが分かる。高運動量に対する FTK の RMS は 3.9 % (Offline) ⊕ 11.5 % (FTK) であるのに対し、低運動量では 2.6 % (Offline) ⊕ 2.3 % (FTK) となることから、今回使用したパターンや SS サイズなどの FTK 本来の分解能が高運動量領域で劣化していることが示唆された。したがって今後は、より分解能が良い SS サイズの追究する必要があることが分かった。

4.2.4 低ルミノシティ評価の纏め

以上から、本論文で使用したパターンでは、リアルタイムの飛跡再構成ながら Barrel 部でオフラインの 80 % 以上の再構成能力を持っていると結論付けることができる。

また、Track のインパクトパラメータについてはオフラインとほぼ同等の再構成能力であり、2 次崩壊点までの距離を用いて b ジェットの同定を行うには十分な分解能を持っていると言える。

最後に横運動量の分解能では低運動量領域においては FTK に由来する分解能が 3 % 程度であり、ほぼオフライン並の分解能であるのに対し、高運動量領域では今回使用したパターンや Super Strip サイズでは FTK がもつ分解能が悪く、オフラインに対して劣化が見られた。

今後は更に精度を向上させるにはより良いパターンの生成、Super Strip サイズの最適化、新たなアルゴリズムを使用することが考えられる。また、実機の動作テストから情報をフィードバックすることで、シミュレーションにおける新たな問題点が明らかにし、より精度の良いシミュレーションで再評価することが必要である。

4.3 高ルミノシティにおける Track の評価

FTK は高ルミノシティ下でも Track を再構成し Track 情報を安定して LVL2 に供給する必要がある。FTK が飛跡情報を安定供給することで、LVL2 でより洗練されたアルゴリズムを使用し、より物理的な情報を用いて高ルミノシティで問題となるトリガー閾値を上げることなく観測データを取捨選択が可能となる。このような高ルミノシティ下での安定動作のために現段階から評価を行い実装に向けてより Track の精度を良いものにする必要がある。特に、我々が観測したい物理現象に対してハイパフォーマンスで安定した動作が必要となる。

そこで本論文では将来、標準理論の検証に使用することになる Vector Boson Fusion(VBF) 過程の $H \rightarrow \tau\tau \rightarrow lh$ 事象において、高ルミノシティ下における FTK が再構成する Track を評価する。

データセットは表 4.3 に記載するように、ルミノシティ $1 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ は FTK 実装予定である 2014 年を想定し、ルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ は 2020 年の予定最大値から更に大きな場合を想定した場合である。また、低ルミノシティのサンプルは 4.2 節で使用したのと同じものである。この高ルミノシティ下における再構成率を式 (4.2) で算出する。

表 4.5: 高ルミノシティのモンテカルロデータセット

事象サンプル	ルミノシティ $[\text{cm}^{-2}\text{sec}^{-1}]$
mc10_7TeV.105334.HerwigVBFH120tautaulh.digit.RDO	$10^{32} \sim 10^{33}$
VBF $H \rightarrow \tau\tau \rightarrow lh$	1×10^{34}
VBF $H \rightarrow \tau\tau \rightarrow lh$	3×10^{34}

$\Delta R < 0.02$ のマッチングにて最高ルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ における各横運動量ごとに算出した再構成率を図 4.30 に示す。4.2 節の低ルミノシティの時と比べ、粒子の種類に依存せず再構成率が 75 % から 85 % と飛躍的に上昇している。これは高ルミノシティでは内部飛跡検出器に大量の信号を残し、FTK では大量の Track を算出するため、 $\Delta R < 0.02$ のカットのみでは truth 情報に由来した Track とマッチできていない可能性がある。

実際、最高ルミノシティ下において $\Delta R < 0.02$ のコーン内に存在する FTK が算出した Track の平均数を各運動量ごとに示したものが図 4.31 である。

この値を参照すると各運動量ごとに平均 2 つ以上の Track が存在するため、 $\Delta R < 0.02$ のカットのみでは正しいマッチングを行うことができていない可能性がある。

そこで ΔR のコーンカットに加え横運動量に対し

$$\frac{|P_T(\text{Truth}) - P_T(\text{reconstruction})|}{P_T(\text{Truth})} \times 100 \leq 18 \%$$

のカットをかけることで、より正しい Track とマッチングすることにした。ここで 18 % という値は 4.2 節の低ルミノシティサンプルで横運動量の分解能の二乗平均平方根が 6 % 程度と算出されたため、高ルミノシティにおいて同じ分解能であると仮定し 99.9 % の取得ができる 3 σ の値として

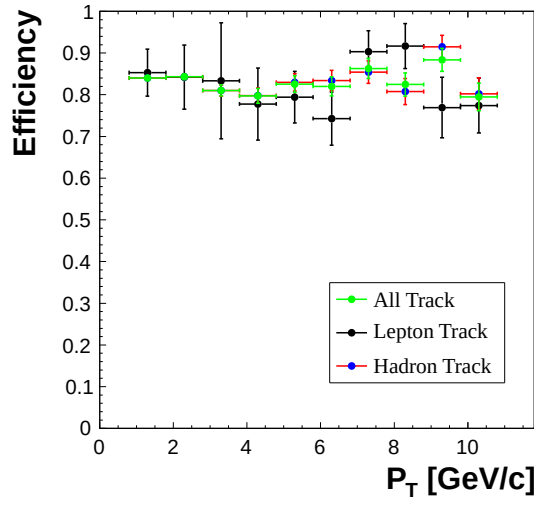


図 4.30: $3 \times 10^{34} [\text{cm}^{-2} \text{sec}^{-1}]$ における VBF 事象での $\Delta R < 0.02$ カットのみの再構成率
 $\Delta R <$ のカットだけではフェイク Track が
多く再構成率を評価できない

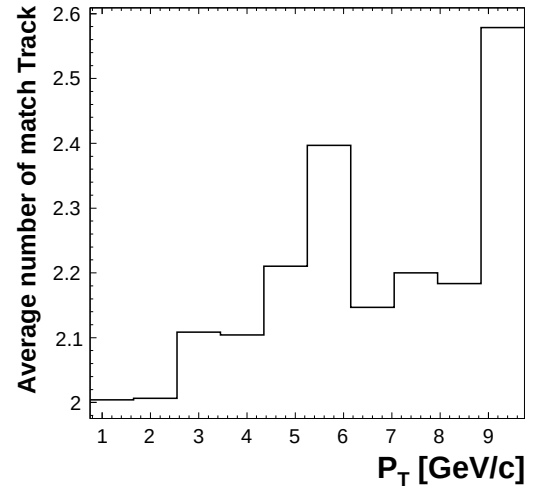


図 4.31: $3 \times 10^{34} [\text{cm}^{-2} \text{sec}^{-1}]$ における角運動量
ごとに ΔR でマッチした再構成 Track の平均数
Track が 2 つ以上存在するため
事象選別を追加する必要がある

18 %を採用した。上記の選別を使用して再構成率を算出した結果を以下に示す。

4.3.1 結果と考察

再構成率の算出結果を表 4.6 に示す。この表から高ルミノシティになるにつれて、再構成率が減少していることが分かる。

表 4.6: 各瞬間ルミノシティにおける再構成率

瞬間ルミノシティ [$\text{cm}^{-2} \text{sec}^{-1}$]	$10^{32} \sim 10^{33}$	1×10^{34}	3×10^{34}
再構成率 [%]	62.9 ± 0.1	41.4 ± 0.3	27.9 ± 0.3

この現象を詳しく調べるために各ルミノシティにおける運動量ごとの再構成率を算出した。図 4.32 はルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$, 図 4.33 はルミノシティ $1 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$, 図 4.34 は低ルミノシティで算出した値をグラフ化したものである。このグラフから、高ルミノシティ下では高い運動量成分において再構成率が顕著に減少することが示された。また、この下降現象はレプトン粒子、ハドロン粒子全てにおいて見受けられるため、高ルミノシティの影響による PileUp 効果によって減少傾向が引き起こされていると考えられる。

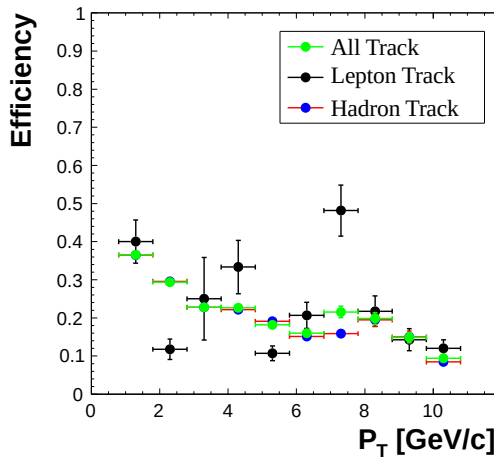


図 4.32: $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ における VBF 事象の ΔR と横運動量カットでの再構成率
高運動量で再構成率が減少

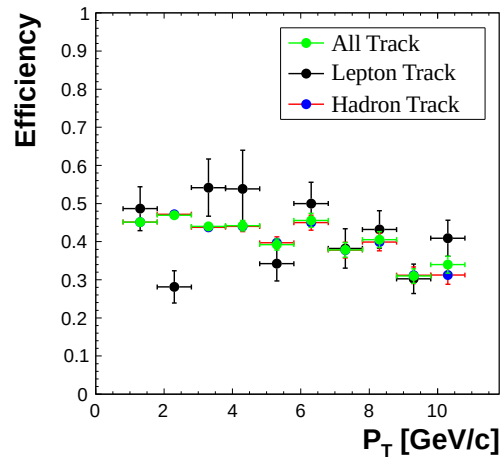


図 4.33: $1 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ における VBF 事象の ΔR と横運動量カットでの再構成率
高運動量で再構成率が減少

一方、図 4.30 から横運動量のカットなしでは再構成率が高い状況であるので、高ルミノシティ下と低ルミノシティ下では運動量に対して振る舞いに違いがあると推察される。

そこで、 ΔR のコーンでマッチングされた飛跡の運動量分解能を、横運動量 $2\text{GeV}/c$ 以下、 $2\text{GeV}/c \sim 6\text{GeV}/c$, $6\text{GeV}/c$ 以上の 3 つに分けて算出した。各結果を図 4.35, 図 4.36, 図 4.37 と、各二乗平均平方根を表 4.7 に記載する。

表 4.7: 各瞬間ルミノシティにおける横運動量の二乗平均平方根

瞬間ルミノシティ $[\text{cm}^{-2}\text{sec}^{-1}]$	$10^{32} \sim 10^{33}$	1×10^{34}	3×10^{34}
横運動量 $2\text{GeV}/c$ 以下の二乗平均平方根	3.3 %	9.7 %	15.7 %
横運動量 $2\text{GeV}/c \sim 6\text{GeV}/c$ の二乗平均平方根	5.2 %	10.4 %	16.9 %
横運動量 $6\text{GeV}/c$ 以上の二乗平均平方根	12.1 %	19.6 %	21.2 %

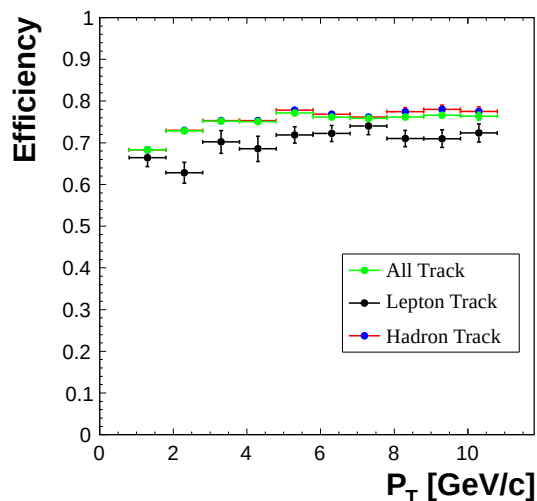


図 4.34: 低ルミノシティにおける VBF 事象の ΔR と横運動量カットでの再構成率
運動量に依らず再構成率は 70 % 以上

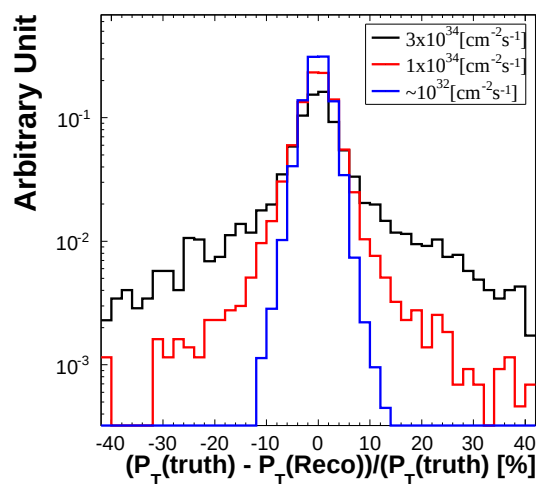


図 4.35: 横運動量 $2\text{GeV}/c$ 以下での運動量分解能

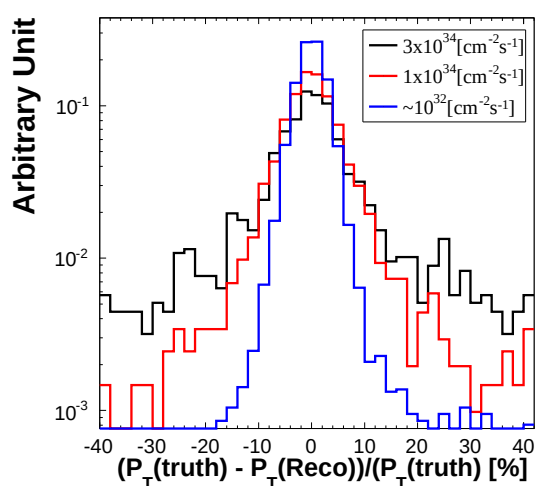


図 4.36: 横運動量 $2 \sim 6\text{GeV}/c$ での運動量分解能
高ルミノシティほど分解能が悪い

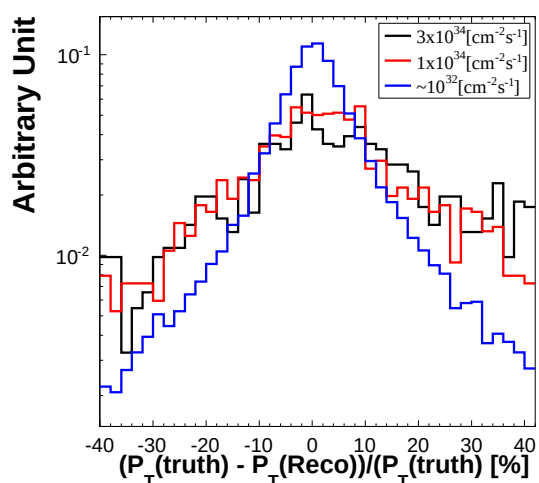


図 4.37: 横運動量 $6\text{GeV}/c$ 以上での運動量分解能
高ルミノシティほど分解能が悪い
ため FTK が PileUp 効果を受けている

上記の結果から2点のことが言える。1つ目は高い運動量になるほど運動量分解能が悪化することである。これは、4.2節の低ルミノシティにおいても起きていた現象であり、高ルミノシティでも引き続き現象は起きている。2つ目に高ルミノシティになるにつれ分解能が悪化することが示される。高ルミノシティにおいて分解能が悪化する要因として以下の原因が考えられる。

高ルミノシティでは低ルミノシティに比べてPileUpによるTrackが増加しSS size内に存在する信号数が増加する(図4.38)。この時、連続したヒット情報を1つにクラスター化し、本来算出すべきTrackから得る信号とは異なる位置情報を使用するためにFTKで算出するTrackのパラメータにずれが生じる可能性がある。

このために高ルミノシティ下において運動量分解能が低下し、再構成Trackとのマッチングする割合が減少したと考えられる。したがって、分解能を向上させる方法を追究すべきであることが本研究で示唆された。

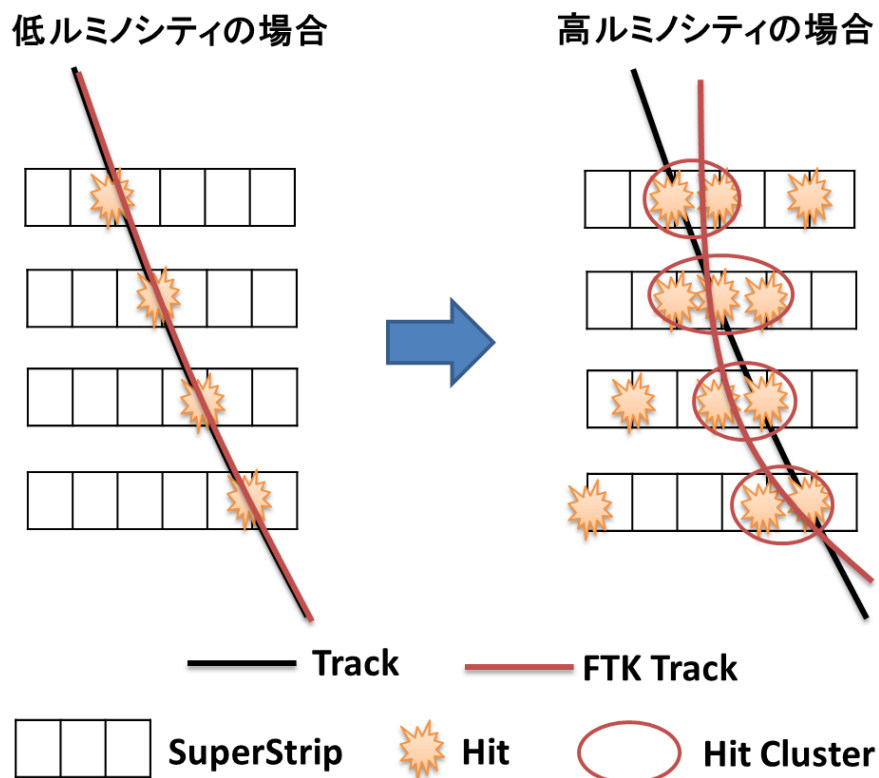


図 4.38: ルミノシティの違いによる FTK Track の違い
 現状の FTK とパターンでは PileUp による信号が連続しているため
 PileUp の Track と認識できず、再構成 Track にズレが生じた

4.3.2 高ルミノシティ評価の纏め

以上から現状のシステムでは高ルミノシティ下において不十分であり改善の必要性があるということが分かった。この改善点として SuperStrip サイズを細かくすることも検討可能である。しかし、サイズを小さくすることは記憶パターン数の増加に繋がり処理時間の増大を招くので細分化だけの方法には限界が存在することに注意しなければならない。

そこで今後行うべき研究の一つはクラスター化方法の再検討である。現在の動作ではヒット情報をクラスター化する際、連続したヒットを1つと考え、そのストリップ情報を Track の算出に使用している。そのため高ルミノシティでは同時に連続して鳴るヒット信号が増え、Track が何本もあるにも関わらず算出する Track が減少したり、Track パラメータがズレた Track を算出する可能性がある (図 4.39 左図)。この現象を解消するために、クラスターサイズの最適化を行い、必要な細分化を検討する。このアルゴリズムによって、より正しい Track が算出できるよう研究を行う必要がある (図 4.39 右図)。

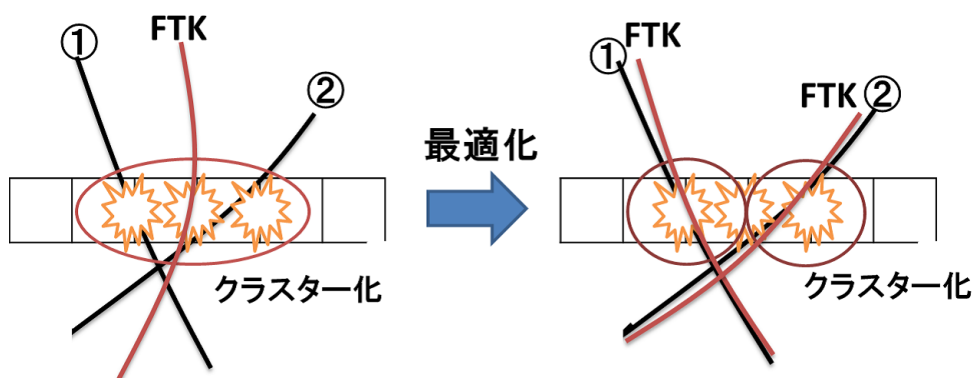


図 4.39: ヒットのクラスター化の概念図

Hit のクラスター化の時点で PileUp 信号との区別をつけることができれば

今回生じた運動量の分解能の悪化を改善することができる

以上から現状では高ルミノシティに対して FTK の十分な力を発揮できない状況だと判明したため、実機挿入前にあらゆる手段を検討し、より精度の良い分解能を求める必要があると示唆された。

4.4 処理時間の計測

前節までは FTK の算出する Track について述べたが、一方 FTK の処理時間も性能評価において重要なパラメータである。なぜなら、LVL1 から 100kHz で情報量を受け取り LVL2 の決定を下す 40msec 以内に FTK は Track を算出しなければならないからである。万一 FTK の処理が遅れて Dead Time を作ることになれば、トリガーの質や効率を下げる要因となる。従って FTK は Dead Time なしに飛跡を多く生む事象でも安定した動作が可能か常に検討しなければならない。

FTK の処理時間は検出器の信号数と認識するパターン数や再構成する飛跡の数に大きく依存する。したがって高ルミノシティになるほど荷電粒子による信号数が増加するため処理時間が大きくなると予想される。このような背景から、本論文では高ルミノシティ下での処理時間の見積もりと評価を行う。

4.4.1 計算方法

FTK のそれぞれの機能で処理時間の計算式を定義することで FTK 全体の処理時間の見積もりを行う。本論文では特に最も時間がかかる DO, AM, DO の受け取り, TF の流れを計算する。

以下に計算方法を述べる。データはある束でデータストリームとして入出力される。そこで図 4.40 のようにデータストリームの先頭が入力される時間を FwIn, 出力される時間を FwOut, 最後尾が入力される時間を EwIn, 出力時間を EwOut と定義し下記の計算式 (4.1)～式 (4.4) のように各機能の処理時間を見積もる。但し、EwIn が 1 データストリームを処理した時間と同義であり、データストリームの出力時間は待ち時間分だけかかるので EwOut になることに注意する。ProcTime は 1 つのデータに対する処理時間、Intime は 1 データの入力時間、Delay はデータストリームの先頭が出力されるまでの待ち時間と定義する。これら 3 つの値はそれぞれのハードウェアのスペックごとに異なる。そこで今回は 2020 年に使用予定のスペックを想定した。そのスペック一覧を表 4.8 に記載する。

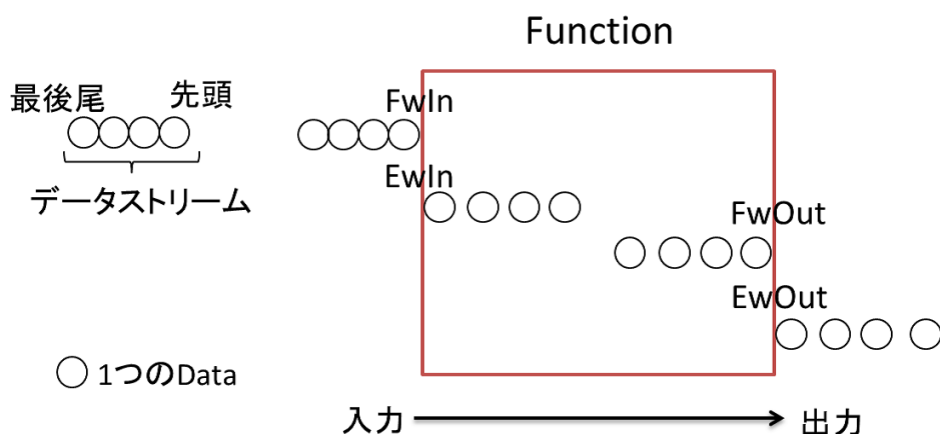


図 4.40: データの入出力の概念図

EwOut の値が 1 つの機能に対する処理時間となる

表 4.8: 各機能における 1 データ辺りの処理能力

機能名	ProcTime(処理時間)[nsec]	Intime(入力時間)[nsec]	Delay(出力待ち時間)[nsec]
DO	5	5	40
AM	5	5	250
DO Read	5	5	200
TF	1	5	200

式 (4.1)～(4.4) 内に Pre と Pre_Pre が表記されているが、これらは 1 事象前を Pre, 2 事象前を Pre_Pre とし、バッファを 2 個、バッファ領域は無視して計算している。次に、DO の FwIn における Pre_EwOut $- 10\mu\text{sec}$ は、LVL1 から 100kHz で受信することから入力時間に $10\mu\text{sec}$ 以内という制限がかかるため、1 事象前の遅延時間を表している。また、各機能はパイプラインで 1 つに繋がっており、FIFO(Fast In Fast Out) で入出力するため、前段の機能の EwOut と後段の機能の FwIn は接続する形となる。最後に AM の処理時間 (EwIn) は AM 内部にバッファが 2 つ存在するため場合分けが生じている。この場合分けの中で最大の時間が処理時間となる。

これらの各機能ごとの計算式によって FTK 全体の処理時間を算出するが、FTK は各機能、各 Board で並列に行われるため、本論文では最も処理の遅い Board の TF における EwOut を FTK 全体の処理時間とした。

●DO

$$\begin{aligned}
 \text{FwIn} &= \max(0, \text{Pre_EwOut} - 10\mu\text{sec}, \text{Pre_Pre_EwOut} - 20\mu\text{sec}) \\
 \text{EwIn} &= \text{Number of Hit} \times \text{InTime} + \text{FwIn} \\
 \text{FwOut} &= \text{FwIn} + \text{Delay} \\
 \text{EwOut} &= \max(\text{FwOut} + \text{ProcTime} \times \text{Number of Hit}, \text{EwIn} + \text{Delay}) \quad (4.1)
 \end{aligned}$$

●AM

$$\begin{aligned}
 \text{FwIn} &= \max(\text{DO_FwOut}, \text{Pre_EwOut} - 10\mu\text{sec}, \text{Pre_Pre_EwOut} - 20\mu\text{sec}) \\
 \text{EwIn} &= \max(\text{DO_EwOut}, \text{Number of SSHit} \times \text{InTime} + \text{DO_FwOut}, \\
 &\quad \text{Pre_EwOut} - 10\mu\text{sec}, \text{Number of SSHit} \times \text{InTime} + \text{Pre_EwIn} - 10\mu\text{sec}, \\
 &\quad \text{Number of SSHit} \times \text{InTime} + \text{Pre_Pre_EwOut} - 20\mu\text{sec}) \\
 \text{FwOut} &= \text{EwIn} + \text{Delay} \\
 \text{EwOut} &= \text{FwOut} + \text{ProcTime} \times \text{Number of Road} \quad (4.2)
 \end{aligned}$$

●DO read

$$\begin{aligned}
 \text{FwIn} &= \max(\text{AM_FwOut}, \text{Pre_EwOut} - 10\mu\text{sec}) \\
 \text{EwIn} &= \max(\text{AM_EwOut}, \text{Number of Road} \times \text{InTime} + \text{FwIn}) \\
 \text{FwOut} &= \text{FwIn} + \text{Delay} \\
 \text{EwOut} &= \max(\text{FwOut} + \text{ProcTime} \times \text{Number of Road}, \text{EwIn} + \text{Delay}) \quad (4.3)
 \end{aligned}$$

●TF

$$\begin{aligned}
 \text{FwIn} &= \max(\text{DOread_FwOut}, \text{Pre_EwOut} - 10\mu\text{sec}) \\
 \text{EwIn} &= \max(\text{DOread_EwOut}, \text{Number of Hit} \times \text{InTime} + \text{FwIn}) \\
 \text{FwOut} &= \text{FwIn} + \text{Delay} \\
 \text{EwOut} &= \max(\text{FwOut} + \text{ProcTime} \times \text{Number of Fit}, \text{EwIn} + \text{Delay}) \quad (4.4)
 \end{aligned}$$

4.4.2 高ルミノシティ下における処理時間

処理時間の検討では表 4.9 に記載するサンプルを使用して処理時間の評価を行う。ルミノシティ $1 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ は FTK 実装予定である 2014 年を想定し、ルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ は 2020 年の予定最大値から更に大きな場合を想定している。予定値より大きなルミノシティで FTK が動作可能であれば、実際に観測される如何なる瞬間ルミノシティにおいて動作可能であると確証するためである。また、 $\text{VBF}H \rightarrow \tau\tau$ 事象を使用する理由は τ 粒子など飛跡を多く生む事象でかつ、我々が取得したい物理現象の 1 つであることから確実に取得できるかを検証するためであり、DiJet の使用理由は高ルミノシティ下では QCD 事象の DiJet が支配的でありランダムで取得

した場合の多くは QCD 事象であることから、常に FTK が動作可能か検証することに注目し使用した。SS サイズは 4.3 節までと同じく Pixel の ϕ 方向 24 ピクセル (1.2mm), z 方向 36 ピクセル (14.4mm), SCT のストリップ数 20 本 (1.6mm) である。

表 4.9: データセット

事象サンプル	ルミノシティ [$\text{cm}^{-2}\text{sec}^{-1}$]
VBF $H \rightarrow \tau\tau \rightarrow lh$	1×10^{34}
VBF $H \rightarrow \tau\tau \rightarrow lh$	3×10^{34}
Di Jet	1×10^{34}
Di Jet	3×10^{34}

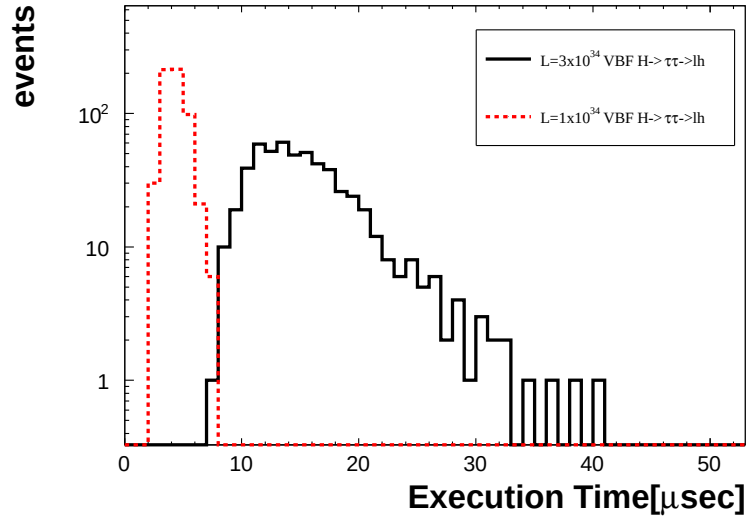
この条件下で処理時間の結果を図 4.41 と図 4.42 に記載し、それぞれの平均値を表 4.10 に記載する。また、図 4.43 と図 4.44 はルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ 下における事象ごとの各機能の処理時間を表した図である。

表 4.10: 処理時間の平均値

	ルミノシティ $1 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$	ルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$
VBF $H \rightarrow \tau\tau \rightarrow lh$	$4.30 \pm 0.04 [\mu\text{sec}]$	$15.7 \pm 0.2 [\mu\text{sec}]$
DiJet	$4.36 \pm 0.04 [\mu\text{sec}]$	$16.7 \pm 0.3 [\mu\text{sec}]$

● 考察

図 4.41 と図 4.42 から両事象共に 2020 年のルミノシティを上回る値 $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ においても約 $40 \mu\text{sec}$ 以下で安定して動作が可能であるということが結論付けられる。また、それぞれの平均値は VBF で $15.7 \pm 0.2 \mu\text{sec}$, DiJet では $16.7 \pm 0.3 \mu\text{sec}$ となり、ほぼ同程度と言える。実際、図 4.43 と図 4.44 を参照すると各事象、各機能に処理時間の揺らぎはあるものの一定以内に処理できている。一方、高ルミノシティ $3 \times 10^{34} [\text{cm}^{-2}\text{sec}^{-1}]$ において分布が広がる要因は処理が遅くなる程の信号量が入ってきたためであるが、遅くなる事象においても $40 \mu\text{sec}$ 以内で処理できているので DeadTime は生じていない。もし DeadTime を生む、つまり処理ができない場合は処理時間を蓄積するため分布が x 軸に水平となるが、今回は最大でも約 $40 \mu\text{sec}$ 以下であるので、現状のデザインで十分な処理能力があると結論づけられる。但し、最高ルミノシティでは処理限界の Road 数や Track 数に近い値を算出し処理時間に広がりが生じているため、今後新たなアルゴリズムの追加によって数が増加する場合には、処理時間の要請からスペックの再検討や追加アルゴリズムの機能制限が必要になることに注意する。

図 4.41: VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象の処理時間

現在の FTK のデザインで最大ルミノシティ時でも 40 μ sec 以内で処理が可能

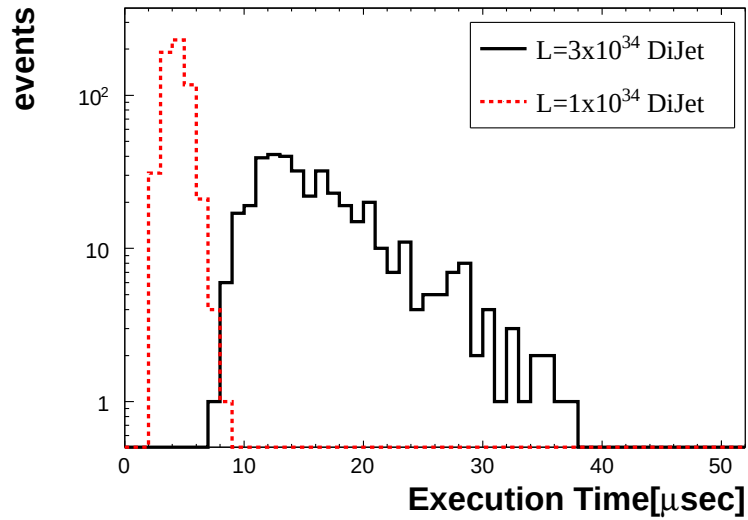


図 4.42: DiJet 事象の処理時間

現在の FTK のデザインで最大ルミノシティ時でも 40 μ sec 以内で処理が可能

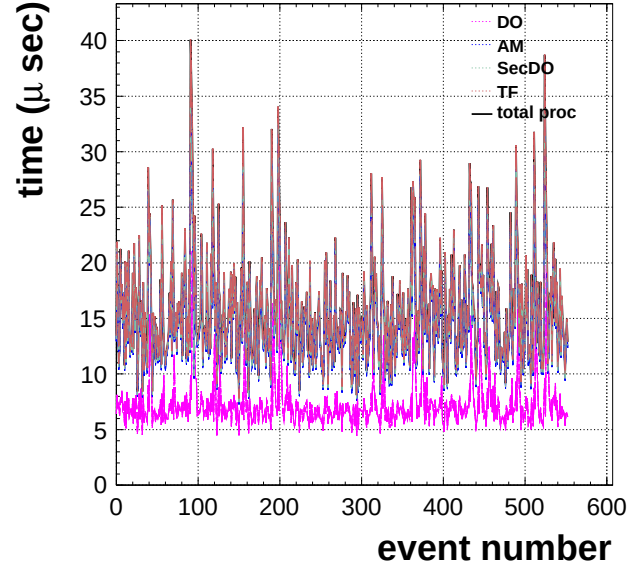


図 4.43: VBF $H \rightarrow \tau\tau \rightarrow lh$ における各事象ごとの各機能での処理時間
事象ごとに処理時間が増加することなく一定時間で処理可能

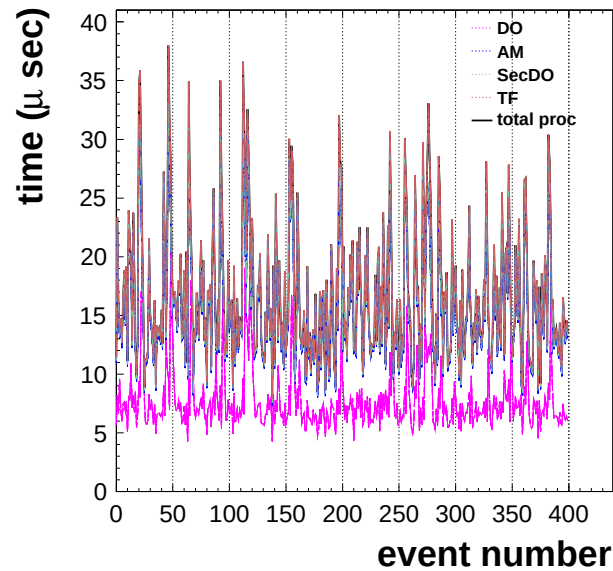


図 4.44: DiJet における各事象ごとの各機能での処理時間
事象ごとに処理時間が増加することなく一定時間で処理可能

5 今後の展望

5.1 性能評価, FTK 実装前の展望

モンテカルロによるシミュレーションを本論文では行った。しかし、実機に実装するにはシミュレーションの妥当性をより確証した上でシミュレーションによる評価で最適化したスペックや SS サイズを使用しなければならない。そこで FTK 全体を実装する前に、FTK の一部分を挿入しデータを取得するテスト (Vertical Slice Test) が 2012 年から開始予定である。図 5.1 と図 5.2 が Test に使用する機能と Board である。AM と TF の部分を 2011 年 9 月に実験終了したアメリカ、フェルミ研究所の CDF 実験で FTK の前身である SVT(Semiconductor Vertex Trigger) にて使用されていた AM チップと GigaFitter(TF の前身) を使用する。動作自体は FTK と変わらず、スペックが古いものであること以外には動作保障されたものである。その Board に対し、ATLAS 用のパターンを記憶させてデータを取得し解析する予定である。

そこで、このテストで得た結果と比較するためにも、取得する実データを入力信号として本論文で用いたシミュレーションで算出される Road と Track を検討する必要がある。これによって実機のテストとシミュレーションでのテストを比較だけでなく、シミュレーションへ還元し、現在より確証を持った状態で FTK 全体の機能を再評価することが可能である。

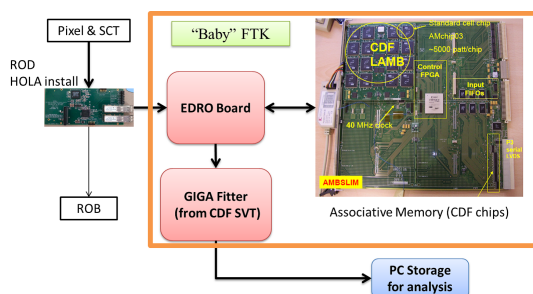


図 5.1: Vertical Slice Test に使用する Baby FTK

AM,TF には CDF のものを使用

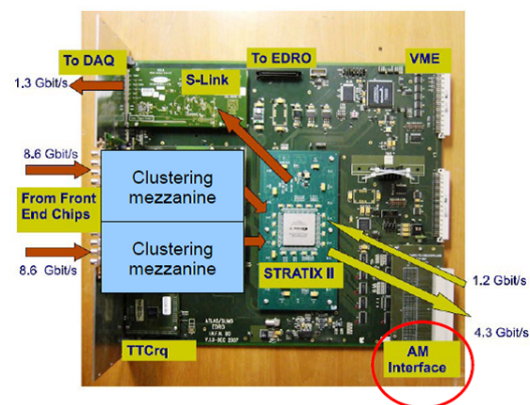


図 5.2: EDRO(Event Dispatch and Read-Out) Board

4.1 節で述べたようにシミュレーションの改良し、2011 年度の実データを用いて算出した Track の分布を示すのが図 5.3 である。使用したデータは 2011 年 6 月 10 日～11 日の MinimumBias Trigger にて得られたものである。この時の FTK が再構成した Track が図 5.3 の赤線であり、黒線は同じデータセットにてオフラインで再構成された Track の数である。この図 5.3 から明らかに FTK は Track を再構成できていない。

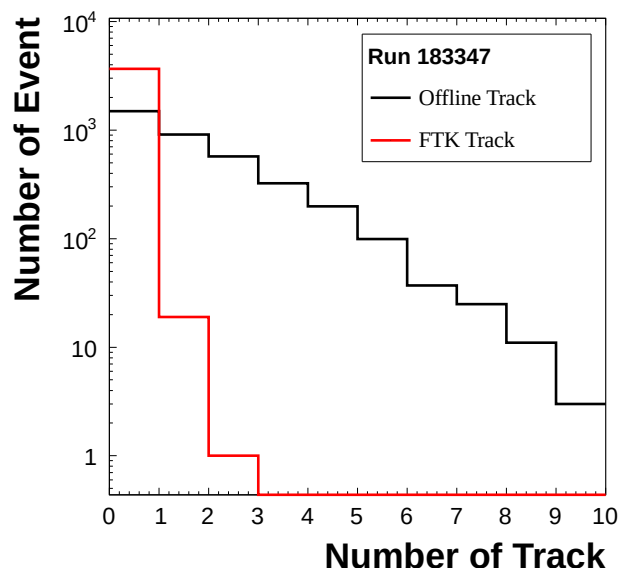


図 5.3: Run183347 MinimumBias Trigger における FTK とオフライン Track 数

現状では取得データで FTK による再構成が出来ていない

原因を追究するために、1 つの Road に対して Hit が何層存在していたか調べたものが図 5.4 である。赤線が上記の実データ、黒線が事象再構成率が高いシングルミュオン事象におけるモンテカルロデータの結果である。FTK の Road は少なくとも 6 層以上存在すればパターンとマッチしたと認識するので、シングルミュオンでは予想通り 6 層以上の事象が約 70 % を占める。一方、実データでは図から信号が 4 層以下に約 85 % を占めており、パターンと認識されていない。

パターン認識が出来ていない理由として以下の 2 つが考えられる。1 つ目は、FTK Simulation のフレームワークが不完全であることである。モンテカルロである RDO ファイルから検出器の信号情報を取り出すことは成功しているが、バイトストリームデータである実データでは信号抽出時における動作を確証していない。すなわち、検出器の位置情報がずれているためにパターンを認識していないと考えられる。この場合は早期改良・解決を行う必要がある。

2 つ目として、ビームポジションによる影響が考えられる。現在使用するパターンはビームポジションを絶対座標系で原点 (0,0,0) とし、 z 方向のみランダムにずらした時のシングルミュオン事象を使用して生成している。しかし現実のビームポジションは図 5.5～図 5.6 に示す通り xy 方向にもずれが生じているので、Track の始点が原点と異なりパターン内の信号数が減少する可能性

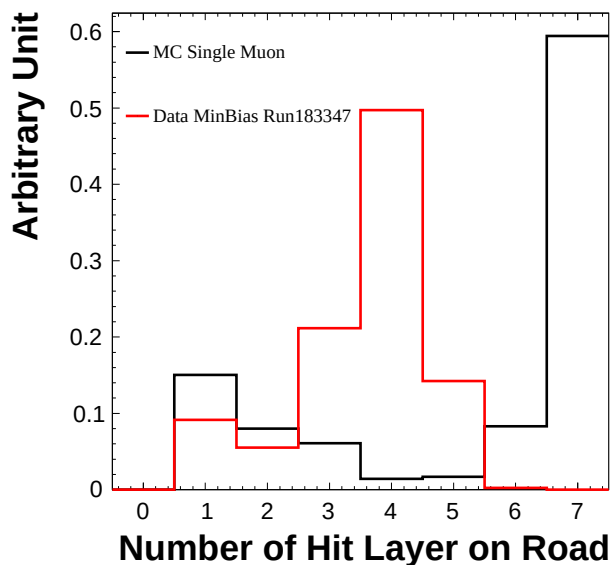


図 5.4: 1Road に対する荷電粒子が信号を残した層の数

取得データはモンテカルロと異なりビームポジションがズレているためパターンとマッチしていない

がある。

この問題を解決するには、実データ若しくはビームポジションをずらした事象でパターンを生成する必要がある。また、パターンの生成後はビームポジションのずれ幅によってパターン認識数や飛跡再構成率がどの程度変化するか調査する必要がある。

今後はこれらの原因や調査を追求し、更なるパターンの最適化を行った上で FTK を実装しなければならない。

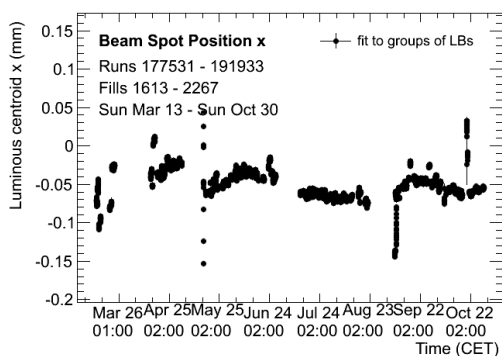


図 5.5: 2011 年度におけるビームの x 方向の分布

常に 0 からズレが生じている

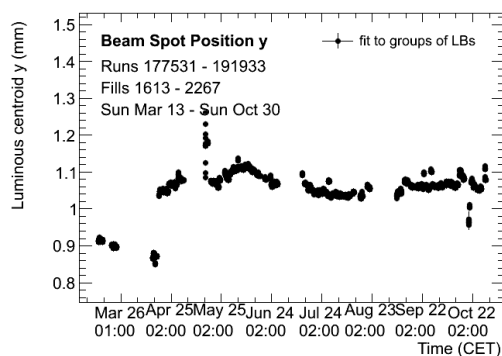


図 5.6: 2011 年度におけるビームの y 方向の分布

常に 0 からズレが生じている

5.2 FTK 実装後の展望

FTK を導入予定の 2014 年度では ATLAS の衝突エネルギーは 14TeV, 瞬間ルミノシティは $1 \times 10^{34} [cm^{-2} sec^{-1}]$ となる予定である。実装後は FTK の動作管理と FTK を使用した物理解析が必要となる。

第一に動作管理の前段階として必要なことは FTK を使用した場合のトリガーメニューの決定である。FTK を導入した場合, ジェットトリガーだけでなく LVL2 の段階で b ジェットトリガーなど高度な情報を使うことでトリガー閾値を上昇させずに事象を取捨選択することが可能になると考えられる。例として図 5.7 に LVL2 で 4 ジェットトリガーを要請した場合と, 4 ジェットトリガーに b ジェットトリガーを追加して要請した場合の閾値とイベントレートの相関図を示す。100Hz のイベントレートでは, 4 ジェットトリガーの閾値が 60GeV のところを, 2b ジェットトリガーの要請で 35GeV 程度に減少可能であることが示唆される。この例のように, 他のトリガーも同様に FTK を使用したトリガーアルゴリズムの検討を行う必要があり, 限られた記憶容量のうち適切な配分でトリガーオペレーションを行うことが必要である。

また, 実データの取得から常に FTK の動作管理を行う必要がある。例としては再構成率の算出である。図 5.8 は 2010 年度のデータを使用して LVL2 と EventFilter における Track の再構成率の P_T 依存性を算出したものであり, 立ち上がりが緩やかな場合は再構成率を一定に保つためにトリガー閾値を上昇させなければならない。それは事象選択を狭めることに繋がるため, 立ち上がりが急で再構成率を良くする必要がある。したがって FTK 導入後も, このような再構成率の立ち上がりや再構成率を管理し, より再構成率の高いパターンを生成方法の追究を行う必要がある。

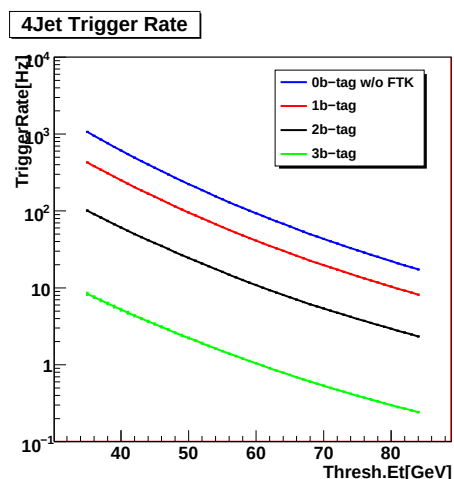


図 5.7: LVL2 における 4 ジェットトリガーとトリガーレート

イベントレートを一定値とすると

b ジェットトリガーを要求すると閾値が減少する

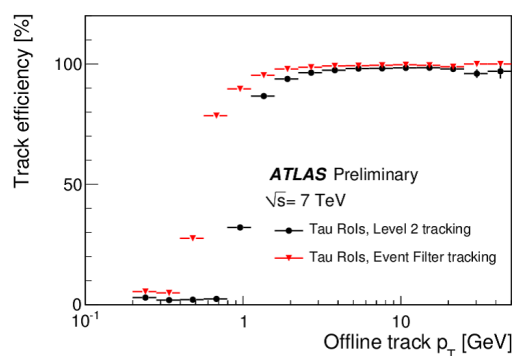


図 5.8: LVL2, EF トリガーにおける飛跡再構成率の立ち上がり

立ち上がりが急である方が良い

第二に行うべき項目は FTK による物理解析への貢献である。例えば FTK 導入により、上記のトリガメニューの選定を行った後は b ジェットや τ 粒子の同定が可能となることで Higgs 粒子から崩壊した b ジェットや τ 粒子事象を捕えることが可能になると考えられる。現状の τ 粒子や b ジェットの崩壊モードは QCD 事象がバックグラウンドになるため、オンラインだけでなくオフラインの解析段階でも選別することができず Higgs 粒子の測定に必要なデータ量となっていない (図 5.9)。しかし、FTK を導入することにより事象をオンラインの時点で多く取得することができ測定が可能となる。これにより Higgs 粒子の発見後に必要となる Higgs 粒子の湯川結合定数などの詳細測定が可能となる。湯川結合定数は標準理論の Higgs 粒子であれば結合定数が図 5.10 のように比例関係となる。このような測定には直接 τ 粒子や b ジェットに崩壊する事象を測定する必要があるが、FTK の導入により測定可能となると考えられる。

以上のように、実装後は FTK の動作だけでなく今まで困難であった物理解析を行うことが可能となり、新しい物理に貢献することが期待される。

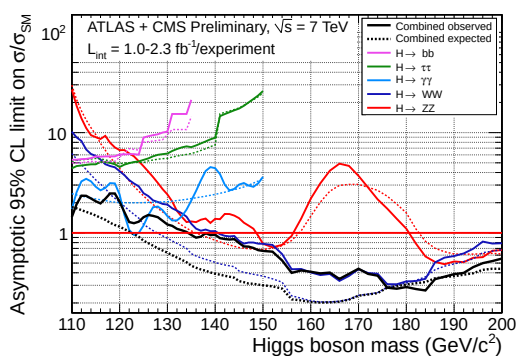


図 5.9: 2.3fb^{-1} における 95 % Confidence Level の Cross Section Limit

QCD 事象がバックグラウンドとなる過程では感度が良くない

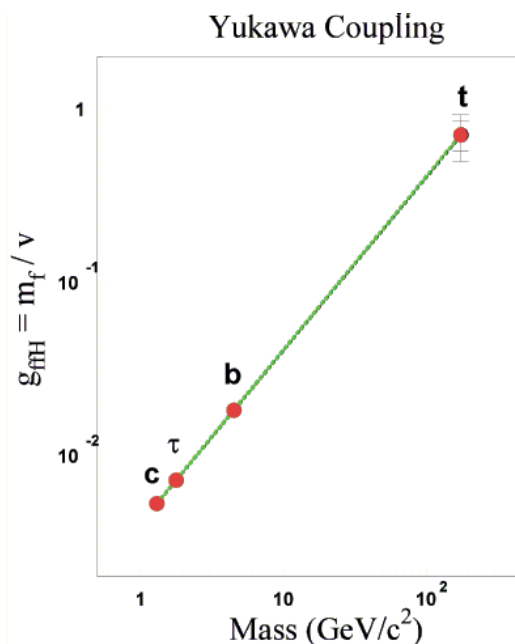


図 5.10: 湯川結合定数とフェルミオンの質量との関係

Higgs 粒子が標準模型由来か判断する場合に必要

6 纏め

本研究では FTK シミュレーションを Athena ソフトウェア上へ導入し、動作テストを行った。初期段階の導入は完成しており、今後の開発では Grid で使用できるスクリプトの開発とユーザーが使用するデータファイル形式の選定が必要である。

更に、2012 年度には FTK の一部を ATLAS 検出器に実装し、動作テストを行う予定であることから、最終実装時に使用予定のパターンサイズと現在のエネルギー・ルミノシティレベルで FTK の再構成能力とオフラインの性能との比較評価を行った。事象再構成率はオフラインとの比で 80 % 以上を誇ることが判明した。また、インパクトパラメータの分解能については $150 \mu\text{m}$ 程度と b ジェットの同定に十分な値であり、オフライン並の能力を発揮することができている。

横運動量の分解能では低運動量用域では FTK による分解能は 3 % 程度であり、ほぼオフライン並の分解能であるのに対し、高運動量領域では分解能がオフラインより 3 倍悪くなることが判明した。以上から今後は高運動量領域における運動量分解能の向上のために、より良いパターンの生成方法や Super Strip サイズの最適化が必要であることが分かった。

最後に実装後、2020 年到達予定の最大ルミノシティ以上の高輝度下における VBF $H \rightarrow \tau\tau \rightarrow lh$ 事象の再構成能力は、ルミノシティの上昇、つまり信号量の上昇につれて $69.6 \pm 0.1 \%$ から $27.9 \pm 0.3 \%$ に悪化しており、今回使用した Super Strip サイズでは高ルミノシティ下において信号処理能力が足りておらず改善の必要性があるということが分かった。一方処理時間の観点から見ると、高ルミノシティ下で得たい事象、ランダムな QCD 事象においても常に約 $40 \mu\text{sec}$ 以下で動作可能であることが示唆された。

今後はシミュレーションと実データを用いて Super Strip サイズの最適化、パターンの改善やクラスターサイズの最適化を行い、処理時間との兼ね合いを見計らいながら高ルミノシティ下でより精度の良いデザインを開発することが必要である。そして、より精度の良い FTK を実装し、トリガーシステムの改善を行うことで新しい物理現象の取得・解析結果に貢献することが期待される。

付録 A FTK シミュレーションの使用方法

以下に FTK シミュレーションの使用方法, Option の説明を行う。但し, Athena 版シミュレーションは現状開発段階のものであり変更される可能性がある。また, standalone 版シミュレーションは完成しており解析用に使用するものである。

パッケージの取得～コンパイル

lxplus にログインし, athena のセットアップを行う。(以下コマンド例を\$ ~で記載する。全て bash で行う。)

```
$ export AtlasSetup=/afs/cern.ch/atlas/software/releases/17.0.3/AtlasSetup
$ alias asetup='source $ AtlasSetup/scripts/asetup.sh'
$ asetup 17.0.3 -testarea= /path/17.0.3/
```

testarea 下でパッケージを 3 つ取得

```
$ cmt co Trigger/TrigFTK/TrigFTKPool
$ cmt co Trigger/TrigFTK/TrigFTKSim
$ cmt co Trigger/TrigFTK/TrigFTKAna
```

TrigFTKPool パッケージをコンパイル

```
$ cd $TestArea/Trigger/TrigFTK/TrigFTKPool/cmt
$ cmt config
$ source setup.sh
$ gmake
```

TrigFTKSim の athena 版 FTK シミュレーションのコンパイル

```
$ cd $TestArea/Trigger/TrigFTK/TrigFTKSim/cmt
$ gmake root_algdictionary このコンパイルはパッケージ取得時の一度だけで良い。
$ gmake
```

TrigFTKSim の standalone 版 FTK シミュレーションのコンパイル

```
$ cd $TestArea/Trigger/TrigFTK/TrigFTKSim/standalone
$ source bootstrap.sh
$ source ../grid/dependencies.sh
$ gmake -f Makefile.grid.mk
```

src ファイルや header ファイルを書き換えた場合は, その書き換えたパッケージの cmt 下で gmake コマンドを入力すればよい。また, standalone 版シミュレーションを grid に job を投げる場合は grid 上でコンパイルされるので, standalone のコンパイルは必要ない。TrigFTKAna については, lxplus 上ではコンパイルできないが, 環境設定上必要なのでパッケージを取得する必要

がある。

Athena 版シミュレーションコマンド (現在, 開発段階)

今後仕様変更の可能性はあるが, 基本的なコマンドラインは変わらないと思われるので記載する。
Reconstruction で RDO ファイルから ESD ファイルに変更する Reco_trf.py を使用する。

```
$ cd $TestArea/
```

```
$ mkdir run
```

コマンドを走らせると色々なファイルが出来上がるため, run ディレクトリ内でコマンドを走らせることを推奨。

```
$ cd run
```

Reco_trf.py をダウンロード (athena は予めセットアップしておく)

```
$ get_files -jo Reco_trf.py
```

実行コマンド

```
$ Reco_trf.py inputRDOFile=/path/rdo.pool.root outputESDFile=
./myESD.pool.root autoConfiguration=everything preExec=rec.doCalo=False,,
rec.doMuon=False,,rec.doJetMissingETTag=False,,rec.doEgamma=False,,
rec.doMuonCombined=False,,rec.doTau=False,,rec.doTrigger=False,,
'rec.UserAlgs=["TrigFTKSim/FTK_jobOptions_forTransform.py"]' maxEvents=2
```

<変更に必要な Option の説明>

inputRDOFile : input する RDO ファイル名 いずれは Grid 上のコマンドに変更

maxEvents : 再構成する事象数

'rec.UserAlgs=["TrigFTKSim/FTK_jobOptions_forTransform.py"]' : 走らせる JobOption。中身は下記に記載する。

< FTK_jobOptions_forTransform.py の説明。「」が説明文である>

```
import AthenaPoolCnvSvc.ReadAthenaPool
from AthenaCommon.AlgSequence import AlgSequence
alg = AlgSequence()
# -----
# FTK algorithm inclusions
「Athena とやり取りする FTKCode を使用可能にする」
# -----
from AthenaCommon.AppMgr import ToolSvc
from TrigFTKSim.TrigFTKSimConf import *
print "Add FTK_SGHitInput tool"
FTKSGHit = FTK_SGHitInput()
ToolSvc += FTKSGHit
```



```

print "Add FTK_SGRoadOutput tool"
FTKSGRoadOut = FTK_SGRoadOutput()
ToolSvc += FTKSGRoadOut
print "Add FTK_SGTrackOutput tool"
FTKSGTrackOut = FTK_SGTrackOutput(OutputLevel=DEBUG)
FTKSGTrackOut.StoreIncomplete = True
ToolSvc += FTKSGTrackOut
print "Add FTK_RoadMarketTool"
FTKRoadMarket = FTK_RoadMarketTool()
FTKRoadMarket.SaveRoads = True
ToolSvc += FTKRoadMarket
print "Add FTKDetector tool"
FTKDet = FTKDetectorTool()
FTKDet.doBadModuleMap = False # True to produce or False to read an existing one
「True の場合は BadModule 有のシミュレーションになる」
FTKDet.useBadModuleMap = False # may disable the use of dead modules mapping
FTKDet.BadSSMapPath = "../config/bad_modules/badSSmap.bmap"
FTKDet.FTK_BadModuleMapPath = "../config/bad_modules/badModulemap_FTK.bmap"
FTKDet.ATLAS_BadModuleMapPath = "../config/bad_modules/badModulemap_ATLAS.bmap"
FTKDet.FTK_BadModuleMapPath = "../config/bad_modules/badModulemap_FTK.bmap"
FTKDet.ATLAS_BadModuleMapPath = "../config/bad_modules/badModulemap_ATLAS.bmap"
ToolSvc += FTKDet

「SS マップや BadModule マップなどのパス設定」
FTKRoadFinder = FTKRoadFinderAlgo( "FTKRoadFinderAlgo" , OutputLevel=INFO)
FTKRoadFinder.NBanks = 8
FTKRoadFinder.NSubRegions = 1

FTKRoadFinder.pmap_path = "../config/map_file/raw_7L.pmap"
FTKRoadFinder.pmapunused_path = "../config/map_file/raw_7L_unused.pmap"
FTKRoadFinder.rmap_path = "../config/map_file/raw_7L.rmap"
FTKRoadFinder.ssmmap_path = "../config/ss_file/raw_50x64x144.ss"
「SSsize を使用するパターンと等しくする」
FTKRoadFinder.ssmmapunused_path = "../config/ss_file/raw_24x20x36unused.ss"
# FTKRoadFinder.badmap_path = "badModulemap_FTK.bmap" # if = "" dont use WC
# FTKRoadFinder.ssmmapts_path = "../config/ss_file/raw_12x10x36.ss"
# FTKRoadFinder.banks = ["0; bank0.patt; -1;","1; bank1.patt; 1000;"]

```

```

FTKRoadFinder.UseTSPBank = False
FTKRoadFinder.DBBankLevel = 1
FTKRoadFinder.TSPSimulationLevel = 0
FTKRoadFinder.TSPMinCoverage = 1
FTKRoadFinder.MakeCache = False
FTKRoadFinder.CachePath = "bankcache.root"
FTKRoadFinder.DoRoadFile = False # Default False:Save SG True: FTKRoad-
Finder.SaveRoads = False
「Road パターンのパス設定」
if FTKRoadFinder.UseTSPBank :
FTKRoadFinder.bankpath = ["/home/volpig/small_bank/patterns_raw_7L_24x20x36_16M.reg"
+str(FTKRegion)+"_sub"+str(FTKSubRegion)+".patt.db"]
else :
FTKRoadFinder.bankpath = [ ];
for reg in xrange(0,8):
FTKRoadFinder.bankpath.append("/afs/cern.ch/user/c/chiba/public/baseDS_v16/patt/
pattgen_raw_7L_50x64x144_1M.reg"+str(reg)+"_sub0.patt.bz2")
FTKRoadFinder.bankregion = range(0,8)
FTKRoadFinder.bankpatterns = [ ]
for reg in xrange(0,8):
FTKRoadFinder.bankpatterns.append(2000000)
alg += FTKRoadFinder

# ----- Track Fitter -----
FTKTrackFitter = FTKTrackFitterAlgo( "FTKTrackFitterAlgo" , OutputLevel=INFO)
FTKTrackFitter.NBanks = 8
FTKTrackFitter.NSubRegions = 1
FTKTrackFitter.pmap_path = "../config/map_file/raw_7L.pmap"
# set bank path
「TrackFitter のパターンのパス設定」
FTKTrackFitter.bankpath = [ ]
FTKTrackFitter.bankregion = [ ]
FTKTrackFitter.banksubregion = [ ]
for reg in xrange(0,8) :
FTKTrackFitter.bankpath.append("/afs/cern.ch/user/c/chiba/public/baseDS_v16/gcon/
corrgen_raw_7L_1M.reg"+str(reg)+"_sub0.gcon.bz2")
FTKTrackFitter.bankregion.append(reg)

```

```

FTKTrackFitter.banksubregion.append(0)
# set parameter
「TrackFitter のパラメータ設定。基本的には Default 値を使用」
FTKTrackFitter.Chi2Cut = 17 # Default 17
FTKTrackFitter.Chi2Cut_Maj = 14 # Default 14
FTKTrackFitter.Chi2Cut_VetoMaj = 0 # Default 0
FTKTrackFitter.Chi2DofCut = 4 # Default 0
FTKTrackFitter.NCoords = 10 # 10 for 7L, 11 for 8L, 14 for 11L
FTKTrackFitter.HitWarrior = 2 # Default 2
FTKTrackFitter.KeepRejected = 0 # more than 0 enable HW debug feature
FTKTrackFitter.FitRemoved = 0 # more than 0 enable debug feature
FTKTrackFitter.HWNDiff = 4 # number of coordinates to define a not match (HW uses
inverse logic)
FTKTrackFitter.MaxNcomb = 1000 # maximum number of combinations to fit in 1 road
FTKTrackFitter.MaxTrkout = 2000000 # maximum number of output tracks in a sub-region
FTKTrackFitter.loadHWConf_path = "../config/hwsdev_file/raw.10.hw" # HW thresholds
FTKTrackFitter.doTrackFile = False
FTKTrackFitter.OutDir="."

alg += FTKTrackFitter
# ----- Merger -----
「Merge アルゴリズムの設定。変更しなくて良い」
FTKMerger = FTKMergerAlgo( "FTKMergerAlgo" , OutputLevel=DEBUG)
FTKMerger.force_merge = True # true only for tests or well known reason
FTKMerger.useStandalone = False
FTKMerger.SingleProces = True
FTKMerger.doMerging = True
FTKMerger.pmap_path = "../config/map_file/raw_7L.pmap"
FTKMerger.NRegions = 8
FTKMerger.NSubRegions = 1
# Set Hit Warrior parameter
FTKMerger.NCoords = 10 # 10 for 7L, 11 for 8L, 14 for 11L
FTKMerger.HitWarrior = 2 # Default 2
FTKMerger.KeepRejected = 0 # more than 0 enable HW debug feature
FTKMerger.HWSDEV_FILE = "../config/hwsdev_file/raw.10.hw" # HW thresholds
alg += FTKMerger

```

Standalone 版シミュレーションコマンド

1.Lxplus 上の Local の場合 (動作テスト用, Region 1つのみで動作テスト)

standalone のコンパイル後, config ディレクトリにて,

```
$ ../standalone/roadfinder roadfinder_onesub_tmpl.in 0 0
```

にて RoadFinder のシミュレーションを実行。roadfinder_onesub_tmpl.in の中身が設定ファイルなので, 上記の Athena 版を参照して Path の変更をする。また, Input ファイルは RDO から抽出した bz2 ファイルを使用することに注意する。bz2 ファイルは以下のリンクに Grid 上の名前が掲載されている。

<https://twiki.cern.ch/twiki/bin/viewauth/Atlas/FastTrackerHighLumiSamples>

次に, TrackFitter の実行は以下のコマンドである。

```
$ ../standalone/trackfitter trackfitter_onereg_tmpl.in 0 0
```

それぞれ, Road の入った root file と Track の入った root file が出来上がる。

2.Grid に Job を投げる場合 (FTK の解析用。利用推奨)

Athena のセットアップはせず Grid の設定を行う。以下を参照。

```
$ source /afs/cern.ch/project/gd/LCG-share/current/etc/profile.d/grid-env.sh
```

```
$ voms-proxy-init -voms atlas -valid 100:00
```

```
$ source /afs/cern.ch/atlas/offline/external/GRID/ddm/DQ2Clients/setup.sh
```

```
$ source /afs/cern.ch/atlas/offline/external/GRID/DA/
```

```
panda-client/latest/etc/panda/panda_setup.sh
```

設定後, TrigFTK/TrigFTKAna/下で

```
$ source setup.sh
```

を実行し, 環境設定を行う。次に, TrigFTK/TrigFTKSim/scripts 下で Job コマンドを流す。以下のコマンドで, RoadFinder~TrackFitter まで同時に Grid 上で動作する。

```
$ ./ftk.py -q grid -mode ftk -username chiba -site ANALY_MWT2
-d raw_7L_24x20x36_v14VBFtautaulh_RECO3E34.grid
-m16 -t user.gvolpi.raw_7L_24x20x36_30Mx300barrel_16NSubs.2011_11_28_12_00_00/
-o baseDS=user.gvolpi.FTKV16_base7L16M_20110905
-o MAXPATTERNS=4000000 -n 10000 -nFilesPerJob 50 -nJobs 4
-inDS user09.MelShochet.v14VBFtautaulh.set01_RECO3E34.rset1.20091114
-o IGNORE_SIGSEGV,strict_error_check -o COPY_WRAPS_FROM_DCACHE,
COPY_BANKS_FROM_DCACHEA
```

< Option の説明 >

詳細は./ftk.py -q grid でコマンドラインにて記載される。以下は変更が必要などのみ述べる。

-d raw_7L_24x20x36_xxx : Output 名となる。SS サイズを変える時は 24x20x36 を, xxx には名前を入れる。

-t xxx : パターンバンク名。Grid 上にあるものを使用

-o baseDS=xxx : 線形一次近似の定数項。Grid 上にあるものを使用。

-n xxx : イベント数

-nFilesPerJob xxx -nJobs yyy : xxx は 1Job で読み込むファイル数, yyy は Job 数を示す。

Grid 上のエラーの関係上 xxx は最大でも 100 程度が好ましい。

-inDS xxx : InputDataSet 名。Grid 上にあるファイル名を指定。

次に、上記のシミュレーションでは SubRegion ごとにアウトプットを生成するため、1つのファイルに統合するコマンドが以下の Merge コマンドである。上記の Panda の Job が終了した後、下記のコマンドを同様に流す。

```
$ ./ftk.py -q grid -mode merge_ftk -username chiba -site ANALY_MWT2 -d
raw_7L_24x20x36_v14VBFtautaulh_RECO3E34.grid
-m16 -t user.gvolpi.raw_7L_24x20x36_30Mx300barrel_16NSubs.2011_11_28_12_00_00/
-o baseDS=user.gvolpi.FTKV16_base7L16M_20110905
-o MAXPATTERNS=4000000 -n 10000 -nFilesPerJob 100 -nJobs 100
-inDS user09.MelShochet.v14VBFtautaulh.set01_RECO3E34.rset1.20091114
-o ftkDS=user.chiba.raw_7L_24x20x36_v14VBFtautaulh_RECO3E34.grid.track/
-o absetacut=1.0,absd0cut=2.0 -o IGNORE_SIGSEGV,strict_error_check
-o COPY_WRAPS_FROM_DCACHE,COPY_BANKS_FROM_DCACHE
```

< Option の説明 >

基本は上記の Option と同じである。変更するのは下記の一点のみである。

-o ftkDS=xxx : xxx に上記のシミュレーションのアウトプット名を使用

inDS, -d, -t 等は全て、シミュレーション時と同じものを使用しなければならない。

付録 B Simulation Code

以下に FTK シミュレーションを実行した場合における Code 内の主要な Function について説明を記載する。

standalone/road_finder.cc

main() 内の read_commands() にて設定ファイルから読み込んだ変数を set と呼ばれる Function を用いて変数やパスの指定を行う。

設定終了後, main() 内の nextEvent() にて src/RoadFinder.cxx の nextEvent() を呼び出す。

この部分は Athena の場合, src/FTKRoadFinderAlgo.cxx 内の Initialize()~Execute() に相当する。Execute() では同じく src/RoadFinder.cxx の nextEvent() を呼び出しているため, nextEvent() から下流は一部部分を除き同じコードが走るように設計されている。

src/RoadFinder.cxx

m_datainput->readData() により, standalone の場合は FTK_RawInput.cxx, athena 版の場合は FTK_SGHitInput.cxx を読み込み Hit 情報を RoadFinder.cxx が用いる変数に詰める。

次に m_datainput->processEvent() により FTKDataInput.cxx を呼び出し, この Function の中で, 先ほど変数に詰めた Hit 情報を読み出し, atIclustering という Function でクラスター化後, Hit 情報を pmap と照らし合わせて Layer ごとに FTKHit ヘアドレスを代入する。そして, for(int ibank = 0; ibank != m_nbanks; ++ ibank) のループ内の pbank->passHits(hitlist) にて FTK_AMBank.cxx の passHits() を呼び出し, Hit 情報を SS サイズで照らし合わせ, Road を算出作業を行う。この部分は下記に示す。Road 算出後は m_roadoutput->eventEnd() にて RootFile へ書き出しが行われ, 1 イベントが実行されたことになる。

src/FTK_AMBank.cxx

passHits() がメインの Function であり, 中は sort_hits(), readout_hits(), routing(), am_in(), am_out() で構成されている。sort_hits() では FTKHit に代入された Hit 情報のうち OverLap 領域に含まれる同じ Hit 情報を 1 つに統合する。readout_hits~routing で SuperStrip に Hit 情報を変換する準備を行い, am_in にて Hit が存在する SuperStrip 情報を算出し変数に代入する。最後に am_out にて SuperStrip 情報の変数を順に読み出し, パターンと照らし合わせて一致した場合を Road として算出し変数に記憶させる。

以上の流れで Road を算出するシミュレーションを行われる。

次に TrackFitter のシミュレーションを下記に示す。ソースコードの構造は road_finder.cc から始まり, 各ソースコード, 各 Function で算出する上記の RoadFinder と同じ形を採用している。

standalone/track_fitter.cc

road_finder.cc 同様, main() 内の read_commands() にて設定ファイルから読み込んだ変数を設定し, nextEvent() にて src/TrackFitter.cxx の nextEvent() を呼び出し実行する。athena 版では track_fitter.cc の代わりに src/FTKTrackFitter.cxx が使用される。中身の Function は Initialize と Execute となっており roadfinder の時と同じである。

src/TrackFitter.cxx

m_roadinput->nextEvent() により Road を使用する変数の初期化を行う。更に processor_init(ibank) にて TrackFitter で算出するパラメータ変数の初期化を行い, m_roadinput->nextRoad(ibank) にて RoadFinder で算出した Road を Standalone 版では RootFile から, Athena 版では変数ポインターから読み出し, Road 1 つずつに対して processor(*cur_road) の Function 内で Track が算出される。processor() では, TF で第一に DO から Hit 情報を得るのと同様に, 最初の for ループで road.getNHits(p) の宣言以降, 1 つの Road に存在する Hit の個数を取得し, 各 Layer ごとに 1 つの Hit を選ぶコンビネーション数を算出する。次に, for(intcomb = 0; comb < (ncomb > m_max_ncomb?m_max_ncomb : ncomb); ++ comb) にてコンビネーションの取り方全てに対し, newtrk.setCoord(ix,(*position[ip])[0]) にて各 Layer の Hit 座標を取得後, current_bank->linfit(sector,newtrk) にて Track を算出する。linfit() は src/FTKConstantBank.cxx に記述されており, この Function の中で, 本文にも記載されている式 (3.1)

$$p_i \simeq \sum_{j=1}^{10} C_{ij}(x_j - x_{0j}) = \sum_{j=1}^{10} (C_{ij}x_j - q_j)$$

及び, 式 (3.2)

$$\chi^2 = \sum_{i=1}^5 \left(\sum_{j=1}^{10} C_{ij}x_j + q_j \right)^2$$

を用いて Track のパラメータと χ^2 を Hit 情報と予め算出した真の情報から算出される。

その他に記載されているコードは, FTK で追加を検討している Option のコードが主である。主に以上のコード部分が AM の Pattern Recognition と TF の Track Fit をシミュレーションするのに使用され, これらのコードを走らせることで Track が算出される。

参考文献

- [1] ATLAS Collaboration:”ATLAS DETECTOR AND PHYSICS PERFORMANCE Technical Design Report”,1999.
- [2] ATLAS Collaboration:”ATLAS High-Level Trigger Data Acquisition and Controls Technical Design Report”,2003.
- [3] ATLAS Photo Page
<http://www.atlas.ch/photos/index.html>
- [4] Aad et al:”The ATLAS Experiment at the CERN Large Hadron Collider”
J.Instrum.,Vol.3,p.S08003,2008.
- [5] Athena. AthenaDeveloperGuide-8.0.0, 2004
- [6] ATLAS FTK Group:”FTK:a hardware track finder for the ATLAS trigger Techniical Proposal”,2010.
- [7] Mauro Dell’Orso:”The Gigafitter in track reconstruction for the search of rare events at hadron colliders”,2008.
- [8] A.Annovi,G.Volpi, 他 FTK Group:”Linear Fit within Missing-Hit Roads in FTK-Sim”,2009.
- [9] ATLAS Level1-Trigger Group:”ATLAS Level1 trigger: Technical Design Report”,1998.
- [10] ATLAS Collaboration:”Combination of Higgs Boson Searches with up to 4.9 fb⁻¹ of pp Collision Data Taken at $\sqrt{s}=7$ TeV with the ATLAS Experiment at the LHC”,ATLAS-CONF-2011-163,2011.
- [11] ATLAS Beam Spot Web Page
<https://atlas-beamspot.cern.ch/webapp/myplots/>
- [12] ATLAS LuminosityPublicResults Homepage
<https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResults>
- [13] ATLAS HLT Tracking Public Results Homepage
<https://twiki.cern.ch/twiki/bin/view/AtlasPublic/HLTTrackingPublicResults>
- [14] 中村浩二:”LHC での Higgs 探索”,2011 年日本物理学会秋季大会口頭発表.
- [15] 中村浩二:”LHC-ATLAS 実験におけるタウレプトン対に崩壊するヒッグス粒子の探索”,2011 年日本物理学会秋季大会口頭発表.
- [16] 長島順清:”素粒子標準理論と実験的基礎”, 朝倉書店,1999.

謝辞

本研究にあたり、学部生から3年間数えきれないほどの指導をして下さった寄田浩平准教授に心から感謝を申し上げます。私が精神的に負い目を感じた時に近くに歩み寄って頂いた時には本当に救われました。時に褒め、時に叱り少しでも成長するようにと熱心に指導して下さい、3年前に比べて大きく成長したと思います。私は産業界に出ることになりますが日本のために使命感を持って仕事を成し続けてゆきたいと思います。本当に感謝致しております。

また、FTKの研究において色々な助言や私の質問に対応して下さいだった木村直樹助教、飲みながらも一緒に思考して頂いた永野間淳二次席研究員、PCの環境設備を整え研究と一緒に思考して下さいるばかりか、私が凹んでいる時にささやかに支えてくれた蛭名幸二助手のお三方には大変お世話になりました。心から感謝しております。

更にイタリアのピサ大学にて1か月間滞在した際に、生活面から研究面まで多忙の中、全てにおいて気遣って頂いたPaola Giannetti氏、FTKシミュレーションの構造、改良方法を熱心に教えて下さっただけでなく、色々な観光地や飲食店など生活面まで私の拙い英語に答えて下さったGuido Volpi氏、美味しいイタリア料理や歴史ある教会に連れて行って下さったAlberto氏とMauro Dell'Orso氏達には、初めての海外研究生活で不安な気持ちの中、温かく見守って頂き感謝致します。

そして、同期の岡本迅人君、鎌塚翔平君、桜井雄基君、長坂優志君、船越雄二郎君、三谷貴志君には数えきれないほどの迷惑をかけたばかりでなく、たくさんの心温かい気遣いに非常に感謝しています。バイトで時間のなかった私にも常に声をかけてくれ、楽しく一緒に飲んだり、ご飯を食べたりできて本当に嬉しかったです。また、特に鎌塚君には、イタリアでの共同生活も含め、解析の思考やプレゼンテーションの資料作りまで研究で多くのことを助けてくれて本当に感謝しています。彼がいたからこそ、研究が捗ったと思います。

上記の皆様が私に協力して頂いたからこそ、無事に修士2年間の研究生活を過ごすことができたと思っています。この場を借りてお礼を申し上げます。本当に有難う御座いました。

最後に、私が生まれてから24年間、毎日休みなく働き、苦しい時期がありながらも温かく支えてくれた両親に心から感謝します。今度は私が両親を支えてゆきたいと心から思います。