UNIVERSIDADE D
COIMBRA

João Carlos Macário Ribeiro

Hardware Tracking for the ATLAS experiment at the HL-LHC

Dissertação no âmbito do Mestrado Integrado em Engenharia Física, Área de Instrumentação, orientada pelo Professor Doutor José Ricardo Morais Silva Gonçalo e apresentada ao Departamento de Física da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Outubro de 2021

UNIVERSIDADE D
COIMBRA

Hardware tracking for the ATLAS experiment at the HL-LHC

João Ribeiro
Integrated Master's in Engineering Physics

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Department of Physics
Faculty of Science and Technology
University of Coimbra

Trabalho com orientação científica de:
Prof. Dr. Ricardo Gonçalo da Universidade de Coimbra

October 2021

Abstract

This Master Thesis presents work developed for the Hardware Tracking for the Trigger (HTT), a system proposed for the Phase-II Upgrade of the ATLAS experiment, at the Large Hadron Collider (LHC). The main project consisted in the study of a new tracking method, that uses the Hough Transform for road finding and a fitting procedure based on an idealized geometry of the detector.

Due to the high number and rate of proton-proton collisions expected at the High Luminosity LHC, the ATLAS trigger has to quickly decide if an event is saved for storage or not. To do so, it will need to perform charged particle tracking at a rate of 1 MHz. This is a tremendous task that has to be performed in the fastest way, with the best efficiency and quality possible, the lowest cost and a not so huge complexity. Finding the best solution is not easy, so various alternatives were proposed.

Recently, a solution based on commodity CPUs and software was chosen to perform tracking in the ATLAS trigger. However, the possibility of using hardware accelerators to assist in this task is being studied, to make use of their higher processing power. The method presented in this thesis could be implemented in FPGAs to quickly perform road finding and give an estimate of the ϕ_0 and q/p_T of tracks to the ATLAS Event Filter.

The results obtained show ϕ_0 and q/p_T resolutions and track finding efficiencies comparable to the HTT baseline system. The downside of the current version of the method is lower performance in $1 < p_T < 2$ GeV muons and higher number of fake tracks (about 36 times more than offline reconstruction) produced in $t\bar{t}$ plus 200 pile-up events.

Keywords: Tracking Trigger, ATLAS, HL-LHC, Hough Transform, linearized fitting

Resumo

Esta tese de mestrado documenta dois trabalhos realizados para o Hardware Tracking for the Trigger (HTT), um sistema proposto para a Fase-II da Upgrade da experiência ATLAS, no Large Hadron Collider (LHC). O projeto principal é um estudo sobre o desempenho de um novo método para reconstruir traços de partículas carregadas, usando a transformada de Hough para encontrar "roads" e um ajuste linear baseado numa geometria idealizada do detetor.

Devido ao elevado número e frequência de choques protão-protão esperados na fase de Alta Luminosidade do LHC, o trigger de ATLAS tem que decidir muito rapidamente se um evento é gravado ou não. Para isso, vai precisar de reconstruir traços a uma frequência de 1 MHz. Isto é uma tarefa com um enorme grau de dificuldade, que tem que ser executada o mais rapidamente, com melhor eficiência e qualidade, e menor custo e complexidade possíveis. Encontrar a melhor solução não é fácil, pelo que várias alternativas foram propostas.

Recentemente, uma solução baseade em CPUs e software comerciais foi escolhida para realizar esta tarefa. Contudo, existe a possibilidade de usar aceleradores de hardware para ajudar o trigger a reconstruir os traços. O método apresentado nesta tese pode ser implementado em FPGAs para rapidamente encontrar "roads" e dar uma estimativa do ϕ_0 e q/p_T dos traços ao trigger.

Os resultados obtidos mostram resoluções dos parâmetros e eficiência em encontrar os traços similares aos mais recentes do HTT. Os aspetos quem ficam aquém dos resultados de base são o desempenho em traços com momento transverso entre 1 e 2 GeV, e o número de traços falsos (cerca de 36 vezes mais que os reconstruídos por métodos offline) obtidos em amostras de $t\bar{t}$ mais 200 eventos acumulados.

Palavas chave: Reconstrução de traços, ATLAS, HL-LHC, Transformada de Hough, ajustes lineares

Acknowledgements

This work would not have been possible without the help of the following people and groups:

Ricardo Gonçalo, Thank you for being the best supervisor one could ask, for all your tireless help throughout the last year, and for making me elevate the quality of this thesis.

Francesca Pastore and Jahred Adelman, Thank you for welcoming me in the HTTSim group, guiding me in this project and patiently answering all the doubts I had.

HTTSim group, Thank you for all the support and help debugging GRID scripts.

ATLAS Portuguese group, Thank you for all the help in our weekly meetings.

I would also like to thank all my friends and family who supported me in these last years, specially my sister for all her love.

I highly appreciate Fundação para a Ciência e a Tecnologia for the partial funding of this project, and the ATLAS Collaboration, for letting me contribute to their amazing work.

Dedication

To Finita, the best mom in the world.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ALFA Absolute Luminosity For ATLAS 9

AM Associative Memory 11, 14, 17, 19, 27, 28, 38

AMTP Associative Memory Tracking Processor 17, 18

ASIC Application Specific Integrated Circuit 1, 11, 17, 27, 28, 38

ATCA Advanced Telecommunications Computing Architecture 18

CMS Compact Muon Solenoid 17, 27–29

CSC Cathode Strip Chamber 9

CTP Central Trigger Processor 10, 11

DC bits "Don't Care bits" 20, 21

EF Event Filter 14, 17, 18, 20, 23, 68

EM ElectroMagnetic 7, 8, 11

FCal Forward Calorimeter 8

FPGA Field-Programmable Gate Array 1, 11, 14, 19, 27–29, 40–42, 68

FTK Fast TracKer 11, 17, 27

gHTT global tracking 17, 18

HEC Hadronic End-cap Calorimeter 8

HL-LHC High Luminosity Large Hadron Collider 1, 12–14, 27, 28

HLT High Level Trigger 10, 11

HT Hough Transform xiv, 1, 28, 29, 38, 39, 42, 47, 51, 68

HTT Hardware Tracking for the Trigger 1, 2, 14, 17–19, 21, 23, 24, 27, 28, 31, 32, 68

HTTIF HTT InterFace 18

HTTSim Simulation of the HTT 1, 24, 31, 32, 39, 68

ID Inner Detector 5, 6, 8, 9, 11, 13, 27, 38

ITk Inner TracKer 13, 14, 17, 19–21, 25, 31, 32, 64

L0 Level-0 trigger 14, 17, 18, 24

L1 Level-1 trigger 10, 11

L1Calo Level-1 Calorimeter trigger system 10, 11

L1Muon Level-1 Muon trigger system 10, 11

L1Topo Level-1 Topological trigger system 10, 11

LAr Liquid Argon 7, 8

LHC Large Hadron Collider 1, 4

LUCID LUminosity measurement using Cherenkov Integrating Detector 9

MDT Monitored Drift Tube 9

MUCTPI Muon to Central Trigger Processor Interface 11

OR Overlap Removal xiv, xv, 20, 22, 25, 32, 47, 50, 62, 64, 65, 68

PRM Pattern Recognition Mezzanine 17–19

rHTT regional tracking 17, 18

RoI Regions of Interest 11, 17

RPC Resistive Plate Chamber 9

SCT SemiConductor Tracker 6

SSID SuperStrip IDentifier 19

SSTP Second Stage Tracking Processor 17, 18

TDAQ Trigger and Data Acquisition System 2, 10, 13, 14, 23

TDR Technical Design Report 18, 21, 23

TFM Track Fitting Mezzanine 17, 18, 20

TGC Thin Gap Chamber 9

TP Tracking Processor 18, 19

TRT Transition Radiation Tracker 6, 13

TTC Timing, Trigger and Control 11

WC "Wild Card" 20, 42

ZDC Zero-Degree Calorimeter 9

Chapter 1 Introduction

In particle collider experiments, it is very useful, and often a necessary requirement, to reconstruct the tracks of charged particles that emerge from the collisions. These are used to estimate the momenta, electrical charge and identity of particle, and the flavour of hadronic jets or corrections to their energy. The accurate reconstruction of tracks involves both the detector and readout electronics, as well as tracking algorithms. All of these are of fundamental importance for experiments such as ATLAS to fulfil their physics goals.

In 2018, the Large Hadron Collider (LHC) [1] had an average of 36 proton-proton collisions every 25 ns, going up to around 70 collisions [2]. This causes billions of particles to exist every second. Over 99% of these events are not the object of study of ATLAS [3], so only a small percentage are stored for later analysis. To decide which events are kept, the complex ATLAS trigger system has to reconstruct the tracks of about 100 thousand events per second, in a very small time period, and in high particle density conditions in the detector. This is a tremendous task, that needs large amounts of processing.

In the High-Luminosity phase of the LHC (HL-LHC) [4], the average number of collisions per bunch crossing (pile-up $\langle\mu\rangle$) will increase up to 200. The current trigger doesn't have the capability to cope with this much tracking, so a new system to perform this task effectively is needed. One of the options proposed to tackle this challenge is the Hardware Tracking for the Trigger (HTT) [5]. There are many different ways to find the tracks and obtain the parameters that describe them mathematically, so an optimum one has to be found for the HL-LHC. The cost, hardware requirements, quality of the fits, latency of the method and other constraints have to be taken into account. This thesis presents the study of a method that uses previous advances and already known techniques to do the tracking in a different way, to determine if it is a better alternative to the already available options.

My Master project was developed within the portuguese ATLAS team, as a member of the working group for the Simulation of the HTT (HTTSim). I developed two distinct projects, both part of HTTSim. The first one was the HTTEventSelectionSvc, a class that chooses the events and simulation objects we wish to use based on user-defined parameters. This was a simple and initial task, with the purpose of understanding how the ATLAS software framework, Athena [6], and the HTTSim code work, while also developing a useful component for the HTT simulation. The second and main project was the implementation and characterization of a new method to obtain two of the parameters of tracks.

The objective of that project was to study the road finding efficiency, quality of the fits, fake track rejection, performance in high pile-up conditions, parameter resolutions that the method achieves, and compare it with the baseline system. The Hough Transform (HT) plus linearized fitting on an idealized detector geometry were chosen so that its implementation in the trigger could be done using only Field-Programmable Gate Arrays (FPGAs). This will remove the uncertainties in using Application Specific Integrated Circuits (ASICs), while at the same time needing a smaller

amount of memory than the baseline system. This will be further explained in section 5.3.

The HTT project was recently abandoned, in favor of a software based solution. The work documented in this thesis contributed to the development and study of the system.

An overview of the ATLAS detector, Trigger and Data Acquisition System (TDAQ), simulation and Phase-II Upgrade is presented in Chapter 2. Chapter 3 is the State of the Art on hardware charged particle tracking. The explanation of the code developed for HTTSim is in Chapter 4, while the description and performance of the tracking method is shown in Chapter 5. The conclusions are presented in Chapter 6.

Chapter 2 The ATLAS Experiment

The ATLAS (A Toroidal LHC ApparatuS) experiment is operated by a collaboration of over 5000 physicist, engineers, technicians, students and support staff from all around the globe. It's a multi-purpose experiment that has the objective of expanding the knowledge and understanding of the world of particle physics. Its detector is integrated in the LHC, which is a powerful circular particle accelerator at CERN (European Organization for Nuclear Research). In the LHC, bunches of up to 10^{11} protons circulating in opposite directions collide 40 million times per second. These collisions have a centre of mass energy of 13 TeV, at a design luminosity (number of protons per area second) of $10^{34}\ cm^{-2}s^{-1}$. This leads to around $\langle\mu\rangle = 40$ interactions per bunch crossing. These high energies and pile-up conditions make the need for a high performance detector, capable of achieving the physics objectives of ATLAS.

The coordinate system and nomenclature in ATLAS is as follows. The origin is in the nominal interaction point, the positive x-axis points to the center of the LHC and the y-axis points upwards, which leaves the z-axis in the direction of the beam. The azimuthal angle ϕ is measured from the positive x-axis, while the polar angle θ is from the positive z-axis. The pseudorapidity $\eta = -ln[tan(\theta/2)]$ is defined as a measure of the polar angle, and can better be understood in Figure 2.1. The transverse momentum p_T is the momentum of the particle in the x-y plane, the transverse energy E_T and missing transverse energy E_T^{miss} are also in the x-y plane. The distance $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$ is a length in pseudorapidity-azimuthal angle space.

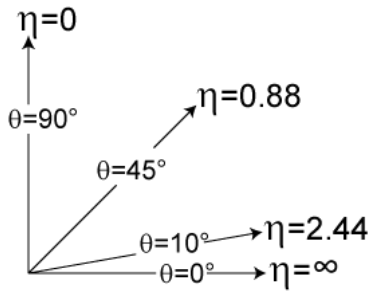


Figure 2.1: Pseudorapidity versus the corresponding polar angles. It starts at 0 for an angle of $90°$, and goes to infinity as the angle approaches zero [7].

A charged particle in a uniform magnetic field will move in a helix. In ATLAS, this helix is mathematically described by 5 parameters: (q/p_T, ϕ_0, η, d_0, z_0). These are estimated at the perigee of the track, it's closest point to the beam, and are, respectively, the charge over the transverse momentum, the angle between the transverse momentum and the x axis, the pseudorapidity, the impact parameter, which is the radius of the perigee, and the z coordinate of this point. These can better be seen in Figure 2.2

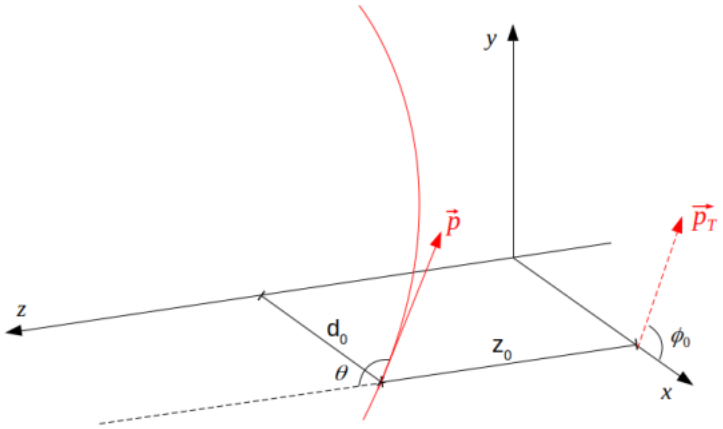


Figure 2.2: Helix parameters of tracks in ATLAS.

2.1 The ATLAS Detector

The ATLAS detector is 25 meters in diameter and 44 meters in length and weights about 7000 tonnes. It's divided in 5 main parts, each with different types of technologies and functions, arranged in a barrel and two end-caps. These are the Inner Detector (ID), the Calorimeters, the Muon Spectrometer, the Magnet System and the Forward Detectors.

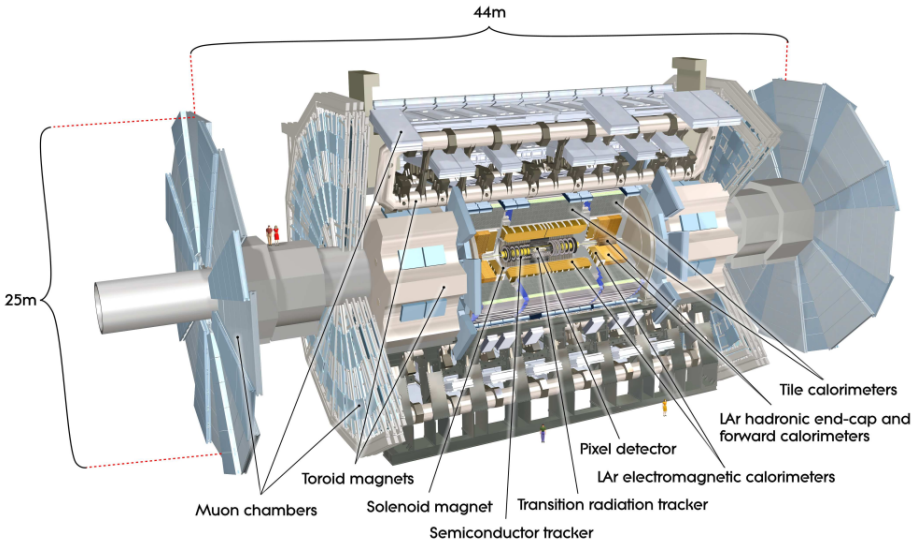


Figure 2.3: The ATLAS detector, showing the Inner Detector, the Calorimeters, the Muon Spectromter and the Maget System. The Forward Detectors are not shown [3].

Inner Detector

This is the closest detector to the beam and is submersed in a 2T magnetic field generated by a surrounding solenoid. It's purpose is to obtain the hits needed for the tracking of electrically charged particles emerging from the collisions. That is, to perform pattern recognition, measure the momentum, direction and charge of particles, and reconstruct the vertices

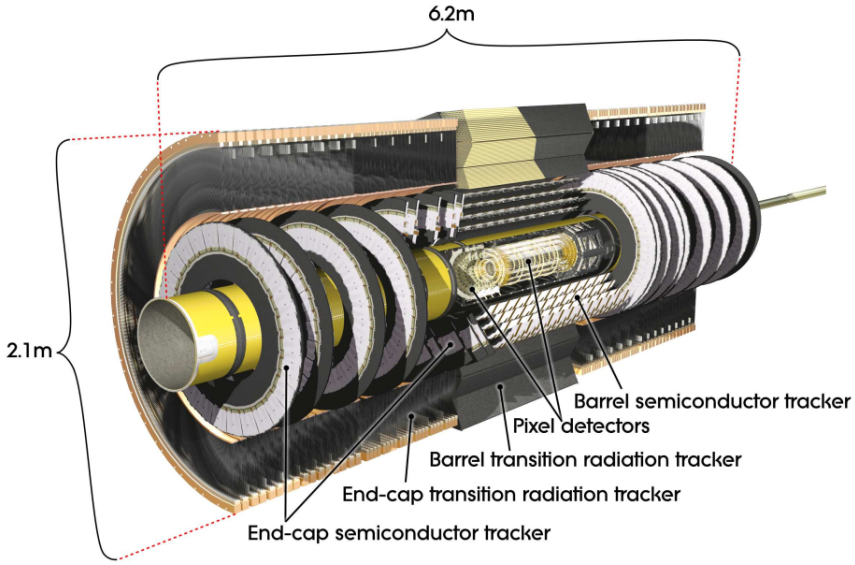


Figure 2.4: The Inner Detector and its sub-detectors, responsible for the tracking of charged particles [3].

where they came from. For that, it uses high granularity and precision measurements from its 3 main components, the Pixel Detector, SemiConductor Tracker (SCT) and the Transition Radiation Tracker (TRT). Each has a barrel part and an end-cap part. The pixels and SCT are responsible for the high precision measurements in a region of $|\eta| < 2.5$, while the TRT helps the tracking process by following the particles up to $|\eta| < 2.0$. In total, the ID has almost 100 million readout channels.

Pixel and Semiconductor Tracker: The pixels and SCT modules are disposed in 4 concentric cylinders each in the barrel, and in the end-caps they make disks perpendicular to the beam. These two sub-detectors provide two coordinates, one in $R-\phi$ and another in z. The pixel sensors have a minimum size in $R-\phi \times z$ of $50 \times 400\ \mu m^2$ and its intrinsic accuracies are 10 μm in $R-\phi$ and 115 μm in z. The SCT sensors are made of 2 sheets of silicon and in the barrel region, each has strips of 80 μm pitch angled at 40 mrad between each other, so that one set of strips in each layer is paralel to the beam. In the end-caps, there is a set of strips oriented radially and another at an angle of 40 mrad, and the mean pitch is also 80 μm. The intrinsic accuracies of each module are 17 μm in $R-\phi$ and 580 μm in z.

Transition Radiation Tracker: The Transition Radiation Tracker consists of straw tubes with 4 mm of diameter, with a 0.03 mm diameter gold-plated tungsten wire in the centre. In the barrel they are 144 cm long parallel to the beam, divided in 2 halves approximately at $\eta = 0$, while in the end-caps the straws are 37 cm long and arranged radially in wheels. This sub-detector only gives the $R-\phi$ coordinate, for which it has an intrinsic accuracy of 130 μm per straw.

One important detail about the silicon detectors is that its modules are arranged in a way so that one end of a module is above the one besides it, and the other end is under the

one on the opposite side. This configuration can be seen in Figure 2.5.

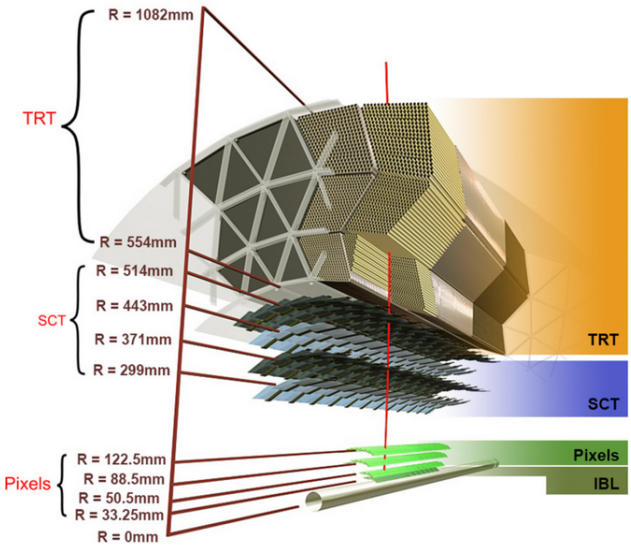


Figure 2.5: Cross section view of the ID, showing the arrangement of the silicon modules [8].

Calorimeters

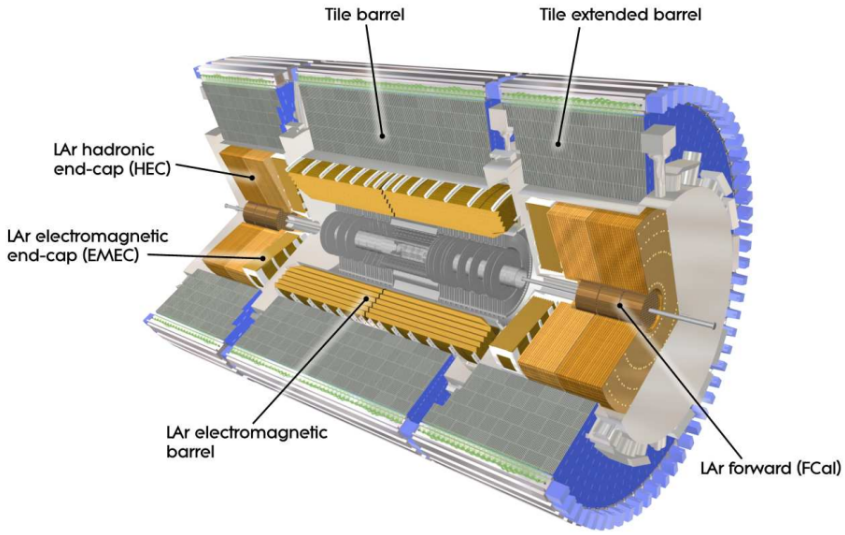


Figure 2.6: The Liquid Argon Calorimeter (yellow) and the Tile Calorimeter (grey) [3].

The objective of the calorimeters is to measure the energy that the particles loose as they pass through the detector. They are designed to absorb most known particles, except muons and neutrinos, consisting of an "absorbing" high-density material that stops incoming particles, meshed with layers of an "active" medium that measures their energy. There are two main parts, the Liquid Argon (LAr) Calorimeter and the Tile Hadronic Calorimeter. The total range covered in $|\eta|$ is 4.9. In the same η region as the Inner Detector, the fine granularity of the ElectroMagnetic (EM) calorimeter allows precision measurements of electrons and photons. The rest of the calorimetry has coarser granularity, which is sufficient to reconstruct jets and measure missing energy.

LAr Calorimeter: The Liquid Argon Calorimeter surrounds the ID. In the barrel, there is the EM calorimeter and in the endcaps there are the Forward Calorimeter (FCal), EM and hadronic endcaps. The EM calorimeter has accordion-shaped kapton electrodes and lead absorber plates over its full coverage ($|\eta| < 1.475$ for the barrel and $1.375 < |\eta| < 3.2$ for the endcaps). The barrel part consists of two identical half-barrels, separated by a small gap (4 mm) at $z = 0$, while the end-cap part is divided into two coaxial wheels. The Hadronic End-cap Calorimeter (HEC) is made of two independent wheels per end-cap. Each wheel is built from 32 identical wedge-shaped modules and is divided into two segments in depth, for a total of four layers per end-cap. These innermost wheels are built from parallel copper plates with 25 mm thickness, while the outer ones have 50 mm thickness. Between the plates, there are 8.5 mm gaps filled with liquid argon, that provide the active medium, making what is called a sampling calorimeter. The HEC overlaps with the Tile calorimeter at around $1.5 < |\eta| < 1.7$ and with the Forward calorimeter between around $3.1 < |\eta| < 3.2$. The FCal is made of 3 modules in each end-cap, the first is copper and optimised for electromagnetic measurements, while the others are tungsten and measure predominantly the energy of hadronic interactions. These modules are metal matrices with regularly spaced longitudinal channels filled with the electrode structure consisting of concentric rods and tubes parallel to the beam axis. The LAr is in the gaps between the rods and the tubes, wich can be as small as 0.25 mm.

Tile Hadronic Calorimeter: The Tile calorimeter is also of the sampling type and extends a region of $|\eta| < 1.7$. It consists of 64 azimuthal modules, and uses steel as the absorber and plastic scintillating tiles as the sensitive medium. The photons produced here are read out by wavelength shifting fibres into two separate photomultiplier tubes. Its objective is to measure the energy of hadronic particles that haven't deposited all their energy in the LAr calorimeter.

Muon Spectrometer

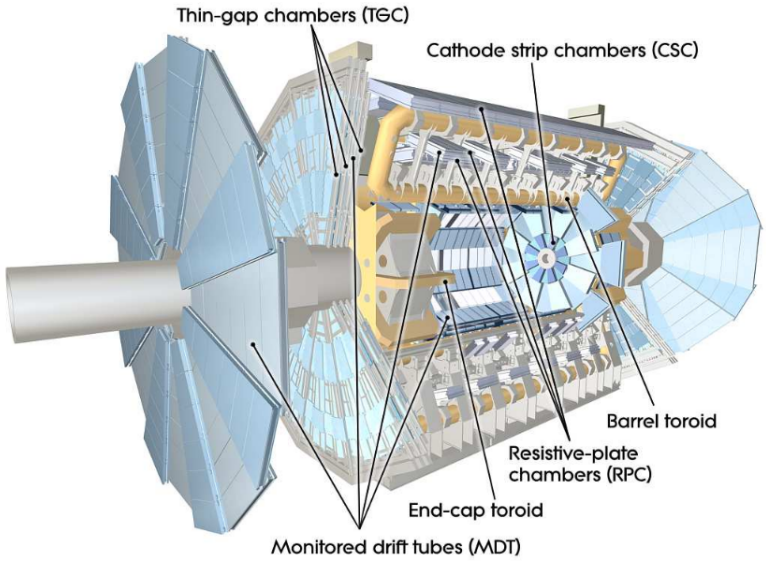


Figure 2.7: The Muon Spectrometer (blue) in and around the Toroid Magnets [3].

Muons pass through the Inner Detector and calorimeters without stopping, so the muon

spectrometer wraps these two detectors to identify and measure the momenta of this type of particle. For that, it uses the toroid magnets to bend their tracks and four types of muon chambers responsible for triggering and high-precision measurements. In the $|\eta| < 1.4$ range, the magnetic field is provided by the barrel toroid, the $1.4 < |\eta| < 1.6$ area, called the transition region, has a combination of barrel and end-cap fields, and in $1.6 < |\eta| < 2.7$, the tracks are bent by the end-cap toroids only. This configuration was chosen to maximize the locations where the field is orthogonal to the tracks.

Muon Chambers: In the entire $|\eta| < 2.7$ region, the Monitored Drift Tubes (MDTs) measure the curve of tracks. At $2.0 < |\eta| < 2.7$, Cathode Strip Chambers (CSCs), which are multiwire proportional chambers with cathodes segmented into strips, measure precision coordinates. Resistive Plate Chambers (RPCs) are used to measure the second coordinate (in the non-bending direction) and for triggering in the barrel, while Thin Gap Chambers (TGCs) serve the same purpose in the ends of the detector.

Magnet System

The ATLAS detector is immersed in two distinct magnetic fields provided by superconducting magnets. It's purpose is to bend the trajectory of charged particles, to enable the measurement of their momentum and charge. The magnets are the Central Solenoid and the Barrel and End-cap Toroids, which are cooled to about 4.5 K (–268°C), to provide the needed intensity of the magnetic fields.

As mentioned in section 2.1 - Inner Detector, the Central Solenoid surrounds the ID and provides a magnetic field of 2 T with a stored energy of 38 MJ, in just 4.5 cm thickness. This is generated by a 7.73 kA current in over 9 km of niobium-titanium superconductor wires embedded into strengthened, pure aluminum strips.

The Toroid Magnets are made of 8 coils each, using a 20.5 kA nominal current in almost 100 km of superconducting wire. This makes it possible to have almost 5.5 Tm of bending power in the barrel toroid and up to 7.5 Tm in the end-cap region.

Forward Detectors

There are also three smaller detectors (not shown in image 2.3) whose main function is to measure parameters of the beam.

Two of them, the LUminosity measurement using Cherenkov Integrating Detector (LUCID) at ± 17 m and Absolute Luminosity For ATLAS (ALFA) at ± 240 m, determine the luminosity delivered to ATLAS. The third is the Zero-Degree Calorimeter (ZDC), which has the objective of measuring the centrality of heavy-ion collisions, is located at ± 140 m

2.2 Trigger and Data Acquisition System

There are almost 1.7 billion $p-p$ collisions per second in ATLAS, that need about 60 million megabytes to be completely stored. However, only a very small percentage of these represent interesting physics processes or even new physical properties. Because of this, it's not necessary, nor possible, to record every single event, so there is a complex trigger that decides whether an event is rejected or sent for storage. This is the task of the ATLAS Trigger and Data Acquisition System (TDAQ), that selects about 1000 of the 1.7 billion events per second.

The trigger uses the detector data to select events in two stages. First, the hardware based Level-1 trigger (L1) reduces the rate of events to a maximum of 100 kHz, then they go through the software-based High Level Trigger (HLT), after which about 1000 are sent for storage. The architecture of the TDAQ system for Run 2 [9] is shown in Figure 2.8.

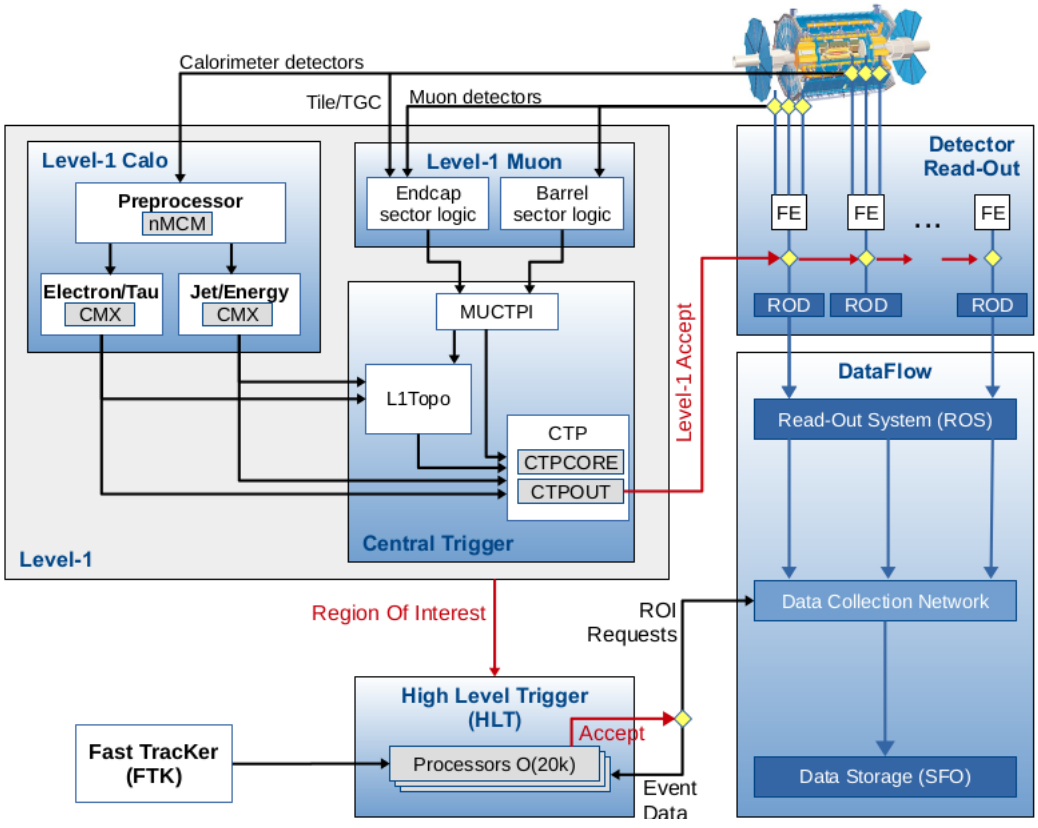


Figure 2.8: Architecture of the ATLAS Trigger and Data Acquisition system for Run 2. [9]

Level-1 Trigger

The L1 trigger [10] is made of custom hardware electronics and uses simple physics objects to quickly reject most uninteresting events, in 2.5 μs, to a maximum trigger rate of 100 kHz. It consists of the Level-1 Calorimeter trigger system (L1Calo), the Level-1 Muon trigger system (L1Muon) and the Central Trigger Processor (CTP), where the Level-1 Topological trigger system (L1Topo) is included.

It takes coarse granularity data from the calorimeters and muon detectors to decide whether or not to send the event to the HLT. If an event is accepted, the L1 trigger forms Regions of Interest (RoIs), which are $\Delta\eta \times \Delta\phi$ sections. These include information about where it found possible physics objects and for example if a threshold was passed. It then sends these RoIs to the HLT, as well as a signal to the sub-detector readout systems.

The L1Calo takes data from the EM and hadronic calorimeters to search for electrons, photons, taus, jets and high total and missing transverse energy (E_T^{miss}). It outputs this information to the CTP through the output merger modules (CMX).

The L1Muon has a part that processes the barrel region and two other parts for each end-cap. It's function is to identify high p_T muons originating from the interaction region, and several p_T thresholds can be programmed. To reduce the amount of fake muons found, the L1Muon also uses coincidence information with the outermost layer of the Tile Calorimeter. The information is passed through the Muon to Central Trigger Processor Interface (MUCTPI), before going to the CTP.

The Level-1 Topological trigger system (L1Topo) uses the objects that the L1Muon and L1Calo found to make further selections based on kinematic information, such as angular separation, and global event quantities, such as the sum of the transverse momenta of all Level-1 jet objects.

The CTP uses all the information obtained in the L1Calo, L1Muon and L1Topo to form the decision whether to reject or pass the current event. If an event is accepted, it sends a signal to the sub-detector readout systems via the Timing, Trigger and Control (TTC) network. Here, it's possible to set up a menu of up to 512 different trigger selections, with combinations of physics objects and thresholds [11].

High Level Trigger

The HLT [12] consist of a CPU farm that runs offline-like algorithms to reconstruct the RoIs found in the L1 trigger. It uses full granularity and precision data from the calorimeters and muon chambers.

The tracking of particles in the ID is performed by the Fast TracKer (FTK) [13]. This is a custom hardware system developed specifically for the rapid execution of this task. It uses Associative Memory (AM), which is a type of Application Specific Integrated Circuit (ASIC), to do the pattern recognition. That is, to find the sequence of hits in the layers of the detector that may belong to a particle. This is followed by a linear fit of the track parameters and a χ^2 on FPGAs. Linearized track fitting was chosen instead of the calculation of the full helical fit, because it gives good results and only needs sums and multiplications, which can be performed very quickly on these hardware accelerators.

2.3 The ATLAS Simulation

The ATLAS detector is very complex and very expensive, so to predict what it's performance is, it's necessary to simulate all of it's components, physics events and their interaction with the detector. A full simulation [14] has been developed in order to study the performance of the detector and the several types of physics, which consists of the entire chain of processes from the $p - p$ collisions to the physics analysis.

To achieve this in the most consistent and simple way, the simulation is divided in 4 main parts: the event generation, the detector simulation, digitisation and reconstruction.

Event Generation

This is the first step, where the proton-proton collisions are simulated with Monte Carlo (standard HepMC formats), and the various decays are generated. These can be known processes or new physics predicted by some theory.

Detector Simulation

Next, the interaction of the generated particles with the detector is simulated with GEANT4. The geometry of the detector is described and used as input, so that it's possible to organize the sensors the in several ways. This is the most critical and time consuming step, due to the high number of interactions of each particle with each piece of matter of the detector. The objective of this step is to obtain the energy deposits of each particle in each layers.

Digitisation

The last step of the detector simulation is to transform the energy deposits in currents and voltages, as the readout electronics would do. This allows the reconstruction to run on data as if it were taken directly from the real detector, which is very useful to obtain the closest results possible to the reality.

Reconstruction

The reconstruction phase of the simulation is when all the physics objects are produced, like tracks, jets, vertices and so on. This gives the information necessary to perform physics analysis, such as finding a particular decay, while rejecting the background.

2.4 Phase-II Upgrade for the HL-LHC

In 2025, it's planned to start the High Luminosity Large Hadron Collider (HL-LHC) [15], where the instantaneous number of protons per area and second (luminosity) will increase up to $L = 7.5 * 10^{34}\ cm^{-2}s^{-1}$, the centre of mass energy to 14 TeV and the average number of

collisions per bunch crossing (pile-up μ) to around $\langle\mu\rangle = 200$.

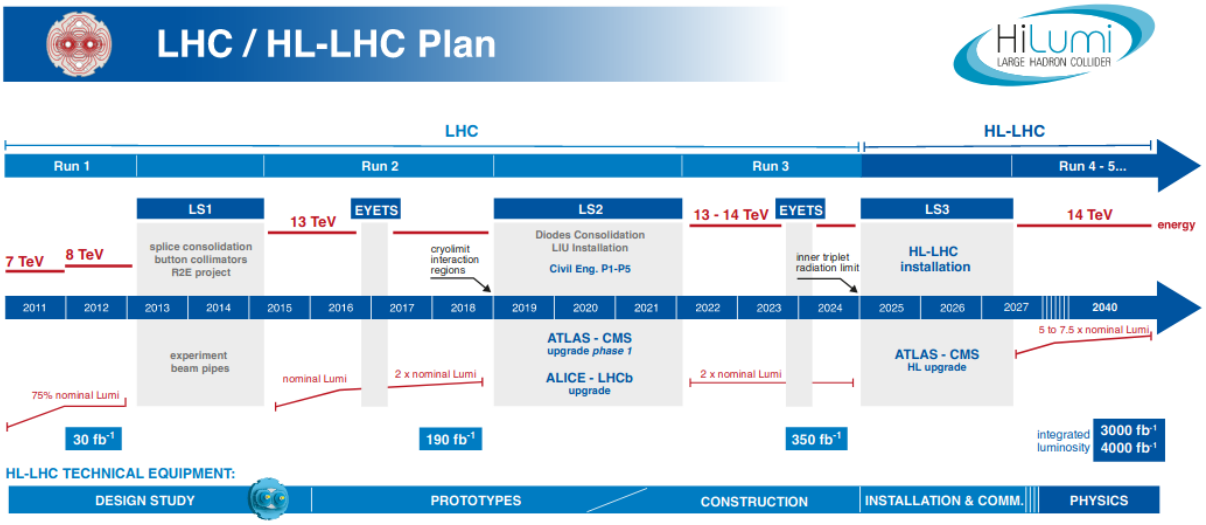


Figure 2.9: Planned timeine for the HL-LHC and its phases [15].

The current detector and TDAQ system aren't capable of dealing with this much data, so an upgrade to these components is necessary to fully exploit the opportunity that is the HL-LHC.

It will start with the Long Shutdown 3, time that will be used for the second major upgrade of the ATLAS detector and TDAQ system, before collisions restart in Run 4. This will be necessary to achieve the trigger rates about 10 times higher the ones expected in Run 3 [5], so that the events associated with the required physics programme can be efficiently selected.

Several parts of the detector will be upgraded, but here the focus will stay on the components this thesis requires.

Detector

The Inner Detector will be replaced by a new all silicon detector, the Inner TracKer (ITk). It will also have pixel [16] and strip [17] sub-systems, but will not have an equivalent to the TRT. The pixel detector will cover the region $|\eta| < 4$ with 5 concentric layers in the barrel and 5 layers of inclined or vertical modules in the end-cap. The strip sub-system will cover the region $|\eta| < 2.7$ with 4 layers in the barrel and 6 disks in the end-caps. A scheme of the position and orientation of its layers is shown in Figure 2.10.

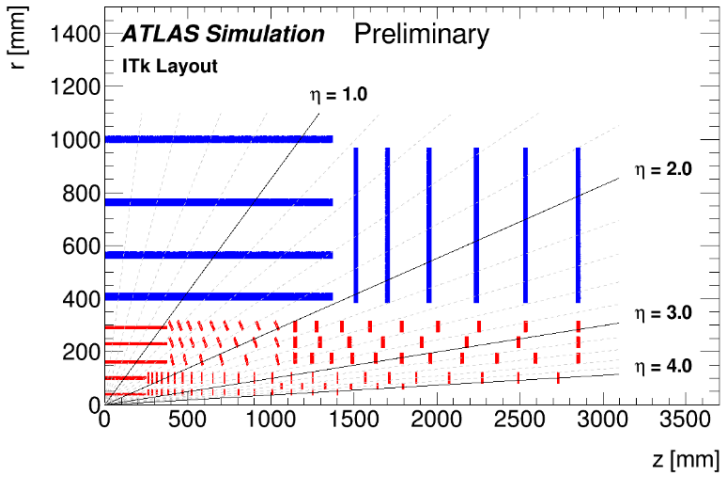


Figure 2.10: A schematic image of the position and orientation of the Inner Tracker layers. The pixels are represented in red, while the strips are shown in blue. [18]

TDAQ

The baseline Phase-II TDAQ system is planned to have a hardware based Level-0 trigger (L0), followed by the Event Filter (EF). The L0 will reduce the rate of events to 1 MHz at a maximum latency of 10 μs. Then, the EF, which consists of a CPU farm and the HTT, will choose the events that are for storage or not.

In the TDAQ, the calorimeter and muon raw data is processed at 40 MHz by the L0 trigger, which gives the decision at 1 MHz and within a latency of 10μs. That triggered data is transferred to the Event Filter, where the HTT reconstructs tracks. There, the decision is made according to the menu (the various physics processes defined in the trigger), and the data is sent to storage at 10 kHz.

HTT

For ATLAS to reach the full potential of the HL-LHC, it will need to quickly reconstruct tracks at a rate of 1 MHz. This is to maintain low thresholds on physics properties and to efficiently select the interesting events. This is a task that takes a significant amount of time in normal processors, so an alternative using hardware accelerators, the Hardware Tracking for the Trigger, is being developed.

This system will reconstruct charged particle tracks in the region covered by the ITk ($|\eta| < 4.0$). It will use Associative Memorys for road finding, and perform a linearized fit for the track parameters and χ^2 in FPGAs.

A more complete description of the HTT system and performance is presented in the next chapter.

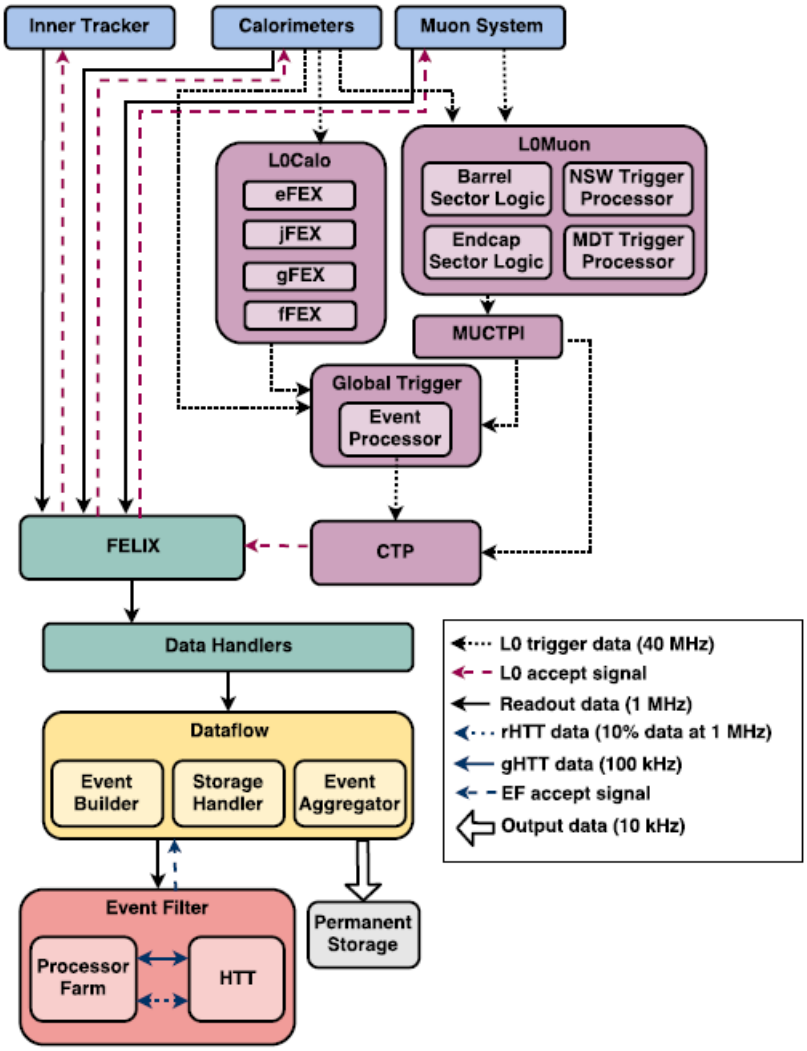


Figure 2.11: The ATLAS Trigger and Data Acquisition System planned for the HL-LHC, after the Phase-II Upgrade [5].

Chapter 3 State of the Art

In this section, three charged particle tracking systems are presented, with focus on the HTT. The Fast TracKer, the system preceding the HTT, and the hardware tracker for the trigger of the Compact Muon Solenoid (CMS) experiment are also discussed.

They all follow the general tracking procedure of clustering hits in the detector close to each other to form spacepoints, followed by a road finding method using combinations of these spacepoints to make track canditades, which are then fitted and finally an ambiguity resolution step to discard fake tracks.

3.1 Hardware Tracking for the Trigger

The HTT is one of the more advanced charged particle tracker project in the world. The challenging aspects of this system are the volume of data, the rate at which this data is produced, the low latency available for computation, the diverse menu of events, the high pile-up environment, how it reconstructs the tracks and the use of the most advanced technologies in computation, memory, data transfer, and several boards developed specifically for this system. It is be composed of the following components and approximate physical characteristics [5]:

- 48 HTT Units, each with
  - 12 Associative Memory Tracking Processors (= 576 AMTPs), each with
    - 1 Pattern Recognition Mezzanine (= 576 PRMs), each with
      - 24 Associative Memory Application Specific Integrated Circuits (= 13824 AM ASICs)
  - 2 Second Stage Tracking Processors (= 96 SSTPs), each with
    - 2 Track Fitting Mezzanines = 192 TFMs
- 385 kW power consumption
- 3.2 Tb/s of input data and about 1 Tb/s output

The HTT will perform two modes of tracking: regional tracking (rHTT) and global tracking (gHTT). Regional tracking will reconstruct tracks of charged particles using only eight ITk logical layers, and in RoIs specified by the L0 trigger. (A logical layer can be a pixel layer or one of the two sides of the strip modules.) This will allow the trigger to perform background rejection from 1 MHz to 400 kHz. Global tracking will use all ITk logical layers (13 in total) to reconstruct charged particles in the entire ITk coverage. It will return tracks at 100 kHz to the EF and will be executed upon request. Its purpose is to give higher quality tracks, with better parameter resolutions.

Functional Requirements

The HTT has a few functional requirements that are associated rHTT and gHTT. The full list of requirements for the Phase-II Upgrade of the TDAQ is presented in section 4 of the HTT Technical Design Report (TDR) [5].

- Regional tracking is needed at the L0 output rate of 1 MHz, for all charged particles with $p_T > 2$ GeV and $|\eta| < 4$.
- Global tracking is required for all charged particles with $p_T > 1$ GeV and $|\eta| < 4$, when requested by the EF.
- The gHTT resolutions of d_0 and z_0 track parameters shall not be more than twice those of offline track reconstruction.

HTT Architecture

The HTT system is planned to be logically organized as 48 units, each of these with 12 AMTPs and 2 SSTPs. Each HTT unit is responsible for reconstructing tracks in an angular region of $\Delta\eta \times \Delta\phi$. Physically, there will be a total of 56 Advanced Telecommunications Computing Architecture (ATCA) shelves (each with 12 blades), 48 for AMTPs and 8 for SSTPs. On a blade fits a general board, Tracking Processor (TP), that handles communications and data formatting and either an AMTP (with 1 PRM) or an SSTP (with 2 TFMs). Every AMTP and SSTP connects to the EF via the HTT InterFace (HTTIF). This can better be seen in Figure 3.1.

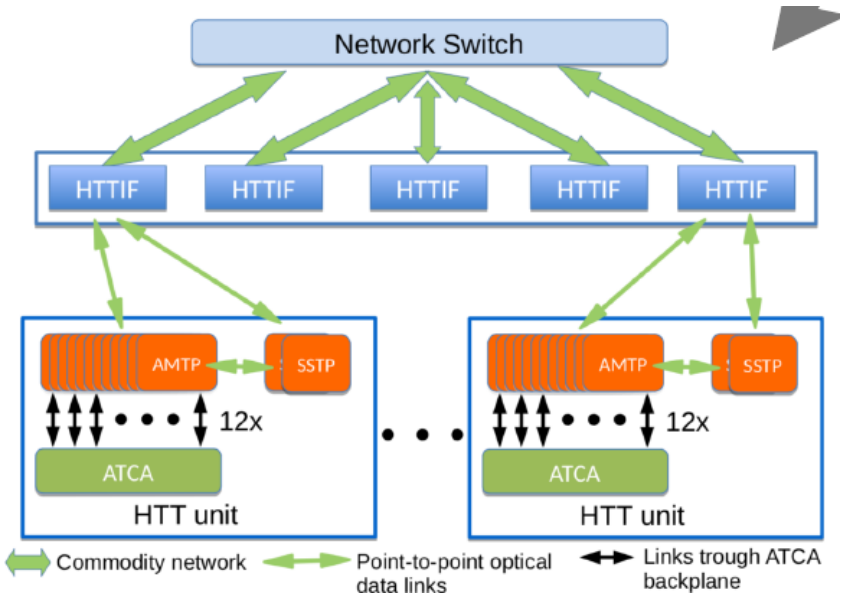
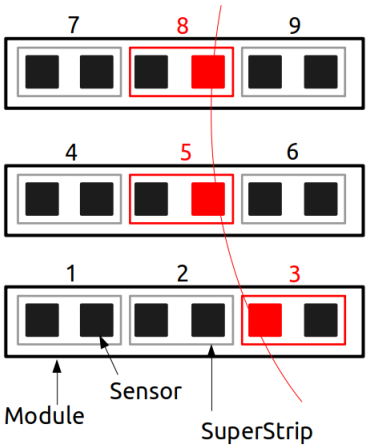


Figure 3.1: Architecture of the Hardware Tracking for the Trigger [5].

Track Reconstruction

The input to the HTT are hits from the Inner TracKer. These hits are clustered into consecutive ITk strip or pixel channels to form so-called "superstrips", each with a SuperStrip IDentifier (SSID). This happens in the TPs. The pixels have coordinates in 2 dimensions, so their superstrips have a width and a length, while the strip modules have only one coordinate, so only a superstrip width is needed. The dimensions (width and length) of the superstrips are related to the number of layers and the actual layers used in the regional tracking, and studies were performed to optimize this number.

Then, there are two stages. The first one happens for both regional and global tracking, and the second only happens for global tracking. The first one happens in the PRMs, and consists of a road finding step followed by a linearized track fit. Roads are sequences of hits in a module of each layer, and to find them, several combinations of the "superstrips" formed with the ITk hits are compared to pre-simulated patterns. These patterns are stored in the AMs and are sequences of SSIDs in each layer. The core elements of first stage processing are these AMs, that each hold a different pattern bank, for a total of over 5 billion patterns. This step can be visualized in Figure 3.2.



P1	P2	P3	P4	P5	P6	P7
1	1	2	2	2	3	3
4	5	4	5	5	5	6
7	8	7	7	8	8	9

Figure 3.2: Illustration of the pattern matching process. A track hitting a detector with 3 layers, each with a module of 6 sensors, will leave hits in the sensors. Each SuperStrip has 2 channels, for a total of 9 SSIDs. The sequence of hit SSIDs (3, 5, 8) is compared to the pattern bank, to search for a match.

When a pattern is matched, a road is formed. In a given module, there can be multiple hits, so all the possible combinations of hits in each layer are considered as a track candidate, for which the track parameters and quality (χ^2) are computed on an FPGA, with the corresponding hit coordinates. The track parameters are calculated with a linear interpolation, as per equation 1, where p_i is the track parameter (q/p_T, ϕ_0, η, d_0, z_0), x_j are the N hit coordinates, and (C_{ij}, q_i) are pre-computed constants. The quality of the fit is calculated with a linearized χ^2 method, as per equation 2, where x_j are the coordinates and (A_{ij}, k_i) are pre-computed

constants. The constants generation process is explained below.

$$p_i = \sum_{j=1}^{N} C_{ij} x_j + q_i \tag{1}$$

$$\chi^2 = \sum_{i=1}^{N-5} \left(\sum_{j=1}^{N} A_{ij} x_j + k_i \right) \tag{2}$$

For global tracking, the second stage is performed in the TFMs, in which the first stage track candidates are extrapolated to the remaining ITk layers and associated to any matching hits nearby. A full track fit is then performed to achieve the best possible track parameter resolution. The duplicate second stage tracks are removed, and the ones that pass the ambiguity resolution step are sent to the EF. The flow of ITk data, first and second stage tracks can be seen in Figure 3.3.

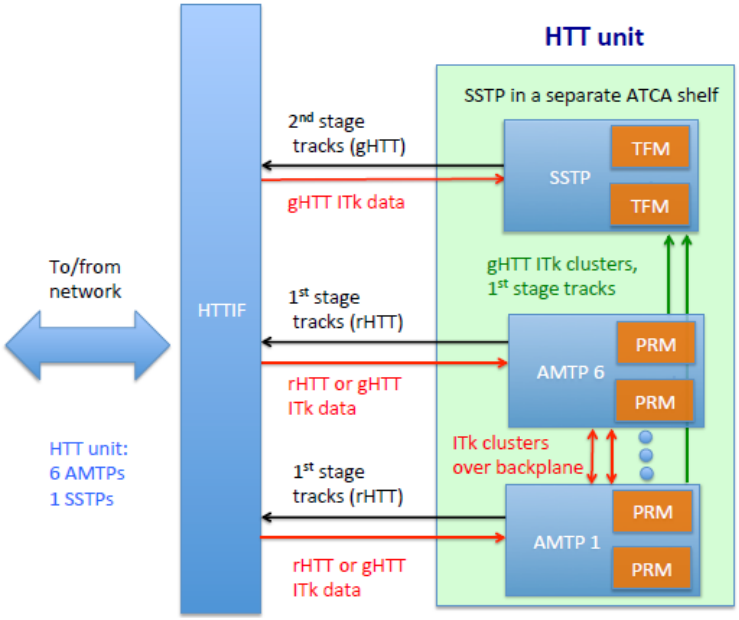


Figure 3.3: Data flow between the Event Filter and HTT, and inside the HTT. The inputs are ITk hits, and the output are rHTT and gHTT tracks [5].

A few details are worth mentioning. If two patterns are very similar, some of the least significant bits of the superstrip number can be ignored and are called "Don't Care bits" (DC bits). The effect is that these two patterns will be combined, increasing efficiency and lowering pattern bank size. When a pattern has fewer than eight layers hit, the missing layers are marked as a "Wild Card" (WC), which make that layer considered as matched. WCs are used to compensate for inefficiencies in detector geometry. The duplicate removal step is done with the HitWarrior algorithm. This algorithm groups tracks according to the number of common hits. In each group of tracks, it then chooses the one with the lowest χ^2, and rejects the rest. This step is also called Overlap Removal (OR).

Simulated Performance

To study the response of the HTT, full simulation of four $\Delta\eta \times \Delta\phi$ regions of 0.2×0.2 for particles with $p_T > 4$ GeV, and one of these $(0.7 < \eta < 0.9)$ with $p_T > 1$ GeV, was performed with 200 pile-up events. In particular, since this last region contains tracks with p_T down to 1 GeV, it provided an important estimate of the gHTT tracking performance. The simulations were done with muons, because they are heavier particles than electrons, so they suffer much less bremsstralhung and multiple scattering, thus producing straighter, cleaner and easier to reconstruct tracks. The number of DC bits used per layer and superstrip dimensions were varied to achieve best efficiency. The results presented next are from the HTT Technical Design Report (TDR).

Using a "7-out-of-8" logic (only need 7 layers match to make a road), an efficiency of around 99% can be achieved in pattern matching. The important value here, and that will be used for comparison with this thesis' results, is the 99.1% efficiency achieved in the $0.1 < \eta < 0.3$ region. This can be seen in Table 3.1

Table 3.1: Simulated pattern matching performance in 4 regions. All numbers are average per event [5].

η range	muon eff.	mean matches pile-up	99% interval matches in pile-up	Superstrip width			DC bits
				pixel	barrel	end-cap	
$0.1 < \eta < 0.3$	99.1%	31	151	33/402	40		21111122
$0.7 < \eta < 0.9$	99.2%	21	93	33/402	40		21111122
$1.2 < \eta < 1.4$	98.8%	42	159	33/402	40	20	21111122
$2.0 < \eta < 2.2$	98.7%	10	56	16/200		10	21111122

At $|\eta| > 3$, the tracks do not hit the ITk strip end-cap disks at all, hence only pixel information is available to make tracks. This significantly reduces the pattern matching efficiency, as was demonstrated by simulating a region in $3.0 < \eta < 3.1$ and with $p_T > 4$ GeV. The efficiency obtained was 95.6%, as shown in Table 3.2.

Table 3.2: Simulated pattern matching at $|\eta| > 3$. All numbers are average per event [5].

η range	muon eff.	mean matches		Superstrip width
		muon	pile-up	pixel
$3.0 < \eta < 3.1$	95.6%	25	8	16/200

To see the effect of inefficient detector channels, 100 – x% clusters were ignored, and 6/8 and 7/8 matches were simulated. This is a worst-case scenario because it is unlikely that whole clusters will be inefficient. Even with 7/8 matches, pattern matching efficiency is still above 95% for 97% hit efficiency, as seen in Figure 3.4.

The roads found in the AM step were used to simulate the tracking performance. The results can be seen in Table 3.3 for the muon plus pile-up sample, in the $0.7 < \eta < 0.9$ region being studied. In the Table, "# roads" is the average number of roads, "# fits" is the average

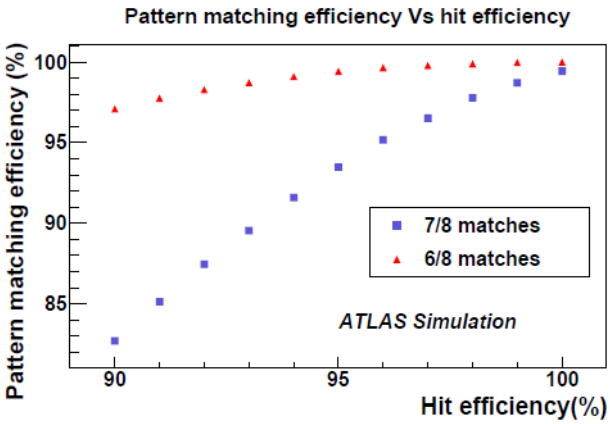


Figure 3.4: Simulated pattern matching efficiency with random inefficient detector channels [5].

number of fits done to cover all hit combinations, "# tracks $\chi^2 < 40$" is the number of tracks that pass the quality cut, "#tracks HitWarrior" is the number of tracks after OR and "#fit constants" is the number of constants for equations (1) and (2) required to fit the event. All numbers are averages per event.

Table 3.3: First stage track fitting performance for $0.7 < \eta < 0.9$ region at pile-up = 200. All numbers are averages per event [5].

particle	min p_T	Eff. (%)	# roads	# fits	# tracks $\chi^2 < 40$	# tracks HitWarrior	# fit constants
muon	1 GeV	99.5	144	1115	55	4.6	73
muon	2 GeV	99.1	79	586	23	1.9	40
muon	4 GeV	99.2	48	313	16	1.2	23
jets	1 GeV		195	1519	77	6.2	97
jets	2 GeV		104	804	29	2.4	52
jets	4 GeV		51	344	13	1.1	26
min-bias	1 GeV		110	842	38	3.6	58
min-bias	2 GeV		48	359	6	0.8	27
min-bias	4 GeV		21	133	1	0.2	12

To estimate track finding efficiencies and track parameter resolutions, studies were performed with muons and electrons. Tracking efficiency for these particles is defined as the fraction of events with at least one track with $\chi^2 < 40$, only for events in which an offline track is reconstructed. To obtain an estimate of the track parameter resolutions, the rms_{95} is used. To compute it, the central 95% of the residuals distributions are used (discarding the 2.5% tails on each side), and then their RMS value is calculated. The residuals are obtained from the difference between the parameter of the track with the lowest χ^2 and the parameter of the truth track. The results can be seen in Figures 3.5 and 3.6, and in Tables 3.4 and 3.5.

The efficiency for muons is approximately flat along the transverse momentum range, while the efficiency for electrons is smaller for low p_T and increases until reaching a plateau for higher p_T. The reason the efficiency at low p_T is smaller for electrons is their high radiation probability, which can reduce their p_T below the threshold of 4 GeV.

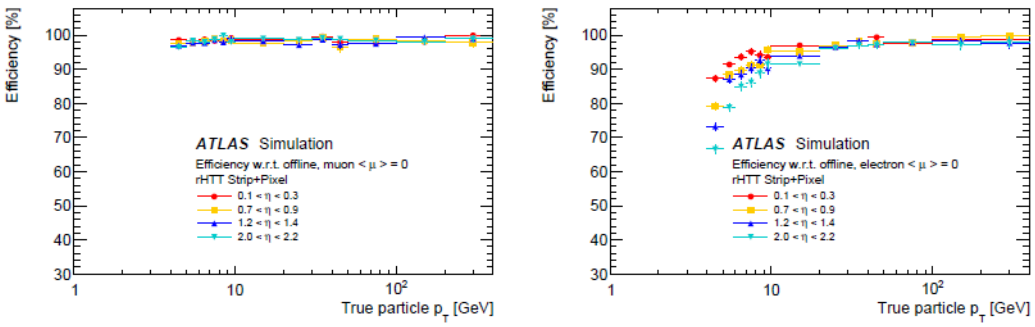


Figure 3.5: First stage track finding efficiencies in 4 η regions for muons (left) and electrons (right) for $p_T > 4$ GeV [5].

Table 3.4: First stage track fitting resolutions (rms95%) for muons $p_T > 4$ GeV [5].

η range	η	ϕ	q/P_t [GeV^{-1}]	d_0 [mm]	z_0 [mm]
$0.1 < \eta < 0.3$	0.002	0.001	0.003	0.20	0.8
$0.7 < \eta < 0.9$	0.002	0.001	0.003	0.20	0.8
$1.2 < \eta < 1.4$	0.004	0.003	0.007	0.33	3.8
$2.0 < \eta < 2.2$	0.004	0.003	0.014	0.71	7.1

Table 3.5: First stage track fitting resolutions (rms95%) for electrons $p_T > 4$ Ge [5].

η range	η	ϕ	q/P_t [GeV^{-1}]	d_0 [mm]	z_0 [mm]
$0.1 < \eta < 0.3$	0.004	0.003	0.021	0.42	2.9
$0.7 < \eta < 0.9$	0.004	0.003	0.031	0.52	4.5
$1.2 < \eta < 1.4$	0.011	0.013	0.048	0.87	19.3
$2.0 < \eta < 2.2$	0.014	0.012	0.059	1.03	22.1

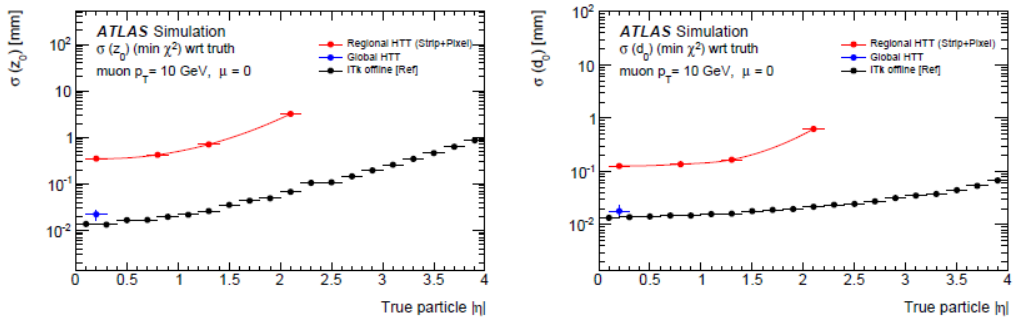


Figure 3.6: Comparison of the z0 (left) and d0 (right) for first and second stage track fitting and offline for 10 GeV muons. For the second stage, only the central region was studied [5].

Most recent results

Since the writting of the Phase-II Upgrade TDAQ TDR, the HTT simulation has evolved considerably. Next are presented the most recent results, used in the official report for the decision on the tracking in the Event Filter [19].

The general efficiency of the HTT for single muon or muon plus pile-up, is the ratio of events with at least one reconstructed track matched to the truth muon over all the events (also written HTT/truth). To consider a track to match, at least 50% of the hits of the

reconstructed track have to be the same as those of the truth track. This efficiency is shown in Figure 3.7 for single muons in the $0.1 < |\eta| < 0.3$ region. This is dominated by the pattern matching efficiency, because the fitting efficiency is usually 100%. The figure shows 3 different cases, in which the pattern bank was made differently. The baseline is the standard one used in HTTSim, and "optimized" refers to a set of pattern banks produced for the study of optimizing the HTT for the L0 trigger, one with a minimum track transverse momentum of 1 GeV and the other of 2 GeV. The baseline configuration was used for the rest of these results.

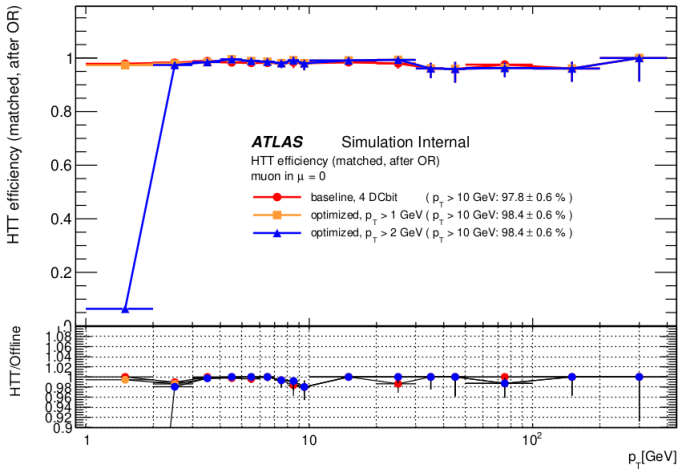


Figure 3.7: First stage HTT/truth tracking efficiency with three different pattern banks for single muons in the $0.1 < |\eta| < 0.3$ region. All cases have similar response, except the blue line because the corresponding pattern bank was made without tracks with $p_T < 2$ GeV. The bottom plot shows the ratio of HTT tracking efficiency over offline tracking efficiency [19].

Tables 3.6 and 3.7 show the different kinds of efficiencies obtained with the simulation for various regions and p_T ranges, for single muons and muons embedded in pile-up of 200 additional collisions, respectively. "HTT/truth", "offline/truth" and "HTT/Offline" refer to the ratio of tracking efficiencies with respect to the reconstruction of the denominator. The values to be used for comparison with this thesis results are the "HTT/truth" efficiencies of the $0.1 < \eta < 0.3$ region for first-stage tracking.

Table 3.6: Tracking efficiencies for single muons in different regions (no pile-up) [19].

η range	HTT/truth [%]			offline/truth [%]			HTT/offline [%]		
	1-2GeV	2-4GeV	>4GeV	1-2GeV	2-4GeV	>4GeV	1-2GeV	2-4GeV	>4GeV
muons, p_T> 1 GeV, no PU, first-stage tracking									
$0.1 < \eta < 0.3$	97.9	98.5	98.9	99.2	99.8	99.7	98.8	98.8	99.2
$0.7 < \eta < 0.9$	95.9	97.3	97.9	99.1	99.9	99.8	96.8	97.4	98.1
$2.0 < \eta < 2.2$	95.5	98.3	98.8	99.4	99.8	99.9	96.1	98.4	98.8
muons, p_T> 1 GeV, no PU, second-stage tracking									
$0.1 < \eta < 0.3$	93.7	96.7	97.7	99.2	99.8	99.7	94.5	97.0	98.0

Table 3.8 shows the rms_{95} resolution of the 5 helix parameters. These were obtained for single muon events at flat $1/p_T$, and fixed p_T of 1, 2 and 10 GeV. Again, the important values to use as comparison are the ϕ and q/p_T resolutions for the first region and first-stage

Table 3.7: Tracking efficiencies for muons embedded in 200 pile up collisions in different regions [19].

η range	HTT/truth [%]			offline/truth [%]			HTT/offline [%]		
	1-2GeV	2-4GeV	>4GeV	1-2GeV	2-4GeV	>4GeV	1-2GeV	2-4GeV	>4GeV
muons, p_T> 1 GeV, PU 200, first-stage tracking									
$0.1 < \eta < 0.3$	97.1	99.3	99.5	98.8	99.5	99.5	98.3	99.8	100.0
$0.7 < \eta < 0.9$	96.4	95.6	97.8	98.8	100.0	99.8	97.5	95.6	98.1
$2.0 < \eta < 2.2$	96.4	97.5	98.4	99.4	100.0	100.0	97.0	97.5	98.4
muons, p_T> 1 GeV, PU 200, second-stage tracking									
$0.1 < \eta < 0.3$	93.1	95.8	96.6	98.8	99.5	99.5	94.2	96.2	97.1

tracking.

Table 3.8: Resolutions of the 5 helix parameters for single muons without pile-up, in different p_T conditions [19].

η range	p_T	η	ϕ	q/p_T [GeV^{-1}]	d_0 [mm]	z_0 [mm]
muons, p_T> 1 GeV, no PU, first-stage tracking						
$0.1 < \eta < 0.3$	flat $1/p_T$	0.0030	0.0027	0.0064	0.49	0.69
$0.7 < \eta < 0.9$	flat $1/p_T$	0.0035	0.0031	0.0075	0.57	1.14
$2.0 < \eta < 2.2$	flat $1/p_T$	0.0071	0.0062	0.0317	1.02	3.04
$0.1 < \eta < 0.3$	1 GeV	0.0054	0.0047	0.0110	0.82	1.21
$0.7 < \eta < 0.9$	1 GeV	0.0065	0.0053	0.0128	0.93	2.29
$2.0 < \eta < 2.2$	1 GeV	0.0126	0.0088	0.0391	1.21	3.34
$0.1 < \eta < 0.3$	2 GeV	0.0029	0.0028	0.0062	0.50	0.65
$0.7 < \eta < 0.9$	2 GeV	0.0033	0.0032	0.0072	0.60	1.03
$2.0 < \eta < 2.2$	2 GeV	0.0067	0.0061	0.0312	1.02	3.20
$0.1 < \eta < 0.3$	10 GeV	0.0011	0.0010	0.0032	0.20	0.30
$0.7 < \eta < 0.9$	10 GeV	0.0011	0.0012	0.0033	0.23	0.39
$2.0 < \eta < 2.2$	10 GeV	0.0026	0.0044	0.0173	0.85	2.48
muons, p_T> 1 GeV, no PU, second-stage tracking						
$0.1 < \eta < 0.3$	flat $1/p_T$	0.0012	0.0012	0.0060	0.05	0.05
$0.1 < \eta < 0.3$	1 GeV	0.0022	0.0023	0.0102	0.09	0.09
$0.1 < \eta < 0.3$	2 GeV	0.0011	0.0012	0.0054	0.05	0.05
$0.1 < \eta < 0.3$	10 GeV	0.0003	0.0004	0.0021	0.03	0.03

The last metric to take into account is the estimation of fakes. To do that, a sample of a $t\bar{t}$ event plus 200 pile-up events was produced in the entire ITk volume. Tracking was performed in the $0.1 < \eta < 0.3$ region, and the number of 1^{st} stage tracks, number of tracks after a χ^2 cut and after OR were counted and compared with the number of truth and offline tracks. The dataflow results are presented in Figure 3.8. The baseline system simulation obtains about three times more tracks (after OR) than offline reconstruction, which in turn produces about two times more tracks than truth.

Constant Generation

As previously explained, the estimation of track parameters and quality of fit are done with linear Equations 1 and 2. The constants for these equations have to be generated prior to their use in tracking, and for this, training single muons are used to reverse calculate the constants. Each muon will go through a set of modules, one in each logical layer, so 8 for the first stage and 13 for the second stage. These sets of modules are called sectors, and each

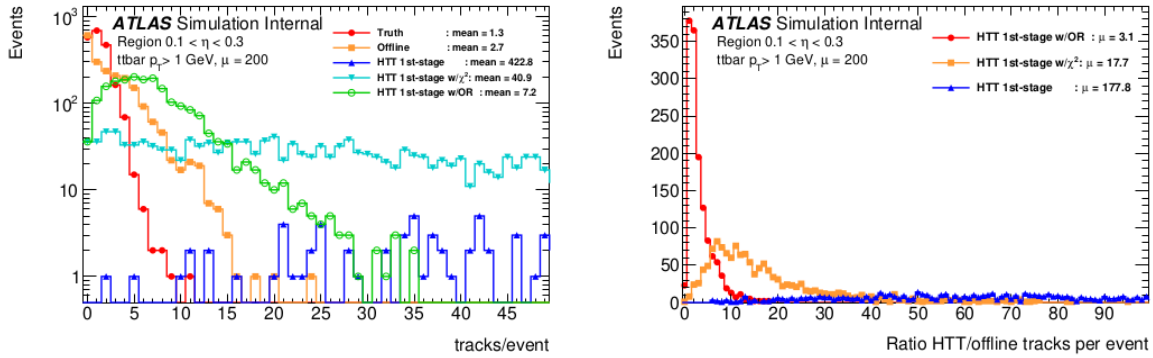


Figure 3.8: Estimation of fake tracks in a $0.1 < \eta < 0.3$ region of a $t\bar{t}$ plus 200 pile-up events sample [19].

sector will have a different set of constants.

To obtain the sectors, the hits of the truth track corresponding to the muon are used. The modules of the hits are organized in ascending order, thus forming a sector. The constants are then computed with the hits' coordinates and the true parameters' values. The objective is to minimize the difference between the fitted ($\tilde{p}_i$) and the truth (p_i) parameters [13], that is:

$$\min\left(\left\langle(\widetilde{p}_i - p_i)^2\right\rangle\right) \qquad \forall i = 1, \cdots, 5 \tag{3}$$

The solution to this problem is based on the covariance matrix of the coordinates:

$$[V]_{lm} = \left(\langle x_l x_m\rangle - \langle x_l\rangle \langle x_m\rangle\right) \tag{4}$$

where the average is done over the coverage of the sector, which is the number of tracks that passed through it.

With this, we can calculate the constants C_{ij} and q_i of equation 1 as:

$$C_{ij} = \sum_m V^{-1}_{lm} \left(\langle x_m p_i\rangle - \langle x_m\rangle \langle p_i\rangle\right) \tag{5}$$

$$q_i = \langle p_i\rangle - \sum_l C_{il} \langle x_l\rangle \tag{6}$$

The constants for the quality of the track fit (χ^2, eq. 2) are obtained from the eigenvalues and eigenvectors of the covariance matrix (V_{lm}, eq. 4):

$$A_{ij} = \frac{u_{ij}}{\sqrt{e_i}} \tag{7}$$

$$k_i = \sum_k A_{ik} \langle x_k \rangle \tag{8}$$

where u_{ij} is the component j of the eigenvector i and e_i its eigenvalue.

3.2 Fast TracKer

The FTK is the system that preceded the HTT. In terms of software, it is very similar to the HTT, it also performs particle tracking with AM ASICs and FPGAs. But it has a bigger number of different boards, which increases the number of connections, amount of firmware needed and overall complexity.

It is interesting to discuss this system, because in 2018, two FTK towers for two $\eta - \phi$ regions were installed and began to perform tracking on real data [20]. This was a proof of concept since we can see if the real results are like the simulations. If so, it would help validate the development of the HTT, since we can have more confidence that it will also have a good performance.

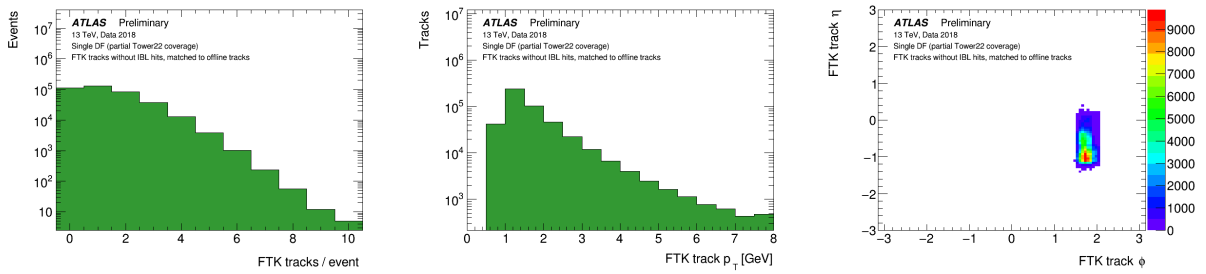
Like in the HTT, the FTK performs tracking in two stages, the first with 8 ID layers and the second with 12. One tower was for the first stage and the other for the second stage. It also groups hits into superstrips and compares with the patterns stored in the AM ASICs, and then fits the tracks.

Performance

For 2018, the 12 layer tower, covering a region which spans $-1.5 < \eta < 0$ and $1.6 < \phi < 2.0$, produced about 500k tracks, with $p_T > 1$ GeV. These first results were good, since the reconstructed tracks were in the correct $\eta - \phi$ region space, the majority very likely corresponded to real particles and were not fake (since they were matched to offline tracks), as shown in Figure 3.9. The track parameter resolutions for p_T, η and ϕ are very close to those obtained by simulation, which means that the FTK is reconstructing tracks with the correct momentum and direction. This can be seen in Figure 3.10.

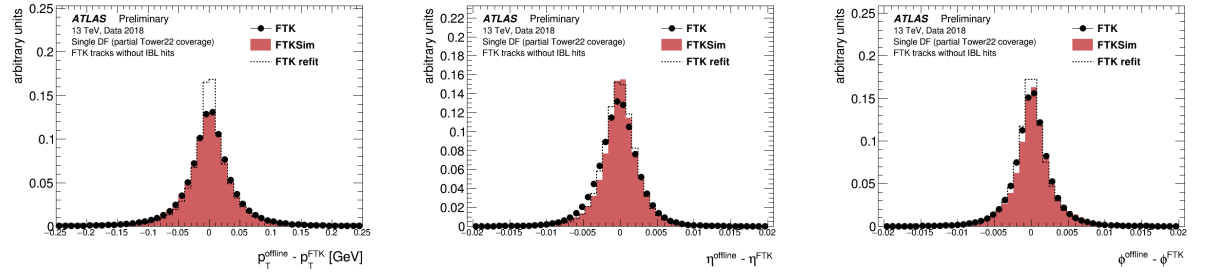
3.3 CMS hardware tracker for the trigger

Another hardware tracker of interest is the one being developed for the CMS experiment, also in the HL-LHC. Its goals are also to perform tracking of charged particles at the trigger level, so that in can efficiently detect events of interest. But, unlike the HTT, it has the goal



(a) (b) (c)

Figure 3.9: Track distributions for the first tracking results of the FTK. (a) number of tracks produced per event, (b) track transverse momentum (p_T) and (c) track distribution in (η vs ϕ) [20].



(a) (b) (c)

Figure 3.10: FTK residuals with respect to the matched offline track for (a) p_T, (b) η and (c) ϕ. The difference between the FTK and matched offline track parameter are shown for the FTK slice (black dots), functional emulation of the FTK slice using FTKSim (shaded red histogram), and FTK refit (dashed line) [20].

of doing so for all proton-proton collisions at the 40 MHz rate, for particles with $p_T > 2$ GeV [21]. This will be possible because of new detector modules with p_T discrimination, which will sufficiently reduce the rate at which data is readout. These modules will consist of two silicon sensors separated by a few millimeters, so a pair of hits, one in each sensor, called stub, will be readout only if their separation corresponds to a track with $p_T > 2$ GeV.

The CMS trigger architecture also consists of a hardware level L1 and a CPU-based High Level Trigger. A key difference is that tracking occurs already at the L1 level. To have an idea of the complexity of each HL-LHC experiment, this system is composed of about 160 boards for 9 nonants, while the HTT is made of about 650 boards for 48 regions.

Three approaches were studied for the development of L1 tracking pattern recognition phase: the tracklet approach, the Hough Transform and the AM approach. The first two were performed in FPGAs and the last one in AM ASICs. Then, all of these were followed by track fitting in an FPGA.

In the tracklet approach, stubs in two consecutive layers or disks are connected to form tracklets (or seeds), given they agree with a track of $p_T > 2$ GeV, $|z_0| < 15$ cm and the shift in position of each of the stub hits must be consistent with the tracklet p_T. Then, track parameters and projections to other layers are calculated for selected pairs and stubs are

matched to the projected tracklet. Lastly, the track is fitted with a linearized χ^2, and the final track parameters are calculated.

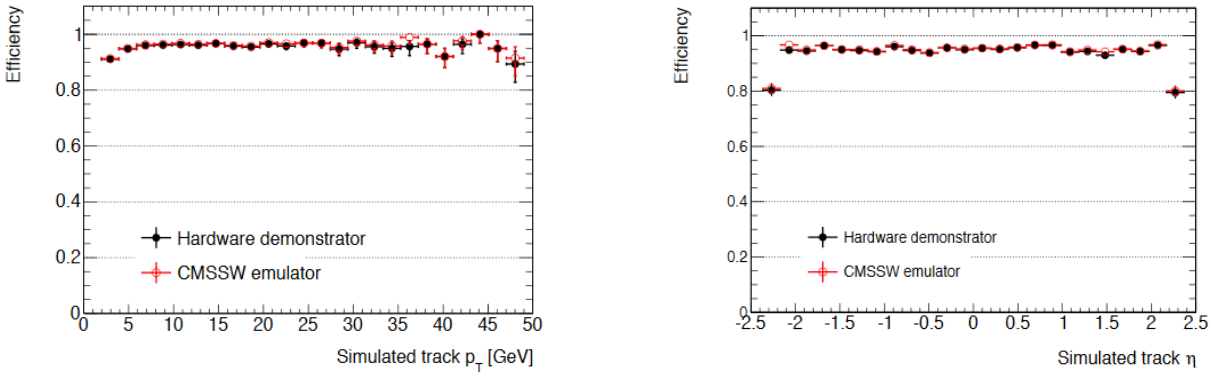
The Hough Transform approach uses the fact that the hit r and ϕ coordinates are known, which is useful for tracking in spherical coordinates. It performs pattern recognition by transforming each hit into a line in the Hough space, and then searches for points where a minimum number of lines cross. The Hough Transform will be explained in more detaield in section 5.1 The fitting and filtering is done iteratively with a Kalman Filter. It starts with a seed estimate of the track parameters obtained in the HT. Then, the trajectory of the seed is projected onto the next layer, and the track parameters are updated with that layer's stub coordinates. This process is repeated for at least four stubs.

The AM + FPGA scenario was similar to the HTT, where the associative memories perform pattern recognition, and track fitting is done in FPGAs, but in this case, a fitting method based on principal component analysis was used.

The CMS collaboration chose to use a hybrid approach, in which pattern recognition is performed using the tracklet approach and the Kalman filter is used for fitting and filtering tracks, with all algorithms running on FPGAs.

Tracking Performance

Here, an emulator and a demonstrator board using simulated data at 200 pile-up interactions were used to study the performance of the CMS hardware tracker for the trigger [22]. Worth mentioning are the graphs in Figure 3.11, which compare emulator and demonstrator results, showing good agreement between the demonstrator and emulator tracking efficiencies.



(a) (b)

Figure 3.11: Track reconstruction efficiency for the primary interaction in $t\bar{t}$ events with 200 pile-up events as function of transverse momentum (left) and pseudorapidity (right). [22]

Chapter 4 HTTSim Code Develpment

As explained in section 2.3, every subsystem and characteristic of the ATLAS detector has to be simulated, in order to understand its response to real collision events and the details of its performance. This is also true of the HTT, where a simulation suite, HTTSim, running in the ATLAS software framework (Athena) was developed to allow the detailed study of this system's performance.

This task had the objective of giving me experience with athena and the code of the full simulation of the HTT, while at the same time developing a useful component for HTTSim. An algorithm called HTTEventSelectionSvc was developed in the context of this thesis, with the purpose of selecting the events to be processed by HTTSim. This selection is based on the running conditions, for example, if running on a simulated muon sample or other types of samples, with or with out pile-up and the region boundaries.

This work happens in the reconstruction step of the ATLAS simulation. The input to HTTSim are files produced by the digitisation step, that contain information of where the ITk and readout systems registered the hits.

4.1 Athena and HTTSim

Athena is the ATLAS software framework. It is based on the Gaudi [23] architecture and on Root [24] objects. It provides the infrastructure to run physics analysis and simulations in a high-level and systematic way, so that one doesn't have to worry about the underlying code that manages the memory, makes the loop running the events, and gets all the configurations needed for the base initialization.

The classes and algorithms are written in C++, to make use of its high performance and proximity to hardware. The framework then creates a python object that corresponds to each C++ class. This is because of the Gaudi architecture, where the configuration and execution of a simulation run, called a job, is done with python scripts, called jobOptions.

There are 3 main types of class that derive from the Gaudi software: Algorithms, Tools and Services, while other classes that usually represent physics objects, such as detector hits and particle tracks, derive from the Root TObject. Algorithms are where the physics analysis or simulations are defined, that is, which manipulations on which data are performed. These manipulations are defined in Tools and Services, and can be for example road finding, hit clustering, event selection, among many other. Typically, Algorithms use Tools and Services to execute common tasks. There is a subtle but important difference between Tools and Services. The latter are singleton classes, which means there can only be one instance of each Service, while Tools can have multiple instances with different configurations. This is because users may want to perform road finding using different configurations in the same algorithm,

for which they would use different Tool instances, but perform event selection in the same consistent way, which would be done by a Service.

Algorithms and Tools have 3 main methods which are `initialize()`, `execute()` and `finalize()`, while services only have the `initialize()` and `finalize()` methods. The `initialize()` and `finalize()` methods of each class instance are executed at the beginning and end of each job. The `execute()` method is executed once per event during the event loop. First, the `initialize()` methods of all the pieces defined in the user's jobOptions file (and their dependencies) are run, so that they load their corresponding properties values. Then, the event loop runs on a certain number of events, calling the Algorithm's `execute()` method each time, where all the calculations and analysis specified by the user are performed. After all events are processed, the `finalize()` methods of all the pieces are called, outputting files, values and cleaning up the environment.

The HTTSim code is built on top of Athena, so that all the algorithms, tools, services and objects derive from Athena or Root classes. It has one algorithm (called "HTTLogicalHitsProcessAlg") considered to be the principal one, where the simulation of the HTT is controlled. This is where the road finding is set to use pattern matching or the Hough transform, the fitting method is chosen, among other definitions. The results presented in this thesis were run with this algorithm.

The HTT simulation starts from ITk raw hits, which where previously obtained by running the ATLAS full simulation on the ITk. The raw hits are then transformed to logical hits by the Mapping service. The difference between raw and logical hits is that the former are as the detector readout system would produce them, while the latter are the raw hits mapped to the HTT logical layers. Each 5 physical ITk pixel layers and each side of the 4 strip layers count as a logical layer, for a total of 13 logical layers. The configuration used in HTTSim was the outermost pixel layer and the 7 outer strip layers for the first stage, and all logical layers for the second stage. Since pixels have two coordinates and strips have one, there are a total of 9 coordinates for the first stage and 18 for the second. With the logical hits, a tool performs road finding to get all the possible track candidates. These roads are then fitted to obtain their track parameters and quality by the track fitting tool. To select the best track and reject fakes, the Overlap Removal tool is run over the track candidates.

4.2 HTTEventSelectionSvc

The HTTEventSelectionSvc is an Athena Service made with the purpose of selecting the events to be processed by the HTT simulation, as mentioned in the beginning of this chapter.

It was implemented in a C++ class, where the member properties hold information such as the region boundaries, that is, the maximum and minimum values for each helix parameter (q/p_T, ϕ_0, η, d_0, z_0). Other member variables hold the type of sample (if the events are single

muons, long lived particles (LLPs), etc) one wishes to run, if it has pile-up or if it's going to perform Large Radius Tracking.

The main method is the `selectEvent()`. Here, it first checks the type of sample, which can be skipTruth, singleMuons, singleElectrons or singlePions, or LLPs. The first case is simple, it passes all the events. If it is one of the single particle samples, it checks the truth tracks of the event (which are tracks known to correspond to a truth level particle generated in the event generation step), to see if there is at least one with parameters inside the region boundaries and with the corresponding PDG code [25], for which it passes the event. The case for long lived particles was not developed by me.

The single muons sample (also used in Chapter 5, unless otherwise specified) was produced with a "Particle Gun", which is a type of event simulator that only produces one particle at the interaction point, with the specified track parameter distributions. The single muons were produced with $p_T > 1$ GeV and had a flat distribution in q/p_T, η and ϕ_0, as shown in Figure 4.1. Ten thousand events were simulated in a region with the following boundaries:

- $0.1 \leq \eta \leq 0.3$
- $0.3 \leq \phi \leq 0.5$ (rad)
- $-0.001 \leq q/p_T \leq 0.001$ (MeV^{-1})
- $-2 \leq d_0 \leq 2$ (mm)
- $-150 \leq z_0 \leq 150$ (mm)

To pass the truth cut made by the HTTEventSelectionSvc, each event was required to have at least one truth track with helix parameters inside the region boundaries, and a PDG code of 13 or -13 corresponding to a true muon. Of the 10000 events simulated, 8812 passed the HTTEventSelectionSvc cuts, and the distributions of the helix parameters of the truth track are in Figure 4.2. The 1188 events rejected by the HTTEventSelectionSvc had a true muon track with at least one parameter outside the region boundaries, as can be seen in Figure 4.3. That truth track was either beyond the $|z_0| < 150$ or $|d_0| < 2$ (mm) limits.

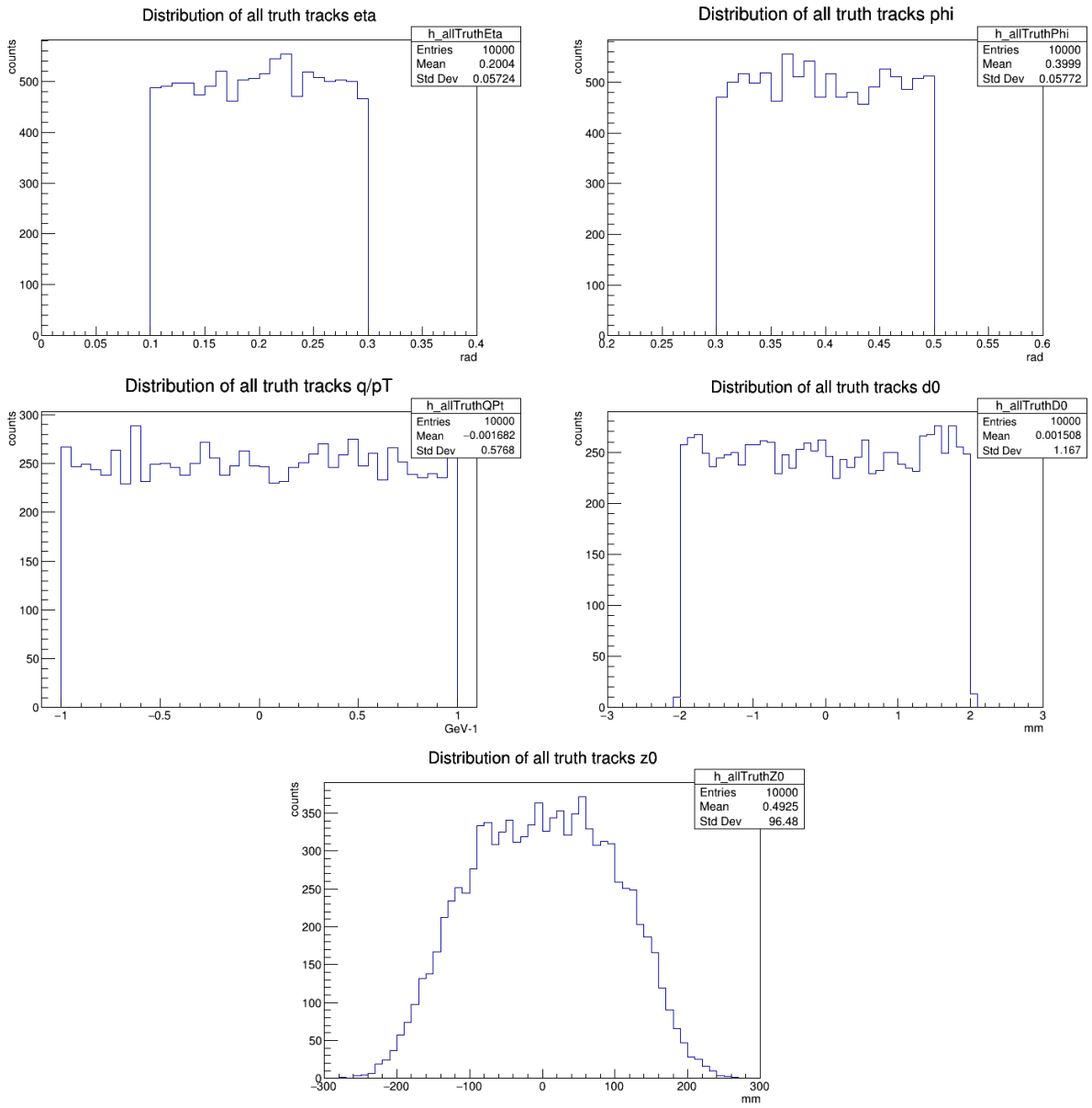


Figure 4.1: Distribution of the 5 helix parameters of the truth tracks corresponding to the single muon. The z_0 distribution goes from -300 to 300 mm, to show that some of the muons were outside the region boundaries.

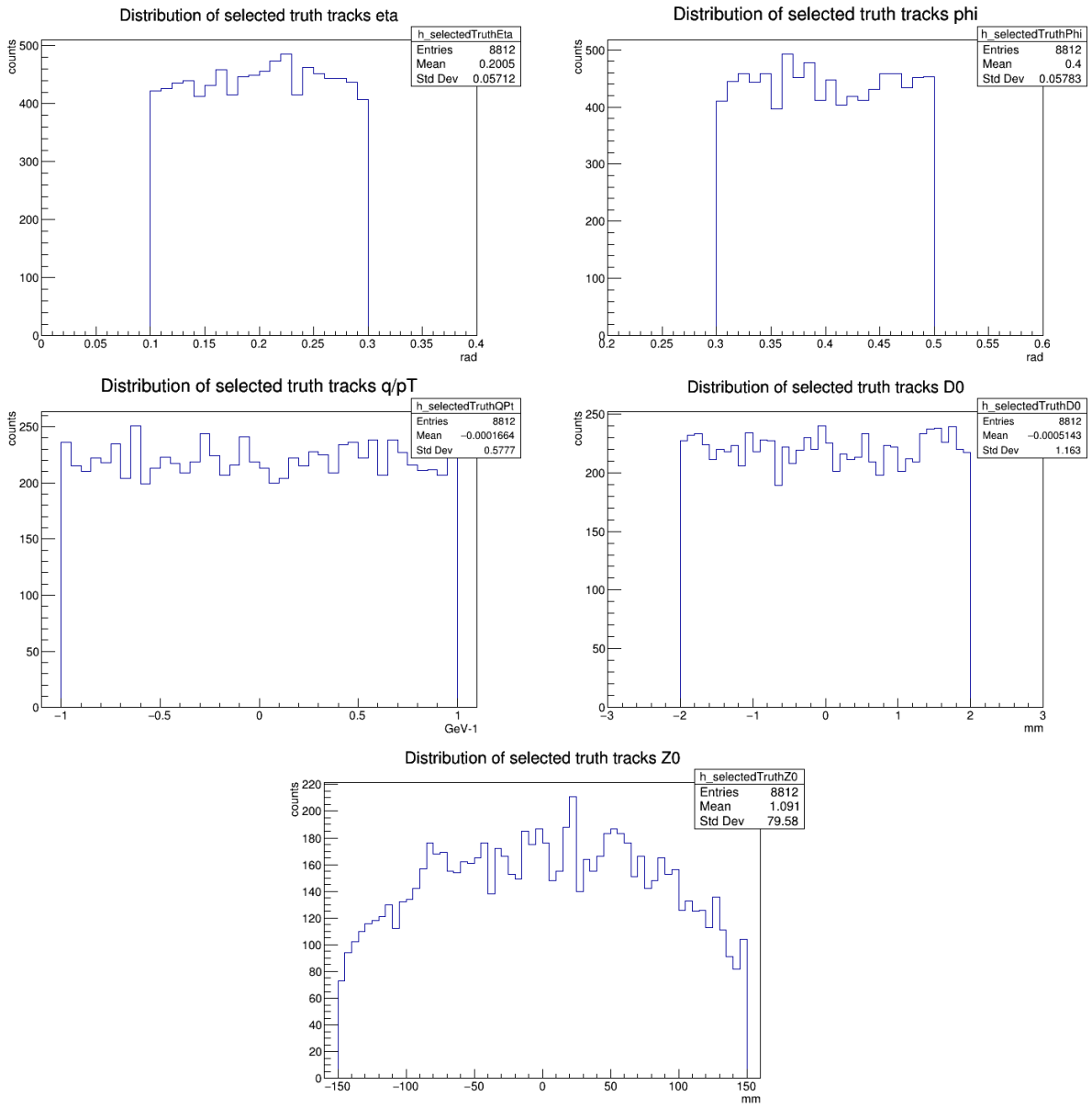


Figure 4.2: Distribution of the 5 helix parameters of the truth tracks that passed the EventSelectionSvc.

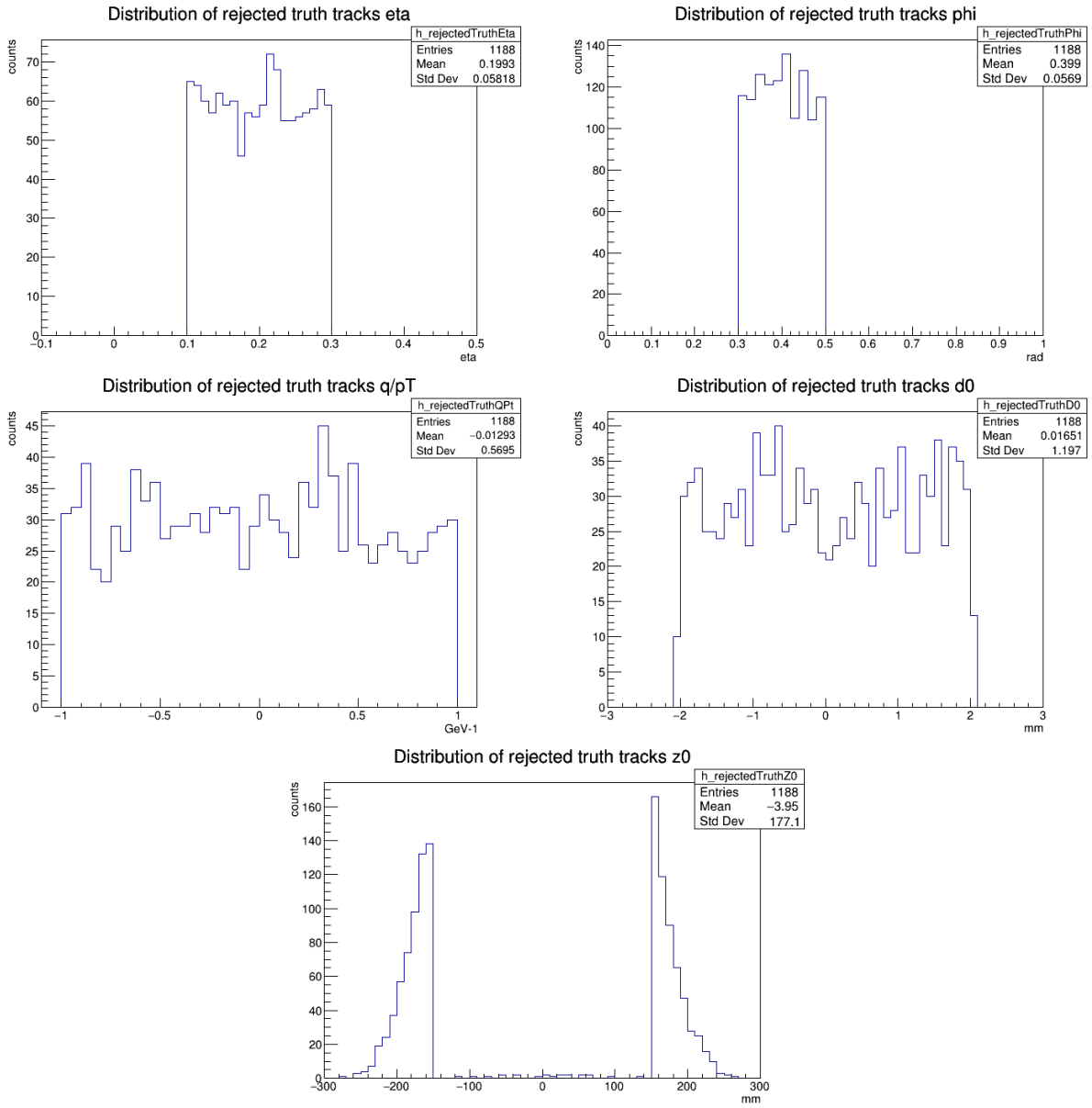


Figure 4.3: Distribution of the 5 helix parameters of the truth tracks rejected by the EventSelectionSvc.

Chapter 5 Tracking with the Hough transform and an idealized detector geometry

This chapter documents the main project of this thesis, which is a method to find roads and determine the ϕ_0 and q/p_T of tracks based on the Hough Transform and an idealized detector geometry. The approximation of the detector to a perfectly cylindrical one allows the use of a different type of sectors and coordinates, as explained in section 5.3. This method was studied and developed as an alternative to the baseline plan that uses AM ASICs for road finding.

5.1 Hough Transform

The Hough Transform is a mathematical tool mostly used in digital image processing to find lines, circles or other shapes. Because charged particles in the uniform magnetic field of the ATLAS ID move in a helix, in the transverse plane their path will be a circle. So, the HT can be used to find roads of the particles, which will have a unique ϕ_0 and q/p_T.

Derivation of the Hough Equation

One can derive the Hough equation to find charged particles in a magnetic filed moving in a circle, starting from the Lorentz force, $\mathbf{F} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B}$. Taking into account that $\mathbf{E} = 0$ and $\mathbf{B} = B\hat{z}$, the force acting on the particle becomes

$$\mathbf{F} = qvB\hat{r} \tag{9}$$

which makes the particle move in a circle, if its momentum is constant, so the corresponding radial acceleration is[1]

$$\mathbf{F} = m\frac{v^2}{r}\hat{r} = \frac{p_T v}{r}\hat{r} \tag{10}$$

Making (9)=(10), we can find an expression for the transverse momentum:

$$p_T = qBr\ [kg \cdot m \cdot s^{-1}] \approx 0.3qBr\ [GeV \cdot c^{-1}] \tag{11}$$

Because we are trying to find circles, one can use the equation for a circle and evaluate it in a pair of points (x_1, y_1) and (x_2, y_2). Subtracting the two, expressing everything in polar

[1] In this illustrative derivation, the particle is considered to be non-relativistic.

coordinates and manipulating the formula further as explained in section 8.1 of [26], we arrive at

$$0.15\frac{qB}{p_T} = \frac{r_1 \sin(\phi_0 - \varphi_1) - r_2 \sin(\phi_0 - \varphi_2)}{r_1^2 - r_2^2} \tag{12}$$

Making point (x_2, y_2) the origin, and knowing that in the ATLAS central tracking volume $B = 2$ T, the Hough equation for this case becomes:

$$\frac{qA}{P_T} = \frac{\sin(\phi_0 - \varphi_1)}{r_1} \tag{13}$$

where A = 0.0003 GeV $c^{-1}mm^{-1}e^{-1}$

The Accumulator

The way the Hough Transform works in this case, is that for each hit in the detector (with coordinates r and φ), there will be a line in $(\phi_0, q/p_T)$ space (also called Hough space), as shown by Figure 5.1. Let's suppose the detector has 4 layers. If there is a point in Hough space with the intersection of 4 lines, then it means there is a track with ϕ_0 and q/p_T corresponding to the coordinates of that point. This is the way the HT is used to find roads, and at the same time obtain an estimate of the ϕ_0 and q/p_T of the associated track.

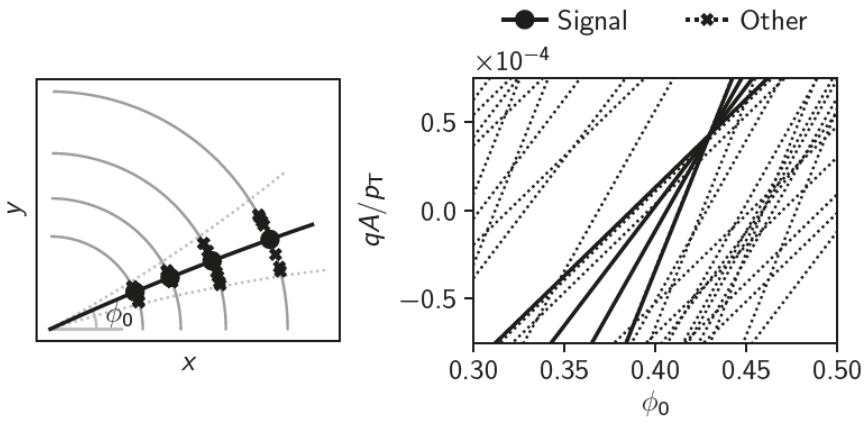


Figure 5.1: Road finding with the Hough Transform [26]. Each ITk hit is transform to a line in Hough space. The intersection of a minimum number of lines indicates a possible road.

But a computer doesn't have infinite resolution, so the Hough space has to be divided into a grid with finite size bins along ϕ_0 and q/p_T. This is what's called the accumulator. In HTTSim, each bin stores the number of lines going through it as well as the corresponding hits. So to find a track candidate, one just needs to set a minimum limit for the number of lines in a bin of this 2D histogram.

In the case of single muons this is very straightforward, but if high pile-up conditions are added, the accumulator gets overflown with lines and it becomes hard to distinguish signal from background. This can be eased by spliting the $\eta \times \phi$ region into slices along z_0, since it extends from -150 mm to 150 mm. By doing this, there will be more accumulators to process, but it will be easier to find real tracks.

5.2 Idealized geometry

As presented in section 2.4, the Pixel and Strip modules in each layer are flat and disposed in a way that makes them overlap with their neighbours at the edges. This arrangement of the modules makes the detector not perfectly cylindrical, which for linearized track fitting will cause nonlinearities. For example, if a particle strikes the overlapping area of two adjacent modules, it will cause two hits with two different sets of coordinates. This will result in different track parameter values than if the detector were an ideal cylinder. Other examples for nonlinearities are discussed in reference [27].

The objective of transforming the location of hits into the ideal detector geometry is to improve the performance of the linearized track fitting, and achieve the best possible utilization of the FPGAs, by minimizing the impact of nonlinearities. For the perfectly cylindrical detector, the hits radial coordinate, R, would not usefully contribute to the principal components analysis performed to the hit correlation matrix, since the radius is constant for each layer. So, this method uses only the φ and z coordinates of the input hits.

Supposing a particle comes from the interaction point (i.e. $d_0 = 0$), it's possible to obtain the coordinates (φ, z) of a hit that would lie in the trajectory of the track. Solving the Hough equation (13) with respect to φ and z:

$$\varphi = \phi_0 - \arcsin(\frac{RAq}{p_T}) \tag{14}$$

$$z = z_0 + \frac{p_T}{Aq} \arcsin(\frac{RAq}{p_T}) \cot\theta \tag{15}$$

The hit transformation equations to the ideal detector coordinates (R', φ', z') were chosen to have a linear dependence on the track parameters, and are the first order approximation of Equations 14 and 15, substituting the real R with the ideal R':

$$\varphi' = \phi_0 - \frac{R'Aq}{p_T} \tag{16}$$

$$z' = z_0 + R' \cot\theta \tag{17}$$

Solving Equations 16 and 17 with respect to ϕ_0 and z_0, and substituting them in Equa-

tions 14 and 15, the coordinates on an ideal layer, in function of the track parameters and the hit's real coordinates, become:

$$\varphi' = \varphi - \frac{R'Aq}{p_T} + \arcsin\left(\frac{RAq}{p_T}\right) \tag{18}$$

$$z' = z + \frac{p_T}{Aq}\cot\theta\left[\frac{R'Aq}{p_T} - \arcsin\left(\frac{RAq}{p_T}\right)\right] \tag{19}$$

However, if this were to be implemented in online track fitting, it would take too much time, because of the arcsin() calculation. So using the series expansion of the arcsin(), which is

$$\arcsin(x) \simeq x + \frac{1}{6}x^3 + \cdots \tag{20}$$

makes the equations suitable for use in FPGAs, where the linear term is refered to as first order and the cubic term as second order:

$$\varphi' = \varphi + \frac{Aq}{p_T}(R - R') + \frac{1}{6}\left(\frac{RAq}{p_T}\right)^3 \tag{21}$$

$$z' = z - \cot\theta(R - R') - \frac{\cot\theta R^3}{6}\left(\frac{Aq}{p_T}\right)^2 \tag{22}$$

5.3 Delta Phis Method

As stated previously, this method is based on the Hough transform for road finding and estimating the ϕ_0 and q/p_T, and on the linearized fit, to correct these parameters. In the constant generation step, the parameters to minimize are:

$$\Delta\phi_0 = \phi_0^{HT} - \phi_0^{truth} \tag{23}$$

$$\Delta q/p_T = (q/p_T)^{HT} - (q/p_T)^{truth} \tag{24}$$

The coordinates in this method are the "delta global phis" ($\Delta\varphi$), which are the difference between the idealized hit global phi (φ'_{hit}) and the ideal expected global phi (φ'_{exp}) from the estimated "track" found using the Hough transform. The latter is the azimuthal coordinate of the intersection of that "track" with the ideal detector layers. So the delta global phis will be $\Delta\varphi = \varphi'_{hit} - \varphi'_{exp}$ for each layer, where:

$$\varphi'_{exp} = \phi_0 - \frac{R'Aq}{p_T} - \frac{1}{6}\left(\frac{R'Aq}{p_T}\right)^3 \tag{25}$$

$$\varphi'_{hit} = \varphi_{hit} + \frac{Aq}{p_T}(R - R') + \frac{1}{6}\left(\frac{RAq}{p_T}\right)^3 \tag{26}$$

With these coordinates, a $\Delta\phi_0$ and $\Delta q/p_T$ are calculated with the linear fit. The final parameters will be:

$$\phi_0 = \phi_0^{HT} - \Delta\phi_0 \tag{27}$$

$$q/p_T = (q/p_T)^{HT} - \Delta q/p_T \tag{28}$$

In a perfectly cylindrical detector, the relative position of the hits of two tracks with the same p_T but different ϕ_0 are going to be the same. I.e. $\Delta\varphi^{layer2} - \Delta\varphi^{layer1}$ only depends on p_T and not on ϕ_0. This is to say that the constants used to fit a track only depend on it's momentum. So, instead of the normal definition of sectors with one module per layer, only a small interval in p_T is necessary, or in this case, a bin in q/p_T. For example, a track with an estimated (from the HT) q/p_T of 0.00095 MeV^{-1} would fall on the bin corresponding to $0.001 < q/p_T < 0.0009$ MeV^{-1}, and use the constants obtained for that bin.

The use of this sector definition immensely reduces the number of sectors and constants that must be saved in the memories of the FPGAs.

5.4 Results

To obtain the results shown in this section, the constant generation was run on a single muon sample, in a region with the same parameter boundaries as presented in section 4.2. For the sectors, 15 q/p_T bins were defined:

{ -0.001, -0.0009, -0.00075, -0.0006, -0.00045, -0.0003, -0.00015, -0.000075, 0.000075, 0.00015, 0.0003, 0.00045, 0.0006, 0.00075, 0.0009, 0.001 } [MeV^{-1}].

The HT was configured with a threshold of 7/8 layers hit, which means some sectors obtained had 1 "Wild Card". This gives a total of 135 (9×5) different sectors.

The objective was to study the same performance metrics as presented in section 3.1, which are for single muons: resolutions, χ^2 divided by number of degrees of freedom ($\chi^2/ndof$) distribution and tracking efficiency. To test the method in more difficult conditions, a sample with a single muon plus 200 pile-up events was used to study the tracking efficiency. The

estimation of fakes was done using the $t\bar{t}$ plus 200 pile-up events sample.

Single Muons

As stated in section 3.1, the tracking efficiency is dominated by the road finding efficiency, and here it was considered to be the number of events with a matched muon dividing by the number of events with a truth muon (HTT/truth). With the Hough transform, the tracking efficiency is 98.434% (8674/8812). This is because the HT didn't find any roads in 138 events, as can be seen in plot 5.2.

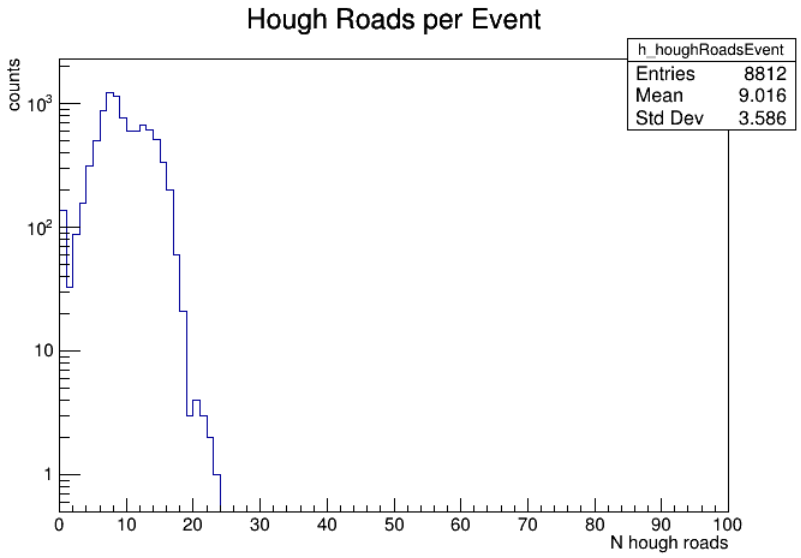


Figure 5.2: Number of roads found by the Hough transform. There are 138 events with zero roads.

It's a lower efficiency than the baseline presented in Figure 3.7, and the reasons for this are the number of hits per event, and the number of layers hit per event. The HT was configured with a threshold of a minimum of 7 layers hit out off 8 and in some of the events with zero roads there weren't even 7 hits. In most of them, there weren't the minimum of 7 layers hit. This happens mostly for muons with low p_T and all this information can be seen in Figure 5.3. The number of events with zero roads, but with 7 or 8 layers hit is about 30. So, if we count only those as inefficient events, the tracking efficiency becomes around 99.5%, which is almost as good as offline.

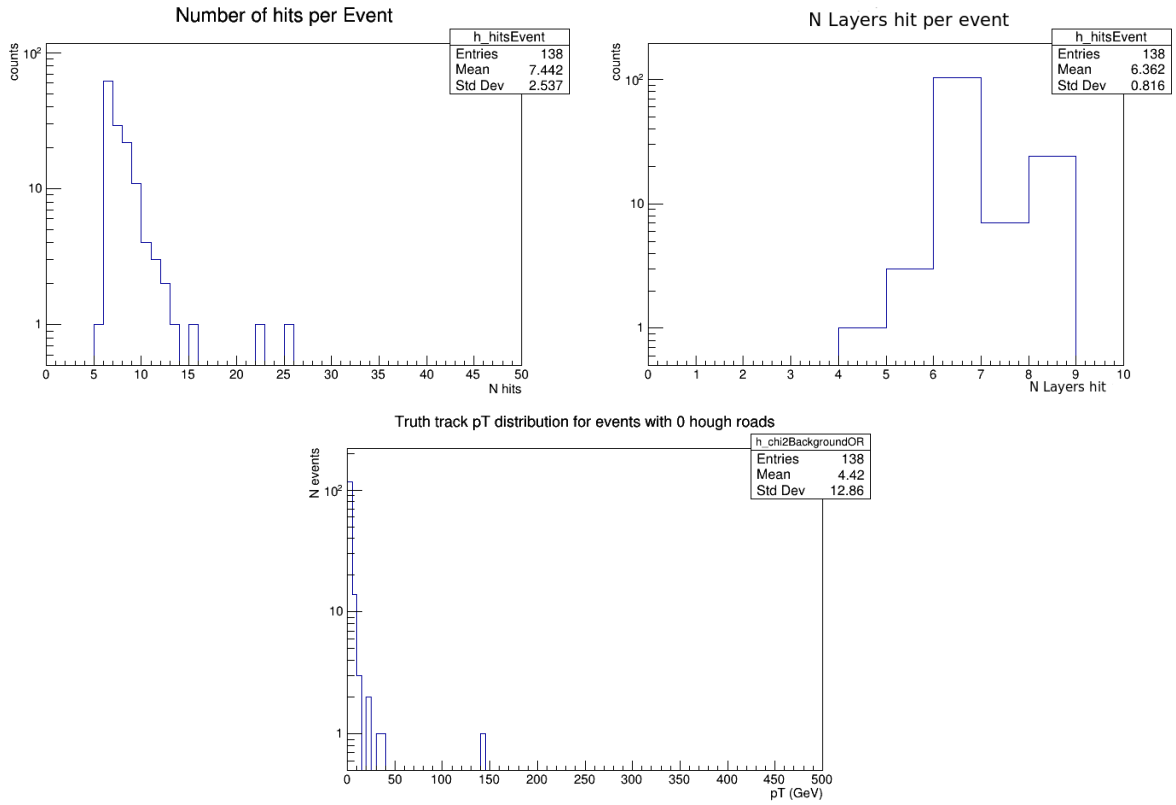


Figure 5.3: For events with zero roads, the distributions of number of hits per event (top), number of layers hit per event (middle) and the p_T of the muons (bottom) are plotted.

The resolutions of the parameters are calculated with the distributions of the residuals (fitted parameter - truth parameter). The 2.5% tails on each side of the distributions are not counted for the rms_{95} calculation. The residuals of the ϕ_0 and q/p_t obtained with this method are shown in Figure 5.4, and their respective resolutions are $rms_{95}(\phi_0) = 0.0053$ rad and $rms_{95}(q/p_T) = 0.0221$ GeV^{-1}.

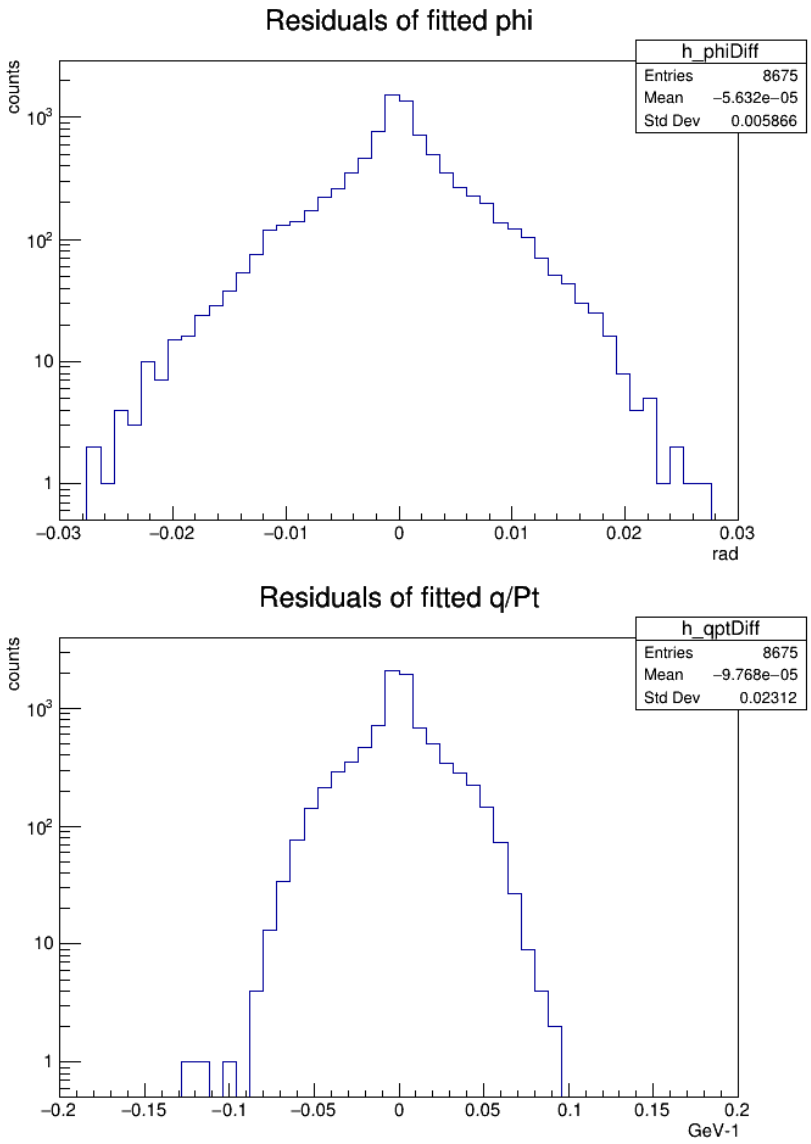


Figure 5.4: Distribution of the residuals (fitted parameter minus truth parameter) of the tracks obtain with the delta phis method. Overlap removal was run without cutting on the chi2. Ideal geometry equations with just the first order approximation.

The $\chi^2/ndof$ distribution of the fitted tracks is shown in Figure 5.5.

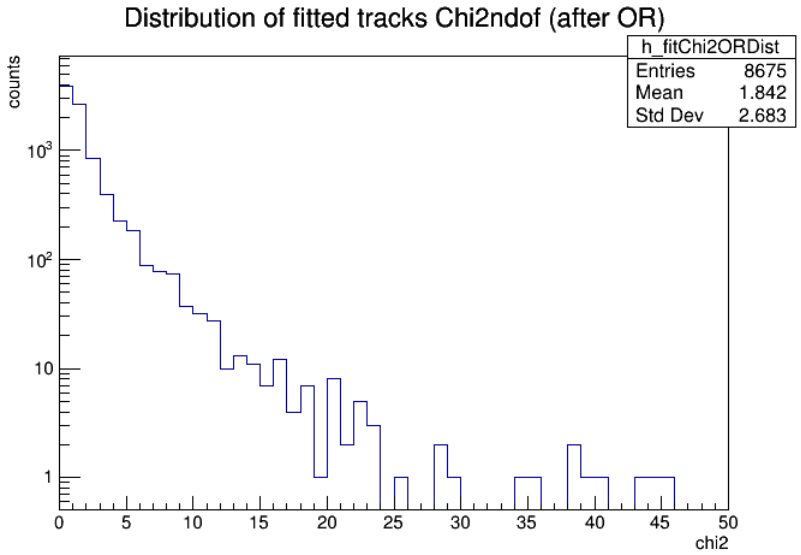


Figure 5.5: Distribution of the $\chi^2/ndof$ of all tracks that passed the Overlap Removal (OR)

On average, the Hough transform finds 9 roads in single muon events with no pile-up. To see if the track that passed the overlap removal is the one with the closest parameters to truth, the plot in Figure 5.6 was produced. It's the difference between the $\chi^2/ndof$ of the fitted track and the $\chi^2/ndof$ of the track with the closest parameters to truth (by eucledian distance in Hough space and the same number of layers hit). Since the Hough transform has a threshold of 7 layers hit, some roads can have 7 layers, while the track chosen by the overlap removal has 8 layers. The overlap removal chooses the track with the highest number of hits first, and then the one with the lowest $\chi^2/ndof$. This plot has a tail on negative values, which means that the track chosen by the overlap removal always has the lowest $\chi^2/ndof$. Or the $\chi^2/ndof$ are equal, as for the bin in 0. (even though the bin goes from 0 to 0.5, there were no events were this difference was positive).

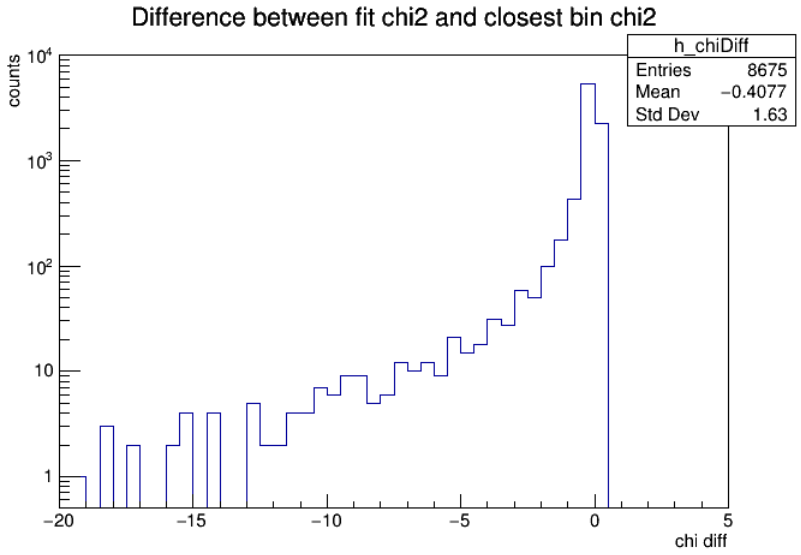


Figure 5.6: Difference between the $\chi^2/ndof$ of the track that passed the OR and the track coming from the HT bin with ϕ_0 and q/p_T closest to truth (by euclidean distance in the Hough space).

Figure 5.7 answers the question of whether the particle found is the real muon. This is called truth matching, and to consider the real particle as found, at least 50% of the track's hits have to be the same as in the truth track. The answer is yes except in one event.

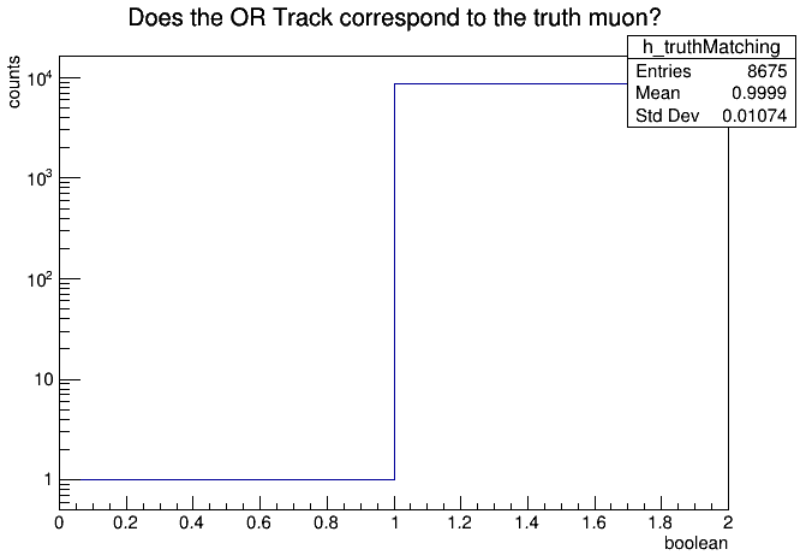


Figure 5.7: Truth matching efficiency for events with at least one Hough road. To consider a track to be matched to truth, it needs to have at least 50% of its hits correspond to the hits of the truth track.

The mean of the $\chi^2/ndof$ distribution should be 1, but it is 1.842 as can be seen in Figure 5.5. Also, just one standard deviation of the residuals of ϕ_0 and q/p_T is already larger than the rms_{95} values presented in Table 3.8. This raises the question if it is consistently doing bad fits, or if there is a systematic error correlated with some property of the simulation. To

find this, plots of the residuals and $\chi^2/ndof$ versus the truth track parameters were produced, and are in Figures 5.8, 5.9 and 5.10. The residuals and $\chi^2/ndof$ are more or less flat in η, ϕ, $d0$ and z_0. The plots over q/p_T are the ones that show where the performance is worse, which is for tracks with $1 < p_T < 2$ GeV.

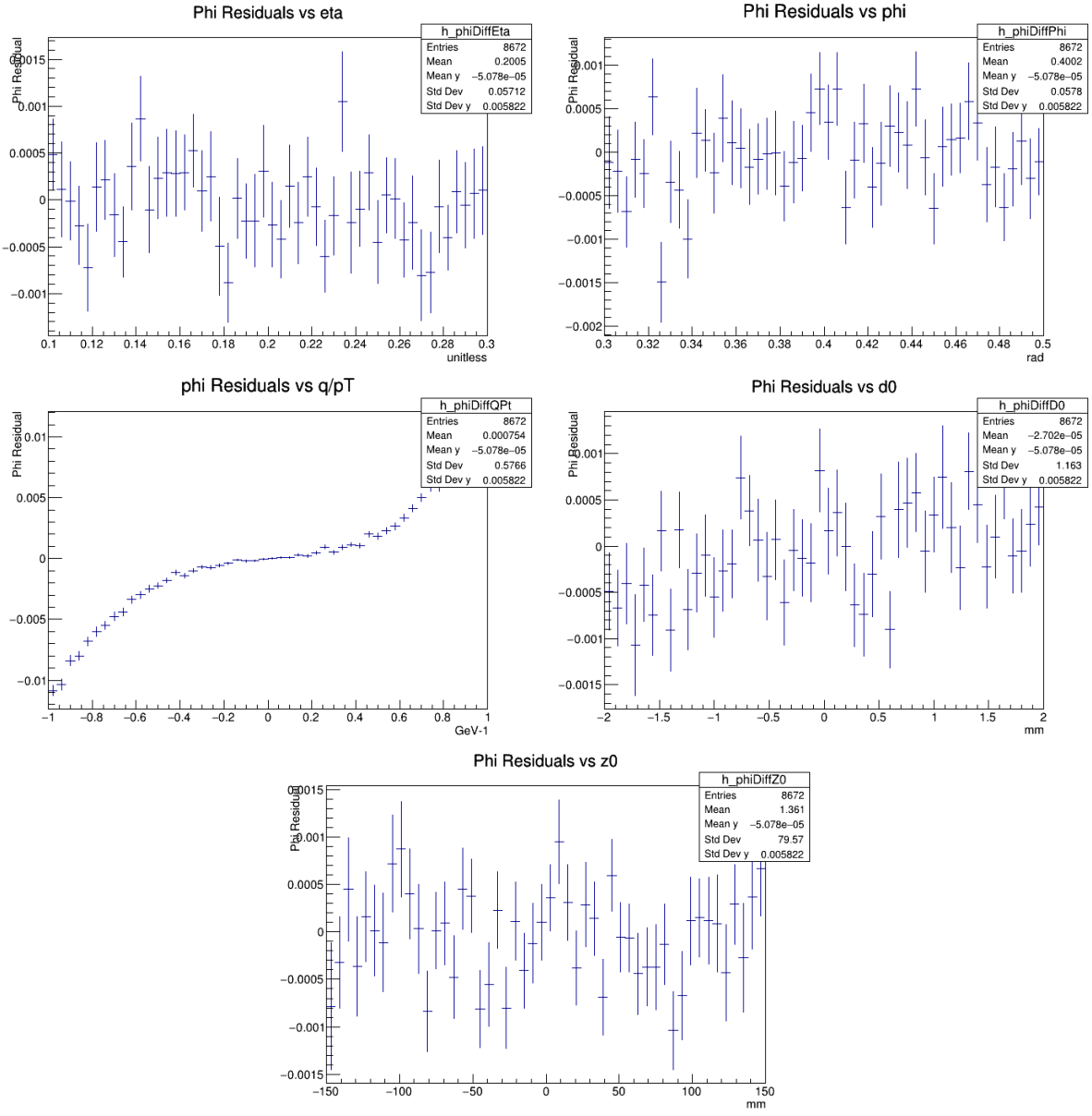


Figure 5.8: Distributions of the residuals of ϕ_0 as function of the truth track parameters.

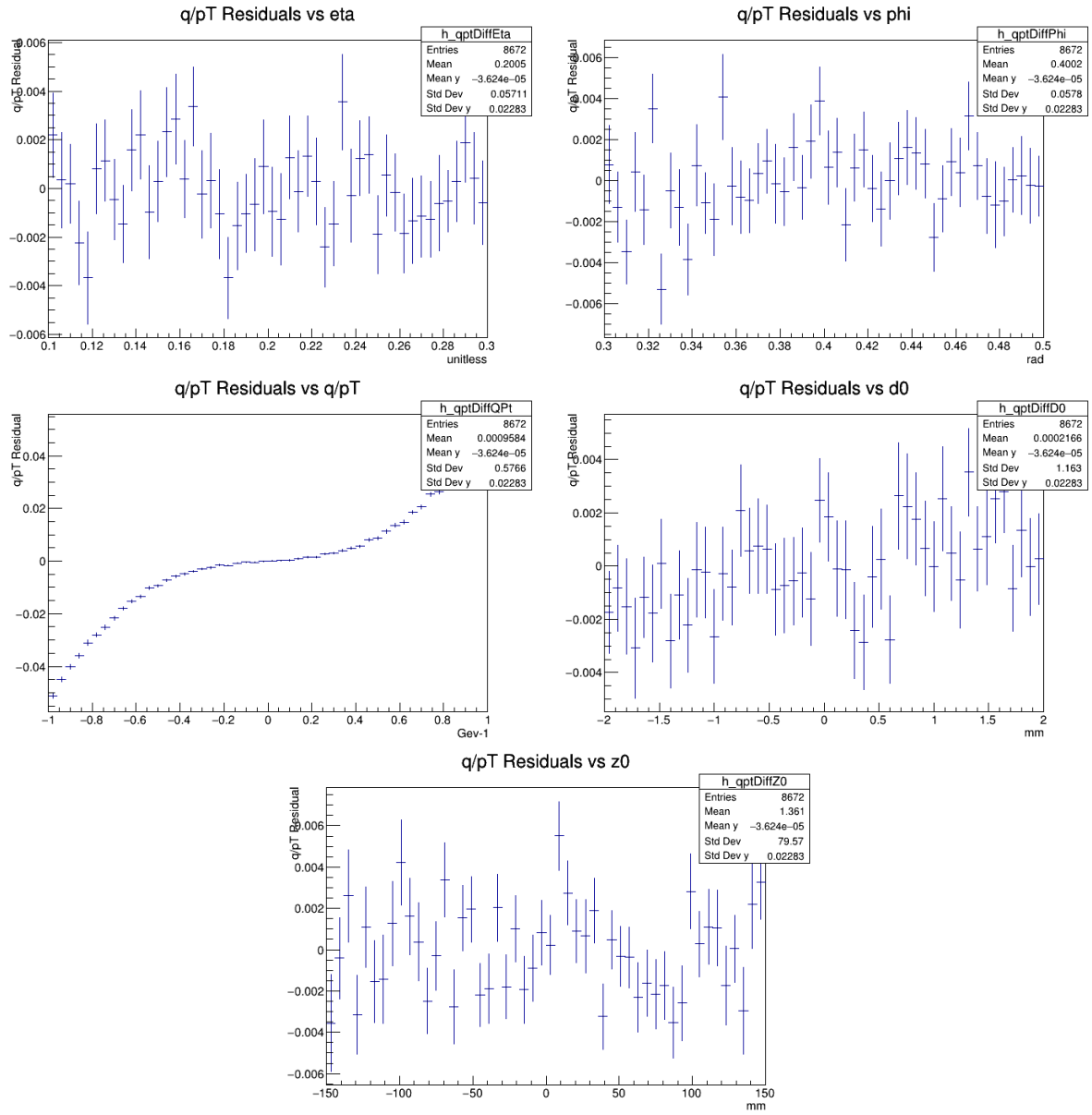


Figure 5.9: Distributions of the residuals of q/p_T as function of the truth track parameters.

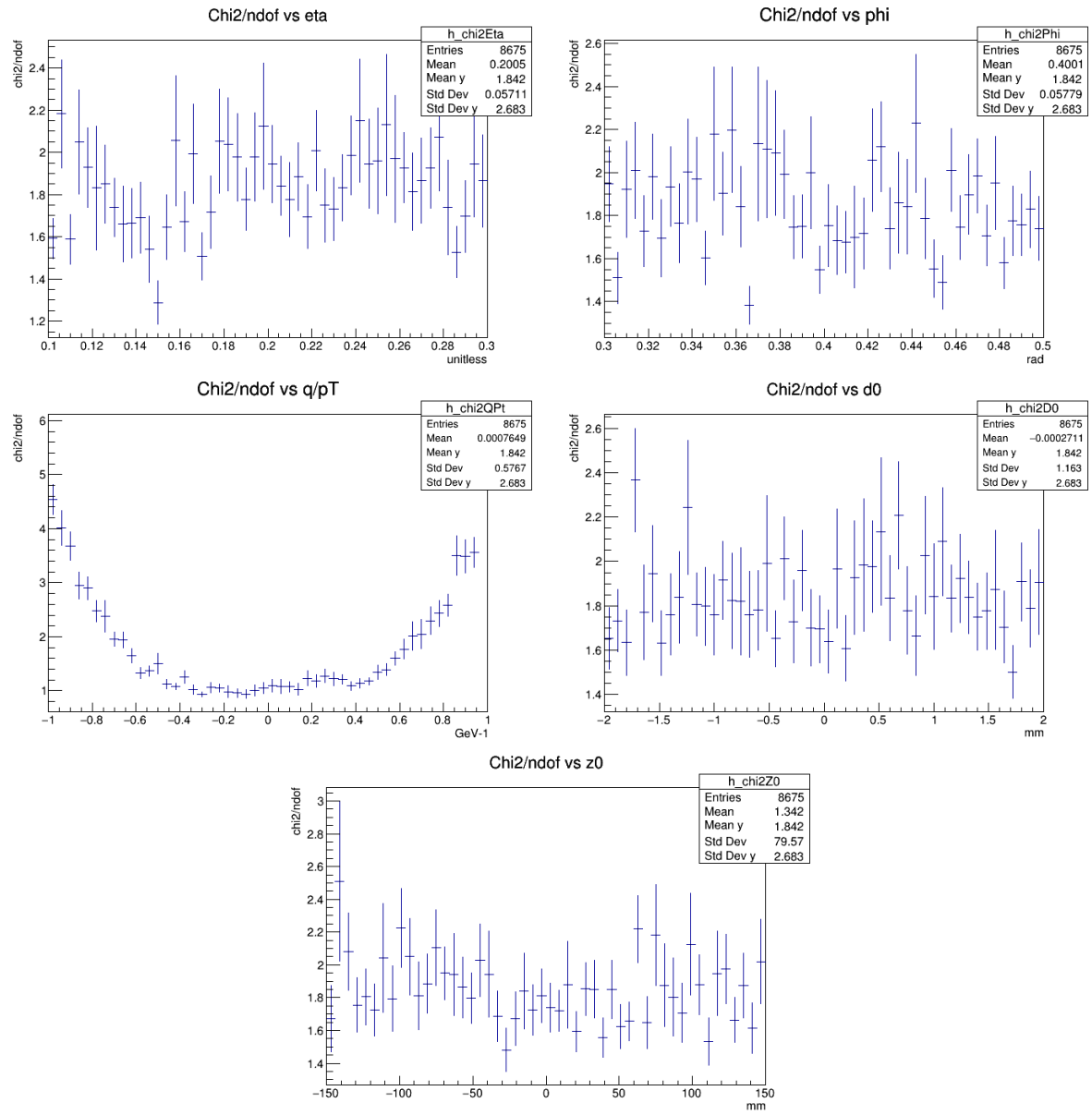


Figure 5.10: Distributions of the normalized χ^2 as function of the truth track parameters.

The Overlap Removal needs a value defined for the $\chi^2/ndof$ cut. So far that value has been very high, to perform the OR only with the common hits between the tracks, and see the tail of the $\chi^2/ndof$ distribution. However, for the $t\bar{t}$ events presented below, a more realistic value has to be found. The objective is to use a $\chi^2/ndof$ cut that achieves 99% truth matching efficiency for single muons. Figure 5.11 shows the truth matching efficiency versus the $\chi^2/ndof$ cut. Using the hit transformation equations with just the first order, the cut that achieves 99% efficiency is 7, and using the second order of the equations, that number is 5. These values will be used in the section for the $t\bar{t}$ results.

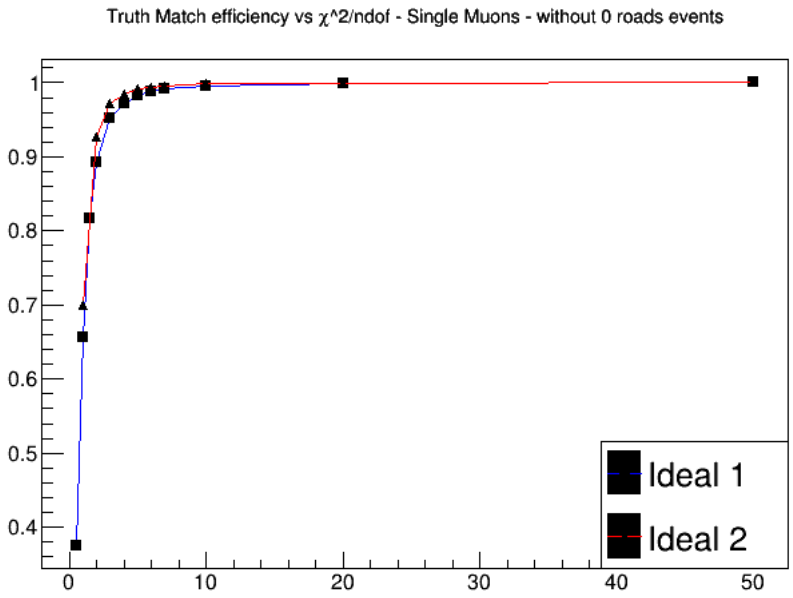


Figure 5.11: In this plot, the efficiency calculated was the number of events where a track was matched to the truth muon over the number of events where the HT found at least one road.

Because the development of this method is only for the first stage tracking (so far), we can restrict the simulation to single muons with $p_T > 2$ GeV. The results obtained for this condition are presented in Figure 5.12, and the resolutions are $rms_{95}(\phi_0) = 0.0020$ rad and $rms_{95}(q/p_T) = 0.0064$ GeV^{-1}. The resolutions are now the same as the baseline system, and the average of the $\chi^2/ndof$ is closer to 1.

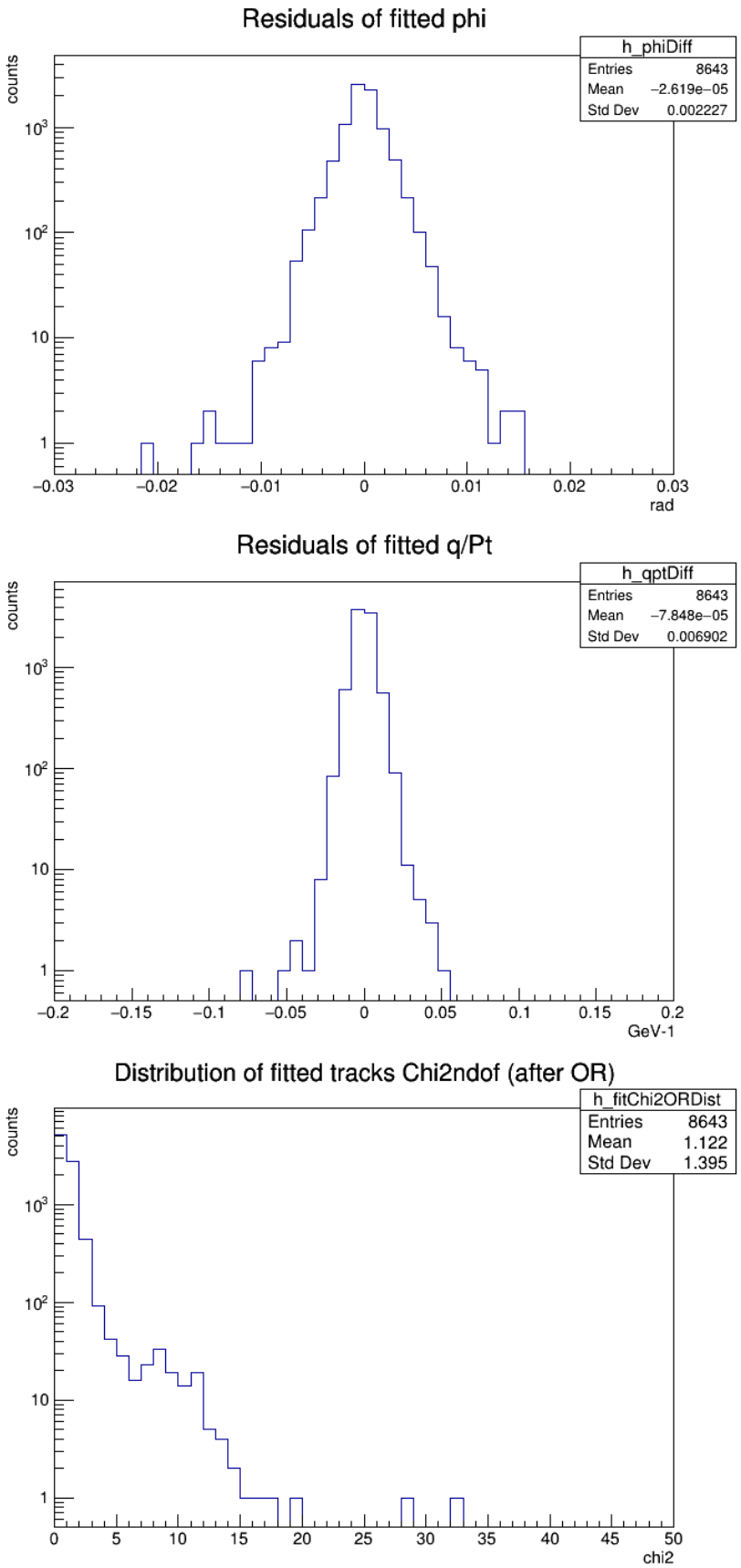


Figure 5.12: Performance of the method on single muons with $p_T > 2$ GeV.

Initially, equations 21 and 22 were being used with just the first order, because the distribution of the delta global phis (the coordinates) were almost identical, as shown in Figure 5.13.

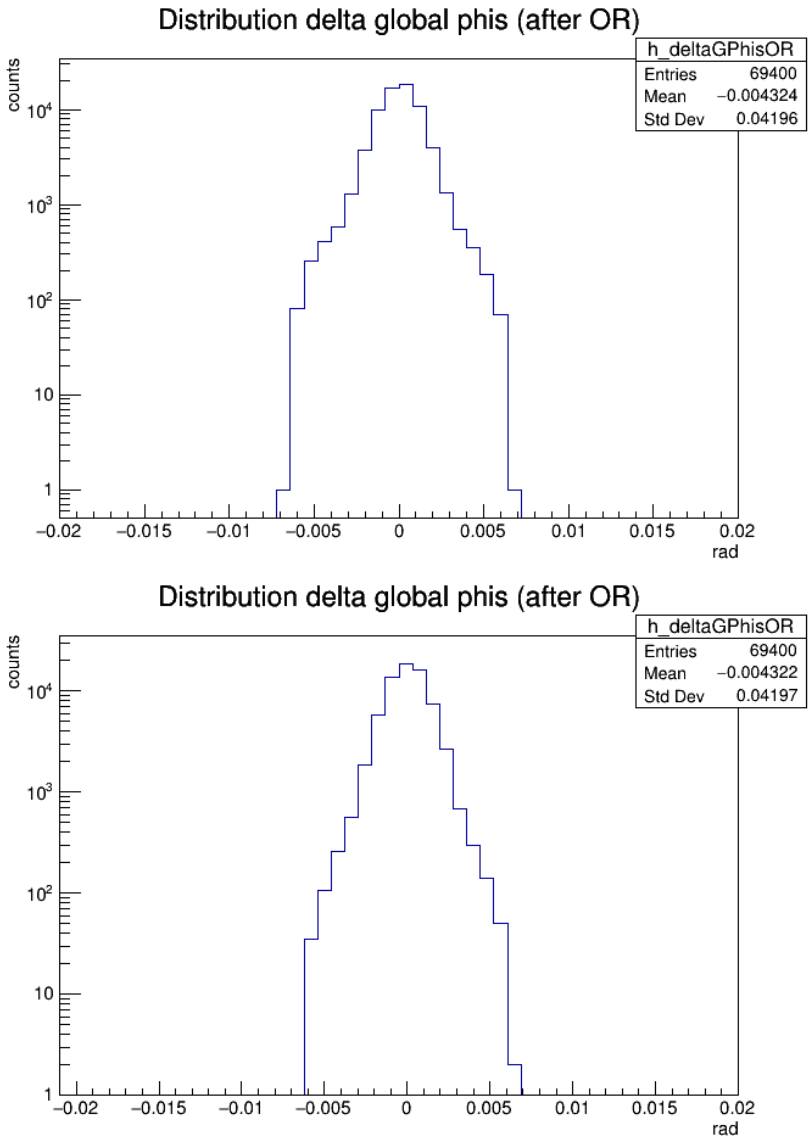


Figure 5.13: Distributions of the delta global phis using only the first order (top) and the second order (bottom) of the idealized detector geometry equations.

For curiosity, the same metrics as previously presented were studied using the second order of the equations and surprisingly, it obtained much better results than the case of only the first order. This can be seen in Figure 5.14 and the resolutions are $rms_{95}(\phi_0) = 0.0035$ rad and $rms_{95}(q/p_T) = 0.0101$ GeV^{-1}

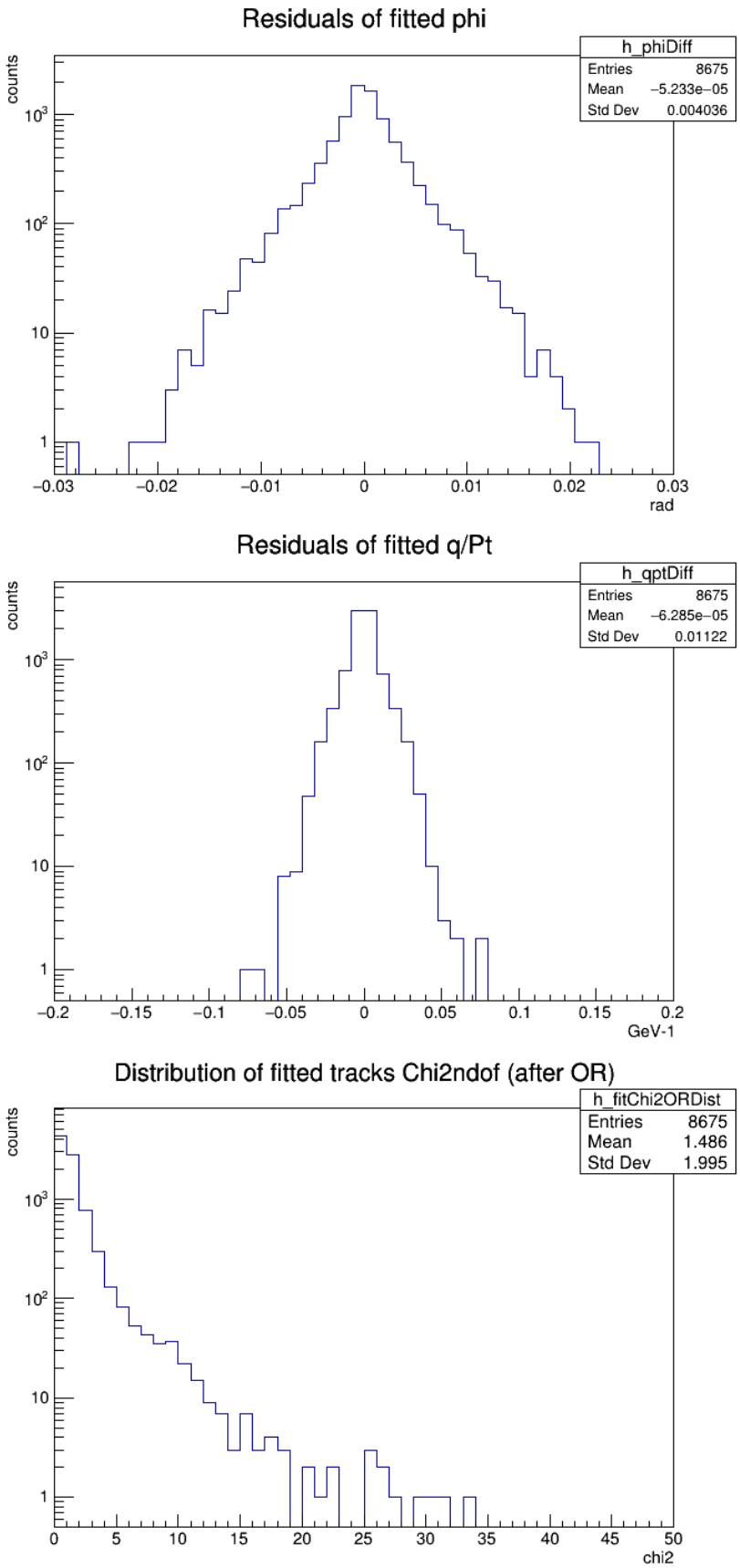


Figure 5.14: Performance of the method on single muons with $p_T > 1$ GeV, using the idealized detector geometry equations with the second order.

And in Figures 5.15, 5.16 and 5.17, the performance as function of the truth track parameters is presented.

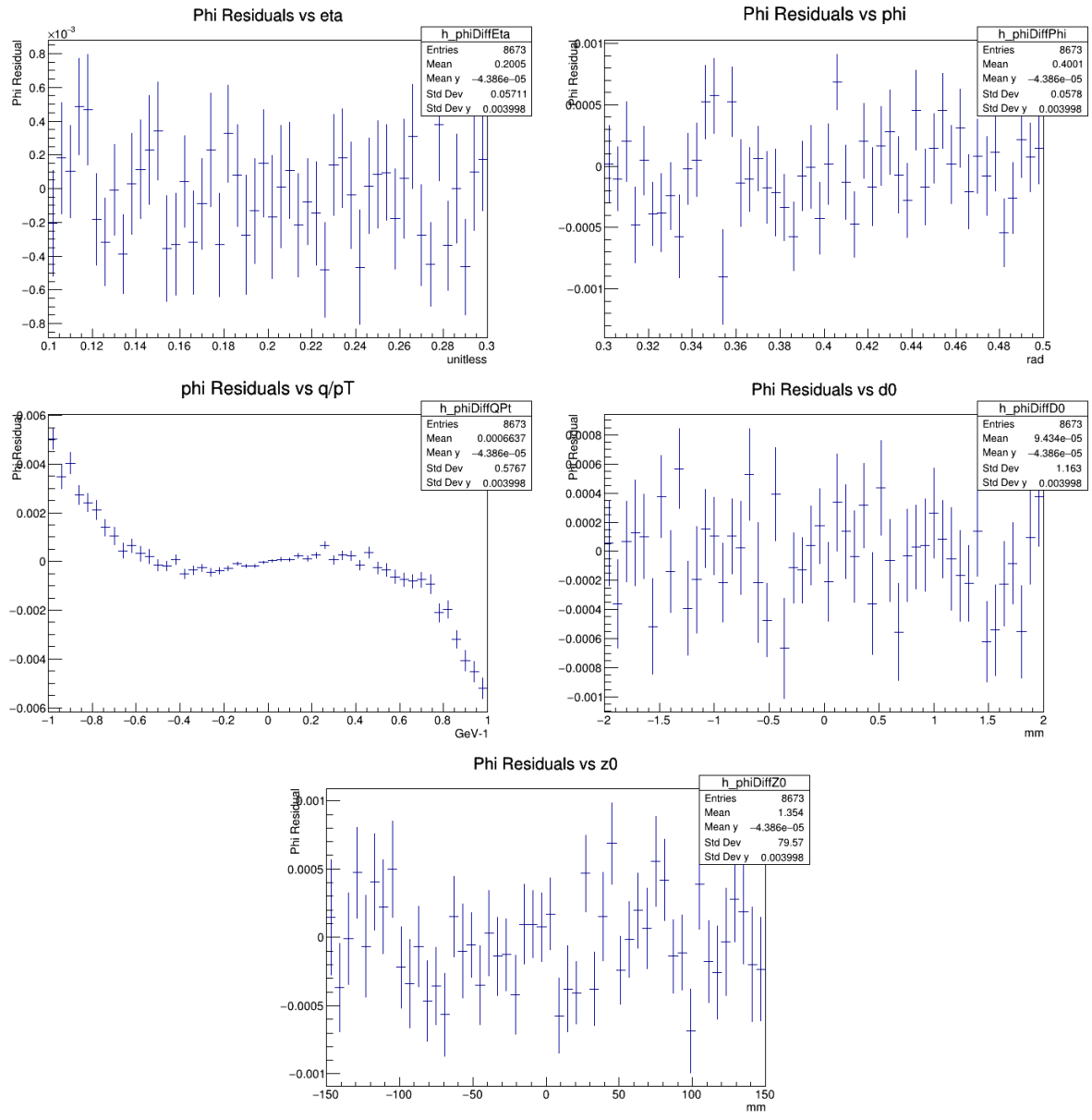


Figure 5.15: Distributions of the residuals of ϕ_0 as function of the truth track parameters. Idealized geometry equations with second order.

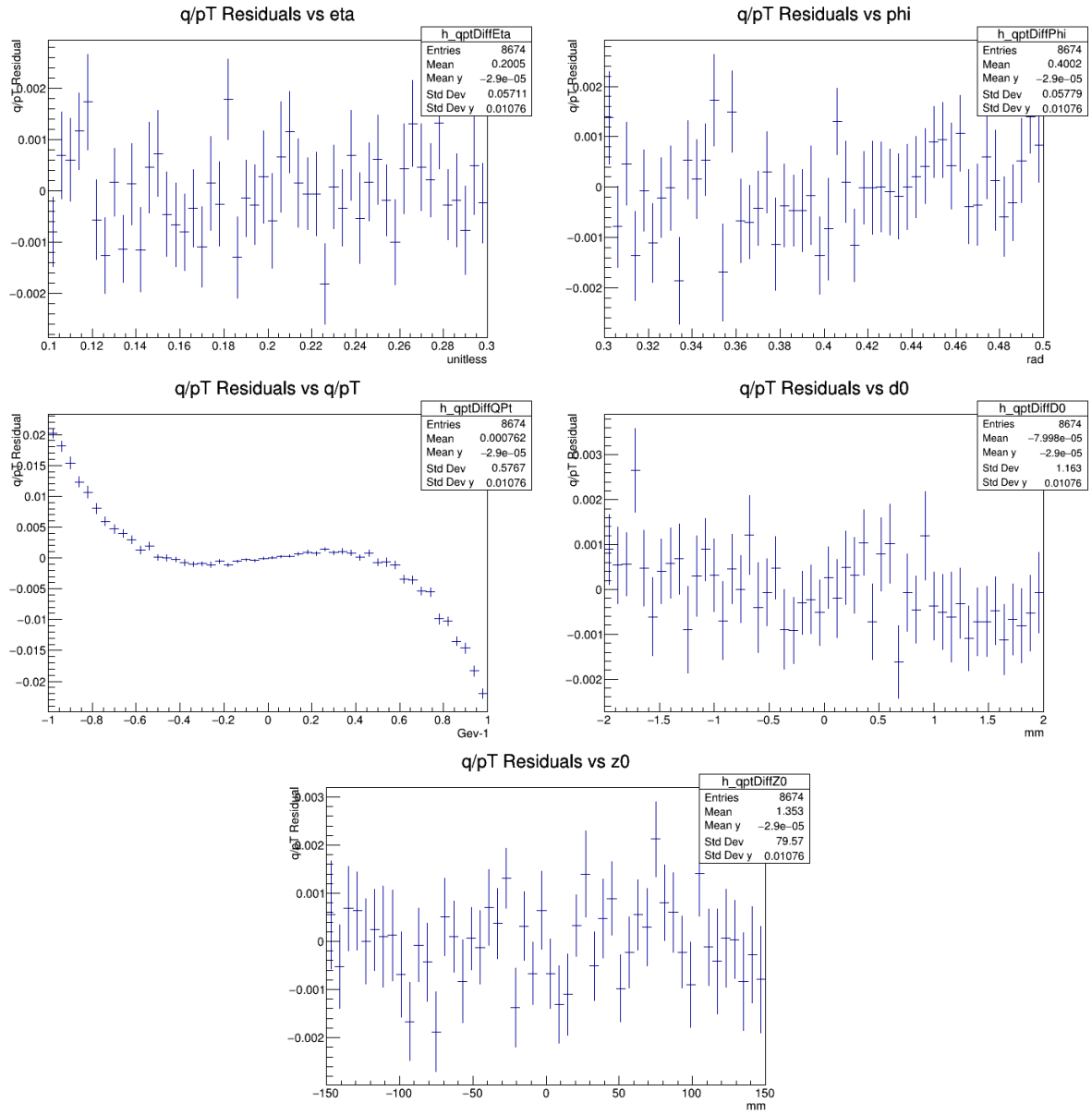


Figure 5.16: Distributions of the residuals of q/p_T as function of the truth track parameters. Idealized geometry equations wit second order.

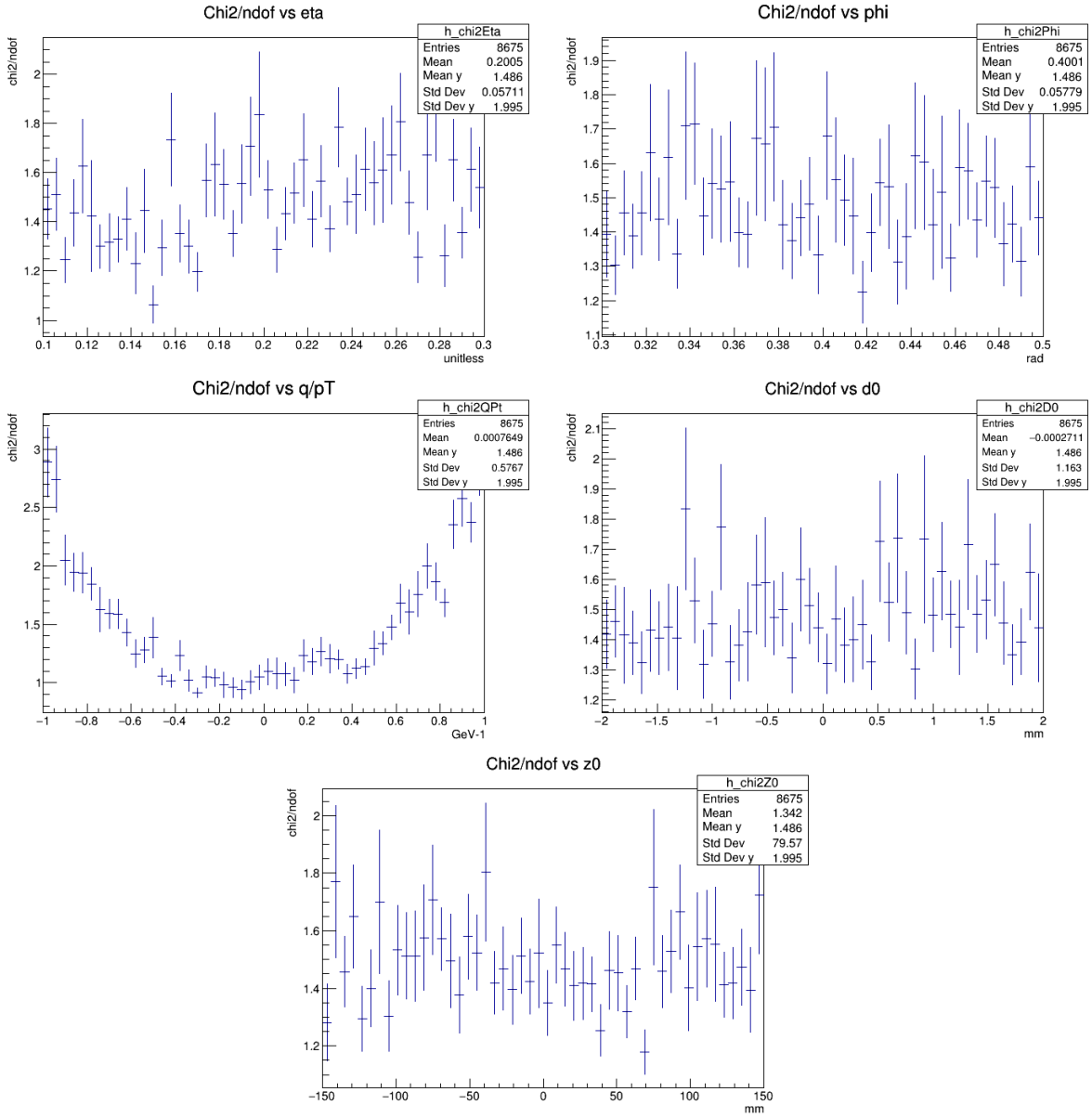


Figure 5.17: Distributions of the normalized χ^2 as function of the truth track parameters. Idealized geometry equations with second order.

Again, the performance is worse on tracks with $1 < p_T < 2$ GeV, so the simulation was run on muons with $p_T > 2$ GeV. The results are presented in Figure 5.18 and the resolutions are $rms_{95}(\phi_0) = 0.0018$ rad and $rms_{95}(q/p_T) = 0.0051$ GeV^{-1}

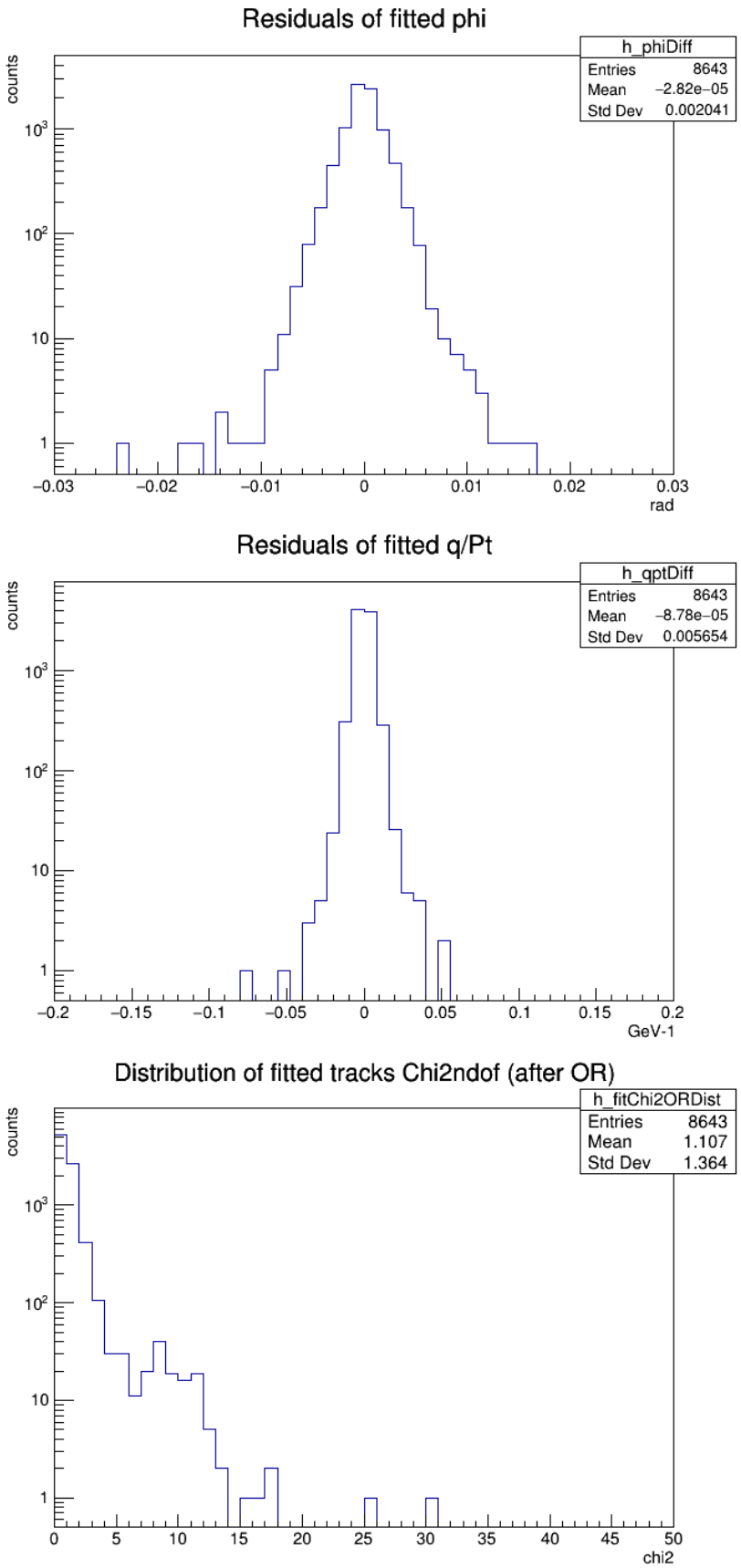


Figure 5.18: Performance of the method on single muons with $p_T > 2$ GeV, using the idealized detector geometry equations with the second order.

The resolutions for the various samples with different p_T settings are presented in Table 5.1, to serve as comparison with Table 3.8.

Table 5.1: Resolutions of ϕ_0 and q/p_T on single muons samples for various p_T conditions. "Ideal 1" and "Ideal 2" refer to, respectively, using only the first order or also the second order of the equations 21 and 22.

p_T		hit transformation equations order	rms_{95}	
			ϕ_0 [rad]	q/p_T [GeV^{-1}]
q/p_T flat	$p_T > 1$ GeV	first	0.0053	0.0221
		second	0.0035	0.0101
	$p_T > 2$ GeV	first	0.0020	0.0064
		second	0.0018	0.0051
p_T fixed	$p_T = 1$ GeV	first	0.0140	0.0592
		second	0.0075	0.0277
	$p_T = 2$ GeV	first	0.0036	0.0121
		second	0.0029	0.0068
	$p_T = 10$ GeV	first	0.0010	0.0046
		second	0.0010	0.0046

As explained in section 5.3, the hit transformations to the ideal detector consider a track with $d_0 = 0$, and the studies shown so far used simulated tracks with $|d_0| < 2$ mm (as were used for the constant generation). To test if the performance was better on tracks with lower impact parameter (d_0), the simulation was run on the same single muons sample, except the absolute value of d_0 was limited to 0.2 mm ($|d_0| < 0.2$ mm). The sectors and constants used were the same as up until here. The results obtain are presented in Figures 5.19 and 5.20.

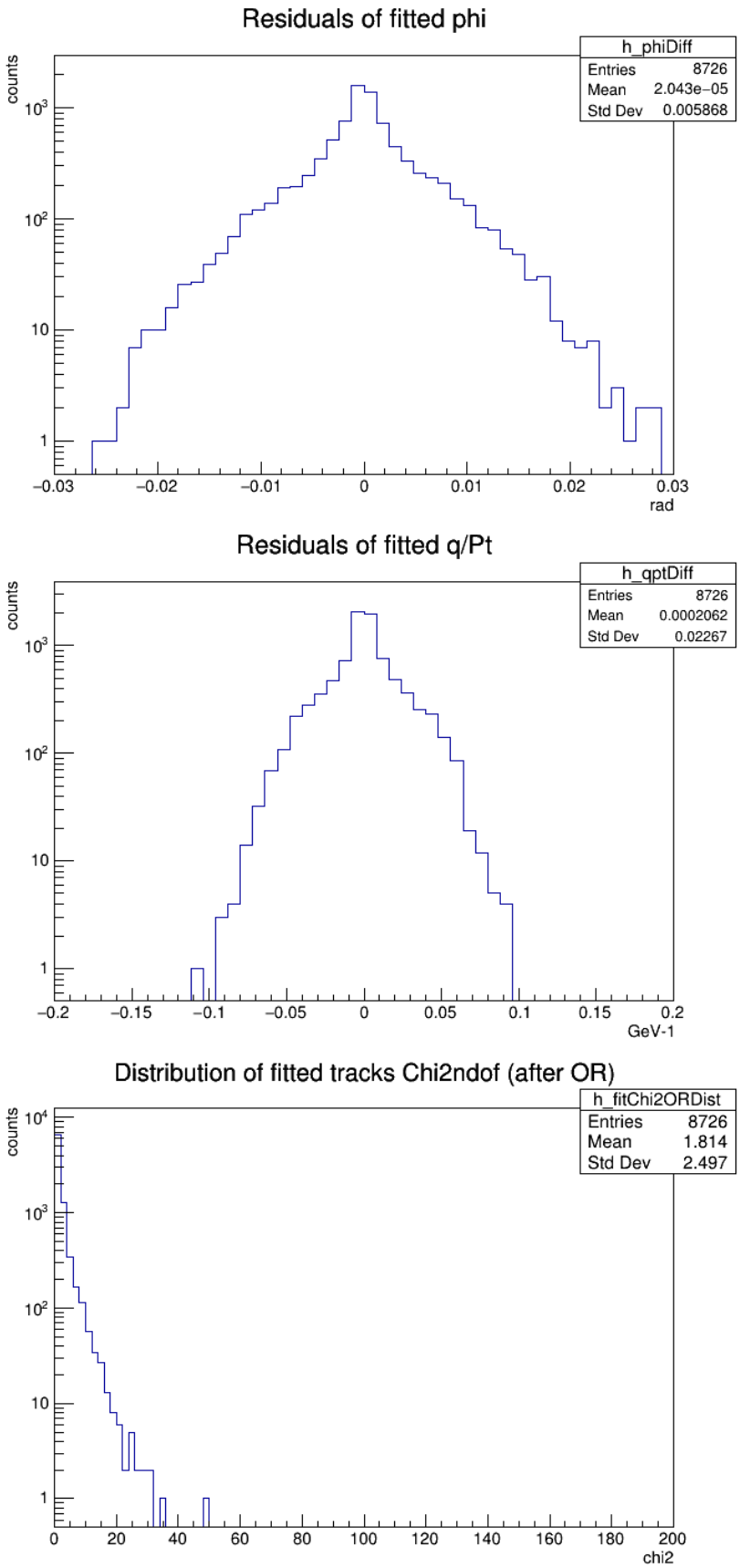


Figure 5.19: Performance of the method on single muons with $p_T > 1$ GeV and $|d_0| < 0.2$ mm, using the idealized detector geometry equations with the first order.

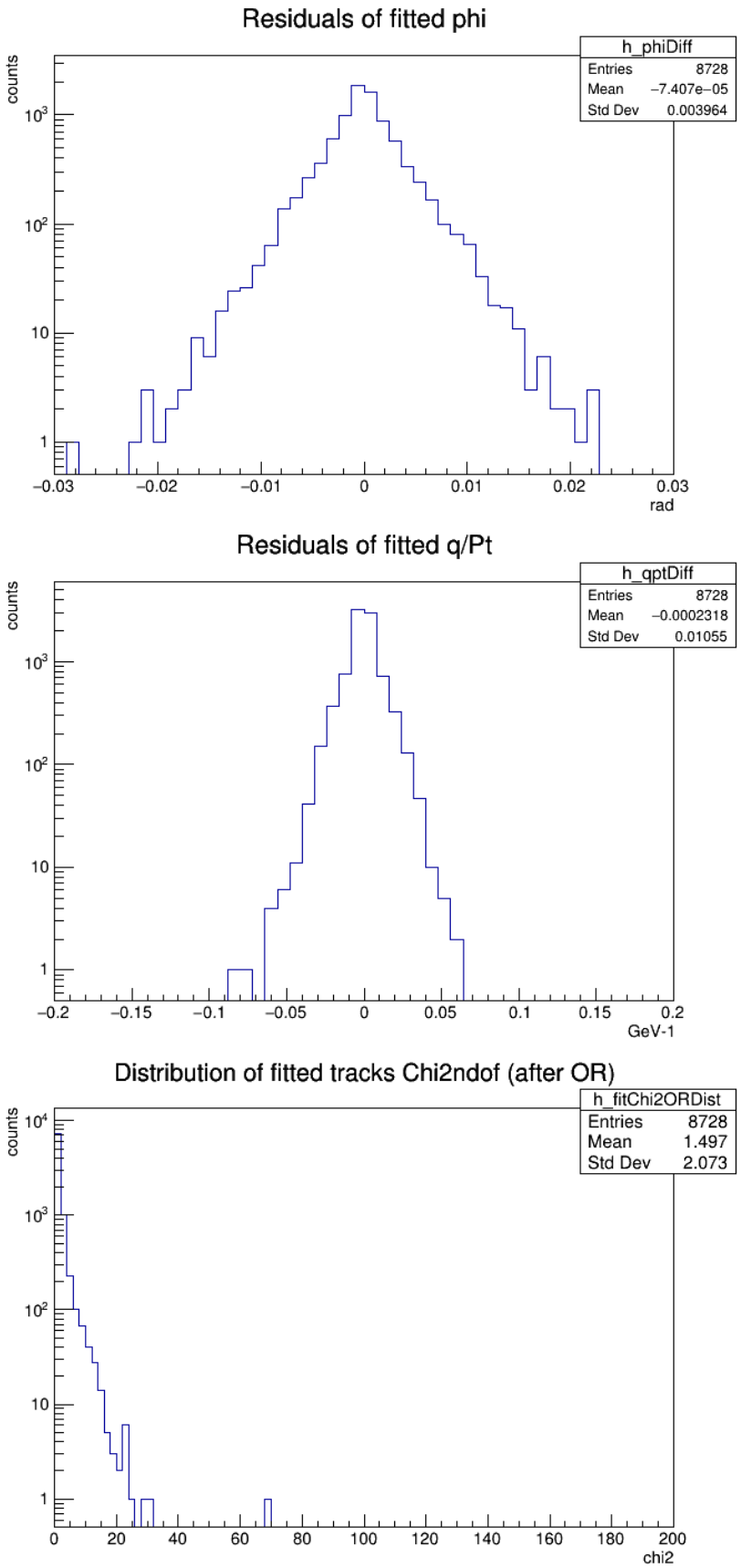


Figure 5.20: Performance of the method on single muons with $p_T > 1$ GeV and $|d_0| < 0.2$ mm, using the idealized detector geometry equations with the second order.

There is only a slight decrease in the average of the $\chi^2/ndof$ distributions and the parameter resolutions are the same.

But maybe the constants could be affecting the performance in this case, because they were obtained for tracks with $|d_0| < 2$ mm. So a new set of sectors and constants was produced with $|d_0| < 0.2$ mm. The distribution of the $\chi^2/ndof$ obtained for muons with $|d_0| < 0.2$ mm using these new constants are shown in Figure 5.21

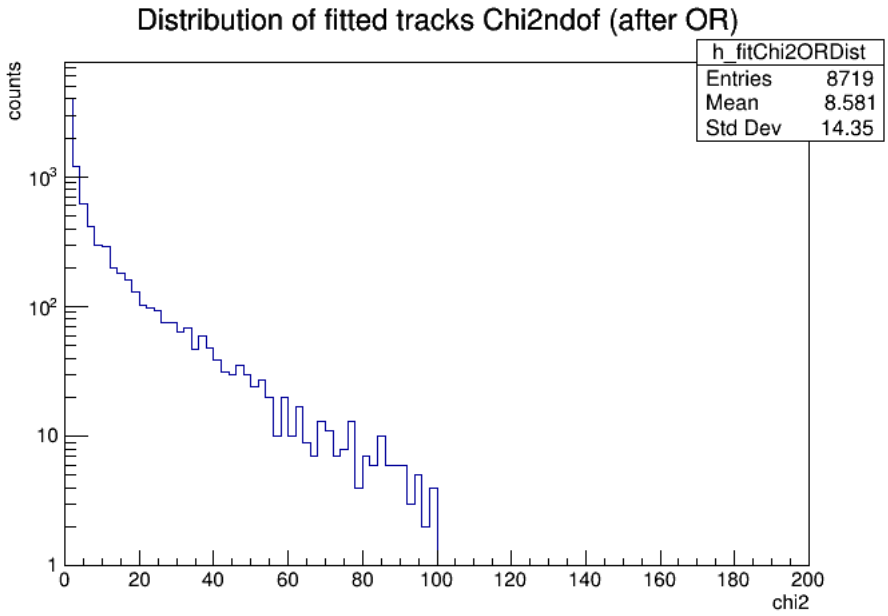


Figure 5.21: Distribution of the $\chi^2/ndof$ of tracks reconstructed for single muons with $|d_0| < 0.2$ mm. The constants used were obtained with muons with $|d_0| < 0.2$ mm.

With these constants and conditions, the method reconstructs tracks with very bad quality, which means using constants averaged for a larger interval in d_0 is better.

Muon + 200 Pile-up Events

This sample contains a muon that is inside the same region boundaries as presented in section 4.2, added to a background of 200 pile-up events. Only the muon has a corresponding truth track, the rest of the information being just the hits in the detector.

This increases the difficulty of tracking, since the higher number of hits increases the probability of reconstructing a fake track.

So the metric being studied here is the tracking efficiency, which was obtained for different values of the cut in $\chi^2/ndof$ used in the Overlap Removal. The results obtained are shown in Figure 5.22.

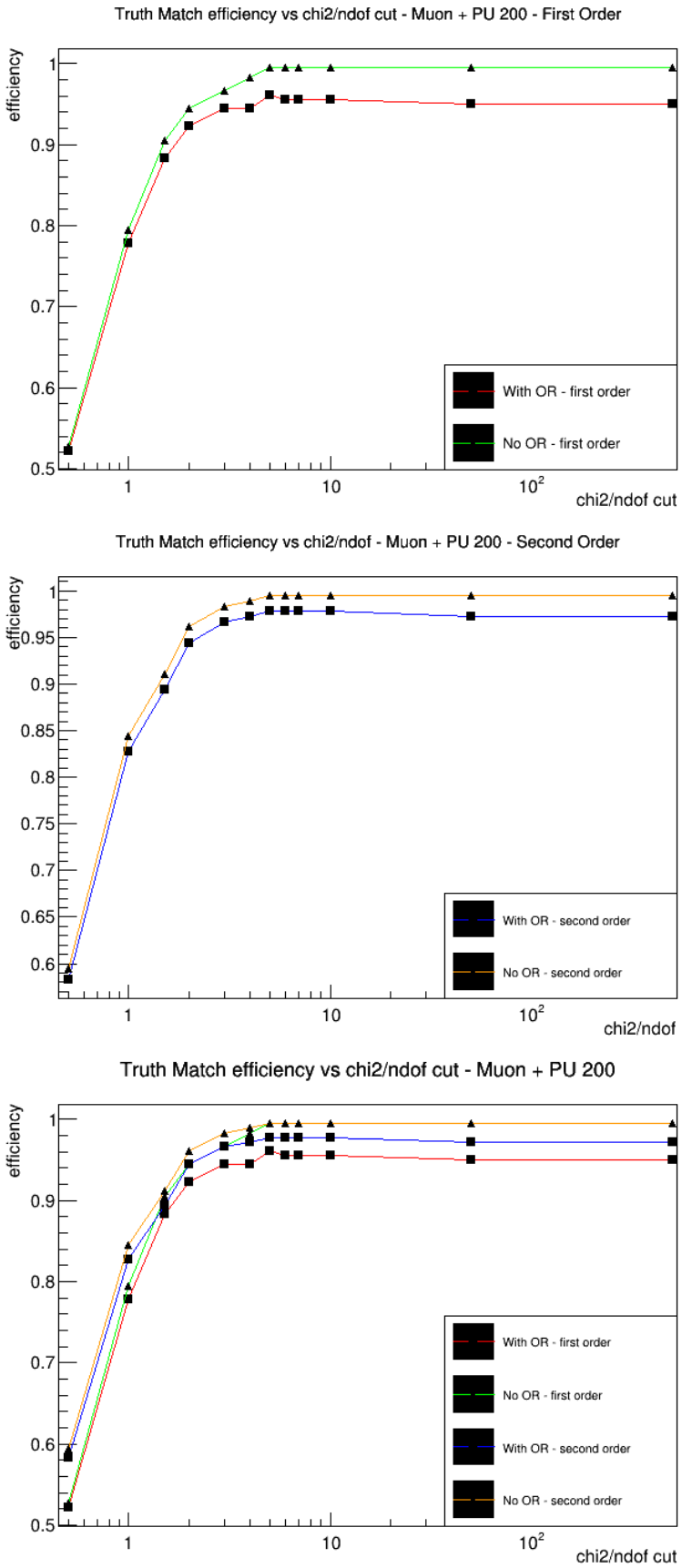


Figure 5.22: Tracking efficiency vs $\chi^2/ndof$ cut for muon embedded in 200 pile-up events. "No OR" means that the track selection was done only with the cut on $\chi^2/ndof$ and without overlap removal. "With OR" means the overlap removal was performed in the standard way.

This makes it seem like tracking without the OR is better, but doing so creates a much higher number fitted tracks that are matched to truth, as can be seen in Figure 5.23. This figure implies that the OR step reduces the number of fake tracks by a large factor of the order of 10-100.

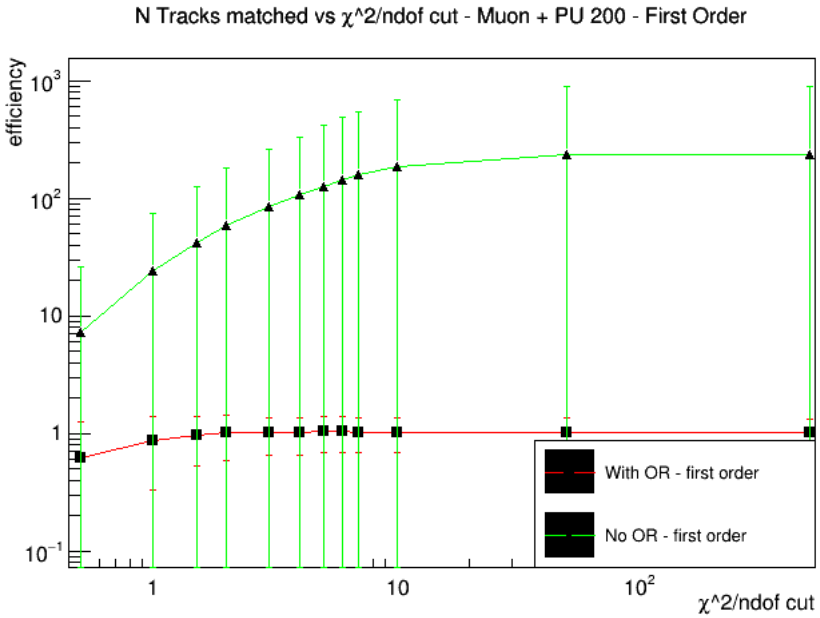


Figure 5.23: Number of tracks matched to the truth muon after the OR or just the $\chi^2/ndof$ cut.

$t\bar{t}$ + 200 Pile-up Events

The $t\bar{t}$ sample was generated in the entire ITk coverage ($|\eta| < 4$). This means that in the region where the method was applied ($0.1 < \eta < 0.3$) there were only a few tracks, or none at all. For these results, only events with at least one truth and one offline track were considered.

Figures 5.24 and 5.25 show the number of tracks per event at the different steps of the simulation. "1st stage" refers to the number of tracks before any OR, "1st stage w/ χ^2" means that only a cut on the $\chi^2/ndof$ was applied and "1st stage w/ OR" are the number of tracks after the normal OR. On average there are 2.057 truth tracks, 2.103 offline tracks, 6785 "1st stage" tracks, 3047 "1st stage w/ χ^2" tracks and 84.4 "1st stage w/ OR" tracks. This method produces about 45 more tracks that offline reconstruction, which is much larger than the baseline system.

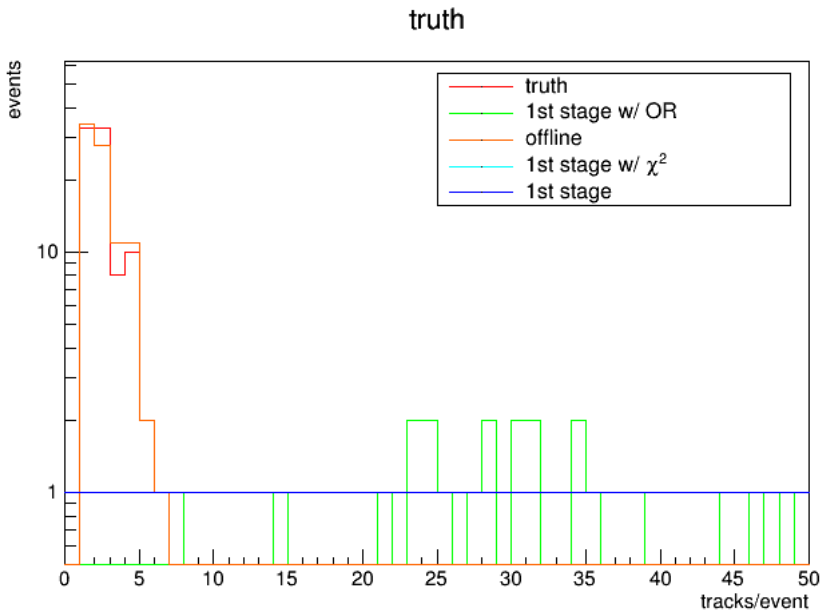


Figure 5.24: Number of tracks in the $0.1 < \eta < 0.3$ region for the $t\bar{t}$ sample. Hit transformation equations with first order.

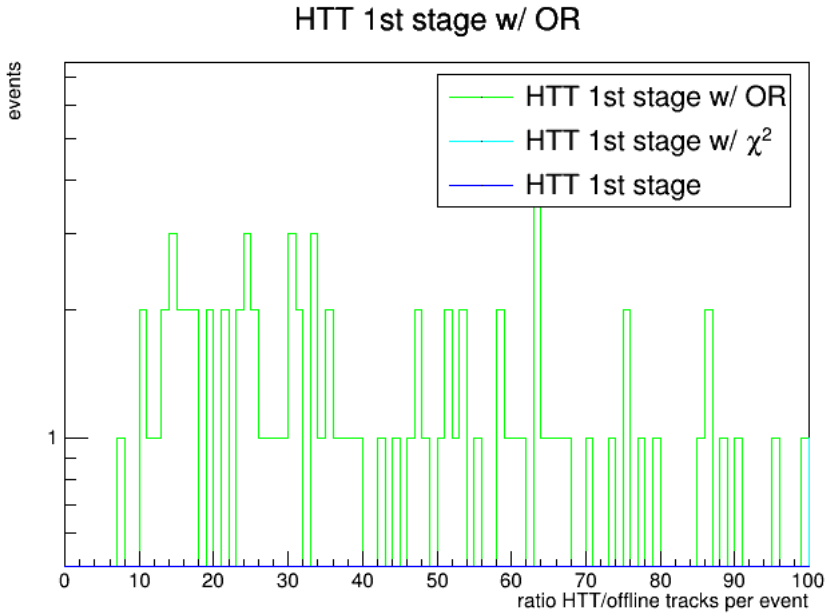


Figure 5.25: Ratio of HTT tracks to offline tracks in the $0.1 < \eta < 0.3$ region for the $t\bar{t}$ sample. Hit transformation equations with first order.

If the hit transformation equations are used with the second order, the $\chi^2/ndof$ cut can be lowered to 5. This reduces the number of tracks in all steps, to about 36 times more than offline after OR. This can be seen in Figures 5.26 and 5.27.

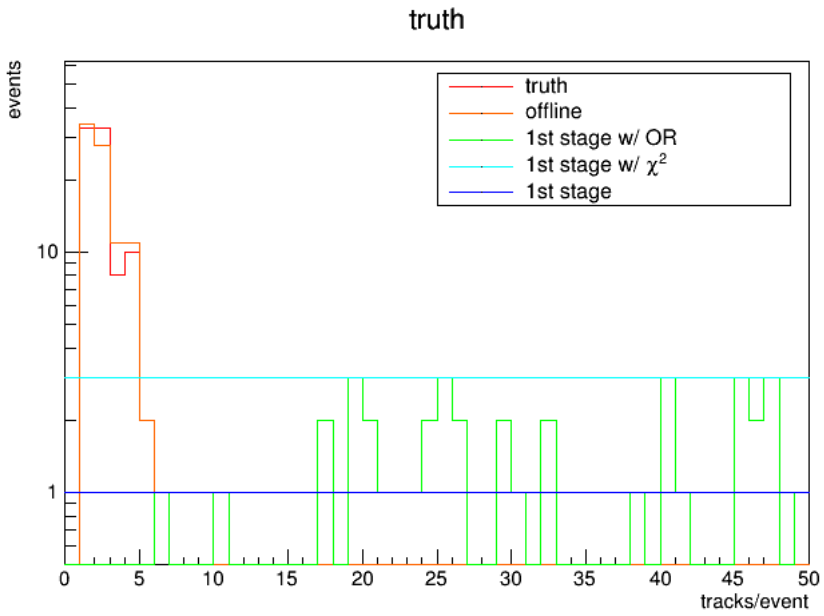


Figure 5.26: Number of tracks in the $0.1 < \eta < 0.3$ region for the $t\bar{t}$ sample. Hit transformation equations with second order.

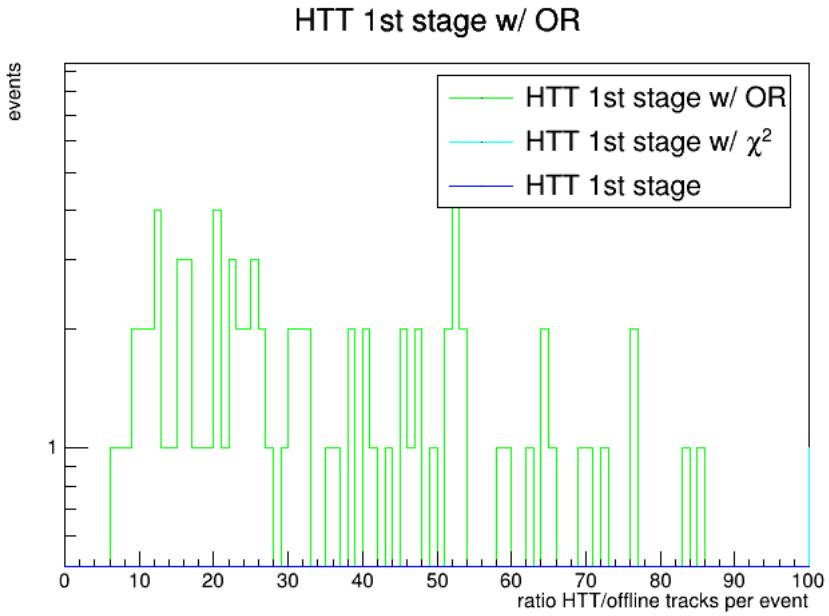


Figure 5.27: Ratio of HTT tracks to offline tracks in the $0.1 < \eta < 0.3$ region for the $t\bar{t}$ sample. Hit transformation equations with second order.

Chapter 6 Conclusions

The new tracking method presented in this thesis was successfully implemented in the HTT simulation. Its performance was studied on single muon events in a track parameter space region defined in section 4.2. The muon track finding efficiency is around 98.4%. This is because the Hough Transform algorithm was configured with a threshold of at least 7 out of 8 layers hit, and some of the events only had 6, 5 or event 4 layers hit. Excluding these events, the track finding efficiency achieved is about 99.5%. The first stage tracking (i.e. $p_T > 2$ GeV) resolutions are $rms_{95}(\phi_0) = 0.0020$ rad and $rms_{95}(q/p_T) = 0.0064$ GeV^{-1}. If the hit transformation equations (21 and 22) are used with the second order approximation, resolutions of 0.0018 rad and 0.0051 GeV^{-1} can be achieved. Including tracks with $1 < p_T < 2$ GeV (to compare with Table 3.8), the resolutions obtained are $rms_{95}(\phi_0) = 0.0053$ rad and $rms_{95}(q/p_T) = 0.0221$ GeV^{-1} using only the first order of the hit transformation equations, and $rms_{95}(\phi_0) = 0.0035$ rad and $rms_{95}(q/p_T) = 0.0101$ GeV^{-1}, if the second order is used. For first stage tracking these are good resolutions, but if low p_T tracks are included, the resolutions are about 1.5 times worse than the baseline system. This means that the method performs worse on low p_T tracks than the baseline system.

The tracking efficiency in muons embedded in 200 pile-up events is 96%, which is about 3% lower than the baseline system. However, an efficiency of 97.8% can be achieved if the hit transformation equations are used with the second order. There isn't a value for the baseline tracking efficiency in the entire p_T range, but comparing with Table 3.7, this method achieves an efficiency close to the average of the "HTT/truth" for the first region.

For the estimation of fake tracks, the number or fitted tracks in the $0.1 < \eta < 0.3$ region was studied. This method produces about 45 (36) times more tracks (after Overlap Removal) than offline reconstruction, using equations 21 and 22 with the first (second) order. According to HTTSim experts, this is an acceptable value.

In conclusion, this method shows promising results as a possibility to assist the Event Filter with tracking operations. The method needs further development, such as a procedure to estimate the d_0, z_0 and η of tracks. Possibilities for this include using the z coordinate of the pixel layer or a Hough Transform based on the $d0$ and ϕ_0 of tracks.

Possible future directions include the aforementioned development of the method, its implementation in FPGAs and interfacing with the CPUs of the Event Filter. Also the extension of the method to the entire detector coverage, optimization of the tools and equations it uses and further studies on its overall performance.

References

[1] Lyndon R Evans and Philip Bryant. *LHC Machine*. In: *JINST* 3.S08001 (2008). This report is an abridged version of the LHC Design Report (CERN-2004-003). URL: https://cds.cern.ch/record/1129806.

[2] ATLAS Collaboration. *Pileup Interactions and Data Taking Efficiency*. URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2#Pileup_Interactions_and_Data_Tak. (accessed: 29/10/2021).

[3] ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider*. In: *JINST* 3.S08003 (2008). Also published by CERN Geneva in 2010. URL: https://cds.cern.ch/record/1129811.

[4] I. Béjar Alonso et al. (Eds.) *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report*. CERN Yellow Reports: Monographs. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2749422.

[5] ATLAS Collaboration. *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*. CERN-LHCC-2017-020, ATLAS-TDR-029. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2285584.

[6] ATLAS Collaboration. *ATLAS Computing: Technical Design Report*. CERN-LHCC-2005-022, ATLAS-TDR-017. Geneva: CERN, 2005. URL: https://cds.cern.ch/record/837738.

[7] JabberWok. *Pseudorapidity*. URL: https://commons.wikimedia.org/w/index.php?curid=5970403. (accessed: 29/10/2021).

[8] ATLAS Collaboration. *Keeping the ATLAS Inner Detector in perfect alignment*. URL: https://atlas.cern/updates/experiment-briefing/inner-detector-alignment. (accessed: 29/10/2021).

[9] A. R. Martínez on behalf of the ATLAS Collaboration. *The Run-2 ATLAS Trigger System*. In: *J. Phys.: Conf. Ser.* 762.012003 (2016). URL: https://cds.cern.ch/record/2133909/.

[10] ATLAS Collaboration. *Level-1 Trigger: Technical Design Report*. CERN-LHCC-98-014, ATLAS-TDR-12. Geneva: CERN, 1998. URL: http://cds.cern.ch/record/381429.

[11] ATLAS Collaboration. *Performance of the ATLAS Trigger System in 2015*. In: *Eur. Phys. J. C* 77.317 (2017). DOI: 10.1140/epjc/s10052-017-4852-3.

[12] ATLAS Collaboration. *High-Level Trigger, Data-Acquisition and Controls: Technical Design Report*. CERN-LHCC-2003-022, ATLAS-TDR-16. Geneva: CERN, 2003. URL: http://cds.cern.ch/record/616089.

[13] ATLAS Collaboration. *Fast TracKer (FTK) Technical Design Report*. CERN-LHCC-2013-007, ATLAS-TDR-021. Geneva: CERN, 2013. URL: https://cds.cern.ch/record/1552953.

[14] ATLAS Collaboration. *The ATLAS Simulation Infrastructure*. In: *EPJ C* (2010). DOI: 10.1140/epjc/s10052-010-1429-9.

[15] CERN. *The HL-LHC project*. URL: https://hilumilhc.web.cern.ch/content/hl-lhc-project. (accessed: 29/10/2021).

[16] ATLAS Collaboration. *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*. CERN-LHCC-2017-021, ATLAS-TDR-030. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2285585.

[17] ATLAS Collaboration. *Technical Design Report for the ATLAS Inner Tracker Strip Detector*. CERN-LHCC-2017-005, ATLAS-TDR-025. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2257755.

[18] ATLAS Collaboration. *Expected Tracking Performance of the ATLAS Inner Tracker at the HL-LHC*. In: ATL-PHYS-PUB-2019-014 (2019). URL: https://cds.cern.ch/record/2669540.

[19] ATLAS Collaboration. *Report of the Task Force Studying a Custom-based Implementation*. Internal report AT2-D-RG-0005 v.1. Reproduced under Creative Commons licence CC-BY-4.0, © 2021 CERN for the benefit of the ATLAS Collaboration. URL: https://edms.cern.ch/document/2591107/1 (private link).

[20] B. Hooberman on behalf of the ATLAS Collaboration. *First tracking performance results from the ATLAS Fast TracKer*. In: *Proceedings of the CTD/WIT,* ATL-DAQ-PROC-2019-010. (2019). URL: https://cds.cern.ch/record/2677874/.

[21] A. Ryd and L. Skinnari. *Tracking Triggers for the HL-LHC*. In: *Annu. Rev. Nucl. Part. Sci.* 70 (2020). DOI: https://doi.org/10.1146/annurev-nucl-020420-093547.

[22] T. James. *A hardware track-trigger for CMS at the High Luminosity LHC*. PhD thesis. Imperial College London, 2018. URL: https://cds.cern.ch/record/2647214.

[23] G. Barrand. *GAUDI - A software architecture and framework for building HEP data processing applications*. In: *Comput. Phys. Comm.* (2001). DOI: https://doi.org/10.1016/S0010-4655(01)00254-5.

[24] R. Brun and F. Rademakers. *ROOT - An Object Oriented Data Analysis Framework*. In: *Proceedings AIHENP'96 Workshop* (Lausanne, 1996). URL: https://root.cern.ch/.

[25] P. Richardson F. Krauss S. Navas and T. Sjöstrand. *Monte Carlo Particle Numbering Scheme*. In: *Particle Data Group* (2019). URL: https://pdg.lbl.gov/2020/reviews/rpp2020-rev-monte-carlo-numbering.pdf.

[26] M. Mårtensson. *A search for leptoquarks with the ATLAS detector and hardware tracking at the High-Luminosity LHC*. PhD thesis. Uppsala University, 2019. URL: http://uu.diva-portal.org/smash/get/diva2:1341509/FULLTEXT01.pdf.

[27] E. Clement et al. *A high-performance track fitter for use in ultra-fast electronics*. In: *Nucl. Instr. Meth. Phys. A* 935 (2019). URL: http://dx.doi.org/10.1016/j.nima.2019.05.018.