Track Quality Monitoring for The Compact Muon Solenoid Silicon Strip Tracker

A thesis submitted for the degree of
Doctor of Philosophy

by

Israel Goitom

School of Engineering and Design
Brunel University

February 7, 2009

Abstract

The CMS Tracker is an all silicon detector and it is the biggest of its kind to be built. The system consists of over 15,000 individual detector modules giving rise to readout through almost 10^7 channels. The data generated by the Tracker system is close to 650 MB at 40 MHz. This has created a challenge for the CMS collaborators in terms of data storage for analysis. To store only the interesting physics data the readout rate has to be reduced to 100 Hz where the data has to be filtered through a monitoring system for quality checks. The Tracker being the closest part of the detector to the interaction point of the CMS creates yet another challenge that needs the data quality monitoring system. As it operates in a very hostile environment the silicon detectors used to detect the particles will be degraded. It is very important to monitor the changes in the sensor behaviour with time so that to calibrate the sensors to compensate for the erroneous readings. This thesis discusses the development of a monitoring system that will enable the checking of data generated by the tracker to address the issues discussed above. The system has two parts, one dealing with the data used to monitor the Tracker and a second one that deals with statistical methods used to check the quality of the data.

Declaration

The work on this thesis, specifically Chapter 4 and 6 are entirely the result of my work while I did my research at CERN and Brunel University. The research on chapter 6 is based on a new statistical model given by [1]. The work on Chapter 5 is the result of my work during the Magnet Test and Cosmic Challenge, where I participated in the data-taking shifts and data quality monitoring process.

Israel Goitom

Acknowledgements

This thesis would not have been possible without the support and help of the following people, and I would like to thank them accordingly.

At Brunel university, my supervisor Peter Hobson for his time to provide me the invaluable guidance on my research and introducing me to particle physics, Steve Watts for the invaluable lessons on silicon detectors, Ivan Reid for the lessons he gave me on ROOT and ORCA to get me started on my work and the help he provided throughout my research and Raul Lopes for inspiring me to study statistical methods that has become the favourite part of my research. The Science and Technology Facilities Council has provided me a three year funding to do this research which I am grateful for.

The work of this thesis was mainly done at CERN as part of the Tracker group. I am grateful for the support I received from the Tracker group in general and especially from Giacomo Bruno, Dorian Kcira and Suchandra Duta for enabling the Tracker_Monitor_Track package to be part of the CMSSW framework. I would like also to thank Ian Tomalin from RAL for his advice in to the development of this package.

Other people who have shed their support outside my academic work are my big family, Elen, Elias, Yordanos and Gideon for believing I had it in me to finish the work and Dad and Mom for being there when I needed them most. Yohannes, Milkiyas and Nathan for their support to get along with my stress. At Brunel Craig, Dan, Matt, Richard, Neil and Tom have been good sport making the research area feel welcoming and listening to the endless stress chats. My time at CERN would not have been as enjoyable with out the 'cernies' and especially Andy, Alan, Dave and the usual suspects of L'Usine. And Nick for

introducing me to Alpine Mountaineering which were helpful during stressful times.

Finally I would like to thank my sister Eden for always believing that I would be able to achieve what I set my mind to and giving me the needed push and endless support during the lowest moments of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Questions such as "what is matter made of" and "what is holding it all together" have been in the mind of humans from as far back as history can take us. The same questions drive scientists in the particle physics community to improve their experimental techniques and theories to better understand matter and its constituents. One such place where scientists from all over the world are working together to find solutions is CERN; the name comes from a French acronym for European Council for Nuclear Research. Although the name has now changed to European Organisation for Particle Physics Research the name CERN still is used as the symbol. CERN is located in the Franco-Swiss border near the Jura Mountain range. It has an underground tunnel where the Large Hadron Collider particle accelerator is housed. The tunnel which is toroidal in shape is around 100 meters underground and 27 km in circumference. This accelerator is used to give particles such as protons (p) and lead (Pb) high energy that gets up 7 TeV and collide them head on at various locations where particle detectors are placed to observe the resulting new particles. There are four major such detectors placed around the LHC tunnel, namely: CMS, ATLAS, LHCB and ALICE. Detectors at CERN are designed to help scientists study particles which are smaller than the size of an atom by magnitudes of around 10^{-8}. One such detector is the Compact Muon Solenoid (CMS), which will be discussed in detail in Chapter 2. Particle Physicists at the moment are studying matter and its constituents to test a theory known as the Standard Model (SM). This theory is briefly discussed in this chapter along with introduction to particle physics. Following this, the chapter will introduce the need for the Data Quality Monitoring targeted at tracks and why such a system needs to be built. Finally an outline of the thesis is given.

1.1 Scope of Thesis

This thesis covers two subject areas. First it discusses the CMS detector used for particle physics research. The main goal of this first section is to give an overview of how the CMS detector, especially the silicon detector, works and the physics studied with this detector. Due to the high complexity of the CMS detector it is not possible to give a full explanation of the details of how the CMS detector works but a detailed description of the Silicon Strip Tracker will be given as a data quality monitoring software was developed for it. The software used for the Silicon Strip Tracker data quality monitoring will be discussed in full detail.

The aim of the second section is to introduce some of the statistical methods used in high energy physics experiments (eg. the χ^2 test and the Kolmogorov-Smirnov test) and how these can be used for data quality monitoring. This section will also introduce a recent statistical test method known as the Energy Test and its implementation for use in monitoring data from the CMS experiment.

1.2 Contribution to Knowledge

The research discussed in this thesis has lead to the introduction of a recently developed statistical test for use in high energy physics experiment. It is our understanding that such test has not been used before in data quality monitoring or any other data analysis of high energy physics experiments. The “Energy Test” has been shown to be reliable and the software that implements it is to be integrated into the next major release of the CMS software.

The research has also unveiled some issues with the statistical test methods used at the moment, which depend on binned data (eg. χ^2). It has been shown through the tests conducted during the research that using unbinned data yields better results. This means that anomalies in the Silicon Strip Detector data arising from detector problems can be detected with less data samples (in the early stages of data taking) in comparison to the data needed by statistical tests that use binned data.

1.3 The Standard Model

"... by convention hot,
by convention cold,
by convention colour;
but in reality atoms and void"
— Democritus (400 BCE)

The standard model is the best physics theory at present which explains the universe - the fundamental building blocks (particles) of matter and how it is held together through the interaction of these particles. The word fundamental implies something that does not have any internal structure and therefore cannot be divided in to anything smaller. Experiments of particle physics have lead to the discovery of around 200 particles most of which are not fundamental. According to the standard model theory all the non fundamental particles are made up from a combination of 6 quarks or are one of the 6 leptons which are considered as fundamental particles. Each particle has its counterpart known as antiparticle. Antiparticles have identical mass to their counterpart particles but have opposite charge and quantum number. The existence of matter and lack of antimatter in our universe is one of the questions the standard model has not been able to explain and is one of the questions scientists at CERN are tying to understand.

Quarks and their properties are listed in Table 1.1. Quarks have fractional charges. Up, Charm and Top have positive charge of $\frac{2}{3}$ where as Down, Strange and Bottom have $-\frac{1}{3}$. Quarks can only exist in groups with other quarks forming composite particles known as hadrons. Depending on the organisation and number of quarks two types of hadrons exist; they are Baryon and Mesons. Baryons are made up of three quarks and Mesons are made up of a quark and an antiquark. Protons and neutrons are baryons. Protons have two up and one down quarks making it positively charged and neutron has one up and two down making it neutral. The π^+ is an example of a Meson made up of an Up quark and a Down antiquark. Only the up and down quarks are noticed on ordinary matter as they are the stable ones. The other quarks have very short life and decay in to other particles.

Particle name	Particle symbol	Spin($\hbar$)	Charge(e)	antiparticle	mass(MeV/c^2)
Down	d	$\frac{1}{2}$	$-\frac{1}{3}$	$\bar{u}$	0.004 to 0.008
Up	u	$\frac{1}{2}$	$+\frac{2}{3}$	$\bar{c}$	0.0015 to 0.004
Strange	s	$\frac{1}{2}$	$-\frac{1}{3}$	$\bar{t}$	0.080 to 0.130
Charm	c	$\frac{1}{2}$	$+\frac{2}{3}$	$\bar{d}$	1.150 to 1.350
Bottom	b	$\frac{1}{2}$	$-\frac{1}{3}$	$\bar{s}$	4.100 to 4.400
Top	t	$\frac{1}{2}$	$+\frac{2}{3}$	$\bar{b}$	$178.1^{+10.4}_{-8.3}$

Table 1.1: Quarks and their properties [2]

Unlike quarks leptons are solitary particles. Table 1.2 gives a list of the leptons with their properties. Each lepton has its associated neutrino particle (ν). Muons (μ) and taus (τ) are very heavy leptons compared to an electron (e^-) and have very short life. Both particles are not seen in ordinary matter. They both decay into electrons and the other neutrinos which are the stable leptons. One such example is a muon decaying into muon neutrino, electron and electron antineutrino.

What holds quarks together to form hadrons, or electrons and the nuclei to form atoms and atoms which are the building blocks of molecules? According to the standard model this happens due to the interaction of the particles through the force carrier particles. A list of known and hypothesised force carrier particles is given in Table 1.3. The interaction between particles can be either attractive or repulsive force. There are four forces known to exist. They are weak, strong, electromagnetic and gravity. Each force has a carrier associated with it or is believed to have a carrier. These force carrier particles are known

Particle name	Particle symbol	Spin($\hbar$)	Charge(e)	antiparticle	mass(MeV/c^2)
electron	e^-	$\frac{1}{2}$	-1	e^+	0.511
electron neutrino	ν_e	$\frac{1}{2}$	0	$\bar{\nu}_e$	$< 3 \times 10^{-6}$
muon	μ^-	$\frac{1}{2}$	-1	μ^+	106
muon neutrino	ν_μ	$\frac{1}{2}$	0	$\bar{\nu}_\mu$	< 0.19
tau	τ^-	$\frac{1}{2}$	-1	τ^+	1780
tau neutrino	ν_τ	$\frac{1}{2}$	0	$\bar{\nu}_\tau$	< 18.2

Table 1.2: Leptons and their properties [2]

as bosons. The electromagnetic force which holds an atom together has a carrier particle know as the photon. Strong force which keeps hadrons and nucleus of an atom together is carried by gluons. The weak force which is responsible for the decay of particles has its carriers which are the $W^\pm$ and Z^0. Gravitational force which keeps us on the planet and the planets in their position in the solar system is believed to be carried by particles known as gravitons. Gravitons have not yet been discovered.

Particle name	Particle symbol	Spin($\hbar$)	Charge(e)	mass(MeV/c^2)
Photon	γ	1	0	$< 6 \times 10^{-26}$
W	$W^\pm$	1	± 1	80.425 ± 0.038
Z	Z^0	1	0	91.188 ± 0.002
Gluon	g	1	0	0(predicted)
Higgs	H^0,$H^\pm$	(0)	$(0, \pm 1)$	?
Gravitons	G	(2)	(0)	?

Table 1.3: Bosons and their properties [2]. Properties enclosed in parenthesis are not experimentally proven, and ? denotes it is not known yet.

Although it seems that standard model explains matter and its constituents, it is an incomplete theory. For example the theory can not explain why certain particles have mass and others do not. For example photons and the Z particles are both force carriers, and while photons are mass less the Z particles are comparatively very massive. The model works well as long all particles are mass less, but mass is unavoidable phenomenon of particle property. SM suggests that particles which posses mass interact with a particle known as the Higgs boson. The particles with higher masses are believed to couple with this particle strongly and the massless particles do not interact with it. The Higgs particle has not been discovered yet. The search for this particle is one of the main goals of the CMS experiment at CERN.

1.4 Motivation for the Data Quality Monitoring System

The work reported in this thesis is involved with Data Quality Monitoring (DQM) system which is closely related to the data acquisition system. The data acquisition systems of particle physics experiments such as the one used in CMS deals with huge amount of data. To quantify this, the detector has to take reading at the rate of 40 MHz from about 10^7 channels which are then read by the data acquisition system through 650 readout electronics. Each readout gives out 1 MB of data. This is such a high amount of data that it is impossible to store. To overcome this data overload, the data acquisition system has to store relatively few selected data believed to be of interest to the physics studies. The decision whether to accept or reject data representing an event (see section 2.7 for description of an event) is dealt with by computers which are part of the data acquisition system known as filter farm. These computers are responsible for reconstructing the events that happen in the detector and they accept or reject an event at the rate of 100 kHz. This reduced rate from 40 MHz to 100 kHz is achieved by the level 1 triggering of the data acquisition system.

The computers assigned with accepting or rejecting the data coming from the detector need to make sure the quality of the data is of acceptable standard. To achieve this the filter farm computers need to run fast and reliable DQM processes that ensure the quality of the data being stored.

To show the importance of monitoring the quality of data let us look at the particle identification process. Identifying secondary particles and measuring their behaviour is used to study a decayed particle created during the *pp* collision. The identification process depends heavily on the precision measurement by the detector's sub systems. In the

CMS detector for instance the inner tracker, which is used to measure the momentum of charged particles, and the electromagnetic calorimeter, which measures the energy of the particles (these sub-detector systems and their measurement mechanisms are described in Chapter 2) can be used to identify an electron. An electron's momentum, charge and direction are measured by the inner tracker and its energy is measured by the Electromagnetic Calorimeter. To identify the electron one of its characteristics can be used, which is given by $\frac{E}{|P|} \sim 1$. Although an electron is given as an example here, this behaviour is expected on most particles at the LHC energies; for example the π. This behaviour in an electron is due to the fact that an electron has very small mass in relation to its momentum, therefore at high velocity its momentum (P) and its energy (E) become almost the same. If the Tracker is not functioning properly, it won't give out a precise measurement. This leads to a possible erroneous measurement of the particle's momentum. This would lead to a false rejection of a valid electron or wrongly accepting of an electron by the data acquisition. In the case of false rejection, valuable information will be lost.

The DQM system helps identify problems such as the ones mentioned above leading to their correction by providing data that can be analysed. When the data is analysed and an error is noticed then the problem can be resolved in one of two ways:

1. Understand the behaviour of the detector to see why there is a problem, and a software adjustments can be made to compensate for the detector errors. An example of this is physical alignment constraints where the reconstruction software is tuned to compensate for the unavoidable misalignment effects.

2. If a problem with a particular sensor is detected then the sensor can be isolated or replaced if possible. An example of this is if a sensor has very high noise or dead strips in the case of the silicon sensors.

The aim of the work in this thesis was to provide a DQM software package that generates the data for monitoring the quality and tools that can be used to analyse the collected data. The Silicon Strip Tracker (SST) sub-detector system was chosen for this work and the data produced from this system was used to develop the monitoring system. Chapter 3 gives a detailed description of this system.

At the start of the project two challenges were identified, first to produce meaningful data that can be used to monitor the quality of the physics objects that are reconstructed by the detector and its software and secondly to provide tools that can be used to analyse the generated data for monitoring purposes. To address these challenges two software packages were developed.

The first software package provides data that can be used to monitor the quality of particle tracks of the SST (see section 3.3). This was developed in agreement with the requirements specified on the Physics and Detector Performance, Technical Design Report, Volume I [3]. Track objects provide data that can be used to monitor both the SST system and the data being generated. They can be used to provide information on how the reconstruction software is performing, or the individual detector modules are behaving. The details of this is given in Chapter 4.

The second part of the software package implements statistical tests that provide tools for analysing the data generated by the first part of the software package. A lack of two-dimensional data analysis tools was noticed in the services provided by the DQM system. This creates a major problem, as some of the quality monitoring data generated by the DQM packages are two-dimensional. This is discussed in Chapter 4 of this thesis. This was the motivation for working on the second part of this thesis, the statistical test tools for two-dimensional data. The two-dimensional statistical tests are relatively unexplored compared to the one dimensional data tests, for which greater familiarity exists. If the DQM software is to run on the data acquisition computer systems a tool for analysing the data automatically by the computer system is needed. This was addressed by studying and testing several algorithms and theories that could solve this problem. This work is discussed in Chapter 6.

1.5 Thesis outline

This thesis documents my work as part of the Tracker group in the CMS collaboration. It introduces the CMS detector in Chapter 2 by giving an overview of the CMS experiment. Chapter 3 gives details of the CMS tracker and the process of the track reconstruction.

This chapter discusses how the sub-detector works, what data is expected from it and how it will be used. Chapter 4 goes in to details of the the track quality monitoring software developed and the data produced by this software. It also discusses tests on the software by generating simulation data that can be used to monitor the quality of tracks. It will also discuss how the software was used in detecting a flaw in the track reconstruction algorithms during the commissioning of the tracker in the "Tracker Integration Facility" at CERN. Chapter 5 describes the usage of the DQM package in the important "Magnet Test and Cosmic Challenge". Chapter 6 gives details on the development and application of a statistics tool based on the recently developed "Energy Test" algorithm and its integration in to the DQM system. And finally chapter 7 will give general summary, conclusions on the research and suggestion for future work.

Chapter 2

The Compact Muon Solenoid Experiment

This chapter will introduce the Compact Muon Solenoid (CMS) experiment at CERN. The CMS detector is one of the two multi-purpose detectors located at the particle accelerator ring the Large Hadron Collider (LHC) of CERN. Modern day experiments of particle physics use accelerators that enable experimenting with high energy particles. The LHC accelerator is one such apparatus. The first section of this chapter will introduce the LHC. This will be followed by the individual sub-detectors of CMS and finally the Software used in the CMS experiment will be discussed.

This chapter is written based on the information given on the "Physics and Detector Performance, Technical Design Report" [3] and a paper on the CMS Experiment [4]. All information including figures should be considered as been referenced from these two documents unless specified.

2.1 The Large Hadron Collider

Particle accelerators are used to make particles energetic by accelerating them to almost the speed of light. These energetic particles when collided with other particles create other particles with greater mass. These newly created particles are then observed using detectors. To accelerate particles large electric field is created by the accelerators which attracts or repels charged particles. This field is then moved along the accelerator pushing the particles. There are two types of accelerators, Linacs and Synchrotrons. The Linacs are straight line accelerators and a good example of this is the Stanford Linear Accelerator Center(SLAC) found in the United States. The Synchrotron accelerators are circular in shape and the Large Hadron Collider (LHC) at CERN is a good example.

LHC at CERN is located in a tunnel 100 meters underground near the Jura mountain ranges on the Franco-Swiss Boarder. It is built in place of the old Large Electron-Positron Collider (LEP). The main goal of the LHC is to shed light on the nature of electroweak symmetry breaking for which the Higgs mechanism is believed to be responsible. Understanding the Higgs Mechanism through experimental study can elucidate the mathematical consistency of the Standard Model (SM) at energy scales of about 1 TeV [5]. There is also a possibility of discovering as yet unknown mechanisms and discoveries that could lead to a unified theory that will join SM and the alternatives of it.

The LEP's chief task was the use of 90–207 GeV e^- e^+ collisions to establish the precision physics of electroweak unification where as the LHC is aiming for a beam energy of 7 TeV and luminosity of $\mathcal{L} = 10^{34}$ cm$^{-2}s^{-1}$ of *pp* collision in a bid to explore new energy (centre-of-mass energy) domains in the region of 1 TeV.

The beam for the LHC starts at the proton synchrotron (PS) where it is injected in to the super proton synchrotron (SPS) with energy of 26 GeV. In the SPS the beam is accelerated to energies of 450 GeV and is injected in to the LHC where it achieves the aimed nominal energy of 7 TeV. The LHC consists of 1232 dipole magnets with r.f. cavities that increase the proton energy by 0.5 MeV/turn. This means that every time the beam is steered by the magnets it gains enery. The beam generated by the PS are grouped in to bunches. In the case of *pp* collision the PS will produce 2808 bunches at a correct spacing of 25 ns. This will achieve a collision rate of 25 ns during the full luminosity producing around 1000 particles per bunch crossing.

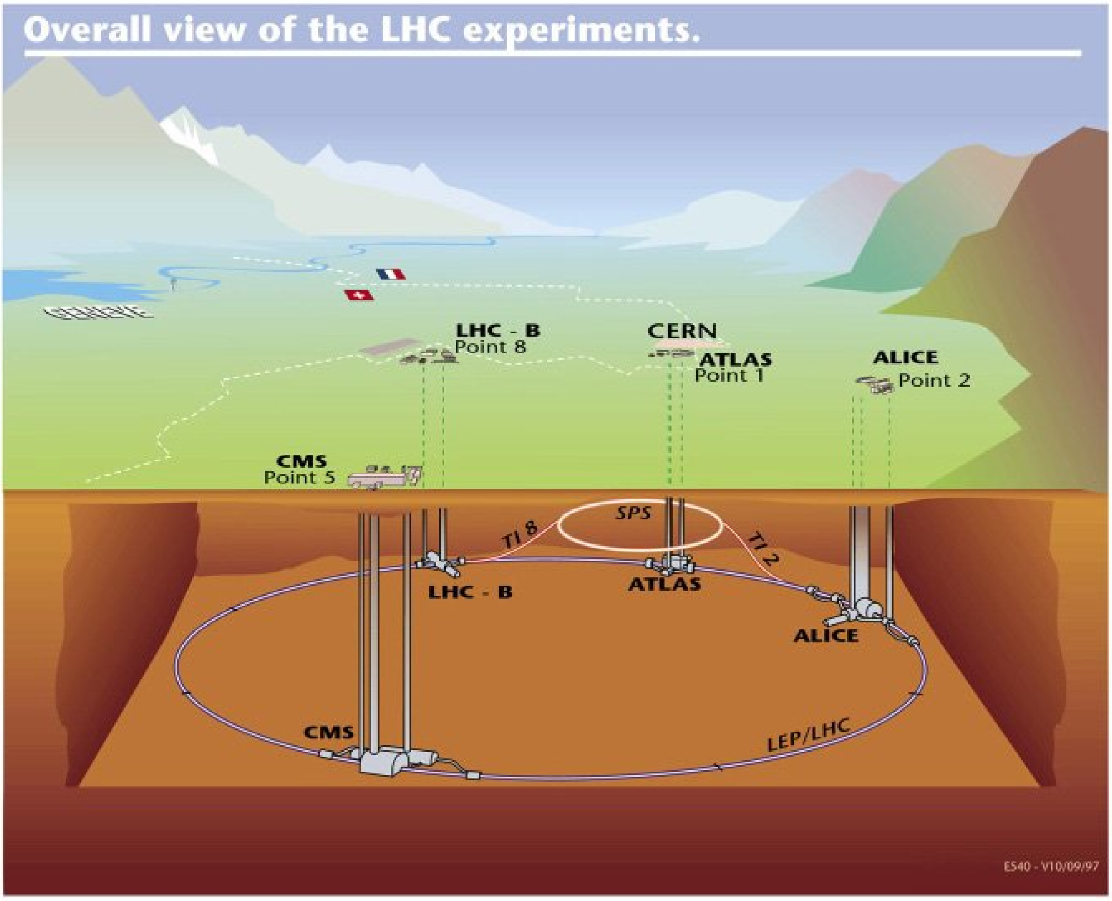

Figure 2.1: The LHC tunnel with the four main detectors [6].

2.2 The Compact Muon Solenoid

The main anticipated physics discovery behind the design of Compact Muon Solenoid (CMS) is the SM Higgs boson. Although CMS is a multipurpose detector and will be used to study physics theories such as extra-dimensions, super symmetric particles, QCD, electroweak and flavour physics; the SM Higgs was used as a benchmark to test the design performance. The ideal signatures of the Higgs particle can be found if it has mass in the region of $2m_Z < m_H < 600\text{GeV}/c^2$ where the ZZ decay can result in further four-lepton decay final state. Although the current lower limit of the mass of Higgs boson is 114.4 GeV/c^2 [3] the branching fractions of the Higgs boson at this mass are dominated by Hadronic decays. These are difficult to use to discover the Higgs particle at the LHC due to the QCD background and poor mass resolution obtained by the jets. For m_H $114-130\text{GeV}/c^2$ the two-photon decay is one of the principal channels likely to yield a significant signal.

The CMS detector has a defined coordinate convention in three dimensional space with x, y and z parameters. The z coordinate is defined to be the line that runs along the path of LHC pipeline through the detector. The y coordinate goes from the center of the CMS detector upward to ground surface. The x coordinate goes from the centre of the CMS detector to the centre of the LHC ring.

The detector was built over ground at SX5 of the LHC tunnel. This was done to advance the work on the detector without regard to the construction of the designated pit. The detector was then lowered to the pit in slices. Like most modern particle detectors, the CMS detector is made up of several layers of different sub-detector systems with an over all shape that is cylindrical. Figure 2.2 shows the CMS detector and its components. At the heart of the detector is the tracker which measures the direction and momentum of charged particles causing very small energy loss of the the incident particle. Outside the tracker are the calorimeters. First the electromagnetic calorimeter (ECAL) followed by the hadronic calorimeter (HCAL). Calorimeters are designed to measure the energy of particles by stopping them. The tracker, the Electromagnetic calorimeter and the Hadronic calorimeter are contained within the superconducting solenoid magnet. The outermost sub-detector consists of alternating layers of iron yoke and muon chambers.

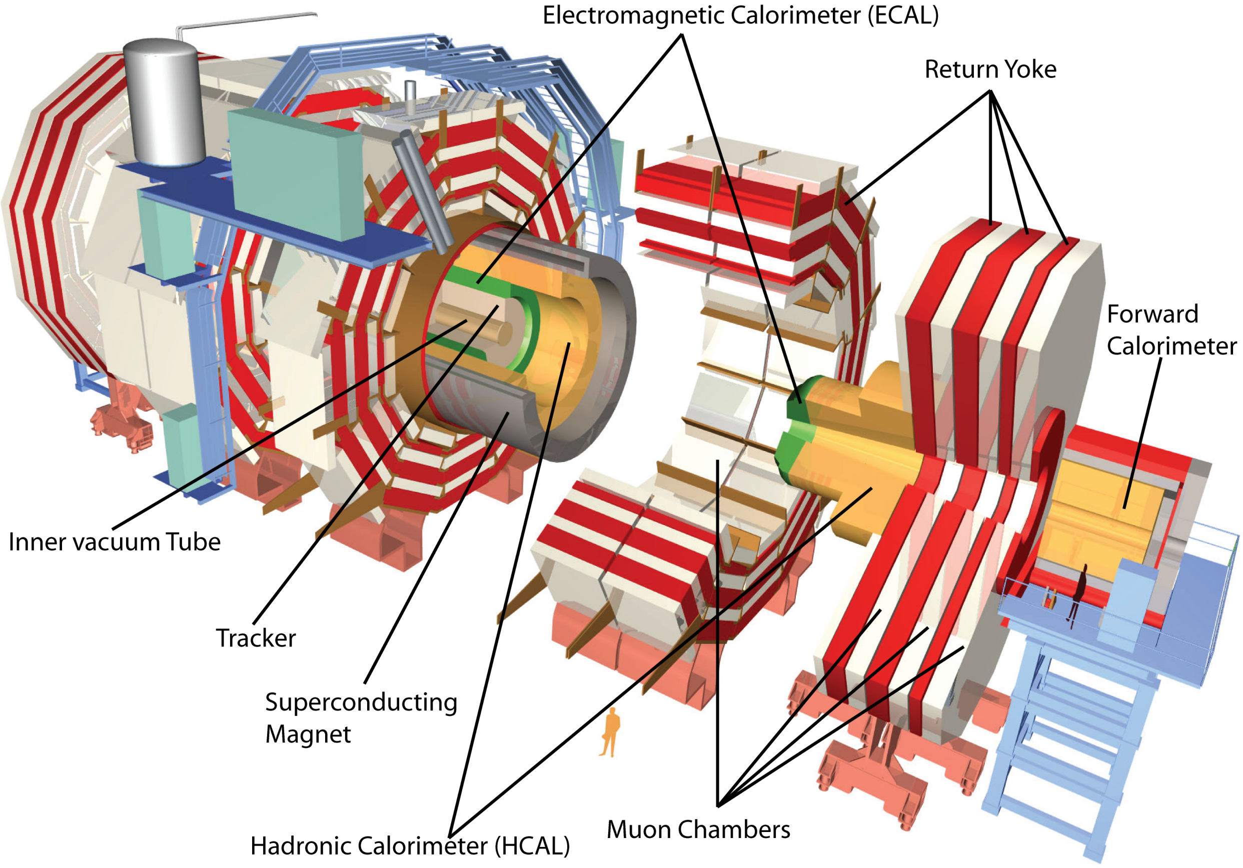

Figure 2.2: The CMS detector and its components [7]

Measuring particle properties is based on the interaction of incident particles with matter. The concept of the CMS design is to have an extremely low density material in the centre of the detector so that particle momentum is measured with minimal energy loss and extremely high density material on the outer layers so that the incident particles can be stopped and deposit all their energy. The different layers of sub-detectors are made of different materials making them useful in identifying particles. Particles interact differently with the various materials of the detector which means having these different sub-detectors in layers enables the identification of particles by studying how they interact with the material. This is illustrated by Figure 2.3 which shows the interaction of various particles with the different sub-detectors. Four particles are given as example in the figure.

The photon is a neutral particle and hence does not leave any track, but it is detected by the ECAL as it interacts through electromagnetic force with the detector material. An electron which is a charged particle curves in the magnetic field, interacts with the inner tracker leaving a track due to ionisation and deposits all its energy on the ECAL through electromagnetic interaction. The proton is a charged particle, it leaves track on the inner tracker and it interacts with the HCAL of the CMS. A muon is not stopped by any of the detector matter and it travels across the whole detector. It leaves a track in the tracker and the muon chambers and it deposits small fraction of its energy on the ECAL and HCAL. The various sub-detector systems are briefly discussed below.

2.3 The Solenoid Magnet

The superconducting solenoid produces 4 Tesla magnetic field. The magnetic field from the solenoid is used to identify the charge and momentum of particles. Oppositely charged particles in a magnetic field move in opposite directions, and this is seen by the particle tracks left on the detector. The magnetic field also curves the path of the particles giving them helical shape. This curvature is used to calculate the transverse momentum of the charged particles.

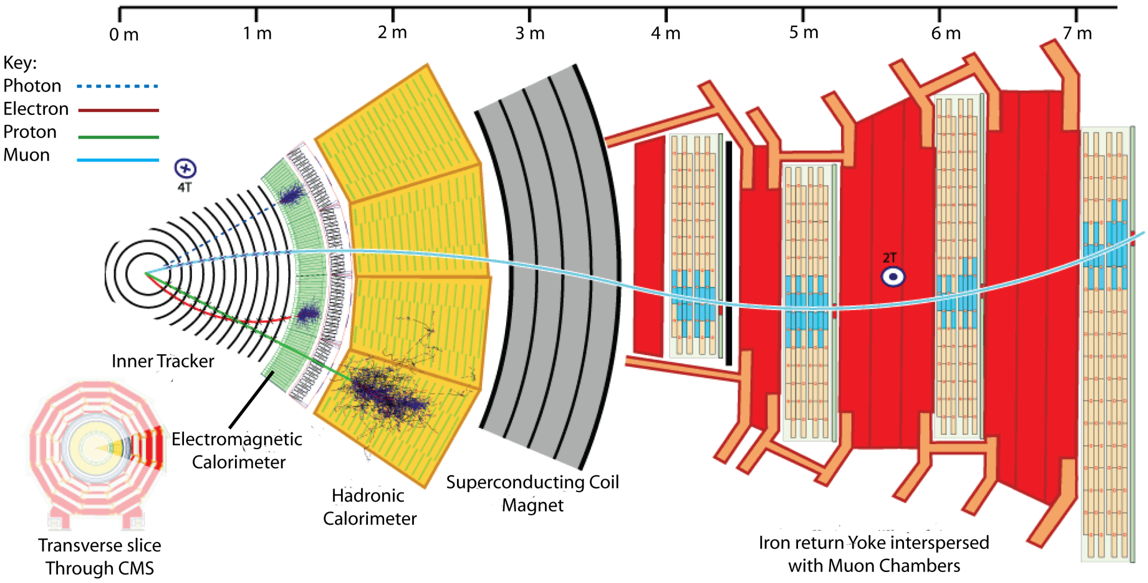
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Photon
Electron
Proton
Muon
4T
2T
Inner Tracker
Electromagnetic Calorimeter
Hadronic Calorimeter
Superconducting Coil Magnet
Iron return Yoke interspersed with Muon Chambers
Transverse slice Through CMS

Figure 2.3: Interaction of particles with the various sub-detectors of CMS [7].

2.4 The CMS Tracker

The CMS Tracker System is a cylindrical shaped sub-detector and it is built in several layers. Each layer of the Tracker consists of several thin silicon sensors. Charged particles travelling through the silicon sensors interact with it to lose a fraction of their energy through ionisation. Although this loss of energy is very small and has very limited effect on the particle momentum, it is what enables the detection of the incident particle in 3D space.

The original design of the CMS Tracker was to have three different parts based on the sensor technology. Two of which being silicon sensors and the third Micro-strip Gas Counter (MSGC) [8]. This design was later changed to make the detector an all silicon detector [9]. The all silicon detector has two different parts arising from the difference in the technology of the silicon sensors used. They are the Pixel Tracker, which is the closest to the beam pipe and the Silicon Strip Tracker covering the rest of the Inner tracker volume outside the pixel detector.

The Length of the of the inner Tracker is 5.4 m and has a diameter of 2.4m. The total weight of the detector when fully installed will be 3 tons [10]. Geometrically the detector modules are divided in to two shapes. The Barrel and the Disks (end-caps). The Barrel Modules are rectangular in shape where as the disk modules are wedge shaped.

2.4.1 The Pixel Tracker

The pixel tracker is the inner most sub-detector system making it the closest to the beam pipe. It has three layers of open ended cylindrical parts making the barrel section and the two disks at each end of the barrel section. Figure 2.4 shows a schematic drawing of the pixel system. The barrel layers are 53 cm long and are located at mean radii of 4.4 cm, 7.3 cm and 10.2 cm from the beam pipe. The two discs on each end of the barrel are located at 34.5 cm and 46.5cm on the z axis. This enables the pixel tracker to cover a pseudorapidity range of $-2.5 < \eta < 2.5$, and the tracker will give 3 tracking points over almost the full η range [11].

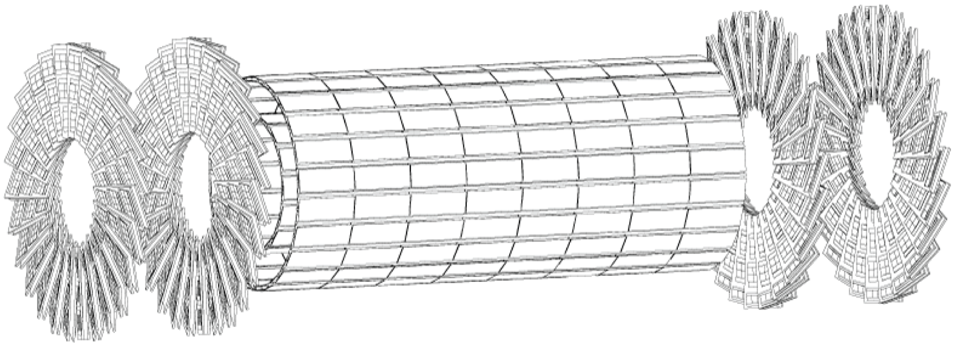

Figure 2.4: A schematic drawing of the pixel tracker. The barrel of the pixel detector is 53 cm long and the outer layer has a diameter of 20.4 cm. Each disk is located at 34.5 cm and 46.5 cm from $z = 0$. [7]

During each bunch crossing of the LHC an estimated 20 hard-core head on pp collision are expected resulting in around 1000 newly created particles [12]. The pixel tracker being placed very close to the interaction point of the CMS means its sensors are exposed to extremely high radiation. Taking this into consideration an emphasis was made on highly radiation tolerant sensors and the n-on-n silicon sensor concept was used. The pixel system is also designed to be accessible every year so that it can be replaced in the unlikely event of it being damaged beyond use. But the design of the system is such that it can be used for two years. It will be the last sub-detector to be inserted into CMS.

The silicon sensors of the pixel tracker have pixels with surface area of 100×150 μm^2. This enables the pixel tracker to measure track position points in the $r - \phi$ and z. One advantage of the pixel detectors over the silicon strip detectors is the high precision measurement of a hit position on the $r - \phi$ and z direction of the 3D space. This enables a 3D vertex reconstruction in space, which is very important for secondary vertices.

The pixel tracker system has pulse height read-out from its sensors which improves the position resolution due to charge sharing because of the Lorentz effect (see section 3.2.6). This allows a spatial resolution in the range of 15-20 μm to be achieved. The Lorentz effect happens naturally in the barrel section of the pixel system as the incident particles path is perpendicular to the magnetic field. As this is not the case for the end-cap modules they had to be tiled by 20^0 to take advantage of this effect giving them a turbine like structure as can be seen from figure 2.4.

2.4.2 The Silicon Strip Tracker

The Silicon Strip tracker is found immediately outside the pixel detector and consists of over 15,000 detector modules. The detector modules used for this sub-detector are made up of silicon sensors with strips instead of pixels. This means that the detector modules will only detect particle position on the x coordinate local to the detector module. The sub-detector system has four sections, two of them make up the barrel section and are know as TIB and TOB, and the other two make up the disk section which cover the barrel section of the sub-detector and are known as TID and TEC.

The work presented in this thesis is closely related to this part of the sub-detector and a whole chapter has been dedicated to it. The detailed description of this sub-detector and how it works is given in chapter 3.

2.5 Calorimetry

Calorimeters, unlike the tracker, are designed to stop the particles from going through so that all their energy is deposited in the detector material. The deposited energy is then measured through the secondary particle shower created due to the interaction of the incident particles with the detector. Calorimeters have several advantages in comparison to the tracker, some of which are given below [13]. But this does not make them better detectors than the tracker as the Tracker has the advantage of enabling the measurement of particle momentum.

1. They provide measurement for both charged and neutral particles unlike the tracker which can only detect charged particles.
2. The measurement precision of calorimeters improves with increasing energy.
3. They have fast time response which makes them suitable for experiments with high collision rate.
4. They respond differently to various particles, and this makes them useful when it comes to particle identification.

2.5.1 The Electromagnetic Calorimeter

Depending on how a particle interacts with a detector material, it creates a shower of secondary particles. These can be either electromagnetic or hadronic showers. Electromagnetic showers are the result of loss of energy of particles interacting with the detector material through the electromagnetic force. Charged particles and photons result in electromagnetic showers upon interacting with the Electromagnetic Calorimeter (ECAL). When energetic electrons are decelerated by the ECAL they lose energy by giving out electromagnetic radiation in the form of photons. This process is known as Bremsstrahlung. This phenomenon is exhibited by all charged particles, but electron energy loss is dominated by this. An incident electron entering the ECAL creates a cascade of secondary particles. These secondary particles are known as an electromagnetic shower and consists of both photons and $e^{\pm}$ pairs coming from the photon conversion. Incident photons to the ECAL gives out electron (e^{-}) positron (e^{+}). The $e^{\pm}$ pair in turn give out γ due to Bremsstrahlung given they have enough energy. The γ then creates further $e^{\pm}$ pair. This cascade can be seen in Figure 2.5. The number of the particles created in the electromagnetic shower from the incident particle is equivalent to the particle energy lost due to the stopping power of the detector material. This method gives a high precision measurement of particle energy.

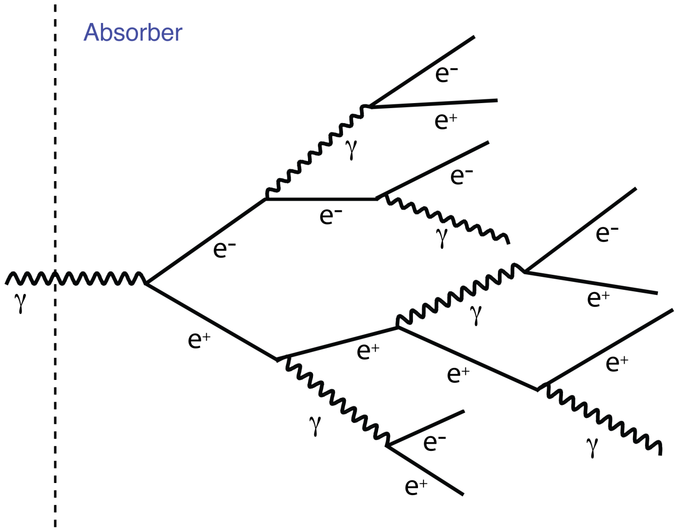

Figure 2.5: A High Energy γ incident to a thick absorber initiates a shower of secondary e and γ via pair production and Bremsstrahlung [13].

The ECAL as with all of the other sub detectors is a hermetic detector. The term hermetic when used in detectors implies that the detector has maximal coverage for particles originating from the interaction point of the detector. When a calorimeter is hermetic it allows the energy measurement of particles in all directions so as to infer the missing energies that escape in the form of neutrinos. The ECAL uses scintillating crystals made of lead tungstate ($PbWO_4$), making it a homogenous detector. A detector is homogeneous when the same sensor is used both as an absorber and detector. The ECAL is a very fast detector as 80% of the light is emitted in the first $25ns$ meeting the collision time of the LHC. The design of the ECAL is driven by the capability of detection of the $H \to \gamma\gamma$ [4].

The ECAL is divided into two based on the location of the detectors. The Tracker Barrel and the end-cap. The barrel part of the detector consists of 61,200 crystals and each end-cap consists of 7,324 individual crystals. The barrel detector modules use avalanche photo diodes (APD) for photodetection. Whereas the end-cap crystals use vacuum phototriodes (VPT).

The structural layout of the ECAL can be seen in Figure 2.6. The crystals of the barrel part of the ECAL are grouped to form a module. Each module is made up of an alveolar structure which contain the individual crystals. The modules are grouped in to super modules. The end-cap of the ECAL sub-detector has two wheels, where each wheel is divided in half making the Dee. The crystals in the end-cap are arranged in groups of 5×5 forming super crystals.

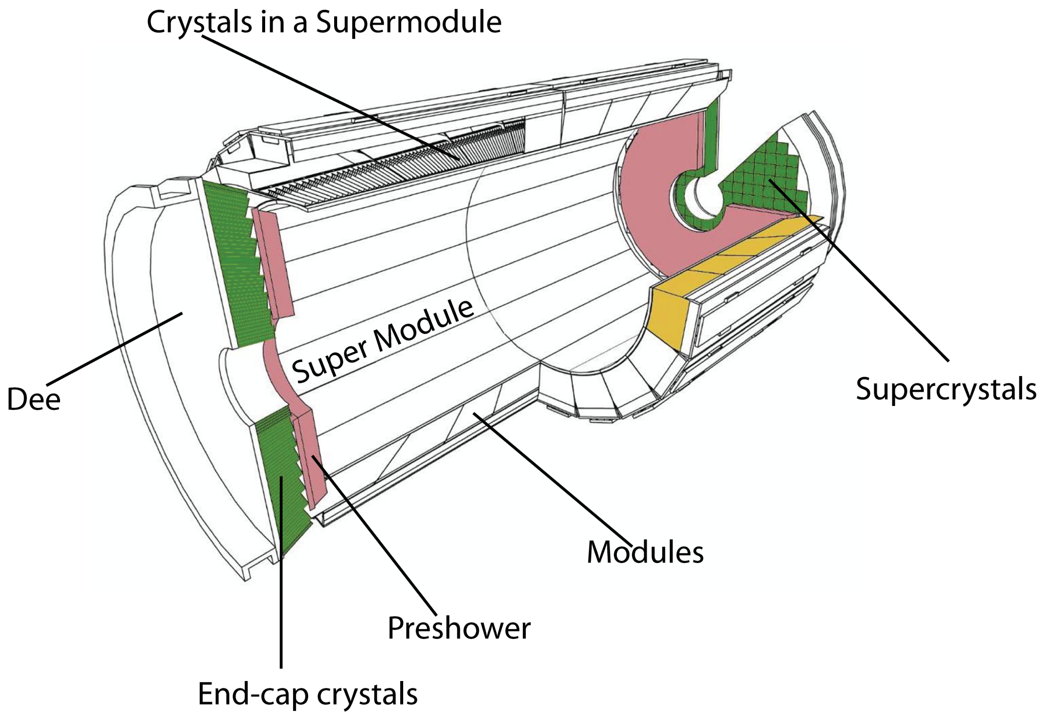

Figure 2.6: View of the ECAL sub-detector. The barrel of the ECAL (at the surface of the crystals) has a diameter of 2.58 m [7].

The performance of a supermodule was measured in a test beam. Representative results on the energy resolution as a function of beam energy are shown in Figure 2.7. The energy resolution, measured by fitting a Gaussian function to the reconstructed energy distributions, has been parameterised as a function of energy:

$$\left(\frac{\delta}{E}\right)^2 = \left(\frac{S}{\sqrt{E}}\right)^2 + \left(\frac{N}{E}\right)^2 + C^2 \tag{2.1}$$

where S is the stochastic term, N the noise and C the constant term. The values of these parameters are listed in Figure 2.7.

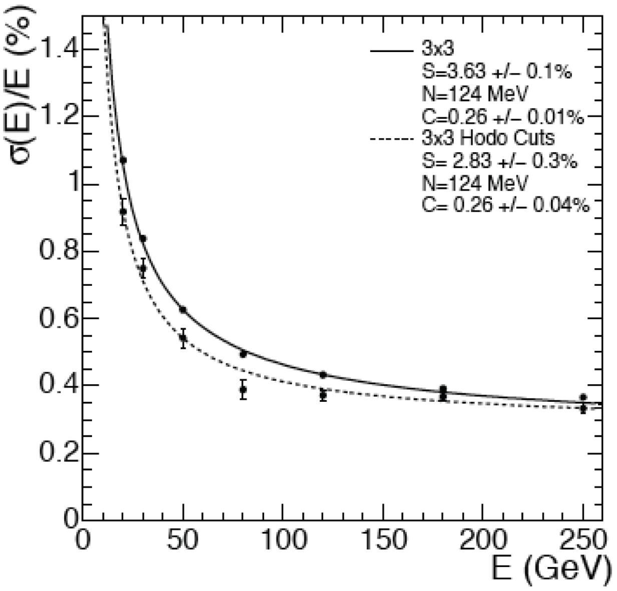

Figure 2.7: ECAL supermodule energy resolution, $\delta E/E$, as a function of electron energy as measured from a beam test. The energy was measured in an array of 3×3 crystals with electrons impacting the central crystal.

Preshower detector

The Preshower detector is designed to identify neutral pions in the endcaps of the ECAL and are placed within the $1.653 < |\eta| < 2.6$ region. It also improves the position determination of electrons and photons with high granularity.

The Preshower is a sampling calorimeter with two layers: lead radiators initiate electromagnetic showers from incoming photons/electrons whilst silicon strip sensors placed after each radiator measure the deposited energy and the transverse shower profiles. The total thickness of the Preshower is 20 cm. The material thickness of the Preshower traversed at $|\eta| = 1.653$ before reaching the first sensor plane is $2X_0$, followed by a further $1X_0$ before reaching the second plane. Thus about 95% of single incident photons start showering before the second sensor plane. Each silicon sensor measures 63×63 mm^2, with an active area of 61×61 mm^2 divided into 32 strips (1.9 mm pitch). The nominal thickness of the silicon is 320 μm.

2.5.2 The Hadron Calorimeter

The concept of measuring energy of hadronic showers is similar to that of the Electromagnetic showers. Incident high energy hadrons interact with nuclei of the detector matter through the strong force giving a shower of secondary particles. The production of hadronic showers is more complicated due to the fact that the shower has both hadronic and electromagnetic components. Figure 2.8 shows the hadronic cascade caused due to the interaction of a neutron with an absorbing material. The Hadronic showers consist of mainly pions (π^0, $\pi^\pm$). π^0 has a very short life and decays giving $\pi^0 \rightarrow \gamma\gamma$ which dominates the electromagnetic component of the shower. Hadronic calorimeters are usually sampling calorimeters. These are layers of passive absorber material alternating with active sensor.

The Hadronic Calorimeter (HCAL) of the CMS consists of a passive absorber material made of brass and plastic scintillators which are the active sensors. They are arranged in a sandwich-like alternating structure. The design of HCAL will enable it to play a major role in the measurement of hadron jets and neutrinos or exotic particles resulting in apparent missing transverse energy [4].

The HCAL has a Barrel (HB) and End-Cap (HE) components. The HB covers pseudorapidity range of $|\eta| < 1.3$. It has 36 identical azimuthal wedges which are divided in to two making the HB+ and HB-. The HE covers pseudorapidity range of $1.3 < |\eta|$. The absorbers of both the HB and HC have openings in between two brass plates that allows

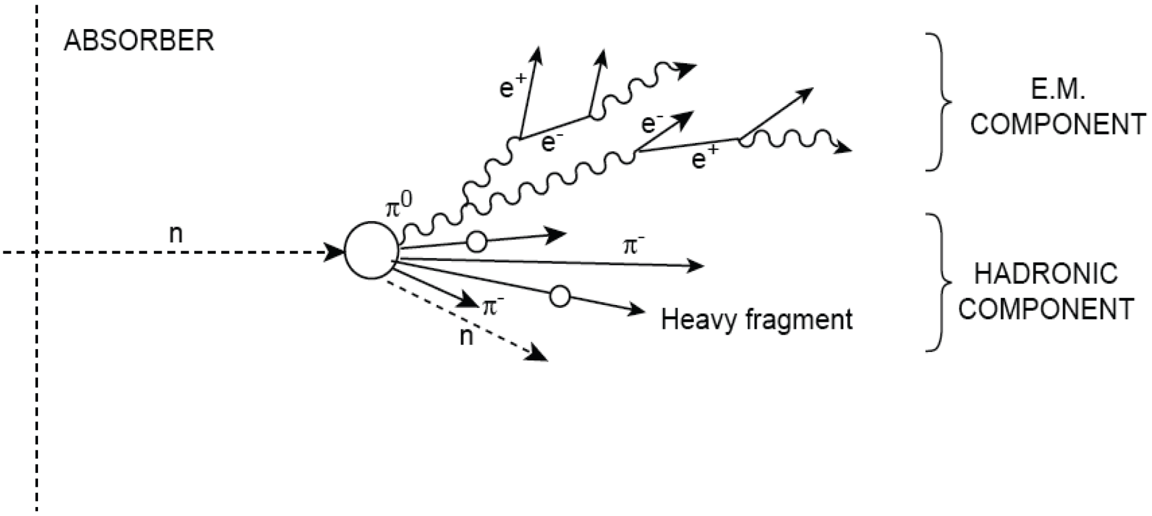

Figure 2.8: Hadronic cascade development as an incident neutron initiates hadronic shower with both electromagnetic and hadronic components [13].

the insertion of the scintillator trays.

In total the HCAL consists of around 70,000 scintillator tiles. Each layer of the wedge in the HB consists of 4 scintillator trays and each scintillator tray consists of 16 towers. The transverse energy resolution of the different parts of the HCAL can be seen in Figure 2.9.

2.6 The Muon System

The muon system, as the middle part of the name CMS indicates, is a very important part of the detector. It has three main functions. First it plays a major role in the Level 1 triggering of events. Second it is the main means of identifying muons and third which is integral to the second is reconstructing the muon tracks and measuring their momentum. The identification and reconstruction of muons is a very important part of the physics studies in the CMS experiment as one of the envisaged signature of the Higgs particle is through its subsequent decays into ZZ leading to the four-lepton end state. The leptons being muons has a very high advantage over the other leptons as muons are less affected by radiative losses in the tracker, and they can give a good 4-particle mass resolution. The

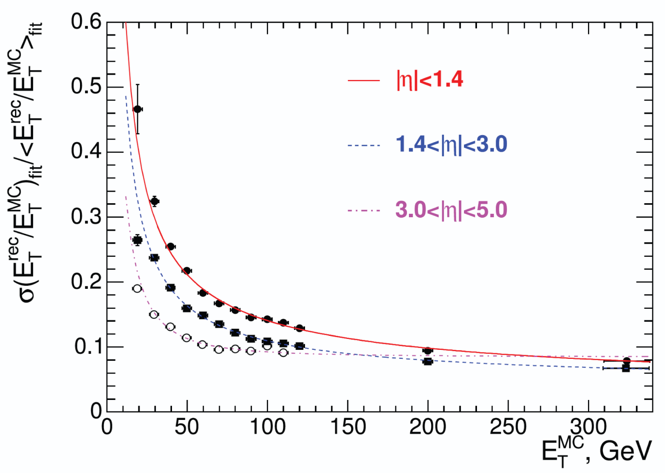

Figure 2.9: Relative transverse energy resolution of the HCAL barrel ($|\eta| < 1.4$), the HCAL endcap ($1.4 < |\eta| < 3.0$) and and the HCAL very forward ($3.0 < |\eta| < 5.0$) as a function of the transverse energy [14].

other advantage of the muon system is, because it is so far out from the interaction point, it is easier to detect muon as the particle flux decreases.

The muon system surrounds the superconducting coil and it is a combination of muon chambers sandwiched between parts of iron yoke. The iron yoke is used to return the magnetic field. As with all the other sub-detector systems, the muon system is designed to be cylindrical due to the solenoid magnet. It consists of a barrel and two end-cap sections. The barrel section is divided in to five wheels. Each wheel has four layers of Drift Tube chambers (DT). Each layer is called a station. The first three stations consist of 12 chambers each and the fourth has 14 where there two extra chambers placed one on the top and another one on the bottom. The DT chambers are positioned in a staggered style to guarantee at least an interaction of an incident muon with three stations. Figure 2.10 shows one of the muon system wheels. The barrel section of the muon system will cover pseudorapidity region of $|\eta| < 1.2$. The end-cap of the muon system will consist of 468 Cathode Strip Chambers (CSC) and it will cover the pseudorapidity range of $1.2 < |\eta| < 2.4$. There is an overlap of area coverage by the barrel and the end-cap which is in the range of $0.9 < |\eta| < 1.2$ where muons are detected by both the DT and CSC. The CSCs are trapezoidal in shape covering either 10^0 or 20^0 and are overlapped to provide contiguous ϕ-coverage with the exception of ME1/3 ring. Besides the DT and CSC the muon system has additional sensors dedicated for triggering, the Resistive Plate chambers. They provide fast response with good resolution but they have coarser position resolution in comparison to the DT and CSC.

The muon system has expected momentum resolution of $6-20\%$ for $P_t < 100$ GeV and $15-35\%$ for $P_t = 1$ TeV depending on the angular position. These values are improved when the inner Tracker system is used, to $2-6\%$ for $P_t < 100$ GeV and $8-18\%$ for $P_t = 1$ TeV [15].

2.7 Trigger And Data Acquisition System

The CMS Trigger and Data Acquisition (TriDAS) system is designed to store selected data events detected by the CMS for an offline analysis. An event in this context is a single bunch crossing at the vertex point. A run which is closely related to event is the time that a given amount of bunches will be injected in to the accelerator. According to this definition a run would have a number of events. The need for data reduction comes due to the extremely high amount of data generated. The collision rate of the LHC which is at 40 MHz would result in approximately 20 simultaneous proton proton head on collision. The aimed rate of data storage is 100 Hz. To achieve this the TriDAS has a trigger system with two parts and data acquisition system that facilitates data storage.

The trigger system of the TriDAS consists of the Level one Trigger (L1T) and the High Level Trigger (HLT). The L1T which is a system of custom made programmable logic electronics will reduce the data rate to 10^5 Hz. The HLT's aim is to reduce the data rate to 100 Hz rate which is the expected nominal rate for the CMS data taking.

The muon system and the the two calorimeters are used in the triggering processes. In the L1T three stages are followed before accepting or rejecting an event. Figure 2.11 shows the hierarchical structure followed by the L1T and the different paths taken by the different triggering sub-detector systems. At the bottom end of the process is the local trigger also called the Triggering Primitive Generators (TPG). These take readings of energy from the

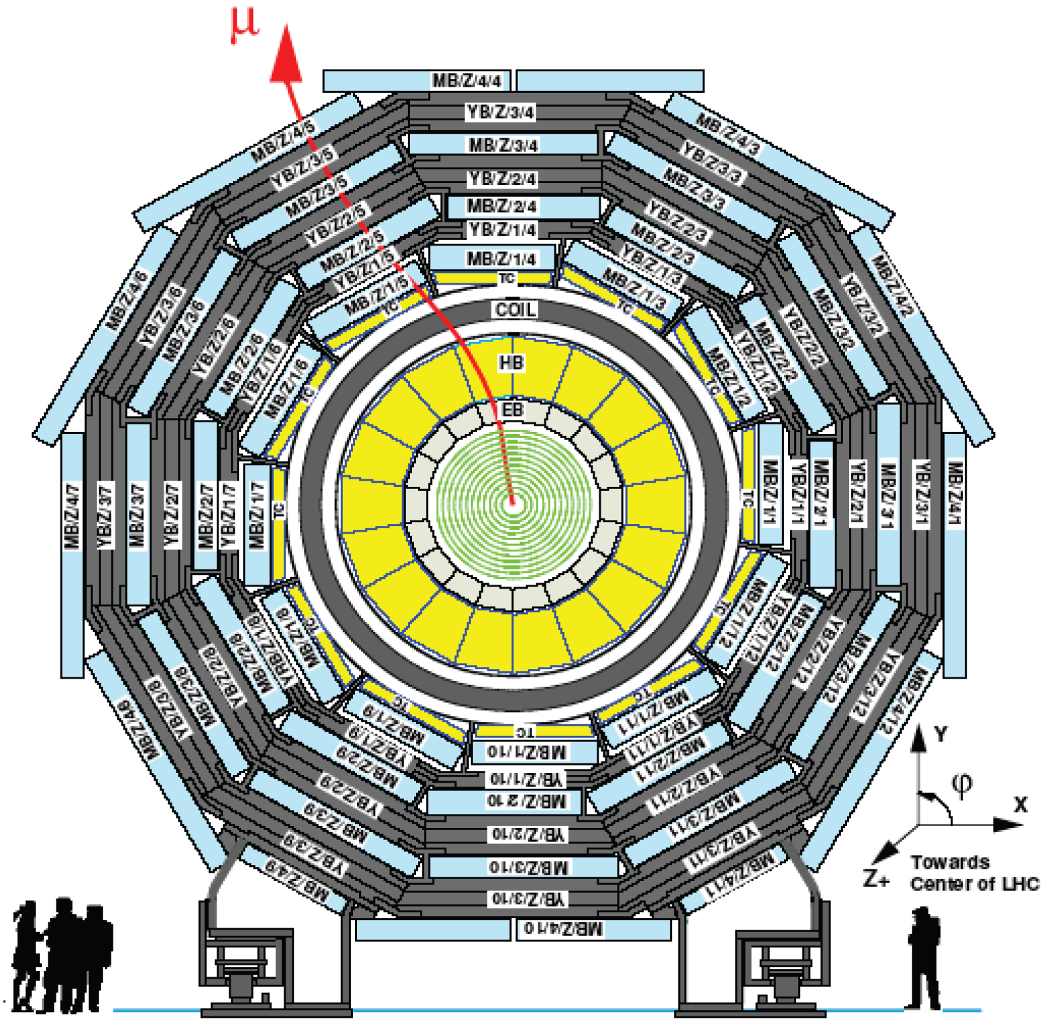

Figure 2.10: Layout of the CMS barrel muon DT chambers in one of the 5 wheels.

calorimeters and/or hit patterns in the muon chambers. The information from the TPG is then further processed by the second stage which is the Regional triggers. This information is then processed to determine particle candidates such as an electron or muon which are ranked using a function of energy in the case of calorimeters or momentum and quality in the case of the muon system. Global Muon/Calorimeter Trigger determine the highest ranking calorimeter or muon system objects for the whole CMS detector and passes this information to the Global Trigger. The global trigger makes the decision on accepting or rejecting an event for further evaluation of the HLT. Depending on the status of the sub-detector readout systems and the Data Acquisition; information provided by the Trigger Control System (TCS), the Level 1 Accept (L1A) signal is relayed to the sub-detector systems using the Timing Trigger and Control (TTC) system. The allowed latency for the L1T between given bunch crossing and L1A signal is 3.2 μs. This is the maximum time allowed for all the sub-detector systems to pipe sensor readings.

The HLT is a software system implemented in a filter farm of about 1000 processors. It runs a faster version of the reconstruction software discussed in Chapter 3. This enables the reconstruction of the fragmented information that was passed by the L1A and saves them as physics objects. An example of a physics object is a particle candidate, or a track of a charged particle. These objects are then studied by the automated triggering algorithms of the HLT to see if the event is of interest. In the case where the the event is accepted, the data acquisition system transfers the data related to the accepted event to the storage database where it can be accessed for further analysis.

2.7.1 Data Acquisition

The Data Acquisition (DAQ) system of CMS is responsible for collecting data and storing it for further analysis. The DAQ is involved from the very start of data taking beginning in the L1T process. At the design luminosity of the CMS, which is $10^{34}\mathrm{cm}^{-2}\mathrm{s}^{-1}$, the estimated head on pp collision is around 20 per bunch crossing resulting in the detection of approximately 1000 newly created particles every 25 ns. This will produce around 1 MB of zero-suppressed data from approximately 650 readout system. The L1T will reduce the event rate to 100 kHz which means the Builder Network should cope with data flow

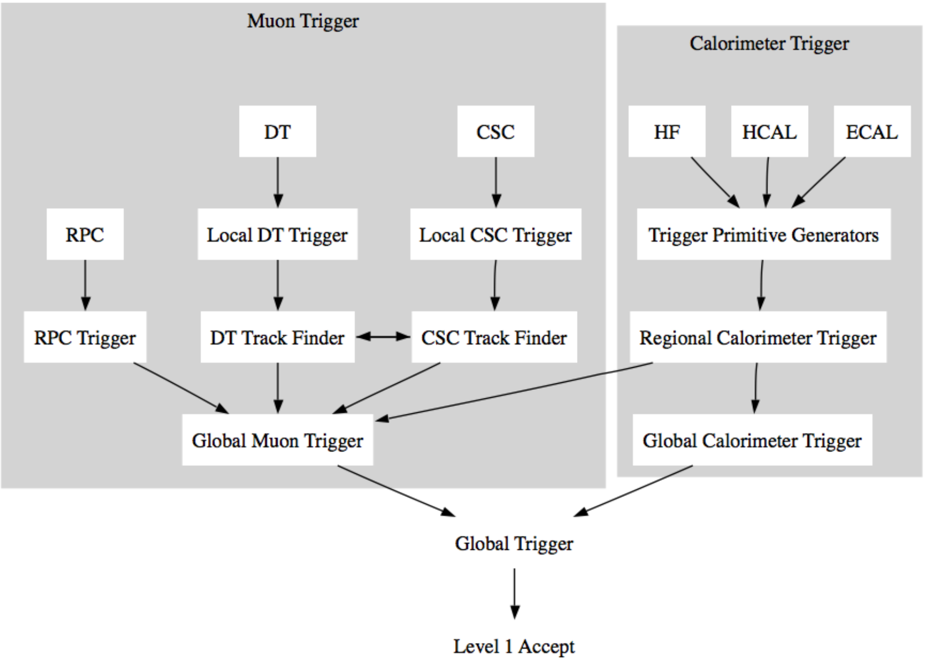

Figure 2.11: The Level 1 trigger of the TriDAS uses the Muon and calorimetry system. In the Muon trigger the Drift Tubes (DT), Resistive Plate Chambers (RPC) and the Cathode Strip Chambers (CSC) make up the primitive detectors. In the Case of the Calorimeters the Forward Calorimeter (HF), the Hardronic Calorimeter (HCAL) and the Electromagnetic Calorimeter (ECAL) make up the primitive triggering generators)

of 100 GB/s coming from approximately 650 data sources. Figure 2.12 shows a schematic drawing of the arrangement of the DAQ system for the data taking. The data collected from the front-end read-out systems is relayed to the filter systems where the HLT software runs.

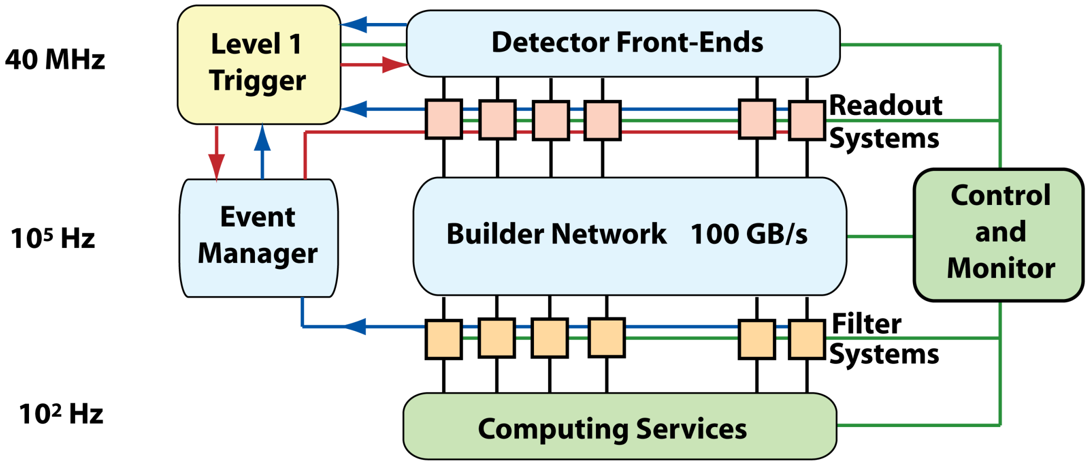

Figure 2.12: The structure of the Data Acquisition system of CMS.

The first step in the filter farm is to build the data coming from the different sub-detector systems. The builder unit of the filter farms is responsible for putting together the fragmented information received from the various sub-detectors. For example the strips of a tracker detector module or crystals of an ECAL module readings that make up a cluster. The data is then read by the filter units (FU), which are part of the DAQ computer farm running the reconstruction software. The data is then used to reconstruct physics objects such as particle candidates or particle tracks. The HLT process needs to be fast to avoid a pile up of events, while reducing the incoming data by factor of 1000. To achieve this the FU running this process needs computing power equivalent to 1,250 nodes with two Intel dual-core processors [4].

2.8 The CMS Software

CMSSW is the framework software used for the various CMS packages. It is divided into subsystems. The subsystems contain packages and the packages contain modules. The modules are the part of the system that do the actual work. In other words the programming code is found in the modules. Each package in the CMSSW framework can have several modules.

There is a sequential flow of data in CMSSW structure. The data acquisition system would take the raw data and store them in the database. Then it is the job of CMSSW to save the data in a ROOT file that can be accessed by the various packages. Figure 2.13 shows how a track object reconstruction process is carried out in the CMSSW data flow. The "Event"[1] is a ROOT file that is produced from the database stored by the DAQ system. The ROOT file is then accessed and/or modified by the first package which is the digitiser. This creates the first ROOT objects known as the digis which can be used in the hit reconstruction of the track. The digis are then accessed by the tracker modules where the track objects are created. Finally the data is stored as an output in to the database where it can be analysed by the data analysis packages.

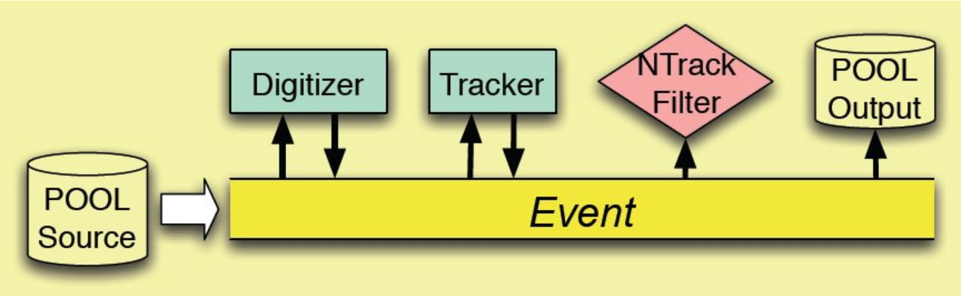

Figure 2.13: The sequential Processing chain of data in the track reconstruction of the CMSSW

The CMSSW was designed according to the requirements that arise from the detector itself and its users. The following requirements have been rephrased from [3]

[1] Event and event should not be confused. The first one with the capital "E" is a name given to a data storage file, where as event is the moment when a collision happens at the centre of the detector.

1. Different software modules should be able to run on different environments. For example a module for individual analysis should run in that environment and at the same time a module for high level trigger (HLT) would run in the HLT environment making it multiple environment software.

2. modules should be environment free so as to enable migration of module from one environment to another

3. In case of change of technology the software should be able to migrate to the new technology without a major rewrite of the software

4. The software should tolerate disperse code development as the authors are both geographically and organisationally dispersed

5. The software should be flexible to accommodate changes without a full re-write of the code, as all requirements can not be known.

6. The software should be easy to use as the users might not be experts of computing. It should not take large amount of time to understand its usage.

Based on the requirements given above the CMS software was designed to have a structure with three components.

2.8.1 Application Framework

The software is designed to have an application framework which defines the top level abstractions, their behaviour and collaboration pattern. It is comprised of two components based on the function of the classes:

1. A set of classes defining the CMS specific concepts such as the detector and event related concepts.

2. A control policy that organises and co-ordinates the module scheduling, control flow and looking after the input/output.

The CMS software application framework goes hand in hand with the event data model (EDM). EDM is the model used for the storage of the source data. The EDM is designed with ease-of-use in mind and taken as priority. The data objects stored are required to be as simple as possible so that access by modules would not require another layer that is specific to reading the stored data. In other words all the data stored need to be persistent. Consistency and ease of use of the software is possible due to the automatic storage of the provenance information of application results by the Framework.

2.8.2 Physics and Utility modules

These are physics specific software components that are written by the different detector groups. They are designed as modules with clearly defined interface (to the framework) so that they can be plugged in to the application framework at run time. As the modules are independent of the computing environment the user has the flexibility over the choice of which module and what version of it to use. These physics modules are independent of each other and can only communicate with each other through the framework's data access protocol provided by the toolkit described below.

2.8.3 Service and Utility toolkits

The service and utilities toolkits can be used by any of the above physics modules. They provide two major services:

1. Physics type services which provides tools that can be used by the modules. An example would be they can provide mathematical algorithm that produces results needed by the module or they can provide histograms that can displays results from the module. These types of services and toolkits are directly related with the work on this thesis.

2. Computer services which enable modules to communicate between themselves or enable them to access the data they need to process.

These toolkits alongside with the application framework make the physics modules independent of the underlying technology hence in the case of the upgrade in the technology the modules would not need a re-write.

2.8.4 Structure and interaction of the various CMSSW components

The design of the CMS software is hierarchical that starts with the system which is CMSSW. The system can be seen as the root folder in a linux file system. It contains all the classes that make up the framework, the physics modules and the services. Each class in the CMSSW is designed as a plugin module. Inside the CMSSW exist subsystems which contain packages. These subsystems define the usage of the packages within the the system. For example, the reconstruction modules are categorised under one subsystem called Reco and the Data Quality Monitoring modules are categorised under a different subsystem called DQM and so on. Each package under the subsystem contains one or more modules. The DQM for instance contains a package named Tracker_Monitor_Track which is a container for two modules, the Monitor_Track_Global and Monitor_Track_Residual. These modules will be used to illustrate the interaction of the different components of the CMSSW. These modules are described in detail in Chapter 4. This illustration will also give an introduction to the module and service toolkit that are used in the work of this thesis discussed in chapter 6. The development of the package named Tracker_Monitor_Track was motivated by the idea to provide data that can be used to monitor the quality of the reconstructed tracks (see section 3.2.1).

The Application framework defines two types of modules that can can be plugged in to it, the EDAnalyser and EDProducer. The EDAnalyser modules are created for the sole purpose data Analysis. This means these modules are allowed to read data from the the Event but are not allowed to write data in to it. To read data the EDAnalyser modules have to use the CMSSW service toolkit. These are provided by the service an utility modules. The EdProducer modules on the other hand are allowed to write back data in to the Event. This is illustrated by Figure 2.13. The modules for the digitiser and Tracker are of the type EdProducer. They have arrows pointing towards them and away from them showing the data flow. In the figure, the process of data flow is from left to right.

The modules of Tracker_Monitor_Track packages are of the type EDAnalyser. Their function is to produce histograms showing the data taken from the Tracker in order to monitor the quality of the tracks produced. For modules to do this the track objects need to exist first. Going back to the data taking of the DAQ the raw data acquired from the detectors are stored in a database after being given the HLT acceptance.. After this the data is accessed by CMSSW and put in to a ROOT file called "Event" (see Figure 2.13). The reconstruction packages which are responsible for creating the physics objects then access the ROOT file for the raw data or the pre-processed data to do the reconstruction. Most of the reconstruction modules are of type EDProducers and therefore can put back the reconstructed objects back to the Event. After all the necessary reconstructions are carried out the data is stored in a data base where it can be accessed by EDAnalysers for further analysis.

The EDAnalysers such as the modules of the Tracker_Monitor_Track then can ask for the modified data through the CMSSW data access protocol where the Event is made accessible to them. The modules can then access the track objects to acquire the needed data for monitoring and store it in another ROOT object (as histograms) or make them directly available for other applications such as the client of the DQM through the CMSSW service and utility toolkits.

Chapter 3

The Silicon Strip Tracker

The goal of the CMS central tracking system is to reconstruct isolated high P_t muons and electrons with a momentum resolution of better than $\Delta P_t/P_t = 0.1P_t$ (in TeV/c) at an efficiency of more than 95% over a rapidity range of $|\eta| < 2.5$ [16], [17]. To achieve this, when fully installed, the Silicon Strip Tracker (SST) will consist of 15,148 detector modules, which will cover an area of 210 m ^{2} with sensitive silicon. These detector modules have readout channels that are close to 10^7 and will be read using 75,376 APV25 chips[18]. The operational temperature for these modules is -20^0 so that they can survive the harsh conditions of radiation caused by the ionising particles flying from the collision point.

My work as a member of the CMS collaboration discussed in Chapter 4 is directly related to the SST, therefore this chapter is intended to give detailed description of the SST. The first section of this chapter will give a description of the physical structure and design of the detector. This will be followed by a section that will give a definition of terms that will be used in describing the reconstruction of the tracker. Finally a description of the track reconstruction process is given.

3.1 The Silicon Strip Tracker structure

The Silicon Strip Tracker (SST) is part of the all silicon Tracker. It is designed to track ionising particles in the 4T magnetic field of CMS. The detector modules of the SST are laid out in layers of cylinders and disks. The cylindrical layers are named the Tracker barrels and the disks are named Tracker Inner Disks and Tracker End Caps depending on their location along the z axis. The whole of the SST structure is supported by aluminium framework.

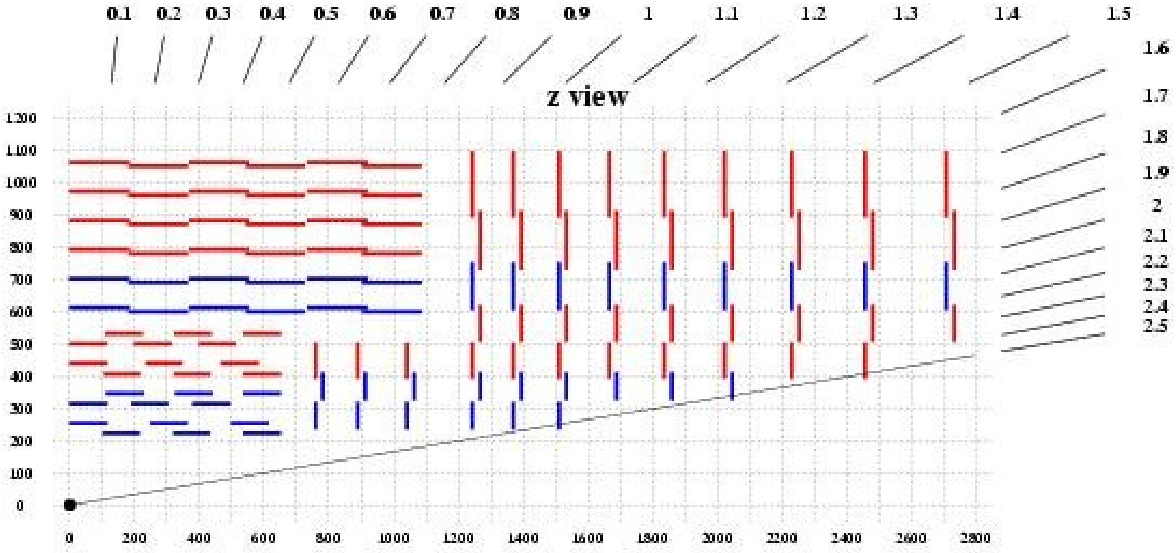

Figure 3.1: Cross section layout of one quadrant of the CMS Silicon Strip Tracker. The dimensions are in mm.

The SST is divided into two equal parts along the z axis at $z = 0$ making the $-/+$ sides of the Tracker. It is also divided across the ϕ plane along the z axis at the $x = 0$ and $y = 0$. Figure 3.1 shows a quarter of the SST. The barrel part of the SST has two sections, the Tracker Inner Barrel (TIB) and Tracker Outer Barrel (TOB). The disks which cover the TIB cylinders are called the Tracker Inner Disks (TID) and the ones covering the TOB are the Tracker End Caps (TEC) [19].

3.1.1 The Silicon Strip Detector Module

A silicon strip detector module consists of an active area made of silicon, a support system made of carbon fiber, cooling pipes and readout electronics. The term detector modules used in this chapter refers to a system consisting of all of the parts mentioned above.

The Silicon Microstrip Sensor:

The active part of the detector module, which will be referred to as the sensor in this chapter, is produced from n-type silicon using the standard industrial process on 6 inch wafers. Although different sensors used in the SST have different sizes, shapes and resistivity (1.5 - 3.0 kΩ cm for the thin detectors, and 3.5 - 7.5 kΩ cm for thick detectors) they have crystal lattice orientation of $< 100 >$ [20]. The various types of the sensors for the SST have been listed in Tables 3.1 and 3.2. This crystal makes the bulk of the of the sensor where the p^+ strips are implanted. These strips make the sensor a position sensitive detector which can provide a 1D position on its surface with accuracy that depends on the pitch size. The pitch is defined as the dead area between two strips (Figure 3.2). The strips have pitch sizes that vary between 80 and 205 μm, but the strips width to strip pitch ratio is always 0.25 for all types of detectors [10]. Figure 3.2 shows a schematic drawing of the silicon sensor.

The process of making the sensor's p^+ strips starts by growing oxide (SiO_2) on the crystal. Trench like depressions are then cut in the oxide where the p^+ is implanted making it a sensitive area. A layer of aluminium is then places on top of the p^+ to provides high voltage supply for bias and it also provides connection for the readout electronics. Detailed description of the process can be found in [21] and [13].

The irradiation of the sensors is inevitable due to the environment they are used in and this affects the sensors in two ways. The first one being the surface damage which

Type	Length 1(mm)	Height (mm)	Pitch (μ m)	Strips	Multiplicity
TIB1	63.3	119.0	80	768	1536
TIB2	63.3	119.0	120	512	1188
TOB1	96.4	94.4	122	768	3360
TOB2	96.4	94.6	183	512	7056

Table 3.1: Inner barrel 320 mm thick sensors and outer barrel 500 mm thick sensors, geometrical dimensions and multiplicities [10]

Type	Length 1 (mm)	Length 2 (mm)	Height (mm)	Pitch (μ m)	Strips	Multiplicity
W1 TEC	64.6	87.9	87.2	81-112	768	288
W1 TID	63.6	93.8	112.9	80.5-119	768	288
W2	112.2	112.2	90.2	113-143	768	864
W3	64.9	83.0	112.7	123-158	512	880
W4	59.7	73.2	117.2	113-139	512	1008
W5a	98.9	112.3	84.0	126-142	768	1440
W5b	112.5	122.8	66.0	143-156	768	1440
W6a	86.1	97.4	99.0	163-185	512	1008
W6b	97.5	107.5	87.8	185-205	512	1008
W7a	74.0	82.9	109.8	140-156	512	1440
W7b	82.9	90.8	90.8	156-172	512	1440

Table 3.2: Geometrical dimensions and multiplicities for 320 mm thick (W1W4) and 500 mm thick (W5aW7b) wedge sensors for TID and TEC: W1 has two different versions for TID and TEC, whereas the TID shares identical W2 and W3 sensors with the TEC [10]

increases the inter-strip capacitance as this affects the oxide-silicon interface charge. The sensors have capacitances which arise naturally. The inter-strip capacitance is responsible for the inter-strip cross-talk. The second effect is the bulk damage (n-type silicon) which causes the bulk type inversion [22], [23].

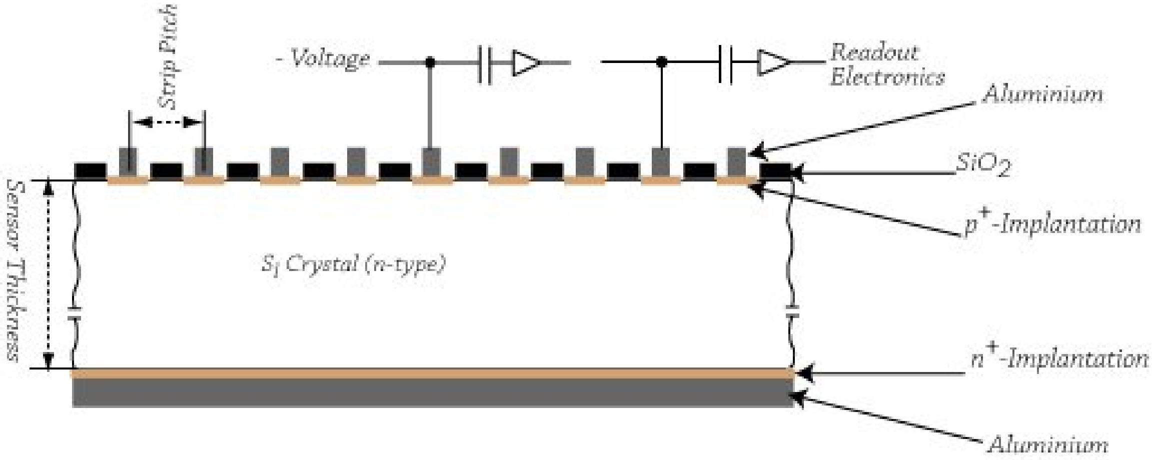

Figure 3.2: A schematic drawing of a silicon micro-strip sensor cross section [13]

The readout electronics:

When a charged particles cross the sensor along its thickness, its ionisation produces potential difference in the silicon sensor and detectable current flows to the strips that are near the interaction point of the incident particle. This current known as signal is picked up by the readout electronics, amplified and stored. The modules read the analogue signal and upon receiving a Level one (L1) trigger the information is sent to the Optohybrid electronics which convert the electric signal to an optical signal that it is subsequently transferred to the front end drivers (FED). The FED converts the optical signal back to electrical signal and furthermore convert the analogue to a digital signal.

The strips of the silicon sensors are read by custom made VLSI chips known as Analogue Pipeline Voltage (APV25) (Figure 3.3). These are deep submicron readout chips that are designed in a 0.25 micron CMOS process to increase the radiation resistance of the CMOS. The APV25 comes with 128 readout channels each of which takes a reading from a single silicon strip. Every channel consists of low noise charge preamplifier, analog gain inverter, 50 ns CR-RC shaper, 192 cell analogue pipeline and analogue pulse shape processor. The signal from the silicon strips comes as charge. A minimum ionising particle passing through a 300 μm silicon detector generates around 25,000 electron hole pairs [24], so the signals arriving at the APV25 inputs are single impulses of current. Upon reaching the channel's preamplifier the signals are amplified and transformed to voltage. The voltage then reaches the 50 ns CR-RC shaper after going through the selective inverter so that its polarity is always positive. The signal is then put to the pipeline.

The APV25 has two types of readout, the peak and the deconvolution modes. The peak mode samples one signal per event which is reserved in the pipeline for readout. This mode is used when the data rate is low where the effects of pile-up of detector signals are not significant. The deconvolution mode is used when data rates are high enough to make the effect of pile-up significant. In this case three samples are taken each at 25 ns rate and reserved in the pipeline for readout. during the readout the a deconvolution process is performed on the signals by the Analogue Pulse Shape Processor (APSP) part of the APV25 chip to determine at which time slot the original signal occurred [25].

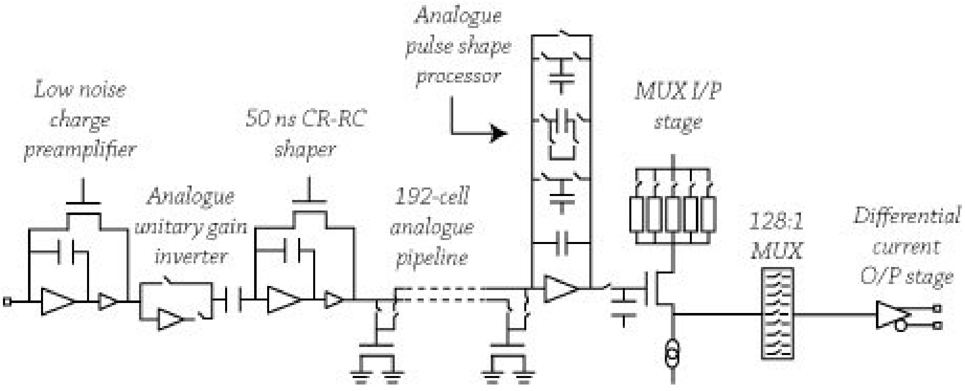

Figure 3.3: Block diagram of one of the APV25 Channels [24]

Support and cooling system: The silicon strip sensors are glued to carbon fiber support system. The readout electronics of the module are also placed on the support system.

The barrel detector modules are arranged with their strips parallel to the beam pipe and the forward modules are placed with the strips perpendicular to the beam pipe. This enables the modules to give a particle position in the 2 dimensional plane of the $r - \phi$. To obtain a 3 dimensional space position of a particle hit, some of the detector modules have been modified to contain two silicon sensors glued back to back. One of the modules is rotated 100 mrad with regards to the z axis. This gives a position of the particle in the $r - z$ plane also. These are called stereo modules.

Depending on their place in the SST the detector modules have various sizes, shape, silicon sensor thickness (see Table 3.3), strip-pitch size and strip numbers. The modules in the barrel contain silicon sensors that are rectangular in shape where as the modules in the End cap are wedged. The list of the modules for each SST subsystem is given in Table 3.3. Figure 3.4 shows the different types detectors used in the SST.

	No. of Dets	Thickness(μm)
TIB	2724	320
TOB	5208	500
TID	816	320
TEC	2412	320
TEC(2)	3888	500

Table 3.3: The distribution of the silicon detector modules in the SST

3.1.2 The Tracker Inner Barrel (TIB)

The inner barrel is located immediately outside the pixel subsystem making it the closest part of the SST to the beam pipe. The TIB consists of 4 layers and each layer is made up of ladder like structured shells that contain the individual detector modules. The shells contain 6 detector modules for each half of the TIB. The TIB barrel is shorter than the TOB along the z axis. This is to stop particles with very shallow angle passing through the barrel detector modules.

The TIB detector modules have thickness of 320μm. The first two layers of the TIB consist of stereo modules whereas the other two are populated with single sided modules. The detector modules on the shells of the TIB are grouped in threes to make strings [26]. Strings are the smallest read-out and control units of the TIB sub-system. They consist

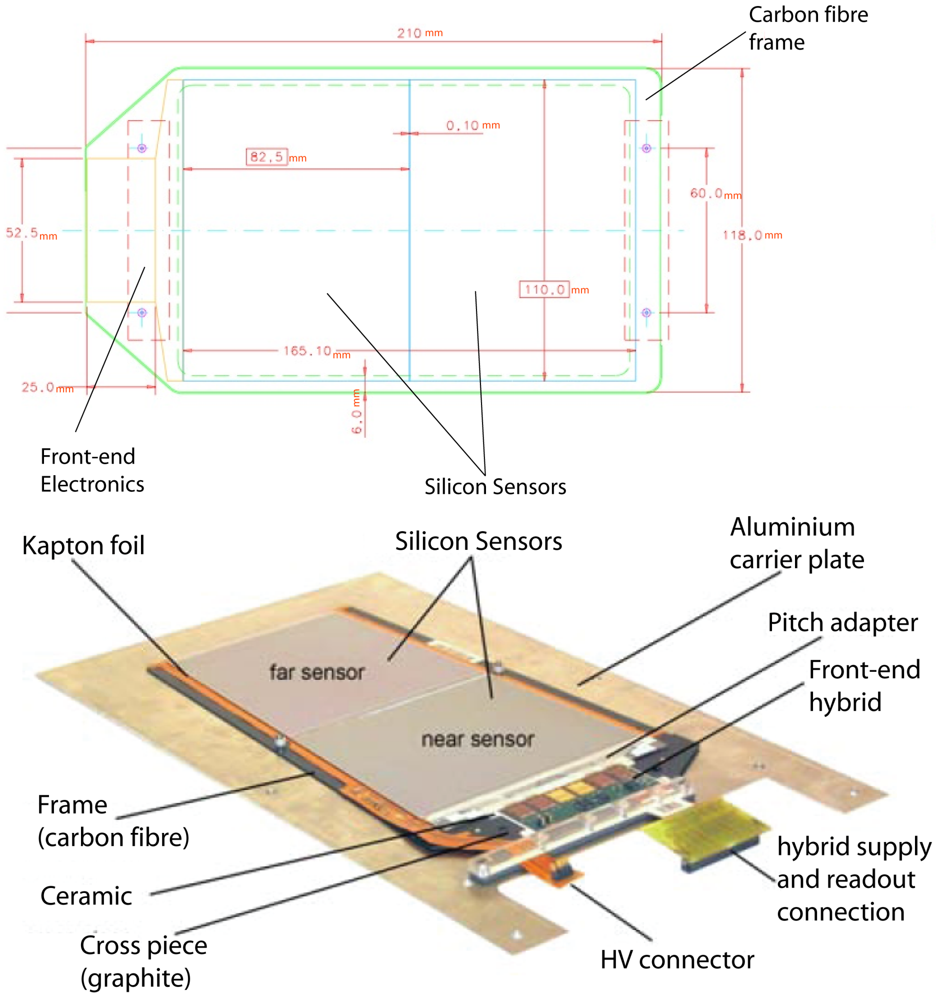

Figure 3.4: The shapes of detector modules for the Tracker Barrel (top) and tracker Disks (bottom) [7] and [4]

of three detector modules connected by a mother cable to a Communication and Control Unit (CCU). The mother cables distribute 40 MHz clock, trigger and communication from the CCU, and High and Low voltages to the detector modules. A schematic diagram of the string readout and control system can be seen on Figure 3.5.

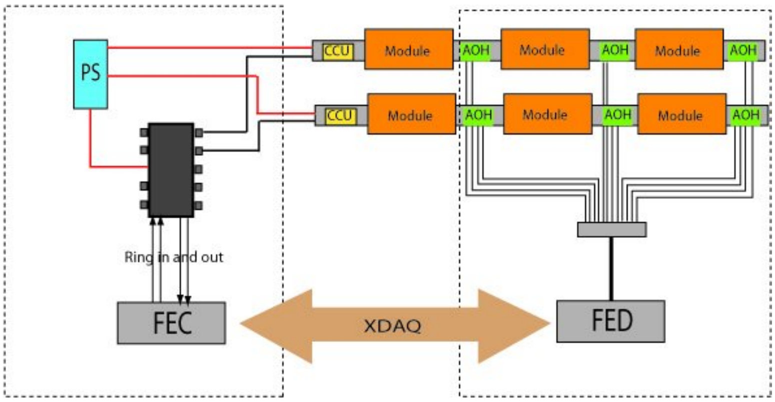

Figure 3.5: A schematic diagram of the arrangement of a string readout and control system of the TIB [26]

3.1.3 The Tracker Inner Disk (TID)

The TID disks are located in the gap between the TIB and the TOB in the z region (see Figure 3.1). The TID consists of 3 carbon fiber disks which house the detector modules [27]. These detector modules are used to track the forward particles. Forward particles are the particles that come from the interaction point at around $|\eta| > 1.5$. The barrel detector modules cannot be used to measure these particles as the impact of these particles will be very shallow to the modules on a parallel plane to the beam. Hence the forward detector modules on the the disks are arranged with the strips of the detector modules perpendicular to the beam pipe and they are arranged in petal like structures to cover the dead area of the detector modules [18].

3.1.4 The Tracker Outer Barrel (TOB)

The TOB consists of 6 layers with detector modules that contain silicon sensors of thickness 500 μm. the first tow layers are populated with stereo detectors. The detector modules are arranged in groups of six on a rods. Each rod contains all necessary services such as the power, cooling and readout electronics for 6 or 12 silicon detector modules. All rod components are contained in an envelope of $159 \times 1130 \times 22$ mm^3, except the four supporting spheres that stick out laterally in correspondence with the two disks of the wheel. The rods are supported by four discs made of carbon fiber where the rods are inserted. The structure that supports the rods is a single mechanical structure (wheel). In total there 688 rods are housed by the wheel.

3.1.5 The Tracker End Cap (TEC)

The TECs on both ends extend radially from 220 mm to 1135 mm and from $\pm\,1240$ mm to $\pm\,2800$ mm along the z-direction. The two endcaps are called TEC+ and TEC- (according to their location in z in the CMS coordinate system). Each endcap consists of nine disks that carry substructures on which the individual detector modules are mounted plus an additional two disks serving as front/back termination. The design of the TEC has an opening that allows the insertion of the pixel detector. The disks are Carbon Fiber Composite (CFC) / honeycomb structures. The detector modules of the TEC are mounted on substructures called petals like the TID to compensate for the dead zones of the detector module. The petal support structures are mounted on to the Disks. The disks in turn have rings where rings 1, 2 and 5 are built up of stereo modules. Petals can be individually removed from the endcaps without uncabling and/or disassembling the entire structure.

3.2 Definition of terms used in track reconstruction

The following terms are used when describing the track reconstruction process.

3.2.1 Tracks

A track is the path of a particle travelling through the SST. In the CMS all particles with the exception of cosmic rays will originate from the center of the tracker ($\sigma_x = \sigma_y = 15$ μm and $\sigma_z = 53$ mm [8]) where the *pp* interaction is expected. Only charged particles leave a track in the SST due to their ionisation properties. The tracks of particles are reconstructed using individual signals left in the detector modules of the SST. These signals are known as hits (Section 3.3.1).

In the CMS a track is defined using five parameters [28]: D_0, Z_0, ϕ, $\cot\vartheta(\eta)$ and P_t. There are further track parameters that contain information which are useful to monitor the quality of the reconstructed tracks. These are the χ^2, number of reconstructed hits and track angles. These parameters are discussed in the following sub sections.

A charged particle travelling in a uniform magnetic field of a solenoid such as the CMS, forms a helical path along the magnetic field. This helical path of a particle (track) can be projected on to two planes, the transverse plane where the azimuthal angle (ϕ) is measured in the $x - y$ coordinate and the longitudinal plane where the θ is measured between the track and the z axis (the beam pipe). Due to the helical shape of a track when the it is projected on to the transverse plane, the track assumes a circular path. If the track is projected on the longitudinal plane it assumes a bending path due to the magnetic field. These are very important characteristics of the helical track, as they are used in the calculation of the track momentum.

3.2.2 Impact Parameters

The track impact parameter or distance of closest approach is the calculated distance between a track and the point of interaction. The transverse impact parameter (D_0) is defined as $D_0 = y_{imp}cos\phi_0 - x_{imp}sin\phi_0$. The D_0 resolution is affected by multiple scattering in the innermost Pixel detector layers and the precision of the impact point is limited by the Pixel hit position resolution. the resolution of the z_{imp} resolution is approximated by $\sigma_z\sqrt{r_1^2 + r_2^2}/(r_2 - r_1)$, where σ_z is the pixel hit resolution and r_1, r_2 represent the radii of the pixel barrel layer.

3.2.3 Transverse momentum

The helical path of a charged particle in a uniform magnetic field along the longitudinal plane makes it possible to project a circular path on the transverse plane. The circular path projected on to the transverse plane makes it possible to calculate the transverse momentum of the particle. The transverse momentum of the charged particle is a component of the particle momentum. Figure 3.6 shows the transverse momentum of a particle originating from the collision point of two particles.

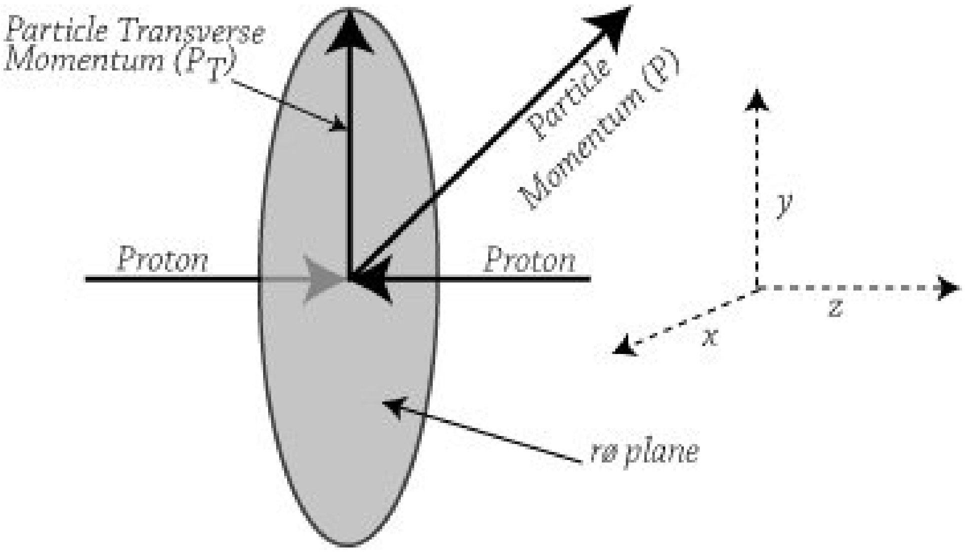

Figure 3.6: The transverse momentum of a particle seen from the longitudinal axis (z)

During the reconstruction of a track (Section 3.3) the transverse momentum of the particle is calculated iteratively. To predict the transverse momentum one has to predict the circular path of the particle first. To predict the circular path three points are needed. Once these three points have been found the path of the track is predicted and using the radius of the circular path of the track one can calculate the transverse momentum (P_t) and it can be calculated as follows:

$$P_t = rqB \tag{3.1}$$

where r is the radius of the circular path, q is the charge of the particle and B is the magnetic field. The radius of the track is calculated as:

$$r = \frac{L^2}{8s} \tag{3.2}$$

where L is the length between the first (P1) and the last points (P3) used in the path prediction and s is the sagitta [29], see Figure 3.7.

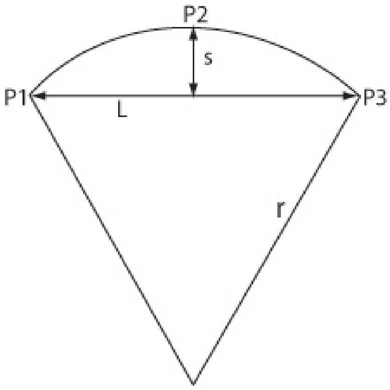

Figure 3.7: A section of the predicted circular path of a particle

3.2.4 Track Angles

In terms of the global representation, tracks are described by the angles ϕ for the $r\phi$ region and θ angle between the track and the z axis which is the beam pipe. ϕ also known as azimuthal angle,and describes a track position on the $x-y$ plane.

3.2.5 Track pseudorapidity

In the proton proton collision the mass of a proton is not equally distributed in the whole particle as it consists of gluons and quarks. This makes the use of the track angle θ less useful and a spatial coordinate, the pseudo-rapidity (η) is used. η is a function of the angle θ which is the measure of the of the track angle with respect to the z axis and is given by:

$$\eta = -\log\left(\tan\frac{\theta}{2}\right) \tag{3.3}$$

3.2.6 Lorentz Angle

For a silicon sensor operating in high magnetic field, the drifting ionization (generated by traversing particles) path is deflected significantly by the Lorentz force $e\overrightarrow{v} \times \overrightarrow{B}$, where $\overrightarrow{v}$ is the drift velocity and $\overrightarrow{B}$ is the magnetic field. The angle of deflection caused by such force in a magnetic field perpendicular to the electric field is known as the Lorentz angle (Θ_L) and is given by Equation 3.4.

$$\Theta_L = \frac{\Delta x}{d} = \mu_H B = r_H \mu B \tag{3.4}$$

Where Δx is the shift of the signal position, d is the drift distance along the electric field. μ_H is the Hall mobility, which is the drift mobility in a magnetic field. The drift mobility without magnetic field is given by μ. the relationship between the two is given by

$$r_H = \frac{\mu_H}{\mu} \tag{3.5}$$

where r_H is the Hall scattering factor by which the Hall mobility differs from the conduction mobility. This factor describes the influence of magnetic field on the mean scattering time of carriers of different energy and velocity. In room temperature the Hall scattering factor has the value of 1.15 for electron and 0.7 for holes. This factor tends to 1.0 for both electron and holes with decreasing temperature [30], [31]. Figure 3.8 shows a diagram of an experimental setup used to estimate the Lorentz angle.

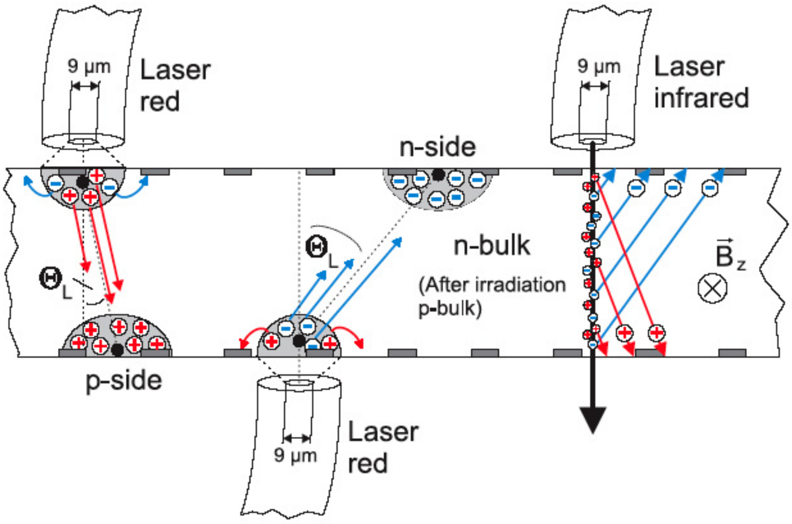

Figure 3.8: Experiment based on Karlsruhe setup for measuring the Lorentz angle of holes and electrons. It is equipped with three fibers delivering laser light to the silicon. The red lasers have a penetration depth of a few μ m. Using the laser pulse on the p-side and n-side side one can measure the drift of the electrons and holes, respectively. The infrared laser generates charge throughout the whole thickness of the silicon. This can be a simulation of a through going particle [32].

The Lorentz angle of drifting ionization has a direct effect on the arrangement and alignment of the silicon sensors in a detector. The Lorentz shift can reach up to 200 μm for electrons in a 300 μm thick silicon detector, which means this side effect must be calibrated out. It can be seen in Figure 3.8 that the Lorentz angle is significantly smaller when the carrier drift is from the n^+ to the p^+ compared to the reversal, showing that a better resolution can be achieved by using the p^+ side for the rϕ position measurement of a traversing particle. In the barrel Tracker the sensor strips are parallel to the magnetic field and their measurement resolution is affected by the lorentz drift. The barrel modules of the SST are tilted by 9^0 along the z axis to compensate for the Lorentz angle [18]. In the case where the strips of the sensor are perpendicular to the magnetic field, like the arrangement of the disk modules, the Lorentz shift does not affect the measurement resolution as the shift is along the strips.

3.2.7 Multiple scattering

A charged particle passing through matter interacts with the constituents of the matter causing its path to be deflected by many small-angle scatters as can be seen in Figure 3.9. This deflection caused mainly by the Coulomb scattering from the nuclei is called multiple Coulomb scattering. For small angle multiple scatters the distribution is roughly Gaussian [2]. For larger scatter angles, this behaves like Rutherford scattering angles giving larger tails than a Gaussian distribution. The scatter angle of a particle in space is given by:

$$\theta_0 = \frac{13.6\ \mathrm{MeV}}{\beta cp} z \sqrt{\frac{x}{X_0}}[1 + 0.038 ln(\frac{x}{X_0})] \tag{3.6}$$

Where p is the momentum, βc is the velocity and z is the charge number of the incident particle. x/X_0 is the the thickness of the scattering matter in radiation length.

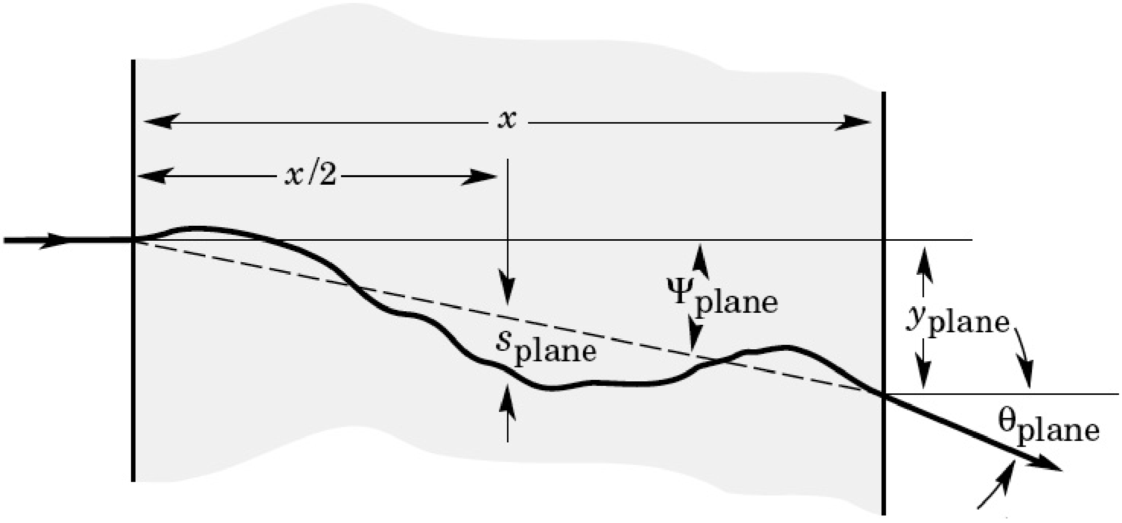

Figure 3.9: A diagram showing the passage of particle through matter and the quantities used to describe multiple Coulomb scattering [2].

3.2.8 Energy loss

The energy released in a detector is what enables the detection and identification of particles. Moderately relativistic charged particles traversing matter loose energy primarily by ionization and atomic excitation with the exception of electrons. This unfortunately is an unwanted side effect experienced on the tracker. Ideally a tracker would be made of matter that would enable a particle to be tracked without it loosing energy. This is not a possibility in real life as the energy loss is what enables the particle to be detected, therefore the lost energy is calculated and accounted for in the equations that estimate the path of the particle. The energy lost by a particle in a given matter can be calculated using the Bethe-Bloch-formula [2]:

$$-\frac{dE}{dx} = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta_\gamma)}{2} \right] \tag{3.7}$$

Where Z and A are atomic and mass numbers of the material, m_e the electron mass, I an effective ionization potential ranging from 13.5 eV in hydrogen to 1 keV in lead, T_{max} is the maximum Kinetic energy which can be imparted to a free electron in a single collision and $\delta(\beta_\gamma)$ is the Density effect correction to ionisation energy loss.

3.3 Track Reconstruction

Hits of around 5x10^3 (5x10^4) per event of the LHC in low (high) luminosity are expected at the CMS Tracker. This calls for a track reconstruction algorithm that is efficient in pattern recognition and fast in its search for hit propagation to the next layer. To deal with these tasks, the hit finding was simplified by arranging the detector modules in layers so that they are hermetic for particles originating from the center of the detector. The propagation of trajectories takes advantage of the fact that the magnetic field is almost constant in a large part of the tracker volume.

The track reconstruction in the CMS Tracker is done in the following five steps:

1. Hit Reconstruction
2. Seed Generation
3. Pattern Recognition or Trajectory Building
4. Ambiguity Resolution
5. Final Track Fit

This method of track reconstruction can be seen as having two separate sections. The first part is involved with measuring the hit positions in the tracker detector and the organisation of these hits in groups believed to be originating from the same particle called "track candidates". These are the first two from the list above. And the second part is on track fitting and filtering which optimally estimates a set of parameters that are used to uniquely describe a Tracker which is a process that involves the rest of the list above [33]. These are discussed in detail below.

3.3.1 Hit Reconstruction

The term hit, when used in the track reconstruction, means the weighted average (centroid) of a cluster of strips or pixels. When a particle passes through the sensitive part of a detector module, analogue signals are produced by strips or pixels close to the interaction point of the sensor. Depending on the particle and its angle of incidence signals with varying pulse heights are generated from the strips. This gives rise to two principal categories of reconstructed clusters: clusters with expected width of one strip and clusters with expected width of more than one strips. In the hit reconstruction process, the group of strips/pixels that are believed to be sharing a charge that belongs to one particle are clustered together. Once a cluster has been determined, the hit reconstruction process estimates its position and uncertainty (error). Based on this definition a cluster of strips is a reconstructed hit (recHit). This signifies that one or more particles have gone through that point of the detector module.

The process of clusterization (reconstruction of a cluster) starts by searching for a seed strip in the Silicon Strip Tracker (SST). This seed strip needs to have a signal to noise ratio of $S/N > 3$. The nearby strips are included to the cluster if they qualify for $S/N > 2$. Gaps are allowed in a cluster for highly inclined tracks. For a cluster to be accepted the total signal size of the cluster must exceed 5 times the square-root of the sum of the rms-noise-squared of the individual strips inside it. The position of a cluster is taken as the centroid of signal height of the strips.

3.3.2 Seed Generation

A seed can be described as starting point for the prediction and reconstruction of a track. It defines the initial trajectory and parameters of a track. The prediction of the track parameters need to be very close to the real values of the fully reconstructed track as they are used in the linear fitting algorithm of the track. The uncertainties of the parameters need to be small, as large errors mean a wide window of search for hits during the track reconstruction.

Seeds can be obtained in two ways: externally, where detectors other than the tracker are used to find trajectories or internally where the search of trajectories starts from inside the Tracker. Although the external method is not used in the generation of seeds due to its poor quality estimation of the trajectory parameters, it is used to constrain the search area of seed hits. This makes it possible to reconstruct seeds only in the region of interest of the detector. The software that uses the algorithm to define this region is called the "TrackingRegion". This algorithm is an important feature in the online software. The TrackingRegion "specifies the direction around which the region is defined, the (signed) inverse transverse momentum range and the allowed position of the track impact point (vertex along the beam line and maximum allowed distance from vertex in the transverse plane and along the beam line)". (Section 6.4.1 of[3]). There are two implementations of the TrackingRegion; the "GlobalTrackingRegion" and "RectangularEtaPhiTrackingRegion".

Seed generation is computed in two levels. The first level is finding the hits and the second level is the estimation of the seed and its parameters.

Hit Finding

There are two types of hit finding that can be used; hit pair finding and hit triplet finding methods.

Hit Pair Finding: Hit pair are defined by two hits from two different layers, where one of the hits has larger radius than the other. The search for the hit pair starts by looking for a hit on the layer that is furthest from the the beam pipe out of the two hits (known as outer hit). The tracking region is used to locate the outer hit. Using the GlobalTrackingRegion is straight forward process as all the hits in the detector layer are used. In the case of the RectangularEtaPhiTrackingRegion the range of the direction of allowed hits is constrained using the η and ϕ. The range of the position is predicted analytically using the minimum allowed momentum, possible direction of the tracks and vertex constraint. The position uncertainties caused by the multiple scattering, hit errors and the non linear projection of the helix are taken in to account to widen the search window. Further constraints are placed on hits which is independent of the tracking region based on their position. The inner hit (second hit) should have a smaller (making it the inner hit) radius compared to the first hit. The analytical prediction of its position is computed taking the vertex constraint in to consideration [28].

Hit Triplet Finding: Hit triplet finding is an extension of the hit pair finding, as the process is based on adding a third hit to the pair. This of course includes the addition of a third layer different to the previously used to find the hit pairs. The search window for the hit in the new layer is defined using the position tolerance ($\Delta\varphi$ and Δz in the barrel and $\Delta\varphi\Delta r$ in the discs). The $\Delta\varphi$ window for the search is centred on the outer hit and the tolerance Δz (Δr) is centred on the line connecting the two hits. The value for the tolerance of $\Delta\varphi$ and Δz (Δr) is given as 0.03 rad and 0.03 (0.02) cm respectively [28].

Seed Generation

The minimal information requirement from the pixel detector when generating seed is a hit pair. Due to the obvious lack of ability to predict the circular path of the the helix track path which is projected on the ϕ plane, when using the hit pair the transverse momentum can not be computed. Transverse momentum, which is one of the five parameters needed to identify a track when using the pattern recognition algorithm. To solve this problem an additional constraint is added to the hit pair to enable the circular path, the track is assumed to pass through a known vertex or the center of the beam spot. The estimation of the seed parameters is an iterative process starting at the beam spot. The estimation uses the equation of ideal helix passing through the hit pair and the beam axis. This is then propagated to the next hit and is updated using the measurements and further propagated to the second hit for further update. The same process is used to generate seeds from triplets with the additional propagation and updated at the third hit.

Seed generation algorithm performance

Using the RectangularEtaPhiTrackingRegion implementation one can reduce the size of the hit search region. Figure 3.10 shows efficiencies of the outer hit searching algorithm as a function of P_t and η. The algorithmic efficiency is better than 99.6% and does not depend on either P_t or η. Figure 3.11 shows the algorithmic efficiency of hit pair finding for a given RectangularEtaPhiTrackingRegion of size $\Delta\eta$ x $\Delta\varphi = 0.2$ x 0.2. The efficiency does not depend on the track P_t but there are minor inefficiencies on the endcap region of less than 1%.

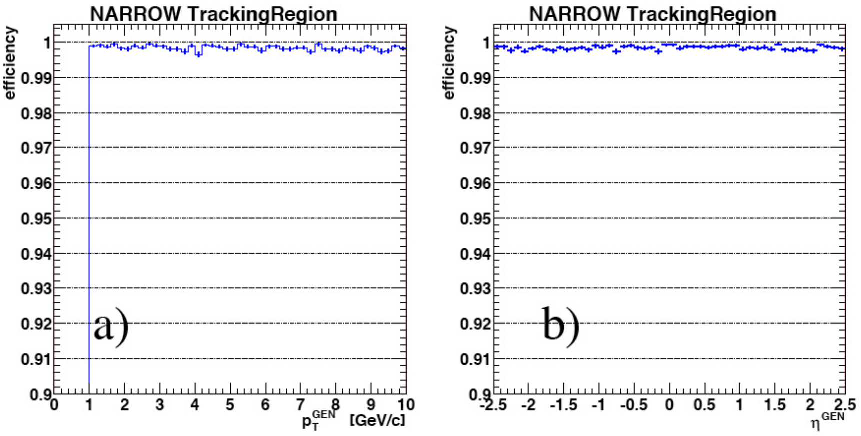

Figure 3.10: The outer hit finding efficiency for the RectangularEtaPhiTrackingRegion as a function of a) transverse momentum and b) pseudorapidity [34]

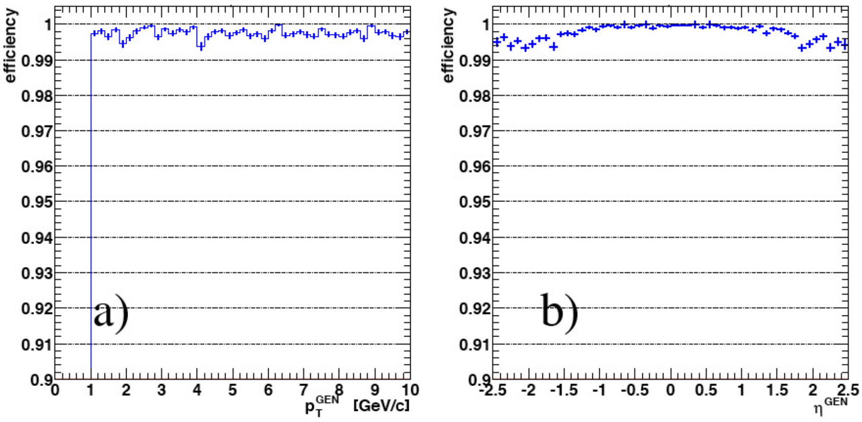

Figure 3.11: The efficiency of the hit pair finding algorithm for RectangularEtaPhiTrackingRegion of size $\Delta\eta$ x $\Delta\varphi$ = 0.2 x 0.2, as a function of a) transverse momentum and b) pseudorapidity [34]

The efficiency of hit triplet finding algorithm compared to all tracks can be seen on Figure 3.12. The efficiency can be seen dropping sharply for $P_t < 2.5$ GeV/c due to the lack of the multiple scattering correction in the algorithm. Comparison of the purity, number of pairs (triplets) and CPU time can be seen on Figure 3.13. The hit pair and hit triplet finding algorithms have similar advantages and disadvantages to that of the seed generation algorithms. Hit Triplets have fewer elements and they use a lot of computing power (CPU time) where as they have superior purity to hit pair. Hit pairs on the other hand have more elements which accounts for efficiency and they use less CPU time.

3.3.3 Pattern Recognition, or Trajectory Building

The pattern recognition or trajectory building part of the whole track reconstruction processes can be seen where track fitting and filtering meet. Track fitting is concerned with the estimation of track parameters. These parameters are carried forward from the seed generation process. This means that filtering is applied to fitting to achieve a better track fit. If a track is taken as a dynamic system then the five parameters that uniquely describe a track (discussed above) are fed in to the filtering process. The CMS Tracker has chosen to use a pattern recognition algorithm based on the Kalman filter and it is called the

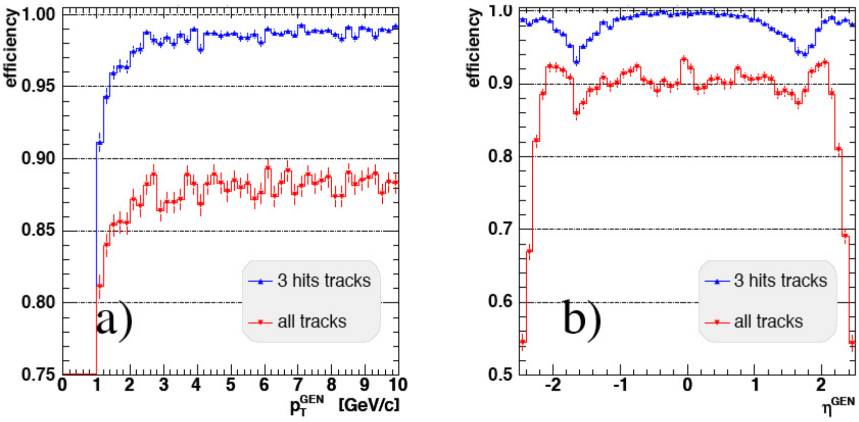

Figure 3.12: The efficiency of the hit triplet. The 3-hit tracks (algorithmic) and the absolute (all tracks) efficiency is plotted as a function of a) transverse momentum and b) pseudorapidity. The constraint of search region in the RectangularEtaPhiTrackingRegion of size $\Delta\eta$ x $\Delta\varphi$ = 0.2 x 0.2, has been implemented in the reconstruction [34].

Combinatorial Kalman filter (CKF) where the Kalman filter is used both for track fitting and filtering due to its recursive nature [28], [35]. The CKF starts from a track seed and it iterates through [36]:

- extrapolate all candidates to the next layer of the tracker
- find compatible hit for each candidate
- for each candidate create a branch with all compatible hits and a missing hit
- drop candidates with low quality. Typically candidates with high χ^2, too many missing hits and are a subset of another candidate

The Kalman filter is used in many applications in tracking the path of objects, eg. radars that track aircraft. The method is very useful when applied in tracking dynamic systems as it addresses the three main problems faced by dynamic systems, namely: filtering, prediction and smoothing. Filtering is used to estimate the present state of a vector using the past measurements, prediction is used to estimate the state of a vector in future and smoothing is the estimation of the vector state at some point in the past using the

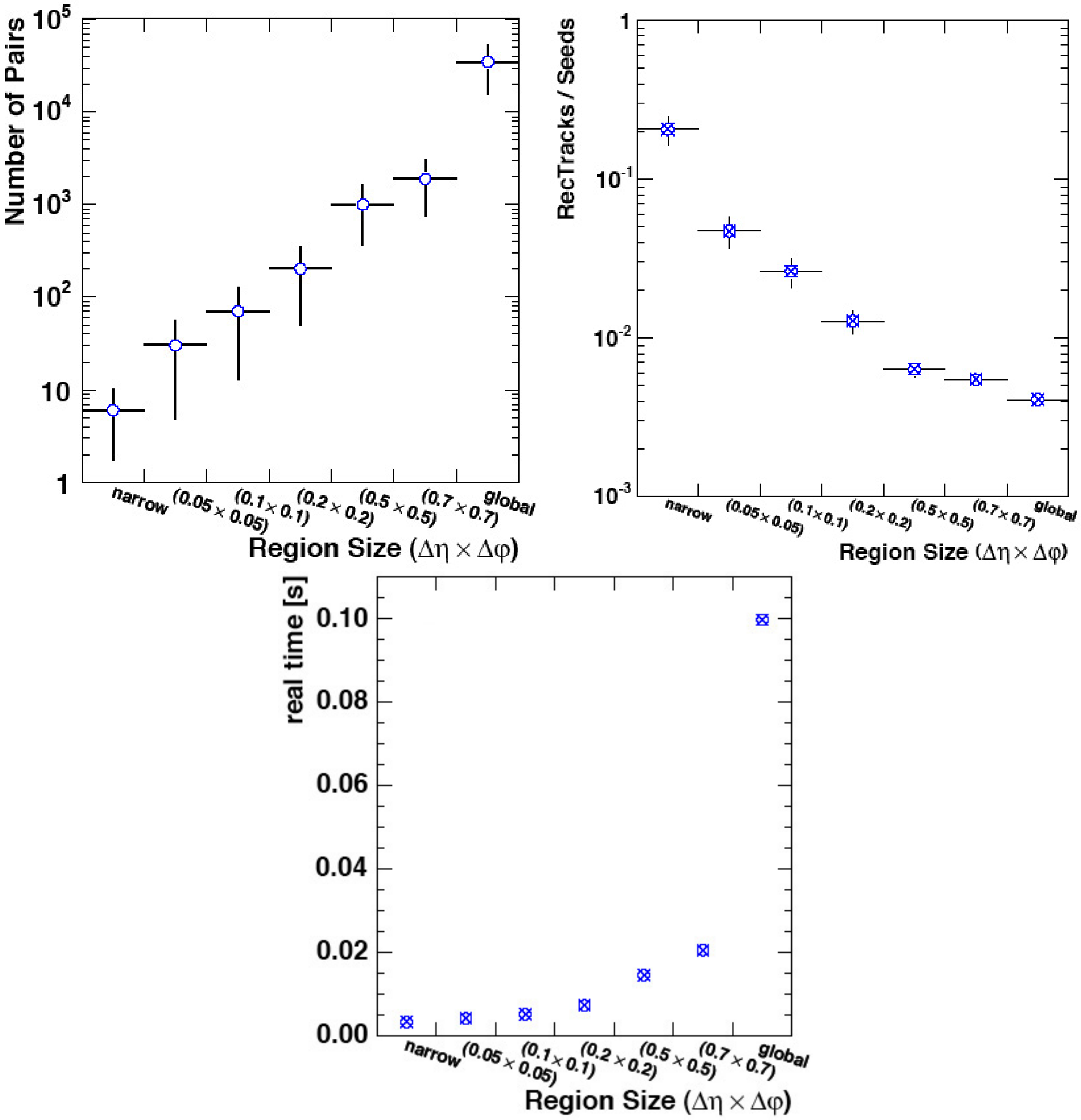

Figure 3.13: From left to right, top to bottom. Number of hit pairs reconstructed, Purity of the reconstructed Hit Pair and the real CPU time taken by the algorithm. (The region is defined by RectangularEtaPhiTrackingRegion 3.3.2) The events are $H \rightarrow ZZ \rightarrow 2e2\mu$ with high-luminosity pile-up [3].

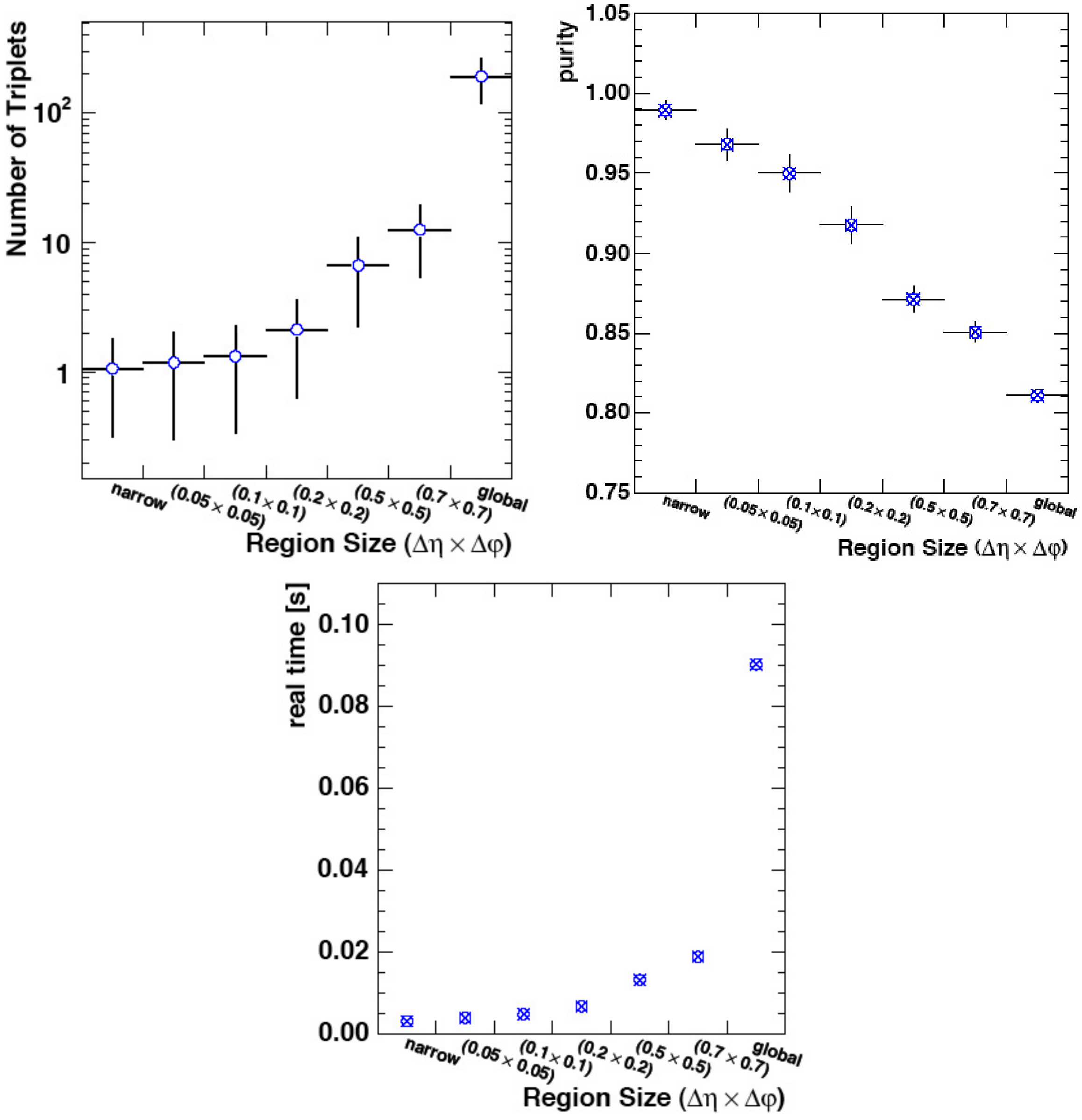

Figure 3.14: From left to right, top to bottom. Number of Hit Triplets reconstructed, Purity of the reconstructed Hit Triplets and the real CPU time taken by the algorithm. (The region is defined by RectangularEtaPhiTrackingRegion 3.3.2) The events are $H \rightarrow ZZ \rightarrow 2e2\mu$ with high-luminosity pile-up [3].

measurements taken up to present[37]. This algorithm is iterative and the precision of track parameters improves with each new measurement (hit on a new layer). The seed generation of the track reconstruction process is part of the Combinatorial Kalman filter [38]. The extrapolation of trajectory to the next layers is done according to the equations of motion of a charged particle in a magnetic field. This extrapolation, part of the dedicated system of the track reconstruction called navigation, takes into account the multiple scattering and loss of energy of particles as they travel through detector modules and the dead zone (support and cooling systems). When the trajectory is propagated to the next layer, new trajectory candidates are created as there may be more than one compatible hits in that given layer. In addition to this another additional trajectory is created using a fake hit to account for the case where a track did not leave a hit on the layer. This hit is called “invalid hit”.

Each trajectory is updated using the Kalman Filter formalism after a new hit is added to it. The updated trajectory is then used as the weighted mean of the combination of the trajectory predicted state and the hit. The weight comes from the errors of the predicted state and the trajectory. The error on the predicted trajectory state has a direct effect on the pattern recognition as this error affects the compatibility of the trajectory with the neighbouring hits. This is because the compatibility of trajectory with the nearby hits is determined by the error on its predicted state. The error on the hits contributes to the determination of compatibility as well. The trajectory at this stage is updated using the Kalman Filter formalism.

Trajectories predicted on the above process are then propagated to the next layer repeating the procedure above. This continues until either the outermost layer of the Tracker is reached or a stopping criterion for the algorithm is reached. To limit the exponential growth of the trajectories on each layer constraints are applied to cut (limit) the number. These cuts are based on the χ^2 of the tracks and the number of invalid hits.

3.3.4 Ambiguity Resolution

The following step is trajectory cleaning by resolving the ambiguities. The ambiguities in the track reconstruction arise due to the large number of mutually exclusive trajectory candidates. These trajectory seeds exist as they are composed to a large extent of the same hits. This causes a track to either be reconstructed from two different seeds or one seed may result in more than 1 trajectory candidate. To resolve this ambiguity a fraction of hits that are shared between two trajectories is used. The fraction of the hits is calculated as follows:

$$f_{shared} = \frac{N_{shared}^{hits}}{min(N_1^{hits}, N_2^{hits})} \tag{3.8}$$

where N_1^{hits} and N_2^{hits} are the number of hits in the first and second track candidates respectively. If the value of this equation exceeds 0.5 then either the track with the lowest number of hits or the track with the highest χ^2 (in the case where both tracks have equal number of reconstructed hits) is removed.

3.3.5 Track Fitting

Finally the trajectory or track is smoothed using a combination of the Kalman filter and smoother. This process takes into consideration all measurements of the track and an optimal estimate of the measurement points is obtained.

3.3.6 Track Reconstruction Performance of the CMS Tracker

The performance of the tracker in general can be evaluated using the study of track reconstruction efficiency and resolution. The studies into this and the results have been documented in [3], [39], [28]. The results from these studies yield similar results as shown below.

In [28] and [39], negatively charged muons and pions (generated by Monte Carlo simulation using particle gun) with transverse momentum of 1, 10 and 100 GeV/c were used. In these analysis default settings for pattern for pattern recognition, ambiguity resolution and fitting are used. It is important to note this information as this will be the basis for this analysis in this thesis. It is also important to note that the tracks are built using the pixel detector. The efficiency is evaluated by dividing the reconstructed tracks by simulated tracks. The following limits have been set for defining a suitable reconstructed tracks. It has to have at least 8 hits with no more than 1 hit missing on all given layers. Its transverse momentum is expected to be $P_t > 0.8GeV/c$

The global track efficiency can be seen on Figure 3.15. The global efficiency is defined as the track reconstruction efficiency for all tracks originating (having their vertex) from the beam pipe. It can be noted from these diagrams that the efficiency of the track reconstruction using the CKF gives about 98%. There are two points that should be noted here. 1) The track reconstruction efficiency drops drops in the $|\eta| < 0.1$ region of the tracker due to the gaps between the sensors in the barrel where the $-/+$ parts of it meet. 2) The reconstruction of pions is not as efficient that of the muons. This is due to the nuclear interaction with the tracker.

The second study into the efficiencies is looking at the resolution of the track parameters as described in this section. The resolution is defined as the magnitude of the Gaussian distribution of the difference (residual) between the reconstructed and simulated track parameters. Figure 3.16 shows the results of such test. Further study results of the Tracker performance can be found in [40].

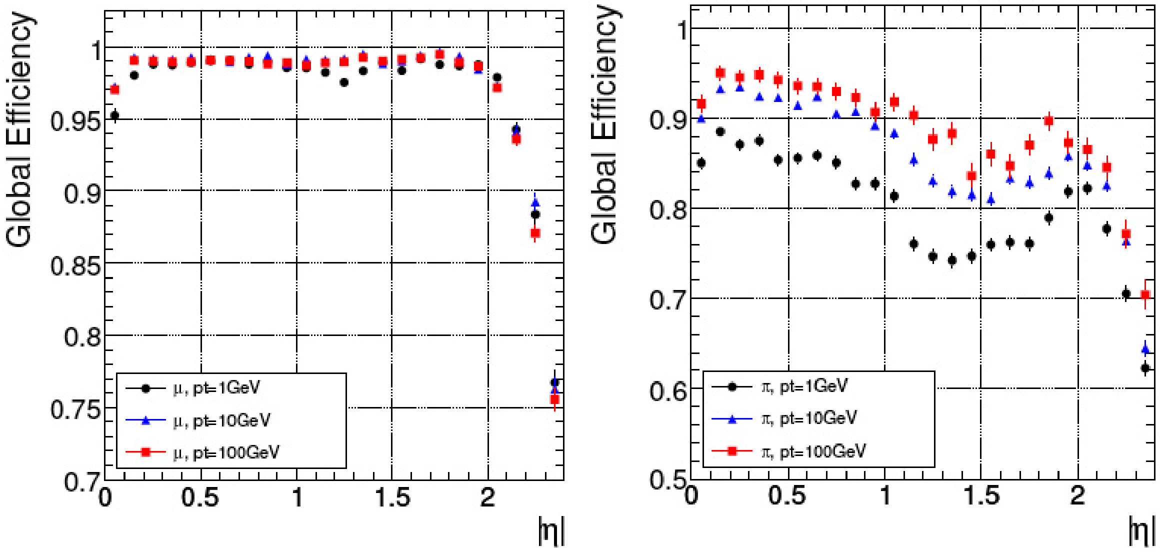

Figure 3.15: Global track reconstruction efficiency for muons (right) and pions (left) of transverse momenta of 1, 10 and 100 GeV/c [28].

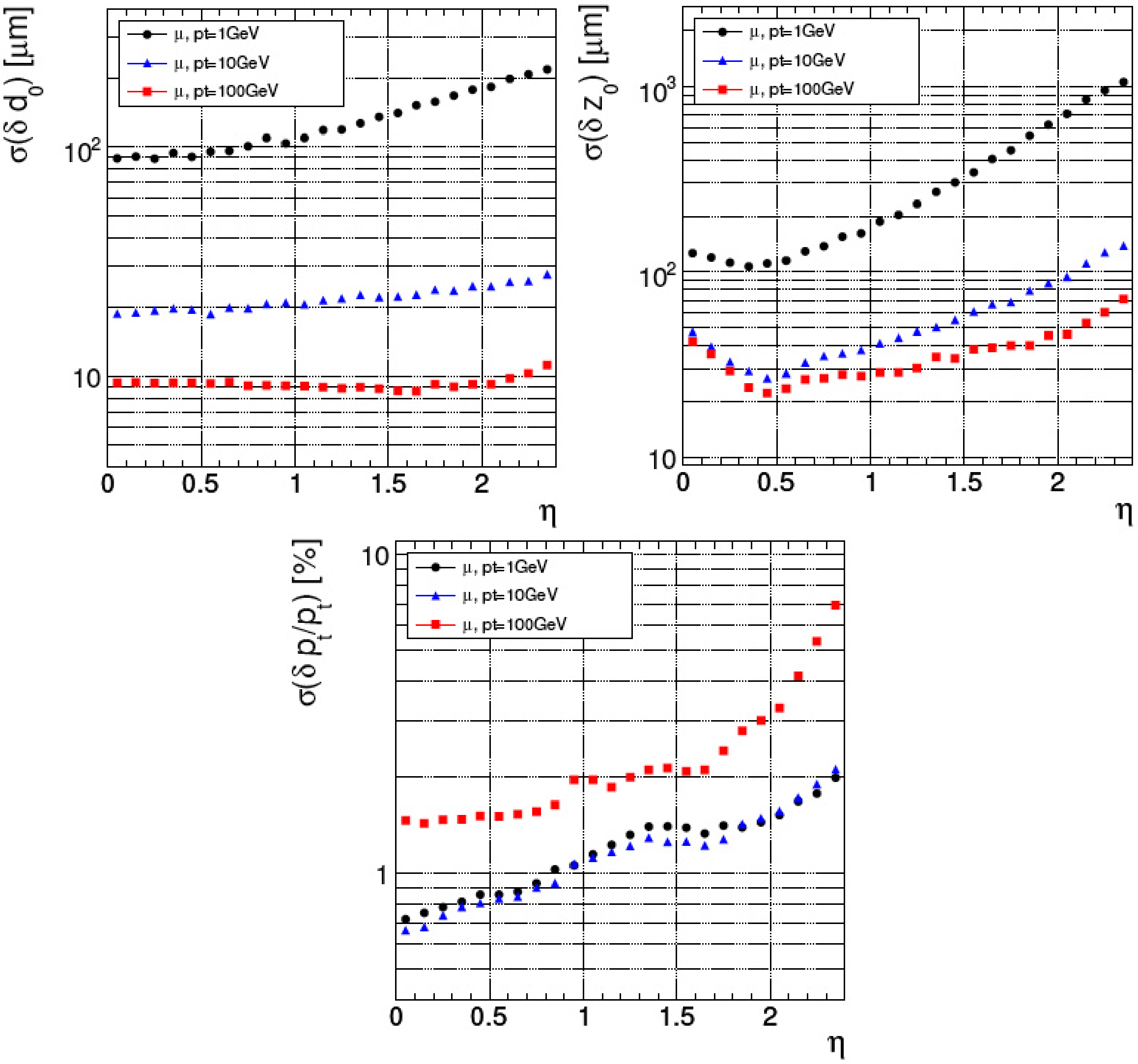

Figure 3.16: Resolution of three track parameters for single muons with transverse momenta of 1, 10 and 100 GeV/c. From left to right a)Transverse impact parameter b)Longitudinal impact parameter and c)Transverse momentum [28].

Chapter 4

The Data Quality Monitoring System

The Data Quality Monitoring (DQM) system is designed to monitor the behaviour of the detector and quality of the data being generated. It has two tasks, the first one being to identify faults in the hardware or software. The second task is to track the change of the detector performance over a long period of time. In the first case Monte Carlo generated data is compared to the acquired data; in the second case data collected some time in the past is compared with the data at hand. Both these data sets used check the quality of the acquired data will be referred to as "reference" data for the rest of this chapter. In both cases, statistical tests are used to compare the acquired data from the detector with the reference data to check if the detector or the reconstruction algorithms are behaving as expected.

This chapter will discuss my development of a DQM package, Tracker_Monitor_Track[1], in collaboration with the Tracker group at CERN. The first part of the package's name implies that it will only monitor Tracks reconstructed by the Tracker. This has recently been changed for the package to include tracks reconstructed by the Muon System after its successful testing during Magnet Test and Cosmic Challenge (MTCC, see Chapter 5). Although the package accesses the data generated by the pixel tracker and the Muon system, the monitoring of these data is not in the scope of this thesis.

[1] TrackerMonitorTrack is how the packages is named in the CMSSW framework, but it has been modified on this thesis for clarity

This chapter goes hand in hand with an implementation of a quality test tool based on the research documented in Chapter 6. This tool will be part of DQM_Services subsystem. DQM_Services provides interface for tools and services such as the ones provided by ROOT to the DQM processes. The two main areas of ROOT used by the DQM_services are the histogramer and some statistical tests. The work on the quality test is an additional software to help test Tracker_Monitor_Track. The DQM_Services system provides tools and services for the DQM packages. Details of these services and tools provided can be found in the physics and data quality monitoring web page [41]. The final section of this chapter describes a method for testing the developed package and the results achieved.

4.1 DQM Structure and data flow

DQM is one of the many sub systems of CMSSW (see Section 2.8). Its task is to provide a monitoring environment for the data-taking by the Data Acquisition (DAQ) system. DQM consists of several packages that facilitate the monitoring process. There are two modes on which DQM can run, they are known as “online” and “offline”. Online mode is when DQM packages are dealing with data coming straight from the detector and in most cases before the High Level Trigger (HLT) decision is taken. DQM is said to be running offline when it is dealing with data that has been stored after a HLT decision has been taken. The HLT was discussed in Chapter 2.

DQM, when running online, has three components [42]: a “Source” which provides monitoring data, a “Client” which analyses the monitoring data and a “Collector” which is the intermediate part where the information coming from the Source is channelled to the Client. A model of the DQM system is given in Figure 4.1. When running online, DQM runs its Sources on the Filter Unit (FU) machines which are part of the DAQ filter farm computing system. In this mode it is dealing with data at the rate of 100 kHz and the system’s processes are involved in the decision making of rejecting or accepting LHC events. Both the Collector and the Client run on worker node machines which are not part of the DAQ system. There is a discussion in CMS however to change this, so that Source and Client can run on the filter farm and Collector to be excluded. It is also discouraged to run DQM online and only monitoring process believed to be essential are to be run in this mode [43]. Some of these processes include monitoring the reconstruction algorithms of the HLT system. Although the change in the DQM structure will not affect Tracker_Monitor_Track, an overview of the planned new structure will be given later.

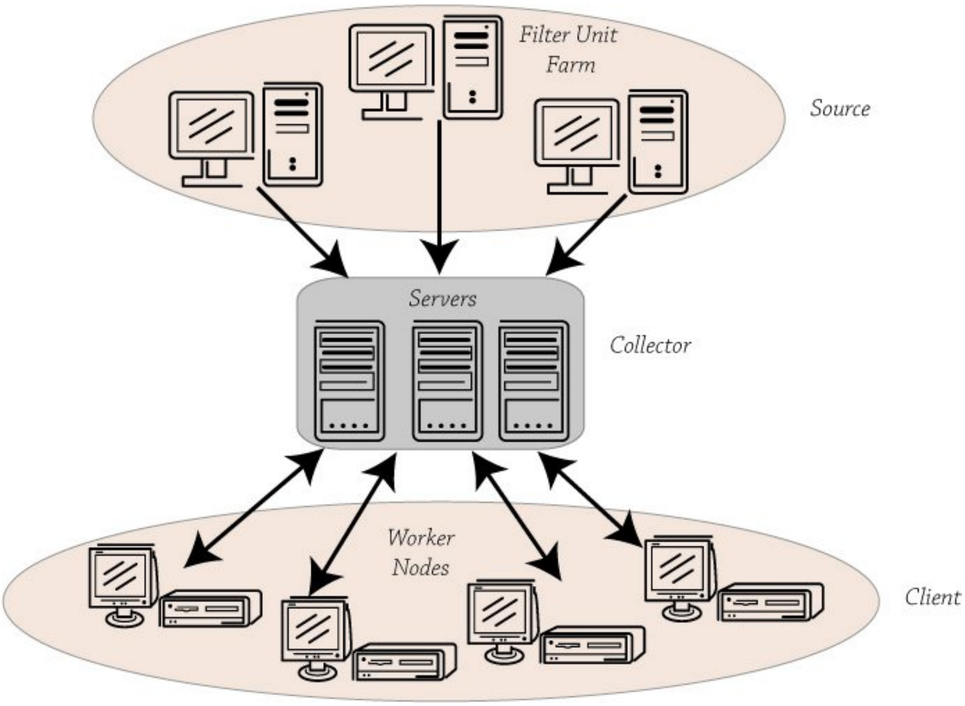

Figure 4.1: The Relationship between the three parts of the DQM. The Collector can not request information from the Source, but the Client can request Monitoring Elements information from the Collector.

4.1.1 Source

Source is a producer which reads the required information from the data produced by CMS and creates objects called Monitoring Elements (ME). MEs contain data which is used to assess the quality of reconstructed physics object. These can be track parameters or measurements taken using detector such as hit residuals of tracks on an individual silicon detector module. MEs are ROOT objects, which are usually histograms, and are analysed by Client modules. Histogram objects are created via the interface provided by the DQM_Services subsystem to ROOT. DQM_Services provides other additional tools and services for filling the monitoring histograms, dynamically modifying the MEs that have been already created and providing an interface to the Collector to update already created MEs. Using the DQM_Services tools MEs are either directly relayed to the Client modules via Collectors or they are saved in a ROOT file for later offline analysis without the need for Collectors.

The processes run by the Source module should be kept at the bare minimum to reduce interference with the other processes running at the DAQ filter farm. The HLT processes include triggering algorithms, calibration and reconstruction. For this reason all the CPU demanding processes of the DQM, such as statistical analysis are dealt by the the Client modules.

4.1.2 Collectors

Collectors are the “middle man” [3] between the Source and Clients. The reason for having this process is to prevent the producer from being affected by the failure of a downstream component in the DQM system [44]. This is important because the producer runs on the filter farm where the HLT processes are running to reduce accepted Level 1 trigger data rate by factor of 1000 putting pressure on the producer process to be reliable [43]. Reducing the effect of the various DQM components on each other increases the reliability as a failure on on one of the downstream components will have no effect on the Source. Two examples of what the Collector prevents from happening are: 1) in the case where a Client process crashes it would not cause the producer to hang waiting for response and 2) the Client processes can be very CPU demanding and direct communication between Client and producer can cause the producer to hang until the processes in the Client is finished.

The DQM system was set up so that a Collector can contact more than one Source but a Source can contact only one Collector at any given time. A Collector can only accept data from the Source but it is not possible to ask for data from a Source using the Collector. This avoids affecting the Source in the case of Collector failure. Source constantly updates the data on the Collector through services provided by the DQM_Services system.

The link between the Client and the Collector on the other hand is both ways, that is, Clients request information and the Collectors provide them. This way a Client can subscribe to MEs of a specific producer through the Collector, the Collector will then relay the information to the Client. Whenever the Source updates or modifies an ME the Collector notifies the Client of these changes, and the Client can then access them if needed.

4.1.3 Clients

The Client process is responsible for making MEs available to end users. These end users can be either humans (scientists studying the monitoring objects) or an automated statistical test which raises an alarm in the case where there is anomaly in the ME data. In the case of human end users the Client provides the end user with GUI and a choice of statistical tools available in DQM_Services [42].

When a Client process is running it can be in either listening or not listening mode. When it is in the listening mode it means that data being relayed from on the Source end of the DQM structure are notified to the Client and all the available MEs provided by through the Collectors. The Clients can then subscribe to the MEs that they need to process. The Clients can cancel or subscribe live MEs using the Collectors. Once an ME is subscribed by the Client the updates on the particular ME by the Source are relayed to the Client without the need to subscribe again. This is stopped only when the Client cancels its subscription.

4.2 Monitoring the Silicon Strip Tracker

DQM has many packages that monitor different aspects of the SST. These can be summarised as following [42]:

- Commissioning: this monitors the pre data taking adjustments, such as synchronisation and calibration.

- Digitised or raw Hits: the charge reading from sensors, their position and distribution is monitored to detect dead or noisy channels
- Reconstructed Hits: cluster charge and distribution of individual detector modules is monitored. These along with cluster size are studied with specific track attached (each hit is dependant on the overall track reconstruction), as clusters are affected by track angle and length.
- Reconstructed Tracks: This monitors reconstructed tracks. Tracker_Monitor_Track is responsible for this.
- Radiation damage parameters: quantities such as the Lorentz deflection, signal trapping and inter-channel coupling will be monitored to study the radiation damage on individual detector modules.

Tracker_Monitor_Track is a DQM package that provides monitoring for the SST tracks. It consists of two modules[2]: Monitor_Track_Global and Monitor_Track_Residual. Monitor_Track_Global provides information of the global parameters of a track. These parameters are the ones used to define a track (see Section 3.3) and a few others that are attributes of a track object. Monitor_Track_Residual provides hit residual (see Section 4.4.1) of tracks for each detector modules.

Figure 4.2 shows a top to bottom hierarchical structure of Tracker_Monitor_Track and its two modules in the CMSSW framework. The model also shows its link with the DQM_Services and Tracker subsystems, which is discussed later in this section. The link between DQM and DQM_Services is not through EDM, this does not break the laws set for inter-package communication in CMSSW discussed in Chapter 2. Service providers can be called using abstract class methods. A diagram of the classes that link the modules is given in Figure 4.3. The classes of the Tracker_Monitor_Track package inherit from a parent class EDAnalyser. This class provides an optional analysis method which can be modified.

The Tracker_Monitor_Track package depends on track reconstruction packages. The CMSSW framework does not allow direct dependence of modules upon each other, therefore

[2] Modules are program codes that run as plugins in CMSSW Framework (see Section 2.8).

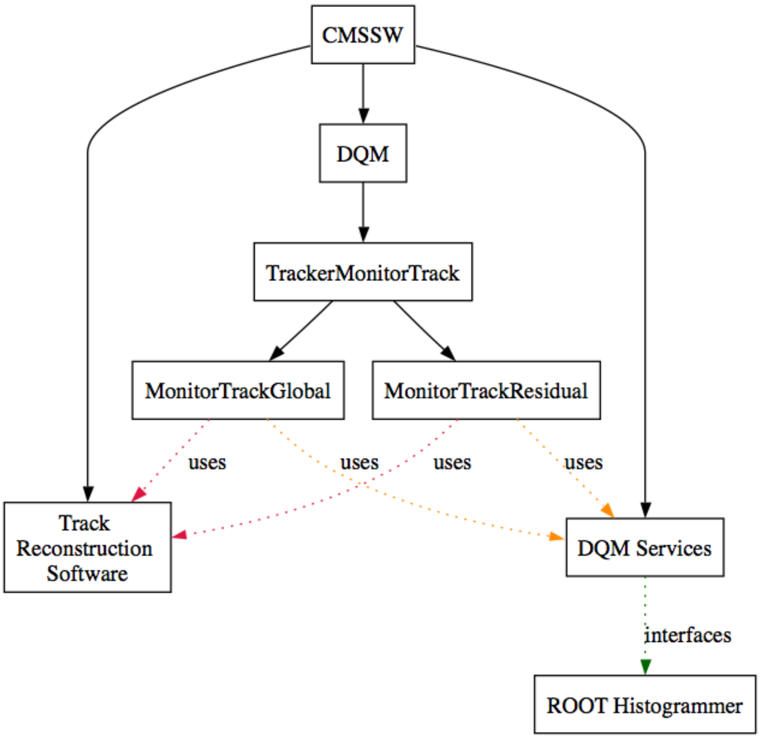

Figure 4.2: Diagram showing the hierarchy of the set up of Tracker_Monitor_Track package in the CMSSW structure. The boxes represent software modules (packages). This diagram also shows the services and tools Tracker_Monitor_Track uses through DQM_Services.

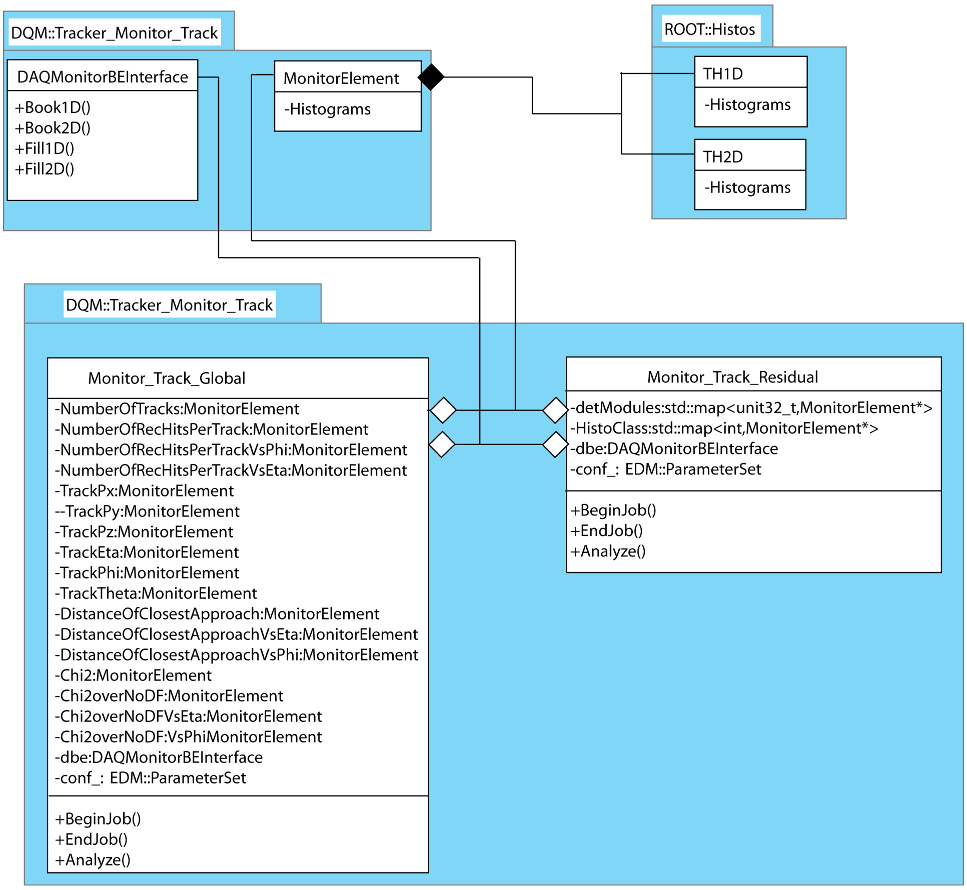
DQM::Tracker_Monitor_Track
DAQMonitorBEInterface
+Book1D()
+Book2D()
+Fill1D()
+Fill2D()
MonitorElement
-Histograms
ROOT::Histos
TH1D
-Histograms
TH2D
-Histograms
DQM::Tracker_Monitor_Track
Monitor_Track_Global
-NumberOfTracks:MonitorElement
-NumberOfRecHitsPerTrack:MonitorElement
-NumberOfRecHitsPerTrackVsPhi:MonitorElement
-NumberOfRecHitsPerTrackVsEta:MonitorElement
-TrackPx:MonitorElement
--TrackPy:MonitorElement
-TrackPz:MonitorElement
-TrackEta:MonitorElement
-TrackPhi:MonitorElement
-TrackTheta:MonitorElement
-DistanceOfClosestApproach:MonitorElement
-DistanceOfClosestApproachVsEta:MonitorElement
-DistanceOfClosestApproachVsPhi:MonitorElement
-Chi2:MonitorElement
-Chi2overNoDF:MonitorElement
-Chi2overNoDFVsEta:MonitorElement
-Chi2overNoDF:VsPhiMonitorElement
-dbe:DAQMonitorBEInterface
-conf_: EDM::ParameterSet
+BeginJob()
+EndJob()
+Analyze()
Monitor_Track_Residual
-detModules:std::map<unit32_t,MonitorElement*>
-HistoClass:std::map<int,MonitorElement*>
-dbe:DAQMonitorBEInterface
-conf_: EDM::ParameterSet
+BeginJob()
+EndJob()
+Analyze()

Figure 4.3: Class diagram of TrackerMonitorTrack package

this dependence exists through the data flow protocol governed by EDM. Figure 4.4 shows the interaction of various modules of different packages. In the figure it can be seen that any process that runs under the CMSSW framework needs a configuration file. This file is part of the CMSSW framework not the DQM and it defines the location of data (the data could be either stored in a local machine or on a distributed computing such as the GRID) that will be analysed by the data quality monitoring modules. The file also defines the parameter settings for the data analysis and the sequence in which the software modules are run. Once this is set up the process is run by calling the CMSSW run command.

Once running in CMSSW framework a process starts by getting Source data from the DAQ database system. The data is transformed into a CMSSW "event"[3], which is a ROOT file. CMSSW follows a strict format outlined by the EDM (see Section 2.8) when creating an Event. An Event contains data in groups that are called objects. These objects have three identification tags: producer name, label and type. Using these tags modules can access specific event objects. After the creation of an event, the modules that have been specified in the configuration file can run sequentially. The sequence by which the modules are run is very important as this can lead to the process crashing. For example Tracker_Monitor_Track is designed to monitor physics objects; tracks. During real-time data-taking ("online") process, the creation of CMSSW event will not have high level physics objects such as tracks. Hence the track reconstruction packages will have to be executed before the Tracker_Monitor_Track. As reconstruction packages are of the EDProducer type (see section 2.8), they are allowed to modify the CMSSW event creating physics objects inside CMSSW event. These objects are accessible to packages such as the Tracker_Monitor_Track which uses them to analyse the track object.

The other important thing to note when running modules is to know the identification tags of the objects that are needed by the modules. For example Monitor_Track_Global uses track objects. To access tracks from event it has to specify which object. An example would be, if the analysis is concerned with tracks that were reconstructed by the "Road Search" algorithm, then it is important to know the name of the object which in this case is "rsTrackwithMaterial". But this might not be enough, if there are more than one track

[3] The word event used in this chapter should not be confused with the LHC event, which is the bunch crossing at the interaction point of CMS

objects then it is essential to specify its type, if not the process will crash.

DQM can also be used in “offline” mode, where the reconstructed objects might have be saved in the “output” data base system discussed in Section 2.8. In this case running the reconstruction packages is not necessary, but if changes have been made (eg. the tracker geometry) then the reconstruction might be run again to apply the changes. In some cases objects created using different reconstruction methods than the one already in the event might be needed. In this case the desired reconstruction packages could be run to create new CMSSW objects. This was used in analysing different reconstruction algorithms on data from TIF (see Section 4.6) where two different algorithms are compared.

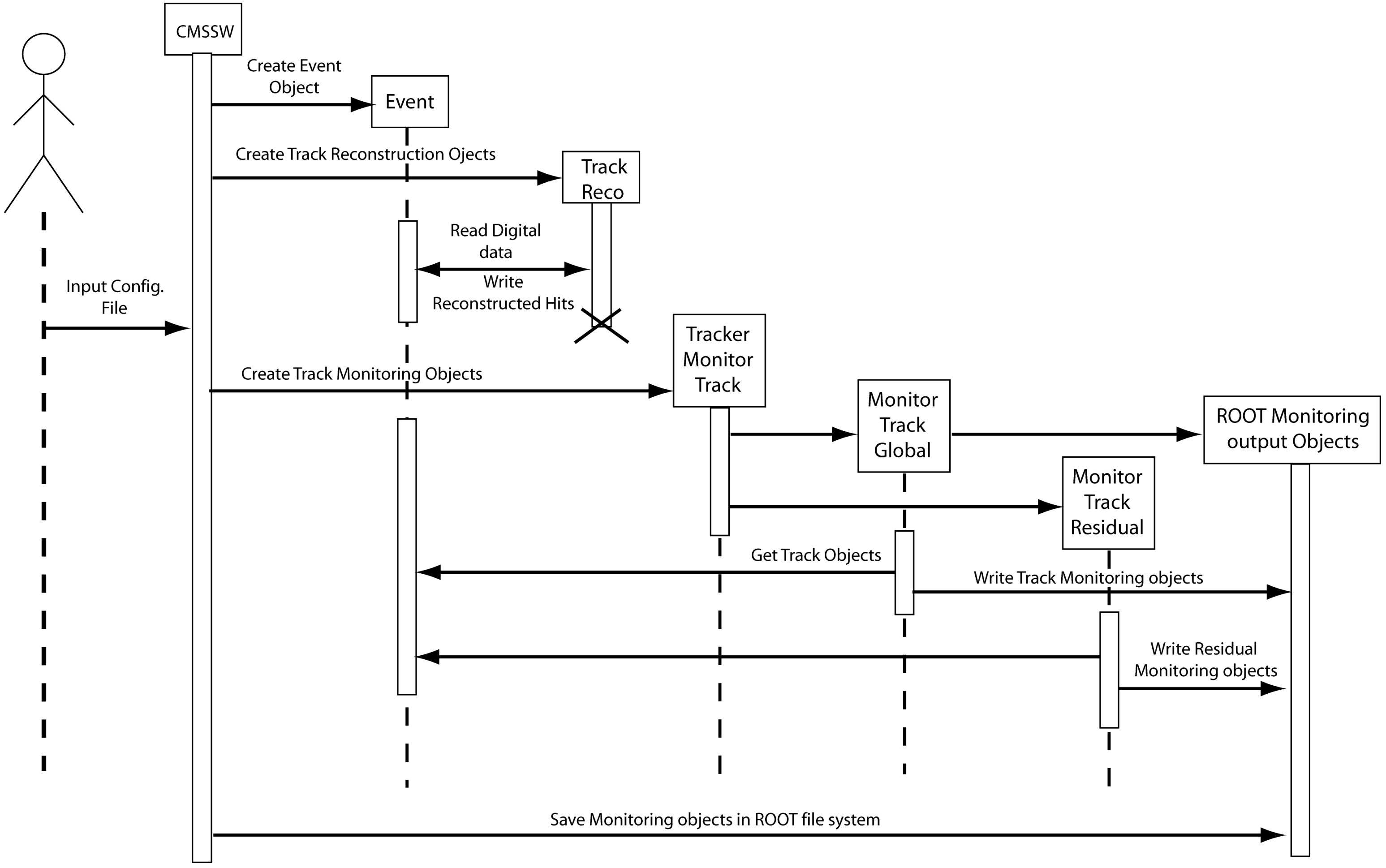
CMSSW
Create Event Object
Event
Create Track Reconstruction Ojects
Track Reco
Read Digital data
Write Reconstructed Hits
Input Config. File
Tracker Monitor Track
Create Track Monitoring Objects
Monitor Track Global
ROOT Monitoring output Objects
Monitor Track Residual
Get Track Objects
Write Track Monitoring objects
Write Residual Monitoring objects
Save Monitoring objects in ROOT file system

Figure 4.4: A sequence diagram showing the flow when a CMSSW module is being run. In this case the Tracker_Monitor_Track package can be seen in the sequence. To run the package several other reconstruction software packages have to run. At the end of the sequence the data is stored as a ROOT file.

4.3 Monitoring Tracks

The software package for monitoring tracks, Monitor_Track_Global, accesses reconstructed tracks from CMSSW event. Tracks have parameters that define them and these parameters are used in monitoring their quality. Table 4.1 gives the full list of the track parameters used in the quality monitoring. These parameters and their contribution to the monitoring of the track quality is described in detail later in this chapter.

Track attributes	Description
P_t	Track Transverse Momentum
D_0	Distance of closest approach projected on the transverse plane
Z_0	Distance of the closest approach along the beam pipe (z axis)
ϕ	Azimuthal angle of the track
η	pseudo rapidity of the track
χ^2	the χ^2 of the fitted track
rechit	Number of reconstructed hits per track

Table 4.1: Track object attributes used in the monitoring of its quality and their representation.

4.3.1 Track Monitoring Elements

The MEs of Monitor_Track_Global module contain distribution of data that is used to monitor the quality of tracks. Based on the requirement given in[3] and private correspondence with the tracker group the following MEs were created and they are to be used to monitor track quality:

- Track Transverse momentum (P_t)
- P_t against the azimuthal angle (ϕ)
- P_t against the polar angle (θ)

- P_t against the pseudorapidity (η)
- Distance of the closest approach projected on the $x - y$ plane (D_0)
- D_0 against ϕ
- D_0 against θ
- D_0 against η
- Distance of the closest approach projected on the $r - z$ plane (Z_0)
- Z_0 against ϕ
- Z_0 against θ
- Z_0 against η
- Track χ^2
- Track χ^2 over number of degrees of freedom (nDoF)
- Track χ^2 over nDoF against ϕ
- Track χ^2 over nDoF against θ
- Track χ^2 over nDoF against η
- Number of reconstructed hits per Track
- Number of reconstructed hits per Track against ϕ
- Number of reconstructed hits per Track against θ
- Number of reconstructed hits per Track against η

The ME produced by Monitor_Track_Global contain both one-dimensional and two-dimensional data sets. Figures 4.5 and 4.6 show eight two dimensional histograms displayed in 3D to show the distribution of the data sample. This information is essential to

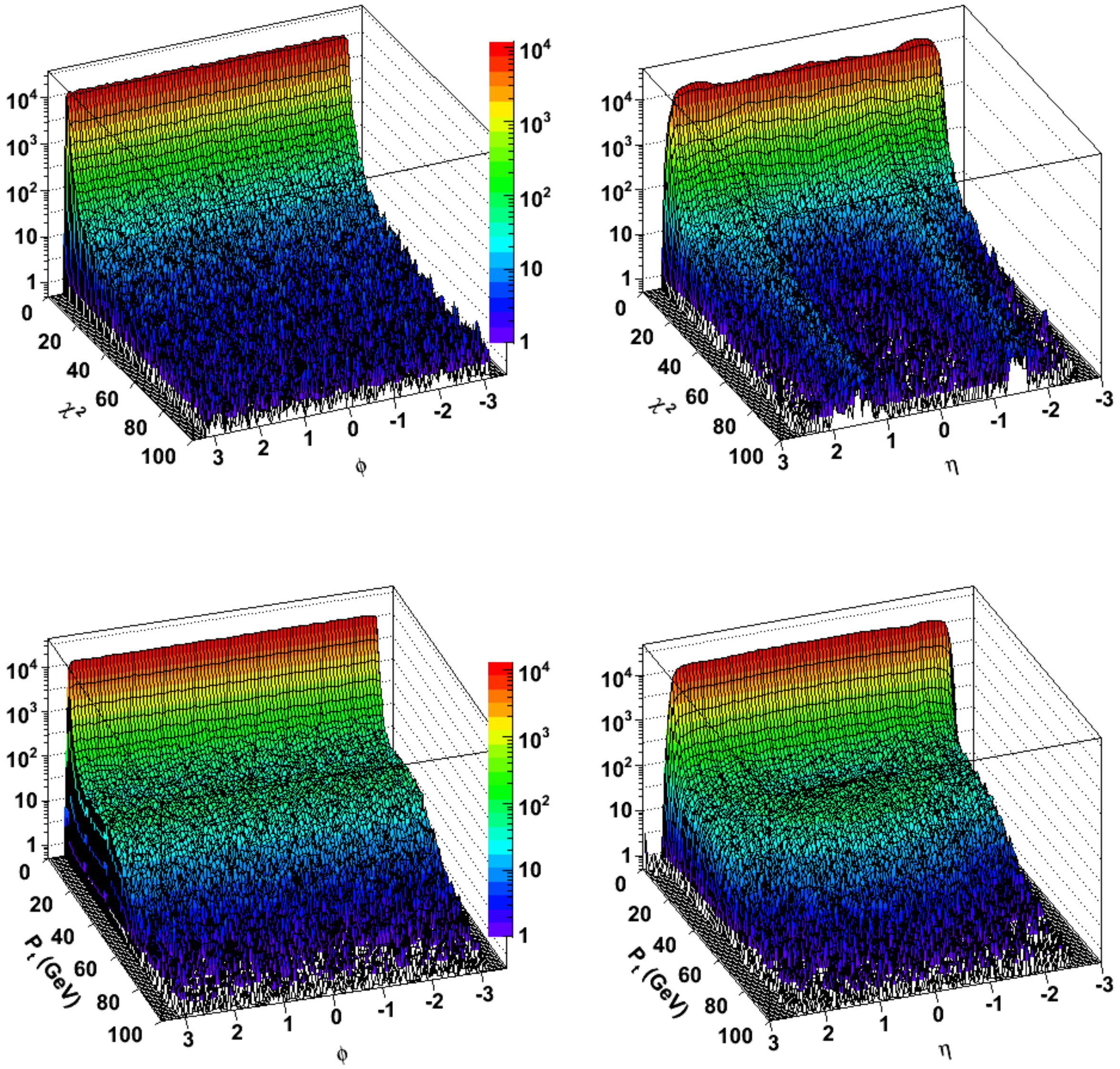

Figure 4.5: The distribution of the χ^2 per number of degrees of freedom and the Transverse momentum seen against the azimuthal angle and pseudorapidity of tracks from the min bias data. This data is a Monte Carlo simulation of *pp* collision using the CMS model in the GEANT simulation package.

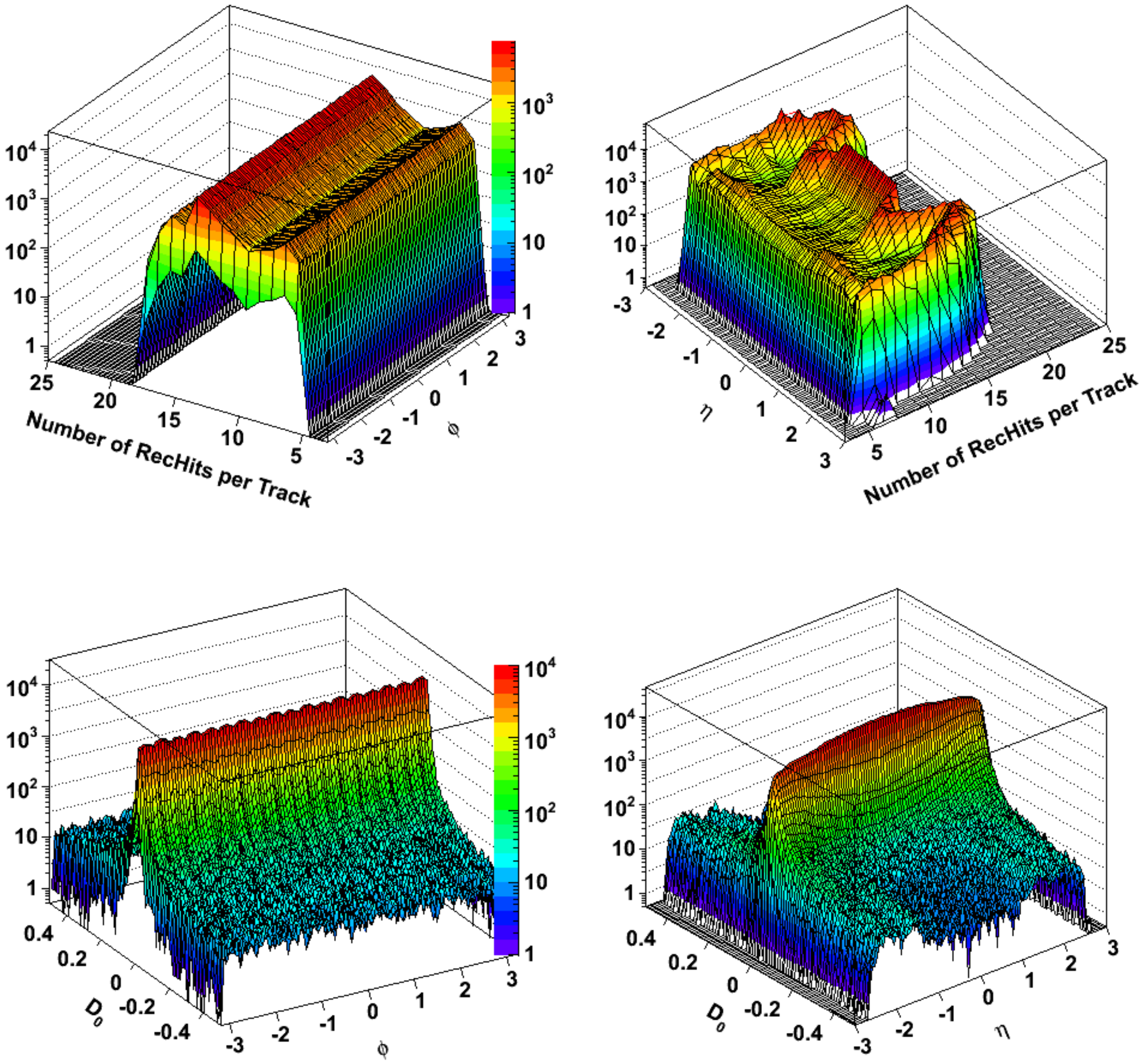

Figure 4.6: The distribution of the Reconstructed Hits per Track and Distance of closest approach (D_0) of tracks against the azimuthal angle and pseudorapidity. This data is a Monte Carlo simulation of pp collision using the CMS model in the GEANT simulation package.

monitoring the quality of the tracks being produced. A change in the normal operation of the Tracker would be reflected in these histograms.

The Monitor_Track_Global also produces one dimensional data as can be seen from Figures 4.7. These have similar use to the two dimensional data, although their usage is slightly different. The two dimensional data show the distribution of data against the pseudorapidity and the azimuthal angle, which give the user (figuratively speaking) an x-ray view of the tracker. This has been demonstrated in the testing section of this chapter using data from the Tracker Integration Facility.

4.3.2 Number of reconstructed hits

Reconstructed hits play a major role in the reconstruction of tracks. Each hit[4] is used to find the next hit on the tracker to identify the path (track) of an incident particle. Tracks reconstructed and fitted using small number of hits are generally of poor quality tracks, as they will include the highest fake hit rate. Fake hits are introduced into the track reconstruction to compensate for hits that might not have been reconstructed. This could be due to the detector module not detecting the incident particle due to dead strips or high noise. The details of this can be found on Chapter 3. Getting reconstructed tracks where a huge proportion of which contain the minimum number of hit requirement, could indicate poor quality of tracks and one would look in this in more detail to understand the reason behind it. The Tracker_Monitor_Track package was used to make such selection during the Tracker integration at the Tracker Integration Facility (TIF) at CERN [45] and Magnet Test and Cosmic Challenge (MTCC) which was a crucial test for testing how the detector modules will behave when fully integrated [46].

There are several reasons for the number of reconstructed hits of a track to be low. Some of the expected reasons are: poor reconstruction algorithm where track reconstruction is relying on fake hits, misalignment of the tracker where hits from a real particle would not be included by the reconstruction, or fault in the detector modules. Faults in the detector module could be due to either noisy electronics or failure to give any readings. In any of

[4] hit will be used instead of reconstructed hits in this chapter where both have the same meaning as Section 3.3.1

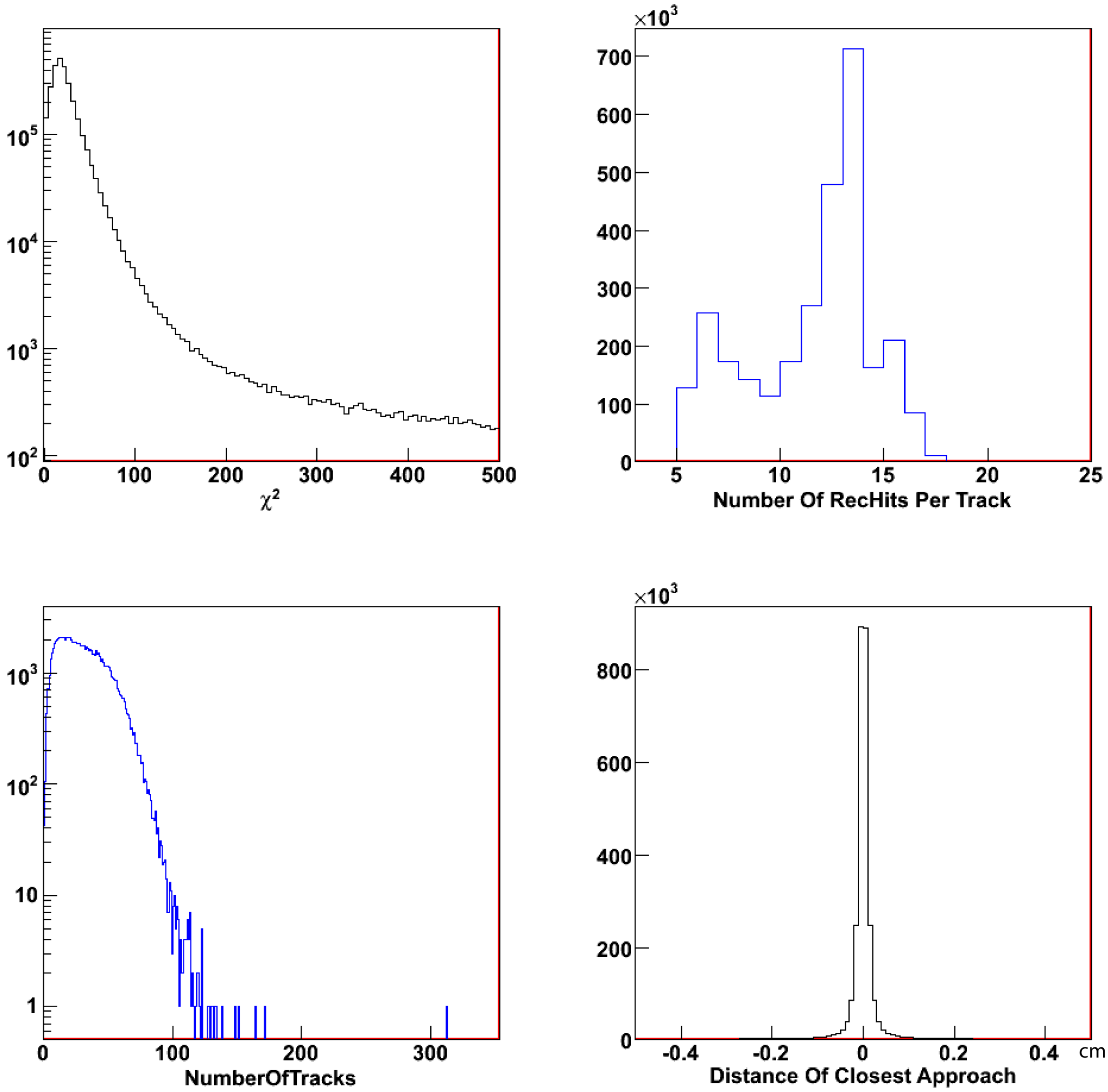

Figure 4.7: One dimensional histograms of the Track monitoring package. From left to right, top to bottom, Track χ^2, Number of reconstructed hits per track, Track Numbers per event and the distance of closest approach projected on the transverse plane.

these cases the result will be poor track quality.

4.3.3 Track monitoring and χ^2

Track χ^2 is the result of the track fitting processes, and it shows the goodness of fit of the track to the associated hits. The fit of the track is improved by the Kalman Filter which uses the χ^2 to achieve the best goodness of fit of tracks to the reconstructed hits. The Kalman Filter algorithm is used to improve the fit of a track to the hits. This is done iteratively. The algorithm uses path estimators from calculations that predict the path of a particle moving through space. Each time a new hit is added to the track, the Kalman Filter algorithm is used to improve the fit to include the new hit. The implementation of the algorithm is given by [37]. Track χ^2 is defined by

$$\chi^2 = \Sigma_{i=1}^{n} \left(\frac{\xi_i - \xi_{(i,a)}}{\sigma_i} \right)^2 \tag{4.1}$$

Where ξ_i is the i^{th} measured coordinate of the reconstructed hit on a particular tracker layer. A track would ideally have a hit for every layer it goes through, but sometimes it is important to create a fake hit to eliminate the possibility where a particle has traversed a layer without being detected. $\xi_{(i,a)}$ is the position of the the projected track that is passing through the given detector module. This is a helix track with a fit parameter $'a'$ used by the Kalman filter to improve the track fit which is the combined result of the chi^2 and other dynamic values arising from the fit algorithm [36]. During the estimation of the track trajectory, the parameter $'a'$ in the function $\xi_{(i,a)}$ is updated until the χ^2 of the fitted track is a minimum. The methods used to achieve this are described in Section 3.3. σ_i is the error on the reconstructed hit measurement which is given by.

$$\sigma_{res} = \sqrt{\sigma_{pred}^2 + \sigma_{hit}^2} \tag{4.2}$$

Where σ_{pred} is the error in the predicted track position on a given detector module known as “trajectory state on surface” (TSS). The error on the predicted position of TSS arises from several contributors. Some of them are: alignment error of the module, error propagated from the previous hit and general error on the assumptions used by the fit equation. The error of TSS on a detector module is given by the difference between the forward and backward predicted state of the particle track. Forward predicted state is the predicted position when the track is reconstructed starting from the inner layers of the tracker and the backward state is when the tracks are reconstructed starting from the the outer layers of the tracker. The σ_{hit} is the error of a reconstructed hit on a single detector module. The error arises from individual errors of a uniform distribution across a single strip of the detector module. The error on individual strip_pitch width is given by

$$\sigma_{hit} = \frac{strip\_pitch\ width}{\sqrt{12}} \tag{4.3}$$

Where strip_pitch is the distance between the centre of two neighbouring strips of a given silicon strip sensor (see Figure 3.2). $\frac{1}{\sqrt{12}}$ is the rms of a uniform distribution (0,1) and is described in detail in [47].

One of the causes for high χ^2, which is the sign of a bad quality fit of a track is misalignment of the tracker. Misalignment causes the creation of fake tracks with unrealistic distance between the projected trajectory of the fitted track on the detector module and the reconstructed hit.

4.3.4 Track Monitoring and Transverse Momentum

The transverse momentum (P_t) of a particle is computed using the curvature of its track (see Section 3.2.3). This makes its precision highly dependent on the alignment of the tracker. It is obvious that a shift in the tracker position will result in the deformation of the curvature of the track which has a direct effect on the magnitude of the P_t. Studying the distribution of the P_t against the azimuthal angle ϕ and the pseudorapidity η can show the effect of misalignment on the P_t of the reconstructed tracks.

4.4 Monitoring the Silicon Strip Detector Modules

The Monitor_Track_Residual module is another part of the Tracker_Monitor_Track package that monitors the silicon strip detector modules performance with respect to the reconstructed tracks. This module works out hit residual of particle tracks that have passed through a given detector module. Hit residuals are explained later in detail.

To calculate the magnitude described as the 'hit residual', both the hit and trajectory positions of a given track on a detector module need to be known. This information is essential in the reconstruction process of the track but not all of it is stored in the reconstructed track physics object. The reconstructed track object is made persistent by removing elements that were used in the reconstruction of the track but are deemed to be not needed during analysis. The trajectory state on surface (TSS) parameter that holds this information is one of the parameters that are not kept in the persistent reconstructed track object. This creates a challenge in getting the fitted track position on a given detector module.

To access the TSS the position of a track on a specific detector module needs to be known, This is achieved by extracting each individual track objects from the event file. From each track that is being studied the hits used to reconstruct it are extracted. Using these hits the track is reconstructed again using the same track reconstruction algorithm as the original track. This time when track is reconstructed and fitted, the TSS object which holds the position information is saved. This methods is repeated for every track whose hit residual had to be calculated creating a CPU time overhead for Tracker_Monitor_Track package. This creates a problem if the package is to be used online. This issue was resolved by making the TSS parameter an attribute of the persistent track object.

The module then accesses both the reconstructed hit and the trajectory state on surface objects to find their position on the detector module. Each track that has its trajectory on the detector modules is used to create data sample of the hit residual information of the detector module. In an ideal situation where the tracker is perfectly aligned and the track reconstruction is precise then the hit residual would be very small.

There are several reasons that make hit residual inevitable. The limit to the precision on which the tracker can be aligned [48], the limit of the precision of the measurement of the position of impact on the detector, the limits on the track reconstruction algorithm, noise on the detector modules and the effects caused by a particle interaction with the detector matter (such as Bremstrahlung and multiple scatting). All these lead to error in the measurement of the track.

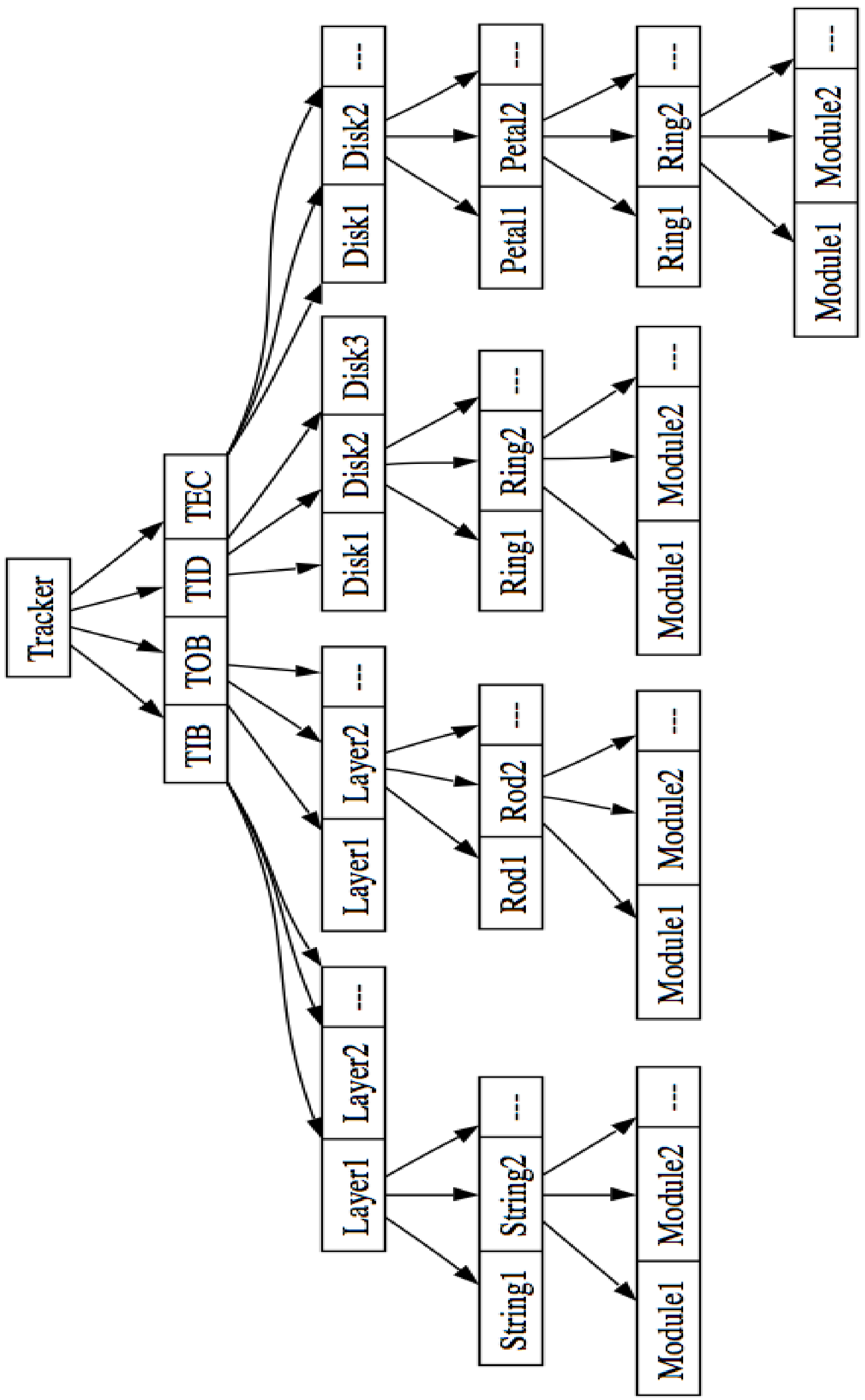

Figure 4.8: The file structure of the hit residual histograms reflects the physical arrangement of the Silicon Strip Tracker Modules.

The use of hit residuals to monitor track quality is fairly straight forward. Hit residual distribution of a given detector module (given there is no fault) has a mean very close to zero with standard deviation ~ 1 (See Appendix A). If the distribution shows significant difference from this, it is possible there is issue with the data quality. This is tested using simulation data with both aligned and misaligned detector geometry to study the change in the detector behaviour in Section 4.5.

4.4.1 Hit Residual of tracks

Hit Residual is the distance between a reconstructed hit and a fitted track's position on a detector module. Figure 4.9 shows a schematic drawing of a hit residual on a silicon strip tracker (SST) detector module. The hit of an SST module is given by the charge measured on individual strips. The strips near a position where an incident particle had interacted with the sensor (see section 4.3.2) give out signals as charges. Due to the design of the SST modules the position of a hit is available only in the 'x' plane of CMS coordinate.

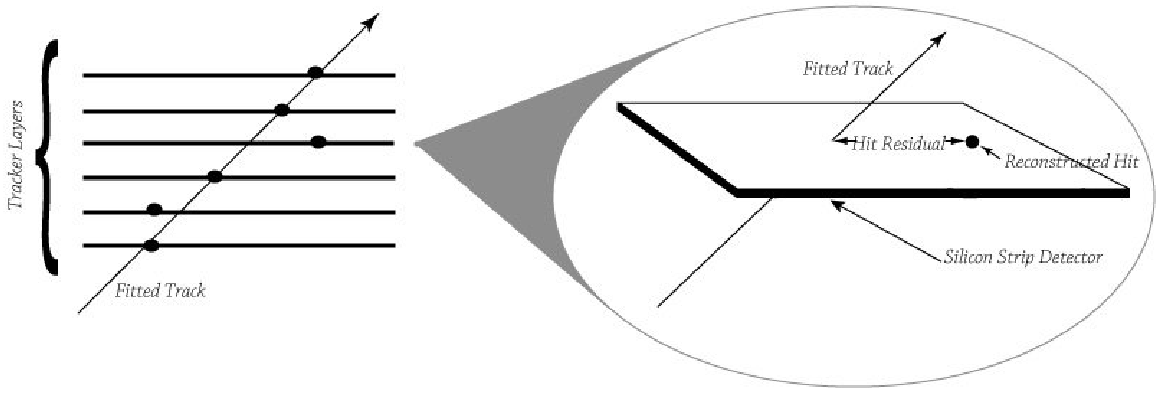

Figure 4.9: Hit Residual of a Track. In a strip detector the reconstructed hit would be a line stretching across the length of the strip not a point

A measurement of a hit position is calculated from a collection of strips known as cluster. The centroid of the cluster charge is used to determine the position of the hit. The hit residual of a track on a detector module is a component of measurement known as 'pull', given by

$$pull = \frac{val}{error} = \frac{X_{pred} - X_{hit}}{\sigma_{res}} \tag{4.4}$$

Where X_{pred} is the predicted position of an incident particle in the detector module. This is calculated by the track fitting algorithm which uses Kalman filter to predict the path of a charged particle travelling in a magnetic field (see Section 3.3). X_{hit} is the position of the reconstructed hit as described above. σ_{res} is the error on the Hit residual. The error of the residual is given by Equation 4.2.

4.5 Tests and Results

After presenting the progress of Tracker_Monitor_Track at the tracker group meeting the tracker alignment group was interested in using the track monitoring package. The alignment group had already developed packages that simulate misalignment scenarios of the tracker geometry. This created an opportunity to test the Tracker_Monitor_Track package using data from a misaligned tracker geometry to see how the distributions of the MEs change. The reason for choosing the misalignment scenario data was to save time as the package already existed and the alignment group has made an extensive study [49] on misalignment scenarios and the package would recreate a realistic case study.

The misalignment package provides two scenarios based on the running stage of the LHC. The first scenario called the short term scenario emulates the time from the very start of LHC beam to the first few months of data taking. This time is critical in terms of aligning the Tracker Geometry. The physical tracker modules won't be aligned to the required accuracy due to engineering limitations. Although these are very small in the range of 50μm, they have an effect on the sensor resolution which is designed to achieve in the uncertainty of 10μm [50]. High momentum particles will be used during this period to align the tracker [51].

The second scenario is the long term misalignment scenario. This covers the period after the short term alignment where the reconstruction algorithms will be tuned to compensate for the unavoidable misalignments. The long term scenario was chosen to test the Tracker_Monitor_Track package as this will be the important time of the physics analysis and the quality of tracks should be guaranteed.

Tracker_Monitor_Track package was tested using CMSSW_1_2_0 version. The package was run using Grid enabled distributed data analysis software such as CRAB[52] and ASAP [53]. These software packages look after the running of the CMSSW framework using data that is available in internationally distributed computing system. As the LHC has not run, Monte Carlo generated simulation data is used to test the monitoring package. The data selected for the test was $Z \rightarrow ee$ generated from the centre of the detector. This data is available on database accessible via the Grid. The package was run on the maximum quantity of data available and it consists of a total of 3,000,000 events.

4.5.1 Hit Residual Results

When the hit residuals on reconstructed tracks were taken from the Monte Carlo simulated data it was observed that the distributions had long tails. This distribution is consistent with a double Gaussian as can be seen from Figure 4.10. This behaviour of the hit residuals is apparent in all the distributions taken from the various parts of the Tracker. To demonstrate this histograms of hit residual distribution from each part of the tracker was selected at random and fitted with single, double, triple and quadruple Gaussian distribution. To make sure the sampling was representative of all detector module types, at least one module from each layer (whenever possible) of the Tracker was taken. The distributions of the samples taken can be seen in Appendix A. All of the distributions did not fit to single Gaussian distribution due to long tails. They all fitted to double Gaussian and above. Although triple and quadruple Gaussian seemed to fit well, their improvement over the double Gaussian was very small.

This behaviour of hit residual distribution (fitting to double Gaussian) can be explained by the fact that hit residual arises from various errors that can be categorised into two types. The first set come from the trajectory fit of the track. These consist of errors propagated from several other hit position measurement and assumptions made by the

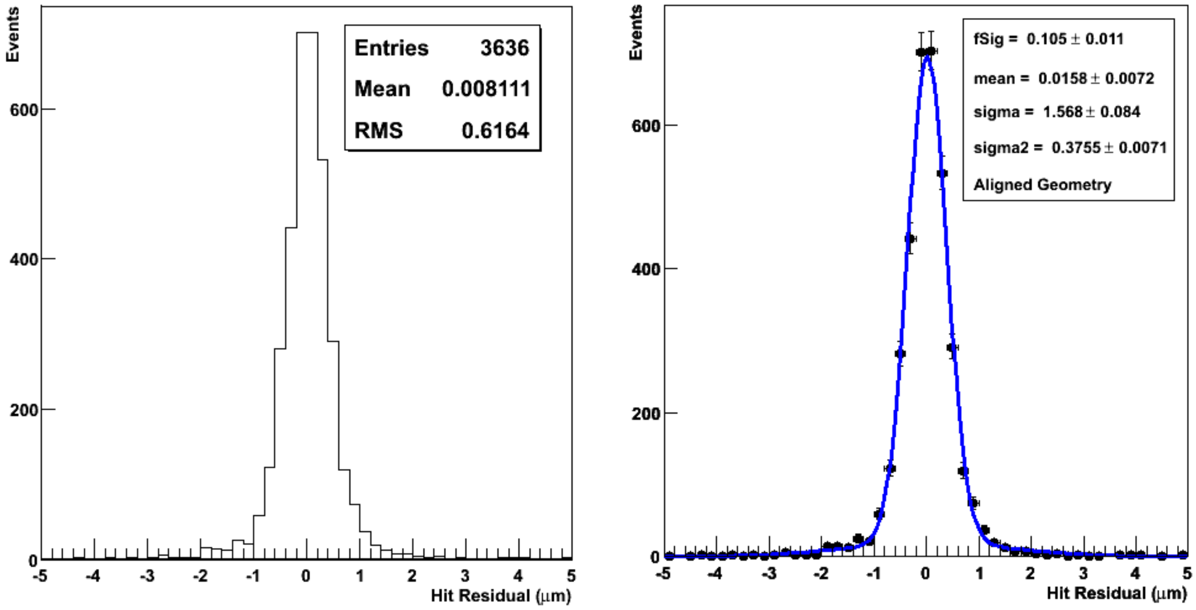

Figure 4.10: Hit Residual of one of the Tracker Inner Barrel modules using the $Z \rightarrow ee$ simulation data. Left the histogram showing the distribution of the hit residuals. Right best fit for the residual is double Gaussian, as the distribution has tails.

track fitting algorithm. The second set of errors come from the error of the measurement of the position of the hit. These set of errors are a combination of the strip_pitch errors given by Equation 4.3 of all the strips added up to make the cluster.

The *mean* of the hit residuals from individual detector modules differed slightly depending on the detector module type. This is shown by Figures 4.11 to 4.14. In these figures the data is collected from different detector modules with similar *strip_pitch* size It can also be noted that as the number of entries per histogram increases the deviation of means becomes smaller. And after the number of entries has exceeded 1000, the means tend to stabilise. This is expected due to the central limit theorem. Due to the variation in the mean of the different hit residual readings the histograms were divided into four groups to represent data of similar detector modules based on the Tracker subsystems.

It was noted that the RMS of the TEC was giving two peaks. A further analysis of the data showed that this behaviour is repeated to the individual detector module level showing that this difference in the RMS was not arising from the difference in the silicon sensors *strip_pitch* size. Figure 4.15 shows the RMS of the various structures inside the TEC. The histogram at the right shows the residual from a single ring which contains identical detector modules. A possible explanation for the existence of two peaks is that there are two types of data arising from different tracks. The electrons from $Z \rightarrow ee$ process are subject to bremsstrahlung process which will affect the track of the electrons. This means that the tracks affected by bremsstrahlung process will give a higher variation in their residual when compared to the tracks not affected by such process (eg. pions and muons). Analysis of the bremsstrahlung effect on the tracks is not in the scope of this thesis and it could be starting point for future analysis.

Because the MC simulation used for this test did not have tracks at high *eta* it was not possible to run a reconstruction so that all the detector modules can get equal amount of readings. As a result some of the modules in certain part of the tracker have very small number of entries and they were left out of this test. A good example of this are the number of entries for the TID modules were noticeably small. This is explained by the distribution of the tracks along the η. Top right of Figure 4.6 shows the distribution of rechits per track along η. It can be seen from this figure that some parts of the detector

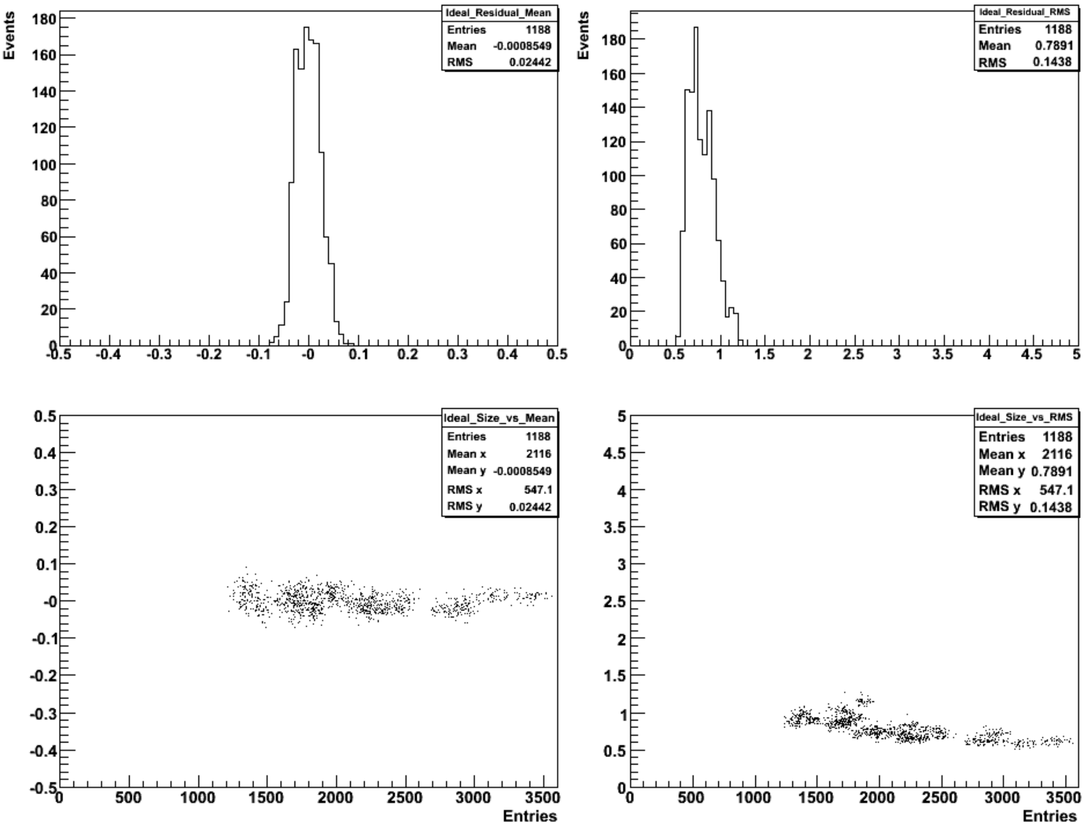

Figure 4.11: The distribution (from top to bottom, left to right) of mean, RMS, mean against number of data entries and RMS against number of data entries of hit Residual of TIB modules.

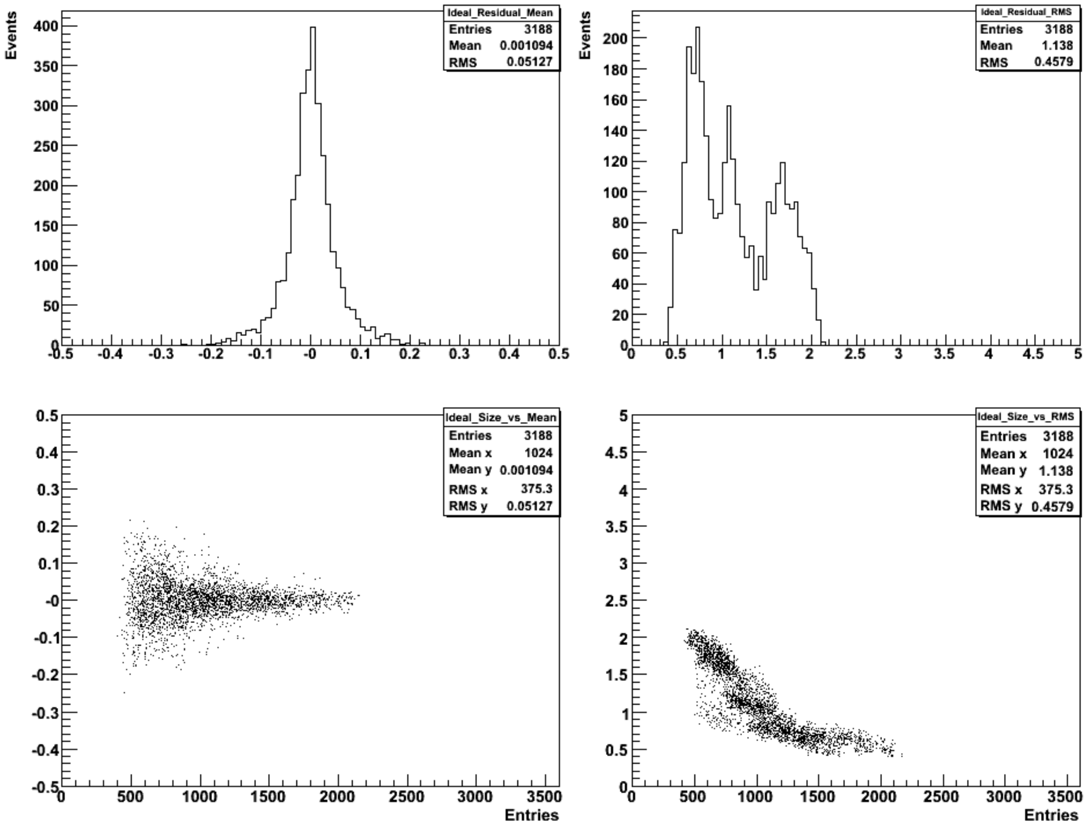

Figure 4.12: The distribution (from top to bottom, left to right) of mean, RMS, mean against number of data entries and RMS against number of data entries of hit Residual of TOB modules. The rms against number of sample entries of the TOB shows a clear correlation; when the number of entries increases the rms becomes smaller.

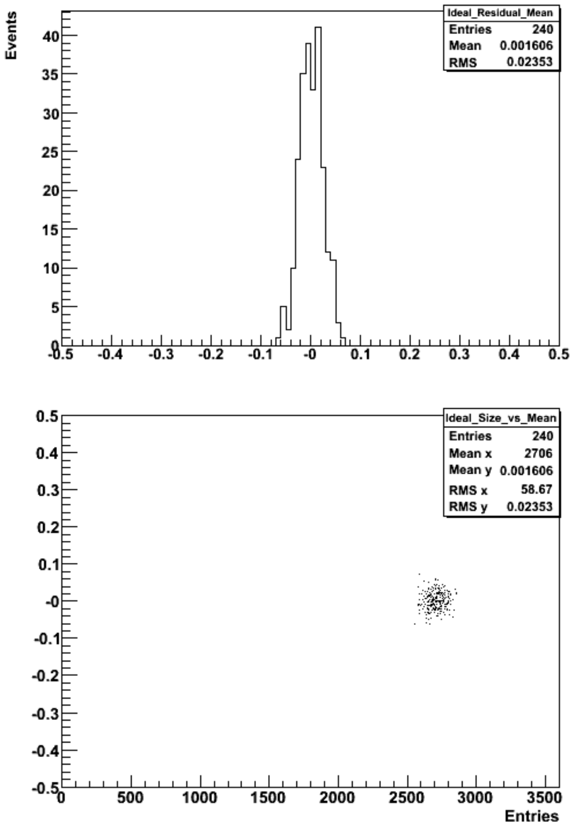

Figure 4.13: The distribution (from top to bottom, left to right) of mean, RMS, mean against number of data entries and RMS against number of data entries of hit Residual of TID modules.

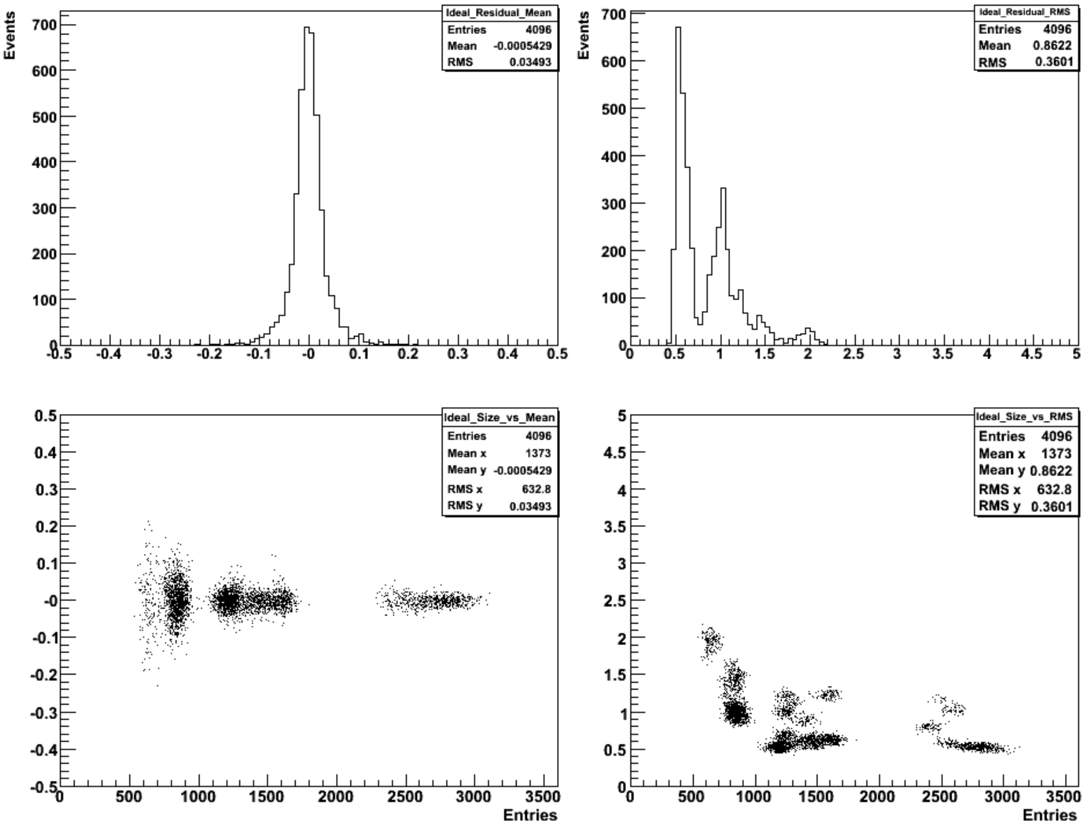

Figure 4.14: The distribution (from top to bottom, left to right) of mean, RMS, mean against number of data entries and RMS against number of data entries of hit Residual of TEC modules. The rms against number of sample entries of the TEC shows a clear correlation; when the number of entries increases the rms becomes smaller (similar to TOB).

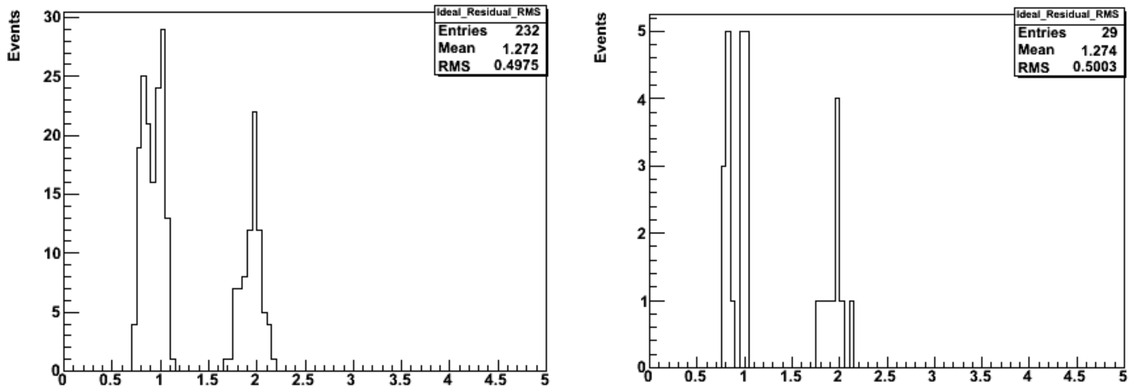

Figure 4.15: The RMS of the hit residuals in the TEC shows two separate data sets through out the structure. The histogram on the left shows the RMS of modules from the same TEC Petal and the right histogram shows RMS of residual from detector modules on the same TEC Ring structure.

were not getting as much data as the rest of detector modules creating a depression on the distribution. This meant that enough data was not collected from the TID system to compare it with the other sub-detector systems. This being the case the distribution of the data seems to be agree with the other sub detector's distribution. The mean for residuals that belong to detector modules with hits > 2500 is grouped around 0 with similar width to the others, the same can be said for the RMS distribution.

To study the possibility of defining a general limit on the *mean* and σ of hit residual, the mean of all individual hit residual distribution with more than 100 entries was studied. Data generated using the aligned geometry of the Tracker was used. The mean was generally very close to zero and with $\sigma \approx 1$. This was even more apparent as the data entry increased. But it was noticed there were slight differences between the different modules of the Tracker. This is clear when comparing Figures in Appendix A where the TID residuals have wider distribution compared to the TIB. To avoid calculating the limits in the σ for each histograms of the 15,000 detector modules, it was decided to to use the χ^2 test. With this test a reference histogram is defined using data from similar detector unit and compare with the actual readings from the detector module during the experiment. The χ^2 test is discussed in detail in Chapter 6 where the same data discussed here is used to show its effectiveness.

To see how misalignment of the tracker would affect the distribution of the hit residual, one detector module was selected from layer 3 of the TIB. A module which read a very high amount of hits was selected to increase the chance that both the data before and after misalignment will be enough to represent the actual distribution of the hit residual. The misalignment of this module was done using the simulation software of the misalignment tools discussed at the beginning of Chapter 4.5. Data was then collected before the misalignment and after the misalignment. To observe it visually both data samples were plotted on the same frame with different colours. Figure 4.16 gives the distribution of hit residuals from both the aligned and misaligned tracker geometry.

It can be clearly seen that the tracker misalignment has a direct effect on the hit residual distribution. The data from the misaligned geometry has shifted mean and relatively higher σ values. Which means that studying the hit residual will indicate anomalies in the case of tracker misalignment.

4.5.2 Results for Track Global parameters

The same data discussed above is used to study the effect of misalignment on the global track parameters. In case of misalignment the modules will not provide an accurate position of the hit, which will have a direct effect on the reconstructed track. One ways this manifests is through low efficiency of the track reconstruction process. In other words fewer particle tracks are found with a misaligned Tracker. When the track numbers for the ideal and misaligned tracker geometry were compared this was seen. Figure 4.17 shows that the track reconstruction efficiency is lower for the misaligned scenario. Although the distribution of the two histograms is similar the fact that data with low hits being removed means that the quality of the tracks will be kept, but less events will be accepted.

Figure 4.19 shows a comparison between the aligned and misaligned data using the χ^2 track parameter. The χ^2 of both data are similar. The reason for this is that the misalignment package introduces a variable called the alignment position error (APE). This variable is the measurement error on the reading of a detector module due to misalignment. This error is combined with the other known errors of the detector module to give the

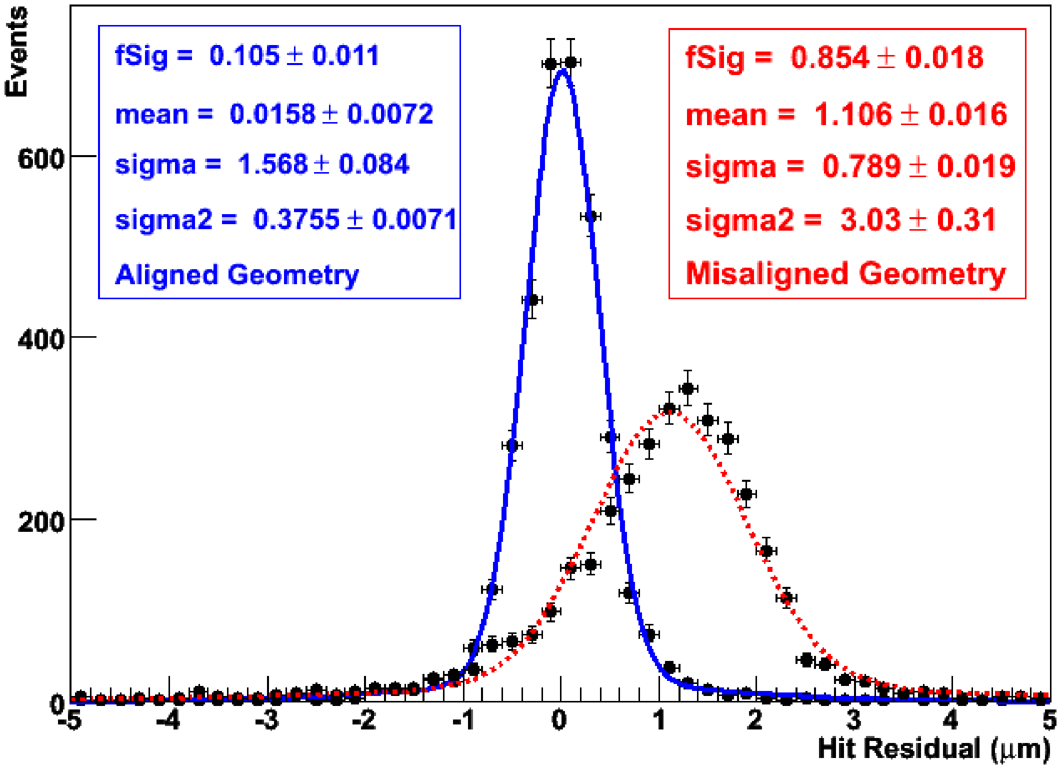

Figure 4.16: Distribution of hit residuals before (blue) and after (red) the tracker geometry was misaligned. Both data were fitted with double Gaussian (the fit parameter for the double Gaussian identical mean was used). It can be clearly seen that the residual monitoring element can identify a misaligned module.

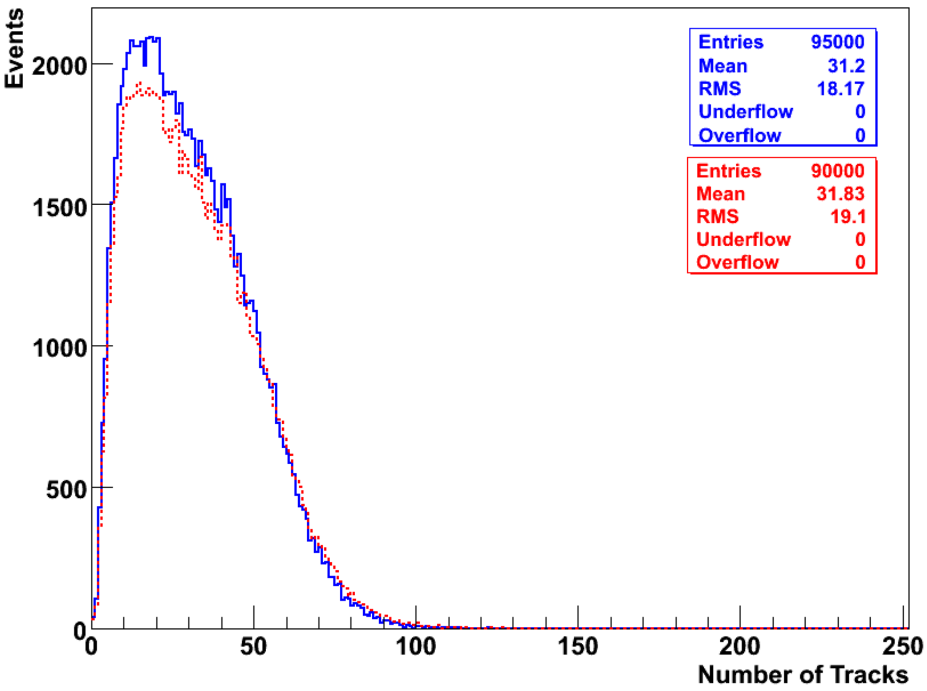

Figure 4.17: Number of reconstructed tracks for Ideal (blue) and misaligned (red) tracker geometry.

total error of the detector module [50]. This has a direct effect in the reconstruction of tracks. APE works by increasing the uncertainty of a hit. The method is illustrated in Figure 4.18. APE increases the uncertainties of a measurement in a given detector module, these uncertainties are the errors given in equation 4.1 where the APE is added to. This way when a track is being reconstructed the hit width will be large enough to compensate for the misalignment. Although this method is useful to compensate the errors caused by misalignment, it can reduce the quality of the tracks as it affects the momentum resolution, track efficiency and fake rate. This problems arise if the size of the APE is too large. The values that would give the optimal values for the APE is given in [50]. The misalignment tools come with an APE value that will compensate for the misalignment created. This is to simulate the real misalignment situation. Therefore it was not possible to see the effects of misalignment on the track χ^2. This will of course change in the real time running of the CMS experiment. Due to time constraints it was not possible to modify the misalignment package to omit the APE.

4.6 Analysing Cosmic Ray data from Tracker Integration Facility

The Tracker was integrated and commissioned in the Tracker Integration Facility (TIF), a dedicated facility at CERN Meyrin site during the period of November 2006 to July 2007 [45]. TIF is a large clean room providing all the services needed for full commissioning and validation of the tracker. This provided a good opportunity for the tracker to be tested using cosmic rays as the tracker was fully accessible at this time. The computing for the test was housed in a nearby building, the Track Analysis Center (TAC), where the facility for data analysis and storage was provided [54].

4.6.1 Tracker Setup at TIF

The Silicon Strip Tracker (SST) was fully commissioned at TIF, but only partial testing was done. This was due to the limited availability of readout electronics and constraints

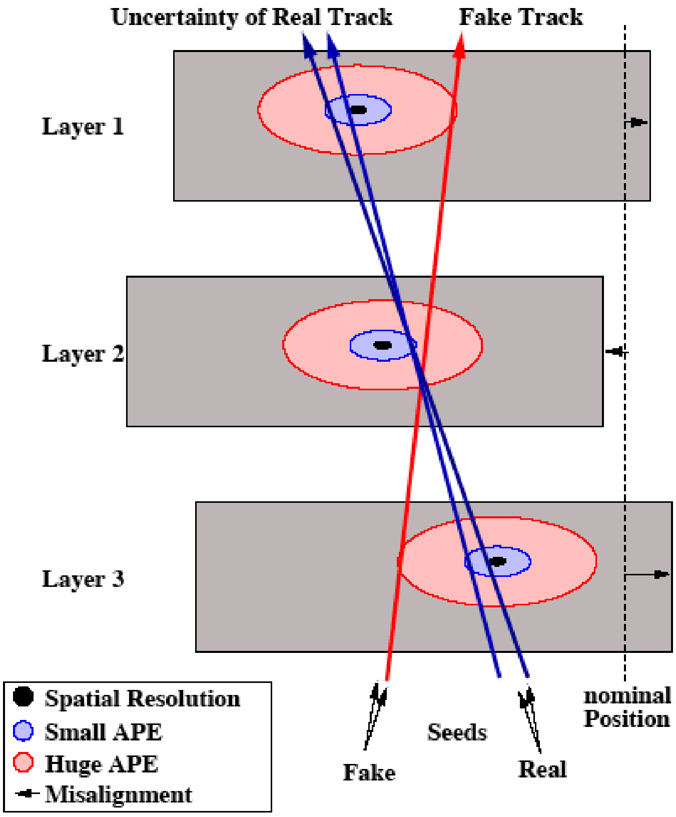

Figure 4.18: Effect of APE settings on Track Efficiency and Fake rate

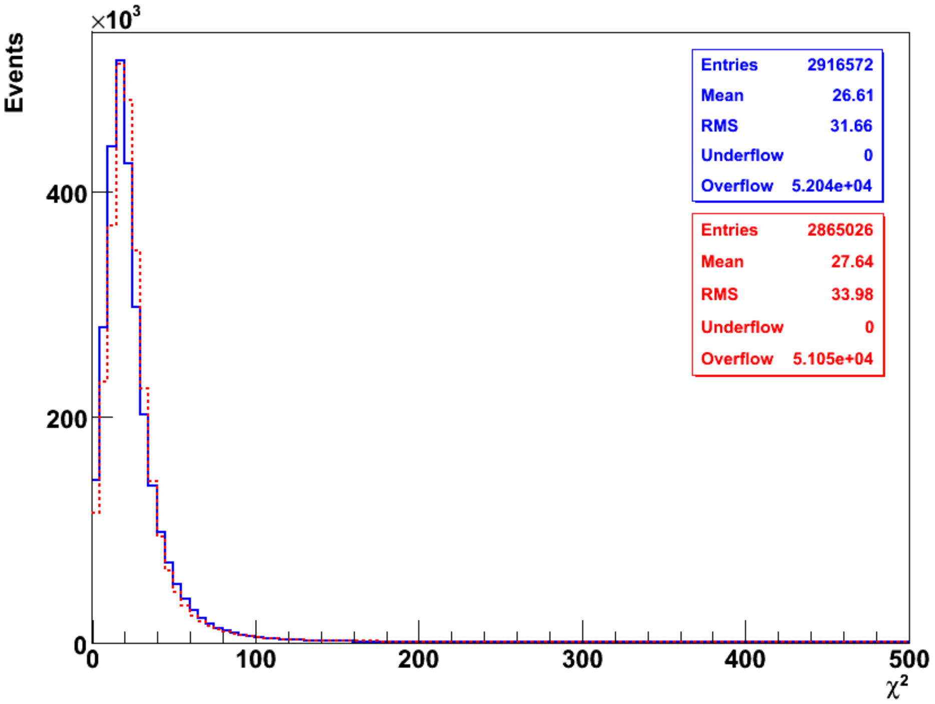

Figure 4.19: χ^2 of reconstructed tracks for Ideal (blue) and misaligned (red) tracker geometry.

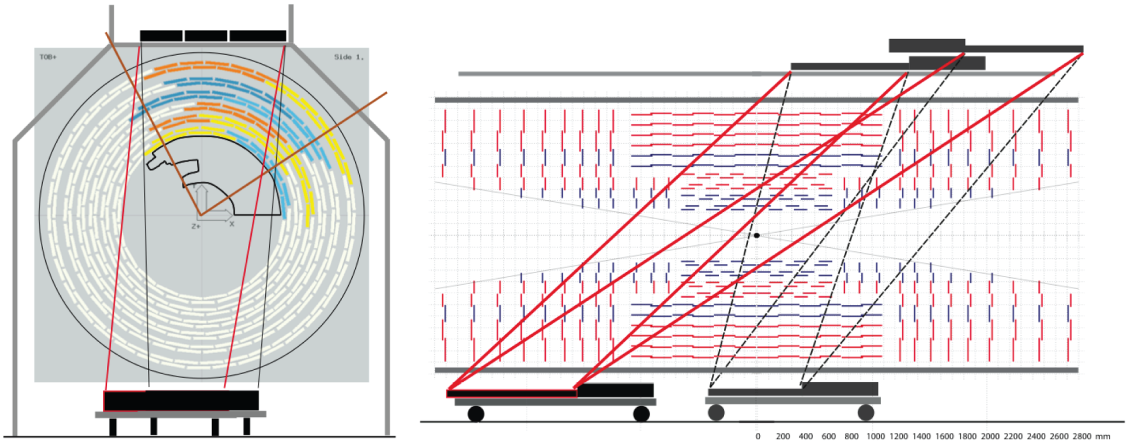
TOB+
Side 1.
Z+
X
0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 2800 mm

Figure 4.20: Layout of the tracker and the triggering scintillator positions during cosmic ray data taking at the TIF. The cross section of the tracker along the $x - y$ plane can be seen on the left. On right the $r - z$ view can be seen. The straight lines show the acceptance region of a cosmic ray.

from the data acquisition along with other mechanical issues. The four sub-detectors of the SST were represented by the partial testing which consisted of around 15% of the silicon detectors that are read using 1.3 million electronic channels. The tracker setup at the TIF did not include the commissioning of the Pixel Tracker.

The SST is not able to trigger on interesting data by itself. To achieve this cosmic muon triggering was provided by scintillating counters placed above and below the tracker. Figure 4.20 shows a schematic view of the tracker along with the triggering scintilators. The acceptance region of the Tracker, which is the area between the straight lines is defined based on the scintillators that are used for triggering. An event is triggered at the coincidence of the top scintillator with any of the bottom ones. To guarantee a minimum of 200 MeV cosmic ray energy a lead plate with thickness of 5 cm was placed on top of the lower scintillator counters [45].

4.6.2 Testing track fitting Algorithms with Cosmic Ray

The description on acquiring the cosmic data taken during the tracker commissioning in the TIF is based on[45]. The period of the data taking spanned March to July 2007. During this time no change was allowed in the setup of the tracker to guarantee consistent data taking for the offline data analysis. The data taking involved grouping the recorded data into different data sets defined by active detector, trigger setup and operating temperature. Based on this nine different data sets were made.

During the period of data taking, a total of 4.7 million cosmic events were recorded. Out of this 4.2 million events were saved for later analysis. This reduction came due to the rejection of data flagged as bad quality. To ensure this each run was checked using online and offline data quality monitoring tools. If an event in a run did not meet the requirement for good quality it was rejected. The parameters used for the quality test are the reconstructed hit per track and track χ^2 for the Tracker_Monitor_Track package.

To test the track quality monitoring package, cosmic ray data reconstructed using two separate algorithms were analysed. The two chosen algorithms are the Road Search (RS) and the Cosmic Track Finder (CTF). Both algorithms were developed during the Magnet Test and Cosmic Challenge which was done before the TIFF. These algorithms are discussed in detail in Chapter 5.

The data selected for this analysis are from a particular run (batch 8055), which were taken using the setup of the tracker shown in Figure 4.20 with only the far right most triggering scintillator (covering the area under the red lines). The temperature for the detector modules was kept at 15^0C. The data was accessed from the storage facility provided by the TAC.

Tracker_Monitor_Track package of the DQM was run offline under CMSSW_1_7_6 version. The data was used to reconstruct tracks using the RS and CTF algorithms. The track monitoring package then generated the ME that were further analysed. It was possible to monitor the global track parameters to study the performance of the algorithms.

During the TIF of data we noticed, as can be seen from left most of Figures 4.21 and 4.22, that the track η and ϕ for both algorithms were not compatible with each other. This should not be the case as the data used for both reconstruction algorithms is the same. This is an indication that both of the algorithms were not giving the right track η and ϕ. Further study into the data showed that the distribution of track η reconstructed

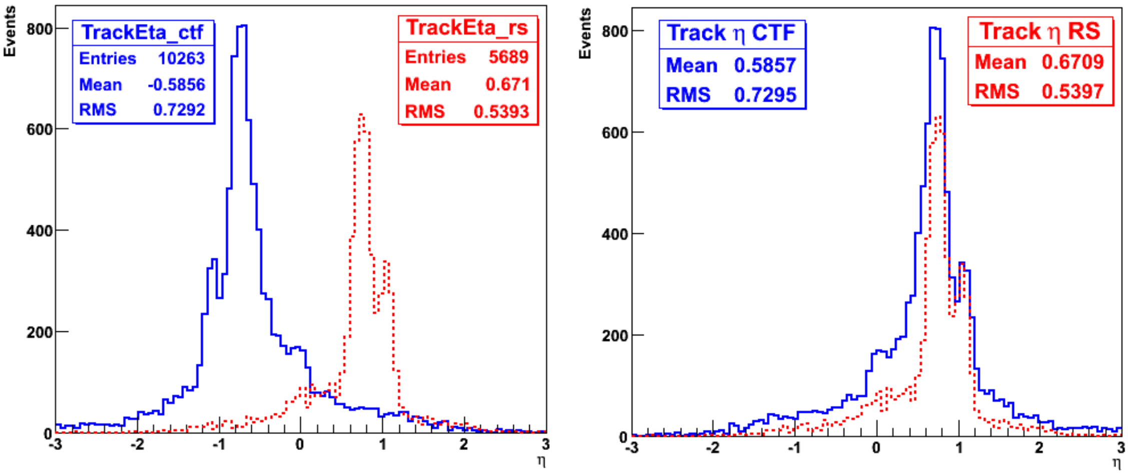

Figure 4.21: Track pseudorapidity of cosmic rays reconstructed using Road Search (red) and Cosmic Track Finder (blue) track fitting algorithms. Before correction (right) and after a correction (left)

by CTF and ϕ reconstructed using RS seem to be flipped along the y axis. From study into the fitting algorithm and private correspondence with the tracker group it was noted an error that has been over looked. When the CTF and RS algorithms were modified to reconstruct Cosmic data, the issue of track angle was not taken into consideration to make the algorithm compensate for this. This was corrected to give data as can be seen from 4.21 and 4.22 on the right most. And the issue of the algorithms not giving the correct track angle an η has been put forward to the tracker group to correct this issue.

Despite this error the data showed that both reconstruction algorithms were consistent with each other. The track η is consistent with the tracker setup shown in Figure 4.20 under the area inside the straight red lines. The peaking around $\eta \approx 0.5$ is consistent with this. The track ϕ distribution showed mean ≈ -1.3 which is consistent with tracks entering from the top section of the tracker, where one would expect cosmic rays to enter the tracker from.

Plotting the other track parameters has shown that the algorithms give similar recon-

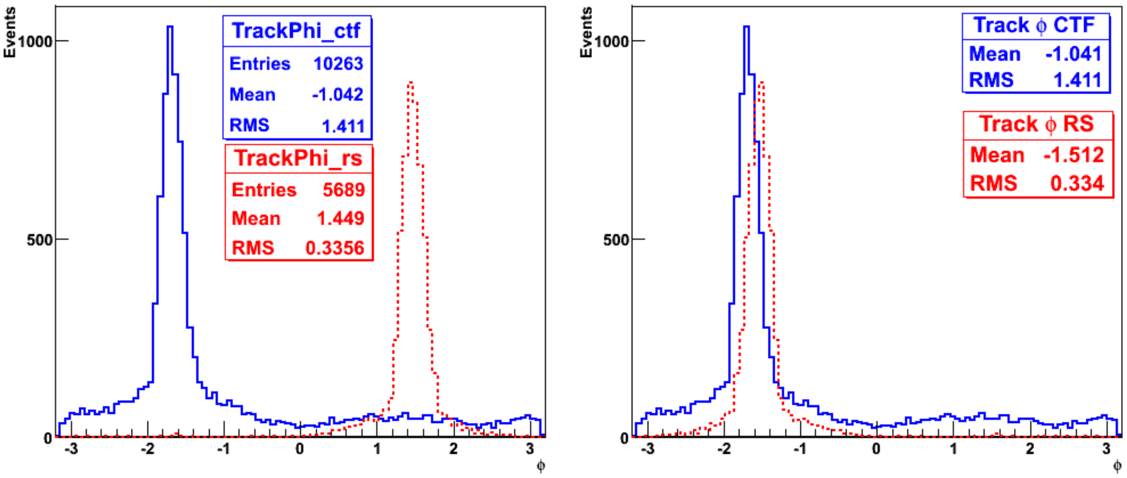

Figure 4.22: Track azimuthal angle of cosmic rays reconstructed using Road Search (red) and Cosmic Track Finder (blue) track fitting algorithms. Before correction (right) and after a correction (left)

struction (relative) efficiencies. The number of reconstructed tracks is lower when the RS algorithm is used. This is explained in detail in Chapter 5 where the reconstruction algorithms are discussed and more plots of the MEs are shown. The number of reconstructed hits bottom in Figure 4.23 shows that the algorithms have the same cutoff region for the minimum number of reconstructed hits per track, which is consistent with the requirement in the [3].

4.7 Conclusion

The Tracker_Monitor_Track package was designed with two goals in mind.

1. collect data samples that would be used to monitor the physics objects, which are tracks

2. provide a means to enable this data to be analysed for quality tests either by end

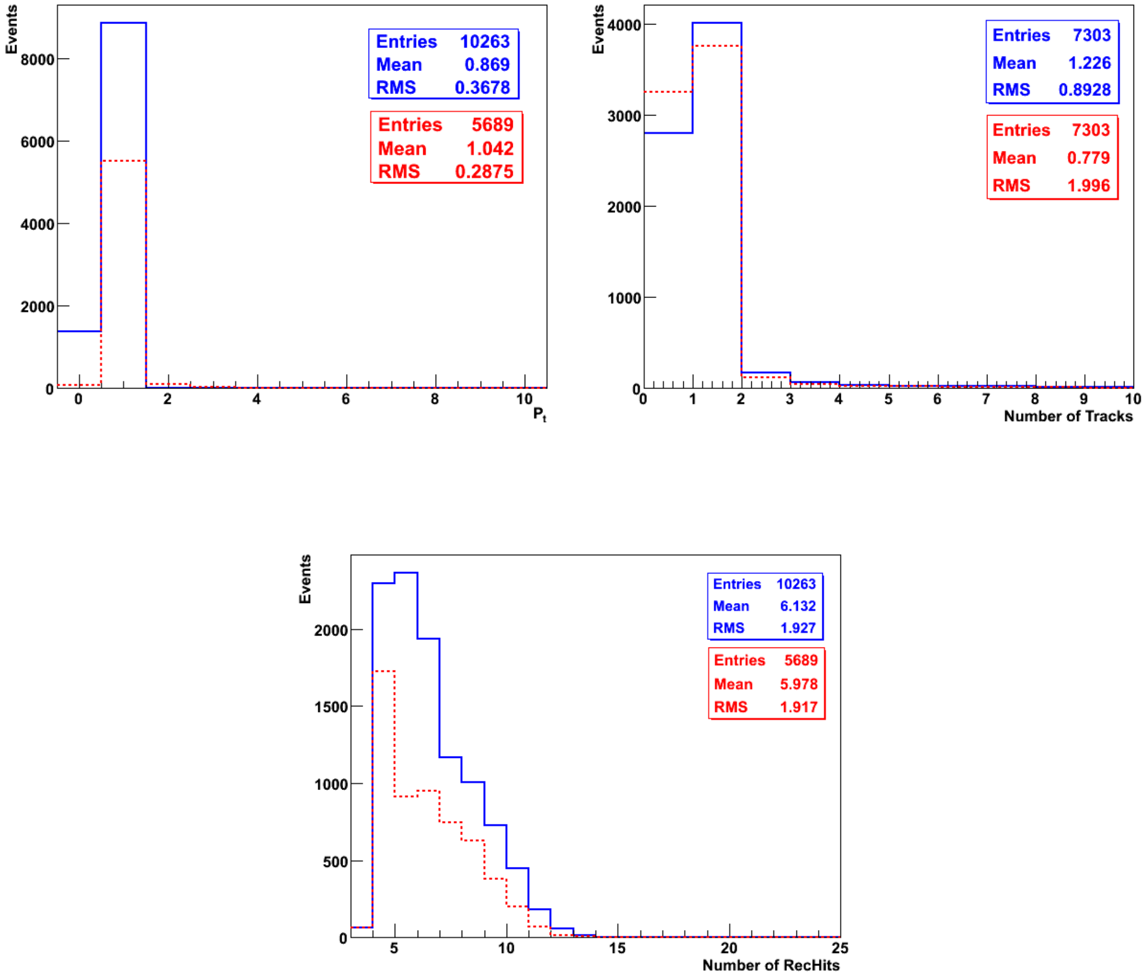

Figure 4.23: Various track parameters reconstructed using the Road Search (red) and Cosmic Track Finder (blue) track fitting algorithms. Track Transverse Momentum (top left), Number of reconstructed track (top right) and Number of reconstructed hits per track (bottom).

user or an automated process

As discussed in Section 4.2, two packages were developed. One to monitor the global track parameters, which are objects that define a track. The second one to monitor the quality of the track with respect to the hits read by the individual detector modules.

Both these tasks were successfully implemented and tested. The test results show that the data generated by the Tracker_Monitor_Track can be used to monitor the quality of the tracks of the tracks being produced. The results of the test have showed any alteration in the detector modules causes a change in the distribution of the data which can then be used to identify a problem.

The package was tested using cosmic rays collected during the Tracker commissioning at TIF. The data from the TIF has showed that the package has fully integrated with CMSSW framework to generate data that was useful in the commissioning of the tracker especially in identifying bad quality data. The package has showed it provides essential information that help identify when things go wrong as can be seen from section 4.6.2.

Chapter 5

Monitoring of Tracks during the Magnet Test and Cosmic Challenge

The CMS detector is highly complex, which means that a lot of unforeseen problems can occur during the running of the experiment, especially at the start. Therefore a test where all the sub-detectors can be checked before the start of the experiment was essential. The commissioning of the solenoid magnet was seen as the best opportunity to perform this type of test[46]. In the summer of 2006 the solenoid magnet of the CMS was scheduled to be installed. Taking advantage of this, the CMS collaboration prepared the partially installed sub-detector systems for a so-called cosmic challenge, where cosmic rays were used to test the sub-detectors. One such sub-detector was the Silicon Strip Tracker (SST). Chapter 3 of this thesis gives detailed information on the SST. The Pixel Tracker, although part of the whole tracker system, was not installed for the Magnet Test and Cosmic Challenge (MTCC) test.

Testing the Tracker during the MTCC was crucial to study the performance of the tracker with the magnetic field and the prototype of the CMS software (CMSSW) which will be used for the data taking, reconstruction and quality monitoring. This was a good opportunity to test the Tracker_Monitor_Track package of the Data Quality Monitoring system as part of CMSSW. As described in Chapter 4, the package has two modules, the Monitor_Track_Global and Monitor_Track_Residual. Both modules were ready for testing at this time. This chapter will discuss how these packages were useful during the test. It will also give a description of the tracker layout and some general information about the trigger system. Furthermore it will discuss the usage of monitoring elements during the MTCC and the performance of the tracker in comparison with the muon system.

This chapter is a summary of a CMS NOTE [46]. My personal contribution during the MTCC included data taking and analysis of the acquired data for data quality monitoring. The work involved managing the data quality monitoring software package discussed in Chapter 4. The test has enabled the further development of the software as it was being tested in real time with real data coming from the partly installed CMS detector. The purpose of the Tracker test during the MTCC and the results achieved along with discussion on these results are discussed in this paper. The tests were conducted by the Tracker group at SX5 point of the LHC ring where I worked as a shift personnel for data-taking and data quality monitoring. Therefore, unless specified, all information including the figures in this chapter are taken from this paper.

5.1 The Tracker setup for MTCC

During the MTCC, the Tracker's fully commissioned read-out electronic channels represented 1% of the nominal Tracker setup which is around 10^7 read-out channels. The active silicon sensor coverage area was 0.75 m^2 of the nominal CMS Tracker setup which will cover an area of 200 m^2. Parts of the Tracker system, TIB, TOB and the TEC were installed. The modules were arranged in space so that they would represent each sub-system of the SST with the exception of the TID (see Figure 5.1). A detailed description of the various parts of the SST can be found in Section 3.1. The TIB structure setup for the MTCC consisted of two prototype mechanical shells that represented Layers 2 (L2) and 3 (L3). The L2 contained 15 double sided modules and the L3 contained 45 single sided modules. The TOB structure represented layer 1 (L1) and 5 (L5). Each L1 rod of the TOB contained six single sided detector modules with strip_pitch 183 μm. The L5 rods contained six single sided modules with strip_pitch 122 μm. The TEC represented the outer 3 disks (8, 9 and 10) of the final End Cap Detector. The details of the detector modules used in the MTCC are listed in table 5.1.

In addition to the Tracker, several other sub-detector systems were also installed. This included 5% of the ECAL final system super-modules, 15 Hadronic calorimeter wedges making the 10% of the final system, and 14 Drift Tube (DT) and 36 Cathode Strip Chambers (CSC).

Tracker	Layer/Ring	Position		Module			Number
Sub-detector		$r(cm)$	$z(cm)$	Type	Pitch (μm)	N. of Channels	of modules
TIB	Layer 2	32.2-35.6	2.9-60.6	$r\varphi$	80	768	15
				stereo	80	768	15
	Layer 3	40.3-43.4	7.5-59.4	$r\varphi$	120	512	45
TOB	Layer 1	59.1-62.9	8.9-98.6	$r\varphi$	183	512	12
	Later 5	94.6-98.4	8.9-98.6	$r\varphi$	122	768	12
TEC	Ring 4	56.2	270-278	$r\varphi$	113/143	512	7
	Ring 5	67.7	267-274	$r\varphi$	126/156	768	5
				stereo	126/156	768	5
	Ring 6	81.9	270-278	$r\varphi$	163/205	512	7
	Ring 7	99.2	268-275	$r\varphi$	140/172	512	10

Table 5.1: Modules mounted in the MTCC Tracker structure

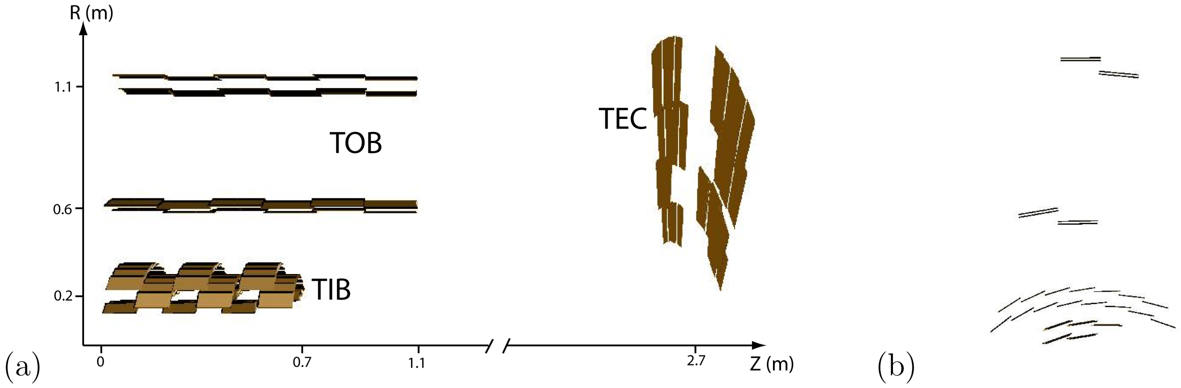

Figure 5.1: Layout of the Tracker MTCC setup: (a) rz 3D view; (b) xy view. The instrumented parts are a fraction of layer 2 and possible of TIB, two rods in layer 1 and in layer 5 of TOB, two petals in disk 9 of TEC.

5.2 Triggering and Data Acquisition

The Level 1 trigger during the MTCC was mainly done by the muon system. This method resembles the triggering that will be used by the CMS detector when it is fully commissioned for data taking during the LHC running. The trigger system of MTCC was set up so that all the triggering signals were routed to a central system. The central system used its triggering logic to make decision on Level 1 triggering. In the case where a Level 1 trigger is decided a global signal was sent out to all the sub-detectors for data readout.

The triggering signals received by the central system came from the Drift Tubes, the Cathode Strip Chambers, Resistive Plate Chambers and the Hadronic calorimeter (HCAL). As it was possible to use the Drift Tubes and the Resistive Plate Chambers to identify the direction of incoming muons, they were useful in selecting muons that were approximately pointing towards the partially installed Tracker. Although triggering of muons pointing towards the tracker was possible, only a few of these muons passed through the tracker.

During the MTCC the tracker readout system used the central data acquisition system for the first time. The process was smooth apart from very few spurious errors on the readout, which was resolved after the MTCC was completed. The raw data from MTCC was transferred to the FNAL Tier 1 center using the CMS computing tools [55] where the raw data was converted into a CMSSW compatible format. The data was further processed at FNAL remote operations center where further reconstruction was done and the data sent back to CERN and stored in CASTOR.

5.3 Track Reconstruction

The MTCC track reconstruction algorithms followed a similar procedure to that described in Chapter 3, which is based on the Kalman filtering technique (see Section 3.3). The reconstruction method starts with seed generation followed by pattern recognition and finally track fitting. Although they are similar there is a slight difference between them. The standard algorithm is designed to reconstruct tracks originating from the center of the tracker, whereas the cosmic rays enter the detector from above. Therefore the algorithm has to take this in to account. Two algorithms are used in the reconstruction of cosmic rays: the Cosmic Track Finder[1] and the Road Search.

5.3.1 Cosmic Track Finder Algorithm

The Cosmic Track Finder [56] was developed for the sole purpose of reconstructing cosmic tracks. This algorithm is a variation of the standard track fitting algorithm of CMS, the combinatorial track finder. It reconstructs single tracks without imposing a region of origin, but it assumes a preferred direction. In the case of cosmic track finder the seed generation (see Section 3.3) uses any pair of reconstructed hits from different layers. This is different from the the combinatorial track finder used for tracking particles from pp collisions where three points are used along with the tracking region constraint (see Section 3.3.2). The method used by the cosmic track finder does not decrease the speed of the track reconstruction due to its extensive hit search, the reason for this being that cosmic rays yield a considerably lower number of reconstructed hits when compared to the pp collisions of the LHC. This is because the number of cosmic rays that will pass through the sensitive part of the CMS detector are very small compared to the number of particles generated during the full luminosity of the LHC operation.

The pattern recognition component of the algorithm starts by ordering all the reconstructed hits with respect to the vertical direction, which is the y coordinate according to the CMS reference system. The ordering of the reconstructed hits along the y coordinate is then used by the algorithm to define track candidates. The trajectory of the track candidates is then used to propagate their path to the next layer where a compatible reconstructed hit is sought for and added to the track candidate if found. The algorithm then tests the compatibility of the added reconstructed hit with the trajectory of the track candidate using the χ^2 estimator. The maximum allowed χ^2 is not a fixed value, and is a adjustable parameter in the estimator. Tracks reconstructed using this algorithm which are used in the MTCC analysis are chosen according to:

[1] It has been decided in this chapter not to abbreviate cosmic track finder so that not to create confusion with combinatorial track finder which is generally abbreviated by CTF. A slight inconsistency is created due to this as the Road Search is abbreviated by RS. It was believed the advantage of not creating confusion outweighs the inconsistency.

- largest number of layers with hits in the trajectory (all four layers if possible)
- large number of hits in the trajectory
- smallest χ^2 value.

5.3.2 Road Search Algorithm

The second track reconstruction algorithm used during the MTCC is the Road Search algorithm (RS). This algorithm, like the combinatorial track finder, is designed to track particles originating from the center of the CMS detector which is the interaction point of the pp collisions. Both the combinatorial track finder and the RS use a Kalman filter (see Section 3.3.3) for the track fitting, but their difference lies in the first steps of seed generation. In the case of RS the algorithm uses "roads" which are predefined detector modules. To form a road, first rings are assumed. Rings are all detector modules that belong to the same $r - z$ plane of the tracker. Two such rings are then used to define a road seed. One of the two rings used to define a road seed must have greater radius. The road seed is then used to find more rings that will be used to define a road that will be used in searching compatible hits in the tracker. When reconstructing tracks using this algorithm, two hits are taken only if they belong in the rings that are used to make a road. The track path is then extrapolated by searching for more hits in rings of the same road.

5.4 Alignment of the Tracker using Hit Residuals

Before the tracker was integrated with the rest of the CMS sub-detectors, it was commissioned at a designated place in CERN called the Assembly Hall. During this time cosmic data was used to check the proper functionality of the tracker. Scintilators placed above and below the tracker barrel were used for triggering. Detailed information on the set up for the commissioning is given in Section 2.1 of[46]. A standalone algorithm for calculating the correction of relative position of TIB with respect to TOB was used. The correction algorithm used three parameters, two translations and one rotation, to correct the position. These are: a shift Δx along the x axis, a shift Δy along the y axis and a rotation $\Delta\phi$ around the z axis.

A sample of 12,340 events was taken, out of which only 3,155 were used for the correction. The reason for the high reduction in the number of events is that only events with hits on all four layers of the tracker barrel section were used. Figure 5.2 shows the hit residual of the data taken before and after correction.

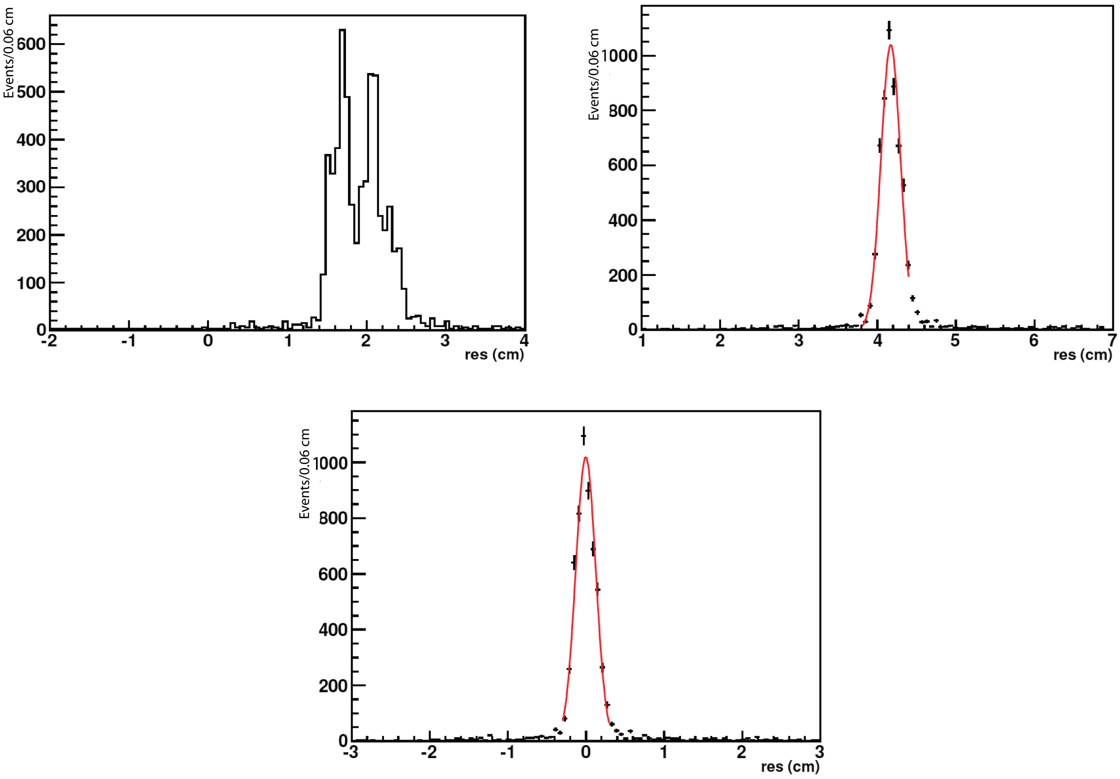

Figure 5.2: TIB residual top left before any correction ($mean = 1.901$), top right after rotation ($mean = 4.191$) fitted with Gaussian (red), bottom after rotation and translations ($mean = 0.02182$) [46].

The rotation of TIB with respect to TOB was done to correct the double-peak on the hit residual distribution that can be seen on the leftmost graph of Figure 5.2. After correction the two peaks merged as can be see on the far right of Figure 5.2. The large shift in the x (the residual axis on the histograms) is the result of the low precision of TIB and TOB structures setup for the MTCC, and does not reflect the ultimate precision of the final tracker setup. After the translation using the shift along Δx and Δy the residual distribution is centred at zero as can be seen from the bottom graph.

The final alignment of the tracker, when installed in the CMS detector during MTCC, was obtained using the Hits and Impact Point (HIP) algorithm. This algorithm aligns individual detector modules to a high precision with respect to each other. The algorithms works by iterating over event samples to improve the alignment resolution of individual detector modules [57]. Using this alignment improved track quality and an increase in the number of reconstructed track was achieved. This can be seen in table 5.2. Here the survey is done using the previously accumulated data during the the MTCC period. The tracks with alignment show a lower χ^2 (which means a better fit), smaller hit residuals and an increased number of hits and tracks. The study of the track reconstruction performance discussed in the next section uses the track data obtained after the above alignment was implemented.

Alignment status	# rec. tracks	$\langle\chi^2\rangle$	⟨#of hits⟩	res. TIBL2 mono [μm]	res. TIBL3 mono [μm]	res. TOBL1 mono [μm]	res. TOBL5 mono [μm]
No alignment	1460	20.1	3.3	526	416	2660	1986
Preliminary alignment	3263	16.5	4.0	518	387	1547	1999
Alignment without survey	4894	6.5	4.3	208	135	389	710
Alignment with survey	4956	6.0	4.3	177	125	357	687

Table 5.2: Most sensitive track quantities for three different alignment conditions. All the numbers are evaluated for tracks with hits in 3 or more layers.

5.5 Data Quality Monitoring and the MTCC

For the test during MTCC, the data quality monitoring (DQM) system was run offline (see Chapter 4). This was possible as the data generated during this test by the tracker was very small (both in rate and volume) in comparison to the expected data to be generated by the *pp* collisions and all the triggered data was saved. When a DQM package is running offline the Source packages are the only part of the system running. The Collector and Client (see Section 4.1.3) parts of the system are not needed as the Monitoring Elements can be analysed interactively from a file generated by the DQM Source package.

One package of DQM used during the MTCC was the Tracker_Monitor_Track described in Chapter 4 which is used to monitor the quality of tracks. At the end of each day, the shift personnel analysed the data to make a decision on whether to accept or reject the data collected for a single run. This decision was made based on the information provided by the DQM software. Tracks with low quality generally have high χ^2 value or a low number of reconstructed hits. The limit on the value of the χ^2 is not fixed. The number of reconstructed hits for a given track should be at least two if the track is to be accepted, as long as its χ^2 value is not too large. Figure 5.3 shows typical data that was taken at the end of each run. In this figure the histograms show the important parameters of the track in determining its quality: the number of reconstructed hits per track, the track angle, and the track pseudorapidity. In this particular case, the generated tracks were behaving as expected as the number of reconstructed hits is greater than two and the track ϕ and η parameters correspond to the direction of tracks expected to be entering the tracker from above and in the region of the tracker which was installed respectively. The number of reconstructed hits were between 2 and 4 which corresponds to the tracker geometry which has four layers in the barrel section and for tracks which came through the TEC section of the Tracker, would have 3 disk parts. The lowest number of hits allowed by the track reconstruction algorithms was 2. This was due to the fact that high momentum cosmic rays will have a straight line through the tracker, and two hits are enough to start searching for track seeds.

5.6 Track Reconstruction during MTCC

One of the goals for undertaking the Cosmic Challenge during the MTCC was to study the performance of the sub-detector modules. This chapter will only discuss the Tracker data

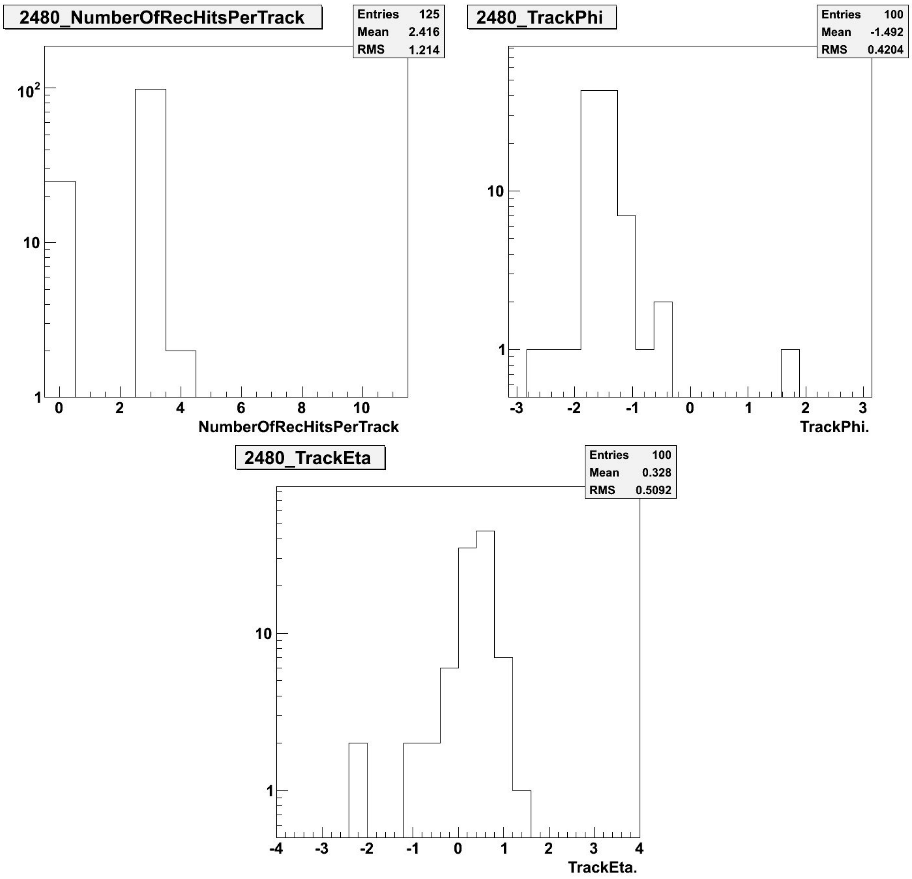

Figure 5.3: Histograms of Global Track Parameters on MTCC data for a run. A run is defined by continuous data taking of events occurring while the tracker is operational.

as the aim is to focus on the achievement of the Tracker_Monitor_Track package during MTCC. Although the comparison had little to do with the Tracker_Monitor_Track package, it was was felt important to give a summary of Tracker performance during MTCC.

During MTCC data was taken both with and without the magnet field. The number of tracks reconstructed using the Cosmic Track Finder and Road Search algorithms after the alignment was applied are given in table 5.3. The number of reconstructed tracks by the Road Search algorithm are less than the ones reconstructed by the Cosmic Track Finder. This is due to the limited geometrical acceptance of the Road Search algorithm.

Figures 5.4 and 5.5 show the "most interesting" quantities of cosmic muon tracks in the

	$B = 0.0\,\mathrm{T}$	$B = 3.8\,\mathrm{T}$	$B = 4.0\,\mathrm{T}$
Cosmic Track Finder	5108	3588	583
Road Search	4737	2343	267

Table 5.3: Number of reconstructed tracks for the Cosmic Track Finder and the Road Search algorithm in the different data samples. The smaller number for the Road Search algorithm is the result of a limited geometrical acceptance of the tracking region in comparison with the slightly lenient one of the cosmic track finder.

$B = 3.8$T data samples. These histograms were generated using the Monitor_Track_Global package that was run in offline mode. The smaller number of reconstructed tracks using the RS algorithm is due to the track seed generation requiring an inner hit in TIB layer 2 and an outer hit in TOB layer 1 or 5. This means if a track has hits on layer 3 of TIB and layers 1 and 5 of TOB, then the RS algorithm will reject this track.

Both the RS and cosmic track finder algorithms have achieved similar results despite the difference in the number of reconstructed tracks. In Figure 5.4 the top figure shows the ϕ distribution of the reconstructed tracks. Here the distribution peaks at $-\pi/2$ which is consistent with tracks which originate from the top of the tracker and travel outside in. The η distribution of the tracks is consistent with the position of the trigger layout during the MTCC. The other quantities of a track parameter (given in Figure 5.5) such as the track transverse momentum, track χ^2 and number of reconstructed hits per track show that both algorithms (cosmic track finder and RS) give similar results. The χ^2, which is a very important parameter in showing the quality of the fit, gives a low value showing good fit.

5.7 Comparison of Track Reconstruction in the Tracker and the Muon System

The tracks reconstructed using the muon system and the Tracker were compared against each other to analyse their relative performance. For the Tracker, tracks reconstructed

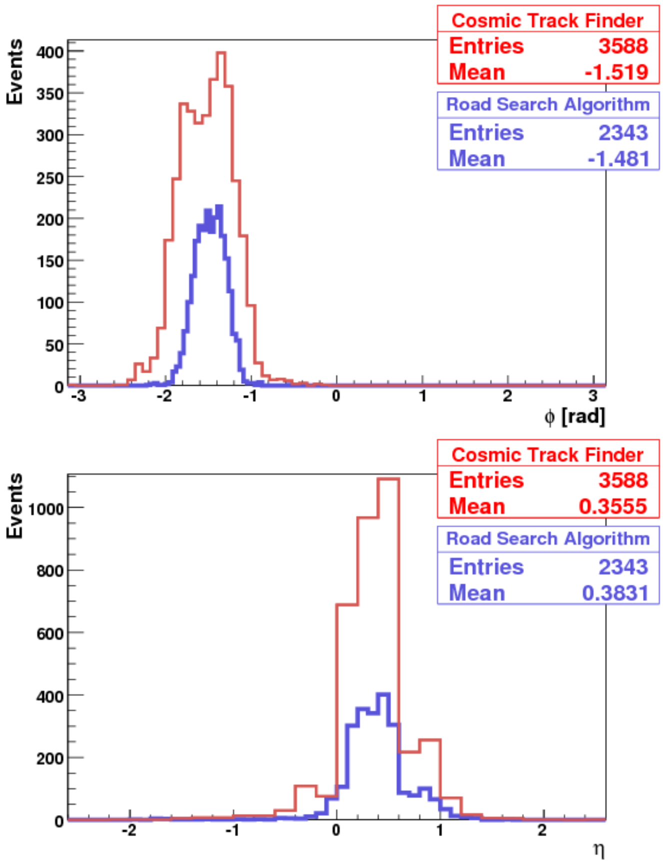

Figure 5.4: Track ϕ (top) and Track η (bottom). Distributions of tracks reconstructed using the Cosmic Track Finder (red) and the Road Search Algorithm (blue) using the $B = 3.8\,\mathrm{T}$ data sample. The distributions for the Cosmic Track Finder are shown for tracks with hits in at least 3 layers [46].

using the cosmic track finder were used. For the muon system, tracks reconstructed using the Drift Tube were used.

The far left of Figure 5.6 shows the correlation of the measured track ϕ of the tracker (ϕ_{tk}) and of the DT ϕ_{DT}. Here both measurements are correlated showing the performance of both sub-detectors was similar. In the case of η (right on Figure 5.6) there is less correlation in comparison to ϕ, whereas when only tracks with hits on all 4 layers were used, the correlation improves (bottom on Figure 5.6). The data discussed above were taken without the magnetic field.

A further analysis for the data taken with the magnetic field on was done. The top part of Figure 5.7 shows the difference in ϕ between tracks reconstructed by the Tracker (ϕ_{tk}) and the muon system (ϕ_{DT}), which decreases with increasing transverse momentum and where oppositely charged muons give different signs. The η correlation however is similar to the tracks reconstructed without the magnetic field. Again in Figure 5.7 lower graphs the correlation of the left graph is improved on the right hand graph where using hits from all the barrel layers of the Tracker improves the track quality.

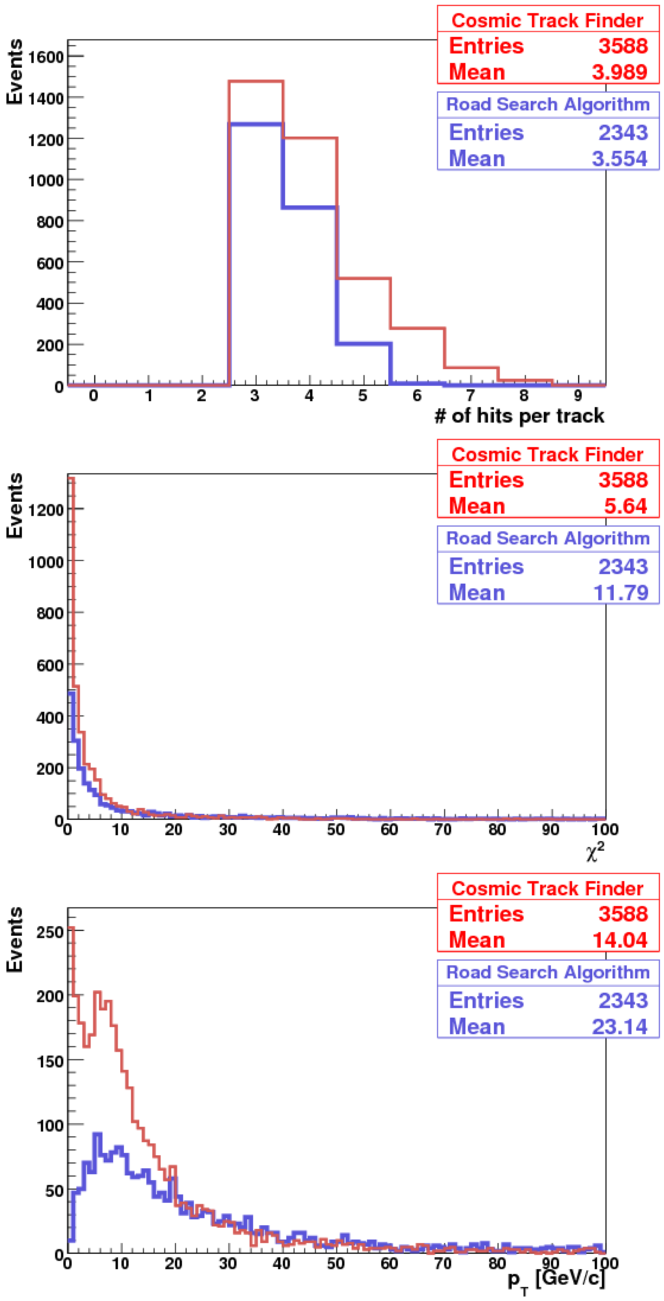

Figure 5.5: Number of reconstructed hits per track (top), track χ^2 (middle) and track transverse momentum (bottom) for tracks reconstructed using the Cosmic Track Finder (red) and the Road Search Algorithm (blue) using the $B = 3.8\,\mathrm{T}$ data sample. The distributions for the Cosmic Track Finder are shown for tracks with hits in at least 3 layers [46].

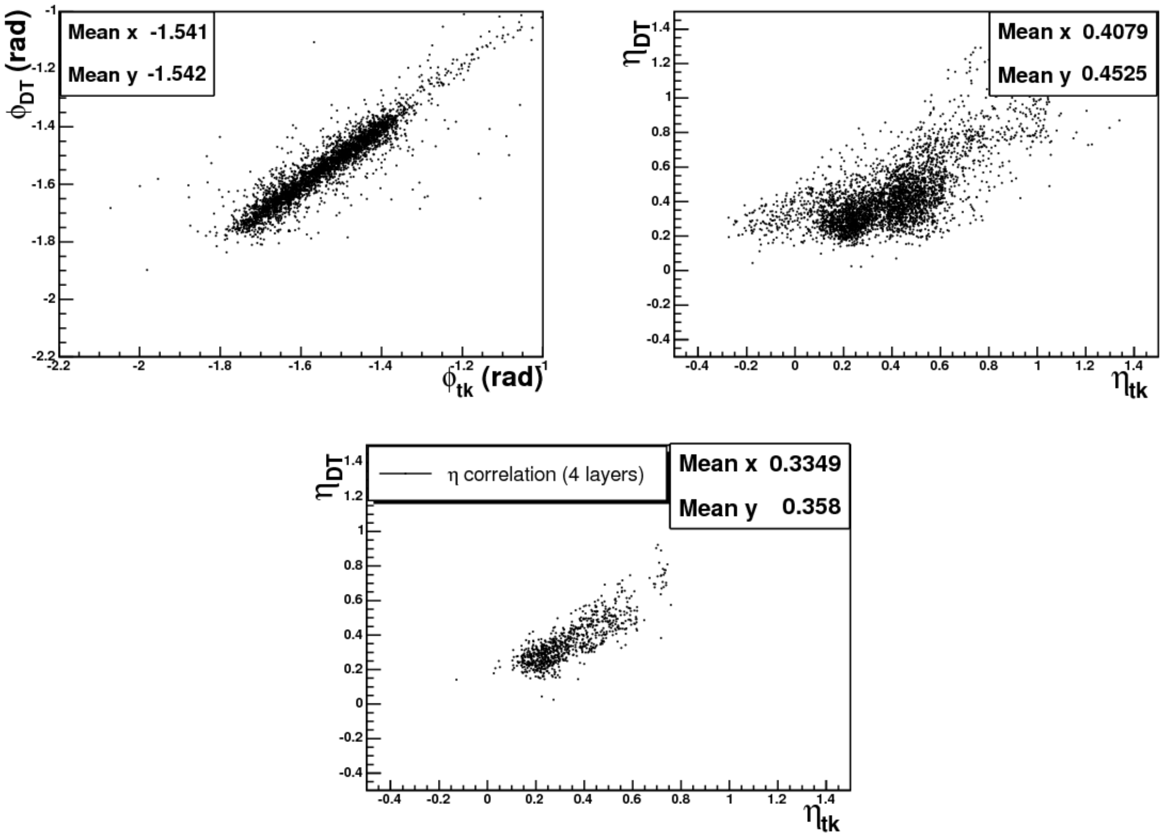

Figure 5.6: Correlations of the directions of tracks, in absence of magnetic field, reconstructed in the Tracker with those reconstructed in the Drift Tubes. On the top left the ϕ correlation is shown, while the top right plot gives the η correlation for all tracks and the bottom plot shows correlation for all tracks with hits in 4 layers. [46].

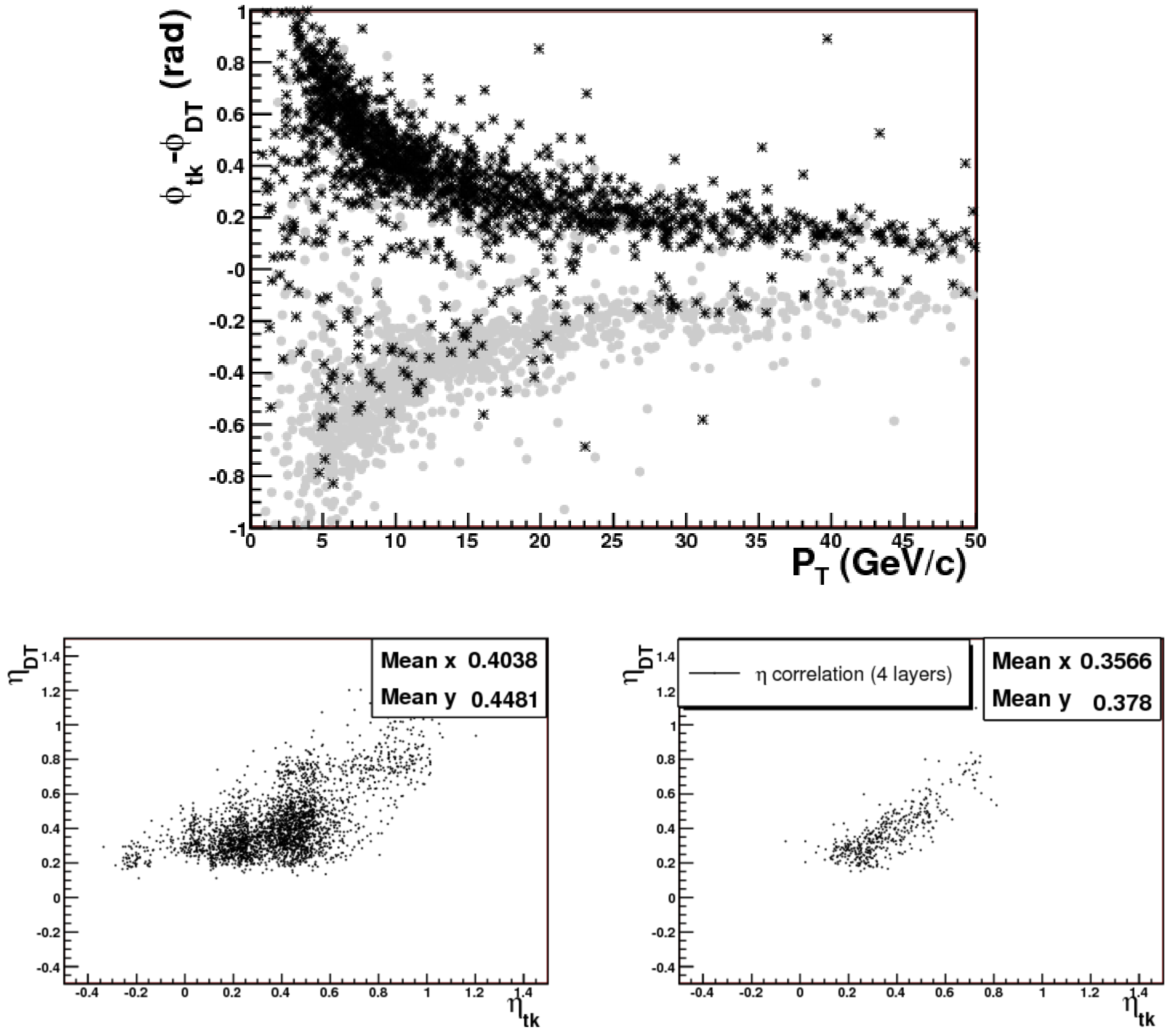

Figure 5.7: Correlations in the direction of tracks, as reconstructed in the Tracker or in the Drift Tubes, for $B = 3.8\,\mathrm{T}$. On the top the difference $\phi_{tk}-\phi_{Dt}$ is correlated to the transverse momentum measured by the Tracker for positive (black) and negative (grey) muons. On the bottom left (right) plot the η correlation for all the tracks (for tracks with hits in 4 layers) is shown [46].

Chapter 6

Quality Test Algorithms and Techniques

This chapter discusses the research done towards finding a suitable two-dimensional statistical test for the data quality monitoring system. The research documented in this chapter has lead to the implementation of a statistical test for the DQM_Services package of the CMSSW framework. After introduction to the motivation for this research, Section 6.2 introduces basic statistics and methods used. Sections 6.3 and 6.4 discusses the methods of implementing some statistical tests and their application for the DQM system. And finally Section 6.5 will discuss the tests on the implemented algorithms along with the results.

6.1 Introduction

The observed data from the CMS Tracker will be used to monitor the quality of the reconstructed tracks and performance of the detector. One way of doing this is to compare the observed data's distribution with a theoretical distribution. In modern experiments theoretical distributions are usually given by Monte Carlo (MC) simulations.

For the CMS experiment the MC data is generated using the GEANT4 [58] model of the CMS. The CMS collaboration has chosen to use the Detector Description Language (DDL) [59] to provide the geometry description of the detector [60]. The geometry created by DDL is used to simulate both the sensitive and "dead" materials. The Tracker geometry simulation is included in CMSSW framework in the form of XML files. This is used to simulate and reconstruct *pp* collisions. The GEANT4 model of the Tracker can be seen in Figure 6.1. Figure 6.2 shows the details (both active and "dead" materials) included in the simulation by taking a TOB layer as an example. There are several general-purpose event generator packages that use this model to generate MC simulation data. Some of these are Pythia [61], Herwig [62] and Sherpa [63]. The objective of the general-purpose event generators, such as Pythia, is to provide a description (as accurately as possible) of what happens during particle collision. At present the main workhorse of CMS is Pythia6.

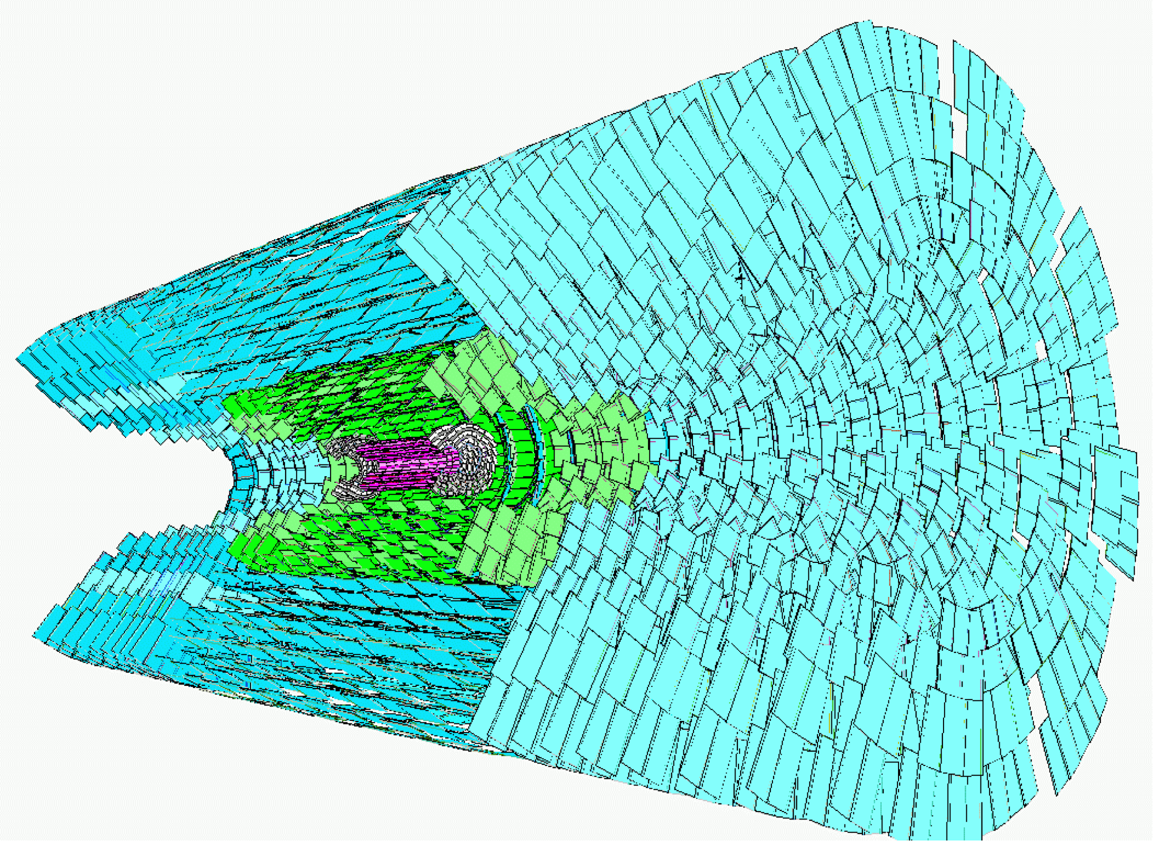

Figure 6.1: The Tracker model in the GEANT4 toolkit as viewed using the IGUANA visualisation software (top) [64].

One of the purposes of making measurement is to verify a theory and the simplest method of achieving this is by using hypothesis testing. The MC simulation data generated by the GEANT model of CMS provides information on how the detector behaves

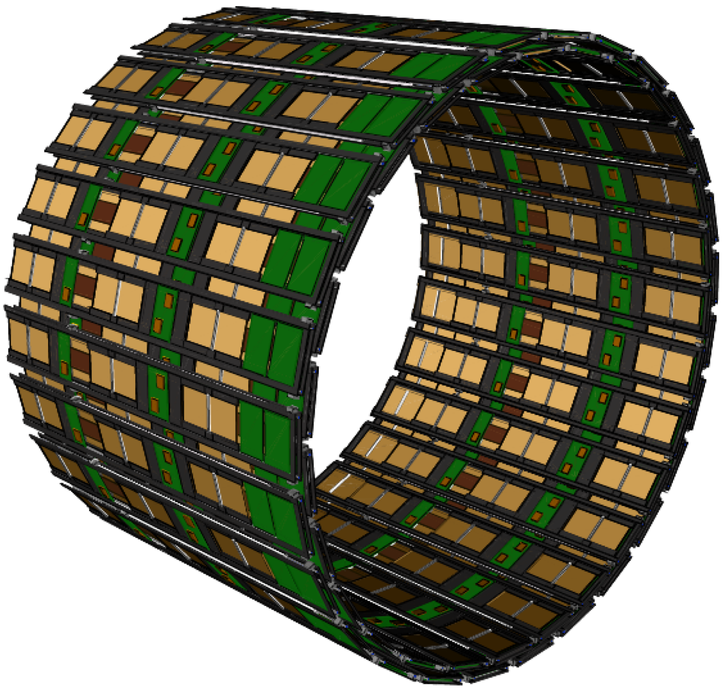

Figure 6.2: The GEANT model of the CMS contains details of active detector modules (golden coloured) and "dead material" which include the readout electronic systems and the support structure as can be seen from the GEANT TOB model [60].

under normal conditions. When generating a simulation data using the GEANT model of the CMS, theoretical parameters are used to generate well understood data. Using hypothesis testing along with a statistical test one can check if measured data from the tracker is consistent with the MC data. The result of this test is then used to make decision on whether to accept or reject the data collected from the CMS detector. Several statistical methods are available that provide such tests. The χ^2 test which depends on binned data is popular among particle physicists. There are other several tests such as the Kolmogorov-Smirnov (KS) and minimum energy tests which do not need binned data which are discussed in this chapter.

At present all the statistical tests for DQM are ported from ROOT[1] but the statistical test tools for multivariate data are not ported at the moment. This poses a challenge for

[1] ROOT is a statistics software popular in the High Energy Physics Experiments [65]

the track quality monitoring package discussed in Chapter 4 which provides two types of data, the hit residuals which are one dimensional data sets and global track parameters which contain both one and two dimensional data sets. The fact that test for bivariate data does not exist in DQM added with the other fact that the bivariate statistical test implementations of ROOT have limitations prompted a reason to study a bivariate test that can give reliable results. The other challenge comes from the fact that the data generated by the DQM is given in histograms which is binned data. This was found to be limiting the exploration of various statistical tests that use unbinned data. We found unbinned data to give more reliable results when using statistical test especially when the data number is small. This is given in the testing and discussion section of this chapter. Unbinned data is sometimes referred to as "raw data". To prevent confusion only the term unbinned data will be used.

6.2 Statistical Analysis of Data

Let $\mathbf{X} = \{x_1, x_2, x_3, ...x_n\}$ be a sample set, where "n'' amount of strings were taken randomly from production line and had their lengths measured. Assuming the machine is producing strings with the same length, one way of estimating the aimed length would be by taking the average mean ($\bar{x}$) of the sample given by

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6.1}$$

Where x_i are the individual measurements. Assuming the distribution of the measured data is Gaussian, the standard deviation will give us on average by how much the individual measurements deviate from the estimated $\bar{x}$. The standard deviation, we will call it σ from here on, is given by

$$\sigma = \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \tag{6.2}$$

Using the information above the length of the intended length of the strings can be predicted with certain confidence. As it is assumed the distribution of the length of the strings to be Gaussian, it can be said that the length of the strings falls in the range of

$$\bar{x} \pm 1.645 \cdot \sigma \tag{6.3}$$

90% of the time. Figure 6.3 shows what the distribution of the string length measurement might look like if it is a Gaussian distribution and the various levels of confidence that we could have on the average length.

σ is sometimes known as the standard error (se) due to the fact that the deviation is a reason of various factors. These can be errors of the measurement apparatus, inaccuracy of the machine and human error [47]. The details of the source of these errors and how they are dealt with will not be discussed in this chapter but it is given in detail in [66]. Knowing the length of the string with certain confidence is only one of the uses of such statistical analysis. One other use would be to check the quality of the product in terms of its length over a period of time. To achieve this one would want to store data taken at a given time for reference to be compared with data that would be taken in the future. The two data samples can then be compared to see if they are consistent. This would give rise to what is known in the statistics world as the two-sample problem. This is a challenge faced by modern experiments; how can one determine if the distribution of sample data is consistent with the MC generated data.

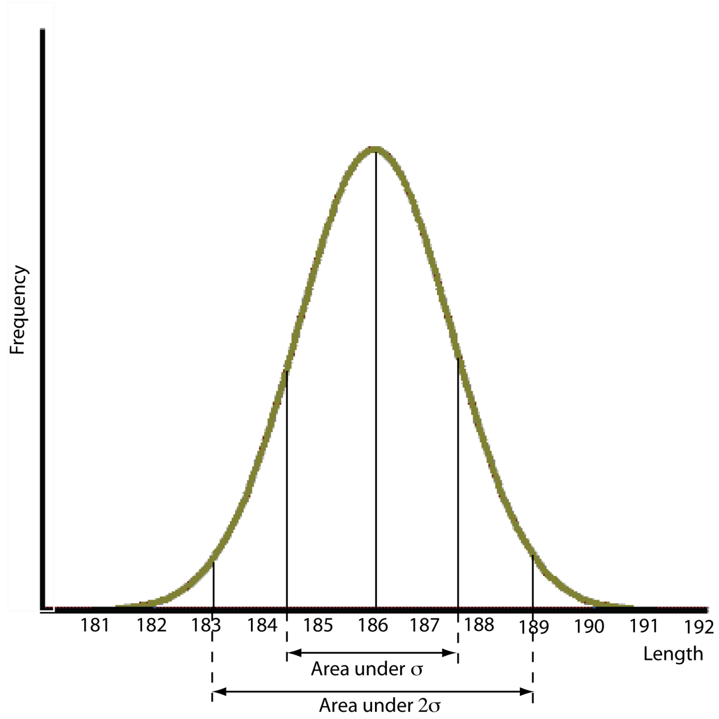

Figure 6.3: A Gaussian distribution showing string length in mm. The area under the 2σ covers $\approx 95\%$ of the string population in the sample.

6.2.1 Two-sample problem

Let $\mathbf{X}$ be a sample set of string length measurements taken two years ago and $\mathbf{Y}$ is a set of measurements taken yesterday. Now we want to find out if the machine is producing strings with lengths that are consistent with what it used to produce two years ago. A reasonable way to do this is to check if the distributions of sample $\mathbf{X}$ and $\mathbf{Y}$ are consistent with each other by comparing their means. The advantage of representing data with a single number or combination of single numbers such as the $\bar{x}$ and σ is that comparison of two data samples is made easier. Of course looking at the $\bar{x}$ and σ might not be enough in certain cases as the number of samples taken has an effect on the accuracy of the estimated $\bar{x}$. So further study into the error of the $\bar{x}$ might be useful using

$$V(\bar{x}) = \frac{\sigma}{\sqrt{N}} \tag{6.4}$$

where N is the number of samples taken. But assuming that enough data samples are taken so that the error on the mean becomes insignificant, then a reasonable test could be comparing the average mean of the two samples. Let $\bar{x}$ be the mean of **X** and $\bar{y}$ be the mean of **Y**, then a test such as $x - y$ can be used to check if there is a significant difference between **X** and **Y**. This test in itself will not be enough, as the average deviation of the individual data samples from $\bar{x}$ and $\bar{y}$ is not taken in to consideration. Therefore we use equations 6.5 to check if the result of the mean difference is significantly high enough in comparison to the deviation for us to believe the machine is producing strings of different size compared to two years ago.

$$\hat{\theta} = \frac{\bar{x} - \bar{y}}{\sqrt{(\sigma_x)^2 + (\sigma_y)^2}} \tag{6.5}$$

Here σ_x and σ_y are the standard deviations of **X** and **Y** respectively. This gives a value, $\hat{\theta}$, that can help make the decision on whether the difference in the mean is significantly high or not. If the value of $\hat{\theta}$ turns out to be small, it can be concluded that the difference, if there exists any, is insignificant to prove there is a difference in the average string length. This raises a further problem on how big or small $\hat{\theta}$ should be to accept or reject the difference in the mean.

In the above example the test was simplified by the fact that the distributions of both the test samples was assumed to be Gaussian. Although it is common to see in particle physics experiments where the r.m.s of sample data is determined through a fit to a Gaussian, this is not always true [67]. In the case where the distributions under test are not Gaussian, using σ becomes less significant as the sigma will not correctly define the variance of the data from the mean [68]. To help make this decision the hypothesis testing method can be used. Both the issues raised above are addressed by using two methods, namely the hypothesis testing and Bootstrap.

6.2.2 Hypothesis Testing

When faced with two-sample problem, one can use a statistical test such as goodness of fit test to achieve a value that says something about the two data samples. The value given by a goodness of fit test is similar to the $\hat{\theta}$ given in Equation 6.5. The decision on what value $\hat{\theta}$ should have is not a straight forward answer. Hypothesis testing is used to help make a yes or no decision when faced with this problem.

In a hypothesis testing one has to set a hypothesis of what is expected. In the example of strings one can say that the quality of the machine has degraded and now the machine is giving different string lengths than what it used to give two years ago. To formalise this let $F(x)$ be a function that describes the data distribution from two years. And $G(x)$ the most recent data. Then the hypothesis test starts by assuming a null hypothesis (H_0) such that

$$H_0 : F(x) = G(x), \text{ for every } x \in \mathbf{R}^d$$

against the alternative hypothesis (H_1)

$$H_1 : F(x) \neq G(x), \text{ for some } x \in \mathbf{R}^d$$

H_0 is a hypothesis that the two distributions are the same, which is the complete opposite of what is being tried to be proved. In this example H_0 would be the case where both data samples give the same average length. For the hypothesis stating that the two distributions are different to be accepted, H_0 has to be completely rejected. If H_0 can not be completely rejected then H_1 can not be accepted.

Both H_0 and H_1 in this case are distribution of values ($\hat{\theta}$) given out by a statistical test of a choice. H_0 is the distribution of $\hat{\theta}$ when both samples are similar and H_1 is its distribution when the samples are different. So when a statistical test is done, the hypothesis is either accepted or rejected based on where $\hat{\theta}$ falls (H_0 or H_1). In other words if the $\hat{\theta}$ falls in H_0 distribution the hypothesis is rejected or if it falls in H_1 it is accepted.

Ideally H_0 and H_1 should be completely different making the decision easy. But this is not the case in reality and there is usually an overlap between the two. Looking at Figure 6.4 which assumes that the distribution of the H_0 and the H_1 are known, an overlap is observed. When there is an overlap a separation between the distributions is made by taking a point where one ends and the other begins. This is called critical point and is noted by $T_{critical}$ in Figure 6.4 where the value is decided by convention (generally either 0.1 or 0.05). In Figure 6.4 if the value of $\hat{\theta}$ falls in the region to the left of $T_{critical}$ then the H_0 is accepted and the hypothesis is rejected and if it falls to the right of $T_{critical}$ then the hypothesis is accepted.

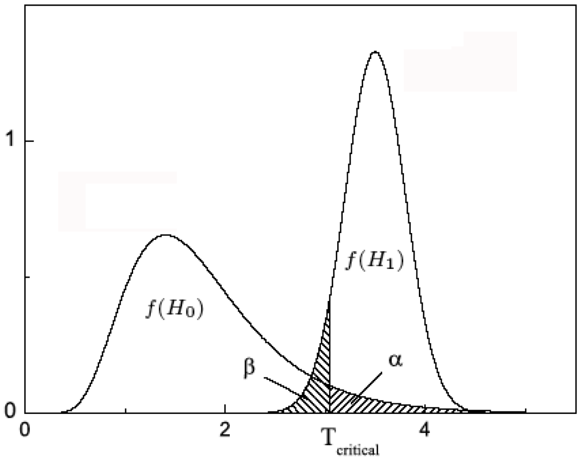

Figure 6.4: The distribution of the H_0 and the H_1 (Adapted from [69])

When making decisions using $T_{critical}$ two types errors arise. The first one, called the $TypeI$ error, is when a true hypothesis is rejected and is denoted by α in Figure 6.4. It is known as the significance level. The second one , called $TypeII$ error, is a false hypothesis is accepted. It is denoted as β in Figure 6.4. $1 - \beta$ gives the power of the test. Power of test is useful when trying to find out how good a statistical test is. High value of the power shows the test is good. For a good test both α and β need to be small. Trying to improve one will increase the other one, therefore it is usual practice to fix the significance level of the test and based on that to accept or reject the hypothesis.

But the issue of having H_0 and H_1 overlapping does not end here. If both have Gaussian distribution then one can say state the confidence level at which the hypothesis was accepted or rejected using Equation 6.3. But these distributions can not always be approximated by Gaussian as will be seen further in this chapter with the energy test.

6.2.3 Confidence Level

Confidence level is based on the probability density function of a data distribution. It is the acceptable range of magnitude of an estimate. Going back to the example of string length measurements, the estimator is the $\bar{x}$. So one can say that if a string was to be taken randomly from the production line, its length will be $\bar{x} \pm \sigma \cdot percentile\_point$, where the percentile point can be found on a table. For example it can be said that 95% of the time the length will lie between $\bar{x} \pm 1.96 \cdot \sigma$. For normally distributed data sample such as the one in Figure 6.3, the confidence level on the statistical estimate and the σ are closely related.

For data samples with a pdf other than the normal distribution, the relationship between the σ and the confidence level becomes meaningless. A good example of this would be a skewed pdf. One suitable alternative to avoid such problem is by using the bootstrap percentile method, where the data sample is ordered and empirical cumulative distribution is computed. The required confidence level is then calculated first by deciding percentage of the data to be considered and then the values where the selected data lies on are computed.

6.3 BOOTSTRAP

The bootstrap is distribution independent statistical test. Due to its iterative nature bootstrap is computationally expensive. For this reason it has only emerged in recent years as powerful computers have become affordable. This section will introduce bootstrap, but if extra information is needed the method is explained in detail in [68].

The method achieves powerful statistical tests by resampling a given data sample. To explain the method let $\mathbf{X}$ be a data sample with elements $\{x_1, x_2, x_3, ...x_n\}$. The process starts by randomly picking m elements from $\mathbf{X}$ where $m < n$. The sampling is done with replacement, that is any x_i which is an element of $\mathbf{X}$ has a chance of being picked more than once. Using a good random number generator can keep this type of repetition at minimum.

The re-sampled data, which is a subset of $\mathbf{X}$, is called a replica, and is denoted by $\mathbf{X}^*$. This resampling is then repeated several times to generate $\mathbf{X}^{*1}, \mathbf{X}^{*2}, \mathbf{X}^{*3}, ...\mathbf{X}^{*B}$ replicas. There is no standard value for B, and the higher its value the better results are achieved. The downside of using too much replicas is that the process becomes expensive in terms of processing power. Therefore the number is usually capped after certain value as the improvement on the statistical test is not significant.

In bootstrap process the replicas are used to achieve the desired statistical test. The *mean* and standard deviation can be used as example to illustrate this. For the data sample $\mathbf{X}$ discussed above, the mean can be worked out using the Equation 6.1 and the standard deviation is given by Equation 6.2.

The bootstarp uses the replica as follows to achieve results as above. The mean of the data sample is calculated by:

$$s(\cdot) = \sum_{b=1}^{B} \frac{s(\mathbf{x}^{*b})}{B} \tag{6.6}$$

and the standard deviation, which is called the bootstrap standard error, is calculated as follows:

$$\widehat{se}_{boot} = \left\{ \sum_{b=1}^{B} [s(\mathbf{x}^{*b}) - s(\cdot)]^2/(B-1) \right\}^{\frac{1}{2}} \tag{6.7}$$

The book by Efron and Tibshirani [68] provides a proof that such a test gives results as reliable as the theoretical equations. One of the advantages of using the bootstrap method is when faced with statistical tests for which there are no known theoretical ways of estimating its standard deviation. One such method is a variations of the Bootstrap test, the Permutation test.

6.3.1 The Permutation Test

The permutation test was introduced by R.A Fisher in the 1930's [68]. The process is identical to that of Bootstrap method except that in permutation test the selection of data is done without replacing. This test is useful when dealing with two-sample problem that have unknown distribution.

For two data sample sets $\mathbf{x} = \{x_1, x_2, x_3, ...x_n\}$ and $\mathbf{y} = \{y_1, y_2, y_3, ...y_m\}$, the process starts by adding the two data samples together to get a new sample set $\mathbf{z} = \{v_1, v_2, v_3, ...v_{n+m}\}$. Then from sample set $\mathbf{z}$ n samples are randomly picked without replacement to make the first half of the bootstrap sample $\mathbf{x}^* = \{v_1, v_2, v_3, ...v_n\}$, the remaining m samples make up the second half of the sample $\mathbf{y}^* = \{v_1, v_2, v_3, ...v_m\}$. These replicas are used to produce the bootstrap replicate statistics $s(\mathbf{x}^*)$ and $s(\mathbf{y}^*)$.

One method to check whether sample sets $\mathbf{x}$ and $\mathbf{y}$ are similar or different is to compute the difference in the statistics values, $s(\mathbf{x}^*)$ and $s(\mathbf{y}^*)$. For illustration we can assume $s(\mathbf{x}^*) = \bar{x}$ and $s(\mathbf{y}^*) = \bar{y}$. The simplest way to make a decision is to compute $\hat{\theta} = \bar{x} - \bar{y}$. So if the value of $\hat{\theta}$ is high we accept they are different, if it is low they are similar.

The other method of checking whether sample sets $\mathbf{x}$ and $\mathbf{y}$ are similar or different is to use goodness of fit tests. A goodness of fit test is generally used to check how well a theoretical pdf fits a sample set. This is discussed in detail in Section 6.4. A goodness of fit test will give a result which serves the same purpose as the $\hat{\theta}$ to check the compatibility of two pdfs.

Using $\hat{\theta}$ by itself to make a decision is not enough as its value has to be determined to be high or low. Most goodness of fit tests have look-up tables to help make this decision but using the Achieved Significance Level (ASL) of the permutation test is an effective method that removes the need for tables. ASL gives the probability that $\hat{\theta}$ comes from the H_0, where the H_0 distribution is given by the permutation test.

The computation of the ASL for two samples $\mathbf{x}$ and $\mathbf{y}$ starts by computing $\hat{\theta}$. This can be achieved using a goodness of fit test. Then the H_0 distribution is defined using permutation method to produce $\mathbf{x}^{*B}$ and $\mathbf{y}^{*B}$ where $B = 1000$. For every $\mathbf{x}^{*b}$ and $\mathbf{x}^{*b}$, where $b = \{1, 2, 3, \ldots, B\}$, the goodness of fit test is computed to give $\hat{\theta}^{*}$. The distribution of $\hat{\theta}^{*}$ is the H_0 distribution. If the the sample number used to make up H_0 distribution is too small then the the distribution will not be a good representation of H_0 but having too much will not improve the accuracy therefore 1000 samples were taken based on the recommendation of [68].

The reason for taking the $\hat{\theta}^{*}$ distribution as H_0 distribution is based on the fact that, when permutation replicas are created, the two original sample distributions are made to be similar. This is illustrated in Figure 6.5 where two distinctly different distributions are taken as an example. The top left histogram shows a bivariate Gaussian distribution with $mean = 0$, $\sigma = 1$ and $\rho = 0$. The top right histogram shows a bivariate Gaussian distribution as well but with $mean = 0$, $\sigma = 1$ and $\rho = 0.9$. But after permutation both samples become the same as can be see from the bottom part of the figure.

This means that the $\hat{\theta}^{*}$ value is what the goodness of fit test result would be if the two distributions were to be similar. This of course is what the H_0 is meant to represent. To make the decision of the hypothesis testing, the ASL value is computed as

$$ASL = \frac{\#\{\hat{\theta}^{*b} \geq \hat{\theta}\}}{B} \tag{6.8}$$

'#' should be read as 'the number of'. The ASL value lies between 0 and 1; 0 meaning that the two sample distributions are completely different and 1 meaning the ASL result has 100% probability that it comes from H_0. When using ASL the H_1 distribution does not need to be determined. The decision is made on a combination of what statistical test is used and the hypothesis test condition. For example if the statistical test of choice gives high value for different distributions and small for similar, then the condition of the

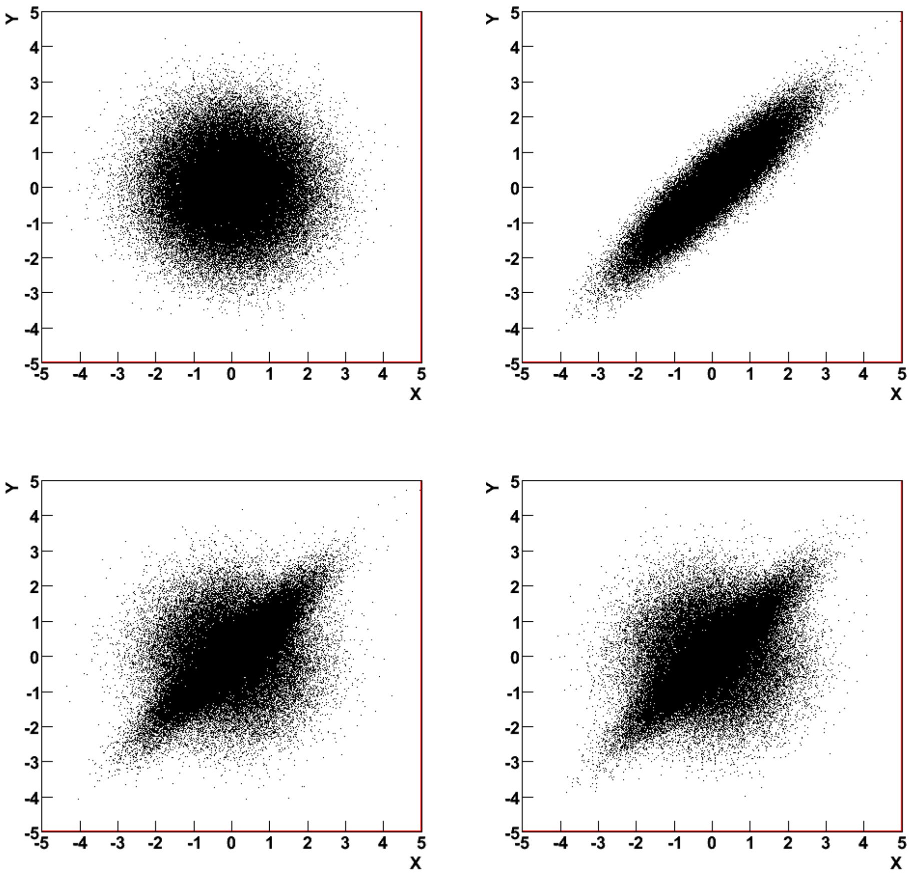

Figure 6.5: Left to Right and Top to Bottom A bivariate Normal distribution with 0 correlation, Normal Distribution with 0.9 correlation, n samples of the permutated data and m samples of the permutated data. On the top two the difference is clear between the two distributions, where as on the bottom two the permutation process has made both samples similar.

hypothesis test is the statistical test value should be greater than the H_0 distribution values with a certain confidence level. The confidence level or significance level is generally set at $\alpha = 0.10$ [68]. The ASL is then computed as in Equation 6.8. Once the ASL is computed it is compared against the decided significance level (α). If the ASL value is greater than α then H_0 is accepted if not it is rejected.

6.4 Goodness of Fit Test Algorithms

One of the main reasons for using the Goodness of Fit (GoF) statistical test is to find out if the distributions being analysed come from the same or different (depending on the analysis) underlying distribution. It is a common practice to compare a data sample ($\mathbf{X}$) against a theoretical distribution ($f(x)$) which is believed to define the sample distribution. In other cases where the theoretical distribution does not have a known model, an MC simulation is used to define the theoretical distribution. In this chapter all the discussed statistical tests are to be used to check the similarity between a given (MC) data set (reference) and measured data set (sample).

The GoF tests can be grouped into two; tests that use binned data, where predefined categories exist for the sample entries, and tests that use unbinned data where each data set entry is independent of any category. Several implementations of both methods exist, but in this chapter we will only study the χ^2 test for the binned data, the Kolmogorov-Smirnov (KS) test for the unbinned data and the Energy test, developed by G. Zech and B. Aslan [1], which has been implemented both for binned and unbinned data samples.

6.4.1 The χ^2 Test

One GoF test which is commonly used in high energy physics experiments is the the χ^2 test. But this test comes with its limits. As the the number of bins used in the test increases (using the same number of data entry), the χ^2 test result becomes less significant [67]. The other limitation of this test comes from the fact that the χ^2 test is a goodness of fit test for data samples that fall into a finite number of categories, hence it can only test binned data. And if two data sets are to be compared using the χ^2 test, they must have the same number of bins.

For a data set $\mathbf{Y} = y_1, y_2, y_3...y_n$ and a function $f(x_i)$ that gives the ideal value of y_i for a given x_i and binned in to m bins, the test is given by

$$\chi^2 = \sum_{i}^{m} \frac{y_i - f(x_i)}{\sigma_i} \tag{6.9}$$

Where σ_i is the measurement error of the observed y_i. The χ^2 can also be used to compare two binned data sets. The result achieved by Equation 6.9 gives a number which is used to whether accept or reject the similarity test. The decision is taken by comparing the χ^2 result against a table, which gives the the probability of the expected value of χ^2 against the number of degrees of freedom. The number of degrees of freedom is simply given by $m - 1$ where m is the number of bins [70].

The χ^2 test is implemented in ROOT. The implementation is based on [71] and the full details of this are given in [65]. The test uses the equation given above to calculate the statistic value between two given distributions. The value is then compared against the results given in a table and the probability of that result happening is computed. The test then gives the computed probability where 1 is the two samples are the same and 0 being they are completely different.

The DQM system has imported the ROOT implementation of the χ^2 test. As the Tracker_Monitor_Track package produces one dimensional data sets, this opportunity was taken to evaluate how the test performed. Two types of data sets were generated using the Silicon Strip Tracker (SST) system. As the LHC has not started real data was not available therefore MC simulated data was used. This data sample is the same as used in Chapter 4.

To evaluate the ROOT package, data samples with the ideal against misaligned Tracker geometry were used. For this test two separate data samples were generated and compared. To start with all data sets were used regardless of their number of entries. From this it was

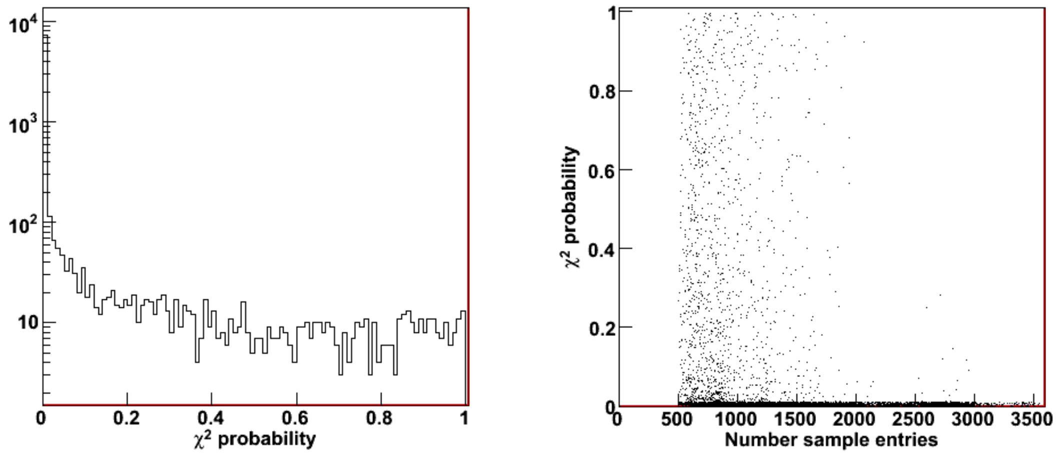

Figure 6.6: The probability of accepting the H_0 when the two samples are different. This test used only samples with data entry greater than 500 and was done using the ROOT χ^2 test.

clear the test performed poorly when the data samples are small, giving high probability of accepting the H_0, which agrees with[67]. As the number of entries per histogram increased the results were as expected, giving lower probability of accepting the H_0. The number of the tests that accepted the H_0 is very low. This is because only very few detector modules had small number of entries.

To avoid the issues discussed above histograms with more than 500 entries were used. This enabled to have a test where each bin would have 5 entries on average. Setting the level of significance at 10% it was possible to reject almost all the data that came from the misaligned geometry. Figure 6.6 shows these results. It can be noticed in Figure 6.6 that very few samples were giving high probability of accepting the H_0. This is because the misalignment tools use random number generator to create a misaligned position of a detector module, and some times the random number generated (on very few detector modules) is very small. This small number means that some of the detector modules end up not being misaligned as the number wont have any visible effect. This will not be the case when the data generated is coming from the detector as this is purely a simulation error.

Based on the results shown in Figure 6.6 we believe the ROOT implementation of the χ^2 can be used to detect distribution that arise from misaligned detector modules. But due to the limitations of the χ^2 test its usage should be done with care. The number of bins per histogram of each detector module is set to 100 by default, which means this test can not be used for histograms with small number of entries. It is not possible to define the optimum number of data entries against bin number for the χ^2 test, but a guidance is given by [67] to always use as fewer bins as possible.

6.4.2 The Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (KS) test is a non parametric GoF test. Unlike the χ^2 test the KS test is binning free, which means the data samples being compared do not have to belong to the same number of categories (bins). In fact binning the data goes against the basis of the test as the K-S test uses every entry in a data set instead of the category where each data points falls in.

For two data samples $F(\mathbf{x})$ and $G(\mathbf{x})$ which contain continuous data the K-S test creates an empirical cumulative distribution (CDF) of the data sets and calculates the maximum absolute difference between the two CDFs. This difference is called the D$_{statistic}$ and is given by

$$D_{statistic} = max|F(x) - G(x)| \tag{6.10}$$

Figure 6.7 shows the computed D$_{statistic}$ of Gaussian (with $mean = 0$ and $\sigma = 1$) and Cauchy (with $mean = 0$ and $\sigma = 1$) CDF. Once the D$_{statistic}$ has been computed for two distributions, the value is compared against a table that gives the probability of the D$_{statistic}$ happening for a test. The table lists the probabilities based on the amount of entry per sample data.

ROOT implements both one and two dimensional KS tests. The test was first imple-

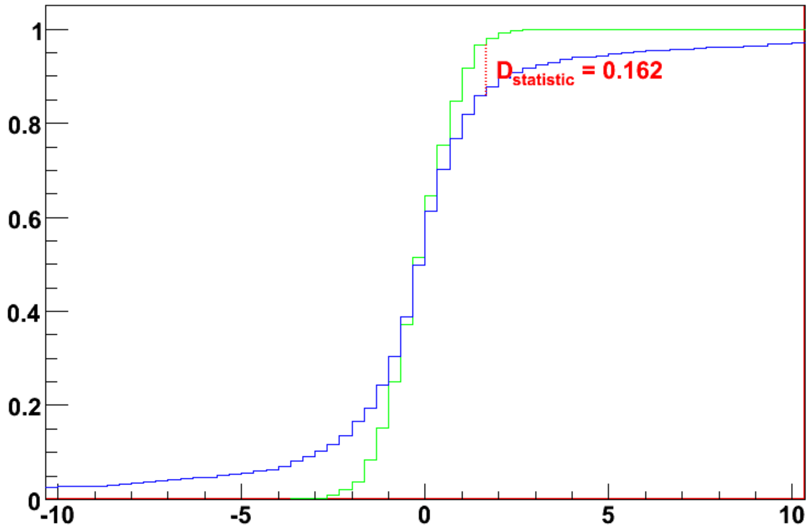

Figure 6.7: The $D_{statistic}$ calculated for the KS test on Gaussian (green) against Cauchy (blue)

mented in the HBOOK[2] package [72] and was later ported to ROOT [65]. However the implementation uses binned data, which is not a true implementation of the KS test [73]. The HBOOK manual warns the user the effect binning will have on the test. It states that the probability distribution given by the test is not "exactly the correct distribution for binned data" [65]. For this reason χ^2 was preferred over the KS test for the data generated by the Monitor_Track_Residual package as it produces binned histograms.

6.4.3 Kolmogorov-Smirnov test for two-dimensional data sets

Using Goodness of Fit (GoF) test for multi-dimensional data sets is fairly new and is generally seen as a challenge. Tests that use binned data suffer from a limitation known in the literature as the "the curse of dimensionality" [74]. That is a data sets in high dimensional space are mostly empty, and tests using binned data sets can only be effective when the data sets used have very large number of entries.

[2] a statistics package that was used before ROOT

The challenge faced when using the KS test for two dimensional data sets is that it orders data so that to work out $F(x) = Pr(\mathbf{X} < x)$. When used for one dimensional data sets the KS test is independent of direction of ordering of the data sets. Where as this is impossible if the test is to be adapted for use in multi-dimensional data sets as there are $2^d - 1$ independent ways of ordering the data to produce CDF [73]. Two implementations of the KS adaptation are tested in this thesis: the ROOT 2D KS adaptation and an implementation done by Raul Lopes [73].

Kolmogorov-Smirnov test implementation in ROOT

The Kolmogorov-Smirnov test implementation in ROOT uses binning to overcome the problem of ordering faced when adapting the KS for two dimensional data sets. This "enables us to define an obvious ordering. In fact there are two obvious orderings (horizontal and vertical) which give rise to two (in general different) Kolmogorov distance measures" [72]. In a histogram this would mean the $\mathrm{D}_{statistic}$ is calculated for CDF based on the ordering of x and y axes separately; then the average of the two $\mathrm{D}_{statistic}$ is taken to calculate the probability of rejecting the null hypothesis.

This implementation suffers from two problems. The first one being the problem inherited from the 1D KS test which is the binning and the second limitation is that it uses the average of two one dimensional tests. This undermines some of the advantages of having multivariate data, as information such as the dependence of the vector elements (e.g. correlation) on each other and their correlation [75] is neglected when the two dimensional data is taken as two one dimensional data sets. These two limitations make the test not effective when provided with certain data types. This is illustrated using two different samples that can be seen on Figure 6.8. These two samples are completely different, but the ROOT implementation of the KS test gives a high probability (99.77%) of accepting the null hypothesis.

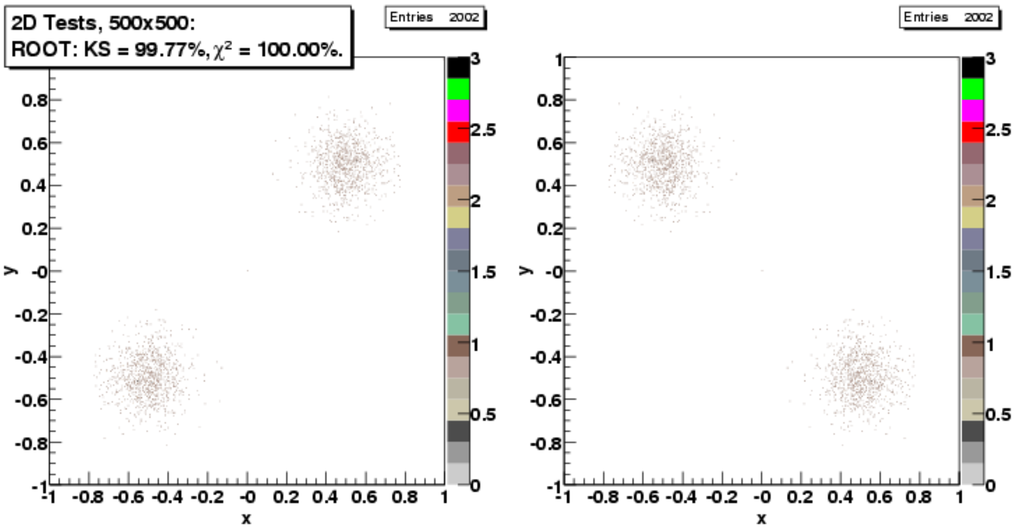

Figure 6.8: When comparing two-dimensional data sets the binning used in Kolmogorov-Smirnov test can give unreliable results. In the above two data sets where they look completely different to the human eye, they have been accepted as the same by the ROOT K-S and χ^2 tests.

Kolmogorov-Smirnov test implementation for two-dimensional data sets

The implementation of the KS test by Lopes is based on Peacock's [76] adaptation of the one dimensional KS test to be used for two dimensional test. The algorithm provided by Peacock is based on making the statistic independent of any particular ordering by finding the largest difference between the cumulative distribution functions under any possible ordering. Given $'n'$ points in a two-dimensional data set, the algorithm then partitions the $'n'$ points in $4n^2$ quadrants where each quadrant is defined by a pair X_i and Y_j. X_i and Y_j being coordinates of any pairs of points in the given samples. Then the maximum absolute difference between cumulative distribution functions in all quadrants is computed. This amounts to calculating the cumulative distribution functions in the $4n^2$ quadrants of the plane [73].

A variation of the Peacock's algorithm is give by Fasano and Franceschini [77] which according to [73] "greatly reduces the lower-bound for its computation". The main difference in the two adaptations of the KS algorithm is that the second one (by Fasano and

Francesichini) is faster. Therefore this test was used to compare with the results achieved by the Minimum Energy test discussed below.

6.4.4 The Minimum Energy test

The idea of the Energy test as given by [1] is based on the theory of electrostatic energy. According to the electrostatic energy theory of charged bodies, "The electrostatic energy of a superposition of a positive and a negative charge distribution is at minimum if the two distributions coincide." This forms the basis of the energy test.

For two charges e_1 stationary at position $\mathbf{r}_1$ and e_2 is displaced from ∞ to $\mathbf{r}_2$ towards e_1, then the work done on e_2 by e_1 is given by

$$W_{12} = -e_2 \int_{\infty}^{\mathbf{r}_2} E_1 d\mathbf{r} = e_2(\varphi_1(\mathbf{r}_2) - \varphi_1(\infty)) \tag{6.11}$$

Where E_1 is the field of e_1 and $\varphi_1(\mathbf{r}_2)$ which is the potential energy between e_1 and e_2 when e_2 is at position $\mathbf{r}_2$ is given by

$$\varphi_1(\mathbf{r}_2) = \frac{1}{4\pi\varepsilon_0} \frac{e_1}{|\mathbf{r}_2 - \mathbf{r}_1|} \tag{6.12}$$

and we arrive at Equation 6.13 as $\varphi_1(\infty) = 0$

$$W_{12} = \frac{1}{4\pi\varepsilon_0} \frac{e_1 e_2}{|\mathbf{r}_2 - \mathbf{r}_1|} \tag{6.13}$$

For a system with 3 charge points e_1, e_2 and e_3 the total potential energy of the assemblage of the three charges is given by [78]

$$W_{123} = \frac{1}{4\pi\varepsilon_0}\left(\frac{e_1 e_2}{|\mathbf{r}_2 - \mathbf{r}_1|} + \frac{e_1 e_3}{|\mathbf{r}_3 - \mathbf{r}_1|} + \frac{e_2 e_3}{|\mathbf{r}_3 - \mathbf{r}_2|}\right) \tag{6.14}$$

Based on Equation 6.14, the total potential energy of a system with n charge points is given by

$$W = \frac{1}{4\pi\varepsilon_0}\sum_{i=1}^{n}\sum_{j<i}^{i}\frac{e_i e_j}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{6.15}$$

$$= \frac{1}{8\pi\varepsilon_0}\sum_{i\neq j}^{n}\frac{e_i e_j}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{6.16}$$

Equation 6.16 has been multiplied by the coefficient $\frac{1}{2}$ as each term is computed twice. For a system of with continuous charge density distribution ρ, its total potential energy is given by integrating over $\rho(r)$ and $\rho(r')$ and then integrating over the integrals of $\rho(r)$ and $\rho(r')$

$$W = \frac{1}{8\pi\varepsilon_0}\int\int\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}d\mathbf{r}d\mathbf{r}' \tag{6.17}$$

If an external charge distributions ρ_{ex} is added to the system the total potential energy of the system becomes the addition of the potential of both distributions and is given by

$$W = \frac{1}{8\pi\varepsilon_0}\int\int\frac{[\rho(\mathbf{r}) + \rho_{ex}(\mathbf{r})][\rho(\mathbf{r}') + \rho_{px}(\mathbf{r}')]}{|\mathbf{r} - \mathbf{r}'|}d\mathbf{r}d\mathbf{r}' \tag{6.18}$$

If ρ_{ex} is taken to be negatively charged and ρ positively charged, and the total charge of the system is fixed at zero so that

$$\int [\rho(\mathbf{r}) - \rho_{ex}(\mathbf{r})]d\mathbf{r} = 0 \tag{6.19}$$

then the potential energy of the system is said to be at minimum. Adapting this theory to statistics with sample distributions of $f(x) = x_1, x_2, x_3 ... x_n$ and $f_0(x) = y_1, y_2, y_3 ..., y_m$ and combining 6.18 and 6.19 the total energy of the two distributions can be calculated using

$$\phi = \frac{1}{2}\int\int [f(\mathbf{x}) - f_0(\mathbf{x})][f(\mathbf{x}') - f_0(\mathbf{x}')]R(\mathbf{x}, \mathbf{x}')d\mathbf{x}d\mathbf{x}' \tag{6.20}$$

where ϕ is the measure of the energy between the two pdfs which is the statistical difference measure that will be discussed in the rest of the chapter. The coefficient $\frac{1}{2}$ is used as each term is computed twice. This is a cascaded effect of the transformation from Equation 6.15 to 6.16. The $R(\mathbf{x}, \mathbf{x}')$ is the distance function. This function is the Euclidean distance of the the two vectors $|\mathbf{x} - \mathbf{x}'|$. Using $R(\mathbf{x}, \mathbf{x}') = \frac{1}{|\mathbf{x}-\mathbf{x}'|}$ makes Equation 6.20 proportional to the electrostatic energy formalism described in Equation 6.18. Expanding Equation 6.20 gives

$$\phi = \frac{1}{2}\int\int [f(\mathbf{x})f(\mathbf{x}') + f_0(\mathbf{x})f_0(\mathbf{x}') - (f(\mathbf{x})f_0(\mathbf{x}') + f(\mathbf{x}')f_0(\mathbf{x}))]R(\mathbf{x}, \mathbf{x}')d\mathbf{x}d\mathbf{x}' \tag{6.21}$$

Where ϕ is the total energy of the system. Each term is an expectation value of R. If each term is denoted by $\mathbf{E}$ such that $E = E(R)$ then Equation 6.21 can be expressed as

$$\phi = \frac{1}{2}E_1 + \frac{1}{2}E_2 - E_3 \tag{6.22}$$

E_3 includes two terms $(f(\mathbf{x})f_0(\mathbf{x}')+f(\mathbf{x}')f_0(\mathbf{x}))]R(\mathbf{x},\mathbf{x}')$, and eliminated the coefficient $\frac{1}{2}$. Taking the first term E_1 which is the expectation value of the distance function of the first distribution

$$E_1 = E(R) = \int g(\mathbf{y})R(\mathbf{y})d\mathbf{y} \tag{6.23}$$

where $R(\mathbf{y})$ is the distance function relative to $g(\mathbf{y})$, $\mathbf{y} = (\mathbf{x}, \mathbf{x}')$ and $g(\mathbf{y}) = f(\mathbf{x})f(\mathbf{x}')$. Based on the law of large numbers; sample mean $\bar{x}$ of n observations from a distribution with mean μ and finite variance, $\bar{x}$ will converge towards μ as n becomes large. Therefore E_1 can be estimated from the sample mean given there is enough data samples.

Statistics deals with finite number of entries in a sample. To obtain the observation $\mathbf{y}$ in Equation 6.23 two independent entries $\mathbf{x}_i$ and $\mathbf{x}_j$ are taken from $f(\mathbf{x})$. Therefore for a sample $\mathbf{X}$ with n entries, the sample estimator E_{1n} of the real E_1 can be constructed by averaging over all possible splitting of the sample into two parts to get

$$E_{1n} = \frac{1}{n(n-1)} \sum_{i<j}^{n} \mathbf{R}(|x_i - x_j|) \tag{6.24}$$

where E_{1n} converges to E_1 in the limit of large n, since it is a consistent estimator of E_1. Generally a GoF test would compare data sample against a known pdf. But in the case of DQM, MC data is used as reference to compare against the sample data. Taking this in to account the sampling version of (Equation 6.22) can be obtained from two samples $\mathbf{X}$ with m number of entries and $\mathbf{Y}$ with n number of entries drawn from f and f_0, respectively:

$$\Phi_A = \frac{1}{n^2}\sum_{i<j}^{n} R(|\mathbf{x}_i - \mathbf{x}_j|)$$
$$\Phi_B = \frac{1}{m^2}\sum_{i<j}^{n} R(|\mathbf{y}_i - \mathbf{y}_j|)$$
$$\Phi_{AB} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} R(|\mathbf{x}_i - \mathbf{y}_j|)$$

$$\Phi_{nm} = \Phi_A + \Phi_B + \Phi_{AB} \tag{6.25}$$

This equation gives the statistic difference between two given distributions. The statistic value is given from the the static energy theory of the distance between two points (charges), as discussed above. The distance between two points can be calculated in various ways as discussed below.

6.4.5 The Distance Function

The Euclidean distance of two multivariate points $\mathbf{x}$ and $\mathbf{x}'$ is given by

$$|\mathbf{x} - \mathbf{x}'| = \sqrt{\sum_{k=1}^{d}(x_k - x'_k)^2} \tag{6.26}$$

Where $k = 1, 2, 3...d$ is the dimension of vectors $\mathbf{x}$ and $\mathbf{x}'$. Three different distance functions given in [69] are

$$\mathbf{R}(|\mathbf{x}_i - \mathbf{x}'|) = \frac{1}{|\mathbf{x}_i - \mathbf{x}'|^k}, 0 < k < \frac{d}{2} \tag{6.27}$$

$$\mathbf{R}(|\mathbf{x}_i - \mathbf{x}'|) = -ln(|\mathbf{x}_i - \mathbf{x}'|) \tag{6.28}$$

$$\mathbf{R}(|\mathbf{x}_i - \mathbf{x}'|) = e^{-|\mathbf{x}_i - \mathbf{x}'|^2/(2s^2)} \tag{6.29}$$

For the DQM test tool the logarithmic function was chosen. The reason for this being the usage of this function by Zech and Aslan for generating their results. It was considered good practice to follow the steps taken by them when trying to reproduce their results. When the minimum energy implementation was run using the logarithmic distance function for data sets with small number of entries, it generated negative energy results. This can be seen in figure 6.9. This behaviour agrees with that in [69].

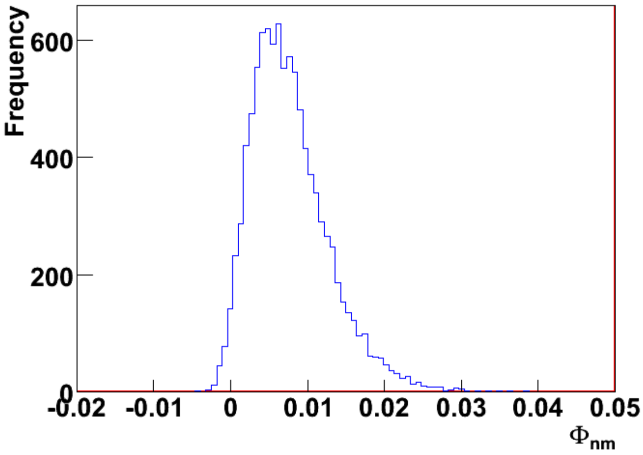

Figure 6.9: Energy distribution for the comparison of Normal distribution with *mean* = 0 and $\rho = 0$ against Cauchy with *mean* = 0 and $\rho = 0$ using the logarithmic distance function. Here 100 samples from the normal distribution and 1000 samples from the Cauchy distribution were taken. The test gives negative energy for small sample sizes.

The fact that the Energy test gives negative values for small entries in data sets can be explained by the looking at the three different parts of the Energy test equation. As the value of the distance start starts to get smaller the resulting values from the logarithmic function will be negative. As expected this behaviour starts to diminish as the n and m approach ∞. This is shown in Figure 6.10 where the energy for data sets with number of entries exceeding 10000 were used.

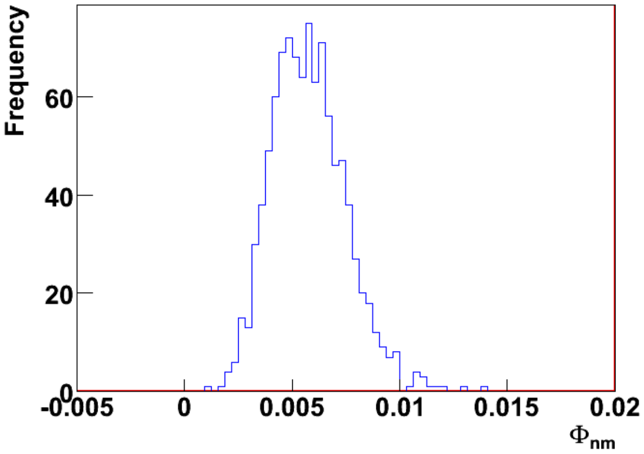

Figure 6.10: Energy distribution for the comparison of Normal distribution with $mean = 0$ and $\rho = 0$ against Cauchy with $mean = 0$ and $\rho = 0$ using the logarithmic distance function. Here 10000 samples were taken from both the normal and the Cauchy distribution. The test does not give negative energy results as the sample size approaches ∞.

6.4.6 The Binned Minimum Energy test implementation

The binned implementation of the Energy test compares "square" ($N \times N$) histograms, but it can easily be generalised to $N \times M$ histograms. When calculating the energy of the first distribution energy between points in the same bin must be taken into account when using binning. To calculate this the average distance between pairs of random points in a unit square was used. The distance of points in different bins are calculated simply as the Euclidian distance between bin centres. The full details of the implementation can be found in [79].

6.5 Tests And Results

The different implementations of the GoF tests discussed in Section 6.4 were evaluated using data samples generated using well defined statistical distributions and MC generated data. The testing was divided into two sections.

The first part is to evaluate the implementation of the Energy test. This was necessary to check if it was possible to achieve the results discussed by Aslan And Zech in[1]. It was attempted to follow the steps followed by them, but it was not possible to ensure similar reproduction of the data as used by[69] and[1] as the generation methods are not described. To the best of our ability we made sure to use widely used random number generators. Two different methods of data generation were used. The first one is, the R statistics package. This is a free software by "The R Foundation for Statistical Computing" and is available to download from [80]. The data generated for use in this test used the R version 2.5.0 (2007-04-23). The R package does not have integrated two-dimensional data generators, therefore a third party library had to be imported. This is the "fMultivar" library[81]. The fMultivar library gives several options for the data to be generated. The options that were used in the generation of the data for this test are the "n" for the number of entries in the sample, "nu" for the type of $Student - t$ distribution and "rho" for the correlation of the data. The data generated is a standard which mean the standard deviation is always 1. The second method for generation was by the GSL library of C++. Classes for generation can be called in a C++ code and data can be generated as standard output. A detailed list of the library and its usage can be found in [82]. Both generators produced data that gave similar results in the GoF and it was finally decided to use data generated using the R statistics package, for its simplicity, as this helped in making the data sample consistent. A list of the generated data using the R package is listed in table 6.1.

The second test was to compare the implemented Energy test against other statistical tests using MC simulation data. The simulation is based on the first 1000000 events after the start of the CMS experiment of the pp collision and is fully described in Section 4.5. The MC generated data was used to reconstruct tracks of the SST using the combinatorial track finder (CTF) algorithm of CMS. This algorithm is the standard reconstruction algorithm

Distribution Type	fMultivar Generator
N(0,I)	rnorm2d(n = 10000, rho = 0)
$N\left(0, \begin{pmatrix} 1 & 0.6 \\ 0.6 & 1 \end{pmatrix}\right)$	rnorm2d(n = 10000, rho = 0.6)
$N\left(0, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}\right)$	rnorm2d(n = 10000, rho = 0.9)
C(0,I)	rcauchy2d(10000, rho = 0)
$student - t_2$	rt2d(n = 10000, rho = 0, nu = 2)
$student - t_4$	rt2d(n = 10000, rho = 0, nu = 4)

Table 6.1: This table gives the command lines used in the generation of data samples for the test of the statistical test algorithm implementation. Here N(0,I) and C(0,I) are normal and Cauchy distributions consecutively with $mean = 0$ and $\rho = I$

and is discussed in Section 3.3. Two types of the tracker geometry were used to generate two different sample of data sets from the same MC simulation. The first set with the ideal Tracker geometry and the second one with misaligned geometry of the Tracker.

6.5.1 Binning-Free Energy test Results

The energy test is a GoF test and returns a statistic that represents the difference between two sample distributions. This value is not a probability of how compatible or different the two samples are. Therefore to make a decision whether the GoF test value accepts or rejects the H_0 the Bootstrap discussed on Section 6.3 is used to get the ASL value. The ASL value is given by the permutation test. This gives the probability of the the statistical test value falling in the H_0 distribution. If the ASL is higher than the significance level (generally $\alpha = 0.10$) then the H_0 is accepted. This method of making decision is discussed in [47]. In other words this test is set up so that the ASL will give the probability of accepting the H_0.

As the distributions used in this test are known to be different, this test is used to evaluate the power of the test described in Section 6.2.2. The power of the test can be

Files compared	$ASL_{(30,30)}$	$ASL_{(50,50)}$	$ASL_{(100,100)}$
C(0,I)#1 vs C(0,I)#2	0.871	0.754	0.527
C(0,I)#3 vs C(0,I)#4	0.53	0.47	0.303
C(0,I)#5 vs C(0,I)#6	0.831	0.872	0.951
C(0,I)#2 vs C(0,I)#5	0.616	0.734	0.524
C(0,I)#6 vs C(0,I)#1	0.967	0.852	0.938

Table 6.2: Results from the binning-free energy test for similar distributions (Cauchy against Cauchy). Data sets generated using fMultivar of the R-package

Files compared	$ASL_{(30,30)}$	$ASL_{(50,50)}$	$ASL_{(100,100)}$
N(0,I)#1 vs N(0,I)#2	0.587	0.85	0.584
N(0,I)#3 vs N(0,I)#4	0.99	0.924	0.201
N(0,I)#5 vs N(0,I)#6	0.728	0.707	0.531
N(0,I)#1 vs N(0,I)#3	0.589	0.686	0.705
N(0,I)#1 vs N(0,I)#4	0.637	0.797	0.695

Table 6.3: Results from the binning-free energy test for similar distributions (Normal against Normal) Data sets generated using fMultivar of the R-package

calculated as $1 - ASL$ for two distributions known to be different. This is discussed in Chapter 14 of the book "Bootstrap Methods and Permutation Test" [83].

To start with, a control test was done. This is to see how the test behaves when two similar data samples are used. For this test, the N(0,I) and C(0,I) were were compared against each other. It is generally acceptable to use values of $\alpha = 0.05$ or $\alpha = 0.1$ as the highest values to reject the H_0. We have chosen to use $\alpha = 0.1$ to reject the null hypothesis. Tables 6.2 and 6.3 give the results of the test. As it can be seen the smallest value of ASL is 0.201 for the specific test $N(0, I)\#3$ against $N(0, I)\#4$ which is too large to reject the H_0. This test has showed the energy test behaves as expected when testing two similar distributions.

Based on the definition of the ASL and the hypothesis testing given in in Section 6.2, it was attempted to confirm the results given in [69] and [1] where it was shown that test was effective for data samples with 30, 50 and 100 entries. In both publications the test was implemented using the permutation test, but the full description of the method was not given. It was not possible to get these information despite attempt of private correspondence with the authors.

It was possible to show that the results given in [69] and [1] can be achieved in certain data samples using only observations between 30 and 100. This can be seen in row 9 of Table 6.4 where the powers $(1-ASL)$ are 0.882, 0.954 and 0.987 for entries 30, 50 and 100 respectively, which is very close to the values achieved by Aslan and Zech. But as the test was repeated for different sets of the similar distributions it was noticed that these results are not consistent. In table 6.4 it can be seen that most of the results can not be used to reject the H_0.

To study if this was a statistical fluctuation a further test was done. The reason for picking this test was based on the calculation of the ASL value. ASL arises from the permutation where the number of the full permutations is:

$$\begin{pmatrix} n+m \\ m \end{pmatrix} = \frac{n+m!}{n!m!} \tag{6.30}$$

where n and m are the number of entries for the two samples being compared and $N = n + m$. It is not possible to do all these permutations as this is computationally expensive. Therefore only 1000 randomly selected values are used as recommended by [68], which can raise a question if 1000 such values can represent the true distribution of the H_0. To evaluate this a particular sample from the Table 6.4 was taken (N(0,I)#6 vs C(0,I)#4) with high ASL value and the permutation test was repeated 1000 times. The result as can be seen in Figure 6.11 shows the ASL value given in the table falls in the distribution showing that the test is accepting the H_0. This test proved that 1000 samples of permutation is indeed enough and the result given by the permutation test is reliable.

Files compared	$ASL_{(30,30)}$	$ASL_{(50,50)}$	$ASL_{(100,100)}$
N(0,I)#2 vs C(0,I)#2	0.193	0.251	0.127
N(0,I)#3 vs C(0,I)#3	0.143	0.284	0.045
N(0,I)#4 vs C(0,I)#4	0.302	0.161	0.669
N(0,I)#5 vs C(0,I)#5	0.456	0.061	0.081
N(0,I)#6 vs C(0,I)#6	0.849	0.607	0.487
N(0,I)#1 vs C(0,I)#6	0.52	0.248	0.275
N(0,I)#6 vs C(0,I)#4	0.966	0.68	0.87
N(0,I)#2 vs C(0,I)#3	0.066	0.041	0.327
N(0,I)#3 vs C(0,I)#1	0.118	0.046	0.013
N(0,I)#5 vs C(0,I)#2	0.468	0.342	0.242
N(0,I)#4 vs C(0,I)#6	0.845	0.701	0.471
N(0,I)#6 vs C(0,I)#4	0.966	0.68	0.87
N(0,I)#2 vs C(0,I)#5	0.579	0.083	0.183
N(0,I)#2 vs C(0,I)#6	0.435	0.775	0.49
N(0,I)#4 vs C(0,I)#2	0.468	0.538	0.079
N(0,I)#1 vs Student-t_4	0.946	0.569	0.125
N(0,I)#1 vs Student-t_2	0.104	0.514	0.002
N(0,I)#6 vs Student-t_4	0.155	0.601	0.184
N(0,I)#6 vs Student-t_2	0.356	0.175	0.028
N(0,I)#2 vs Student-t_4	0.036	0.046	0.063
N(0,I)#5 vs Student-t_2	0.409	0.032	0.004
$N(0,I)\#1$ vs $N\left(0,\begin{pmatrix}1 & 0.9\\0.9 & 1\end{pmatrix}\right)$	0	0.001	0
$N(0,I)\#6$ vs $N\left(0,\begin{pmatrix}1 & 0.9\\0.9 & 1\end{pmatrix}\right)$	0.003	0.002	0
$N(0,I)\#6$ vs $N\left(0,\begin{pmatrix}1 & 0.6\\0.6 & 1\end{pmatrix}\right)$	0.147	0.034	0.003
$N(0,I)\#3$ vs $N\left(0,\begin{pmatrix}1 & 0.6\\0.6 & 1\end{pmatrix}\right)$	0.472	0.095	0.127
$N(0,I)\#4$ vs $N\left(0,\begin{pmatrix}1 & 0.9\\0.9 & 1\end{pmatrix}\right)$	0.101	0.028	0

Table 6.4: Results from the binning-free energy test for different distributions the (# after the file name represents the different file used)

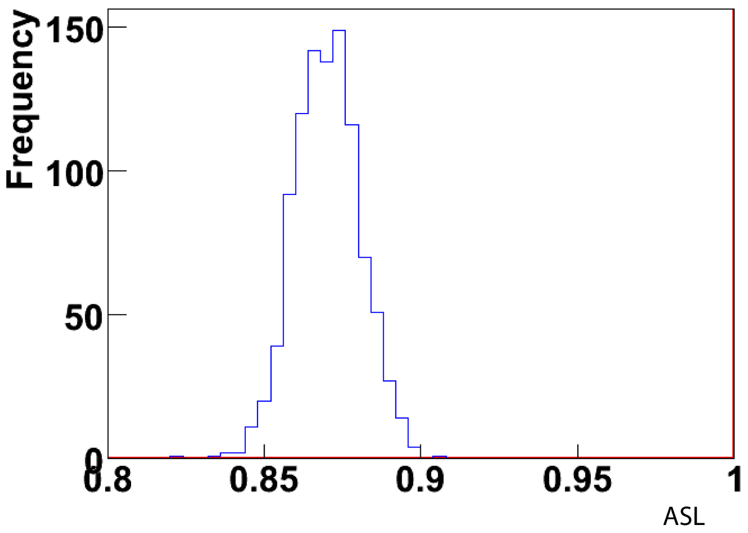

Figure 6.11: ASL distribution for the comparison of Normal distribution file #6 and Cauchy distribution file #4. The Energy test was run 1000 times and the ASL value for each test recorded. For this test data samples with entry of 100 were used.

Size of Files	N6 vs C4	N2 vs C3	N5 vs C2	N4 vs C6	N2 vs C6	N1 vs St4
$ASL_{(30,30)}$	0.954	0.065	0.472	0.81	0.419	0.932
$ASL_{(50,50)}$	0.699	0.046	0.354	0.704	0.769	0.548
$ASL_{(100,100)}$	0.874	0.333	0.215	0.464	0.502	0.097
$ASL_{(150,150)}$	0.841	0.113	0.03	0.457	0.307	0.021
$ASL_{(200,200)}$	0.188	0.151	0.004	0.485	0.038	0.055
$ASL_{(300,300)}$	0.178	0.005	0.003	0.039	0	0.019
$ASL_{(500,500)}$	0.001	0.001	0.002	0.004	0	0.002
$ASL_{(800,800)}$	0	0	0	0.002	0	0.004
$ASL_{(1000,1000)}$	0	0	0	0.005	0	0
$ASL_{(2000,2000)}$	0	0	0	0	0	0

Table 6.5: Results from the binning free energy test for different distributions (N=N(0,I), C=C(0,I), St4=Student-t_4) and the number (eg. N6) correspond to the numbers given in Table 6.4. Data sets generated using fMultivar of the R-package

The behaviour seen in Table 6.4 where two different distributions with sample entries between 30 and 100 giving inconsistent values can only be explained by statistical fluctuation due to high uncertainties of the sampled distribution. To show this, more tests were performed, this time with higher entry. The data sets which were giving high ASL values were selected. The results are given in table 6.5

As the number of entries for the samples increased, the ASL values started to decrease consistently. This shows that the test is reliable as long as long as the sample being tested have enough entries to define their real distribution. It can be concluded from here that for the test to give reliable results, it has to be ensured the underlying distribution can be defined using the given data samples with very small uncertainties. For distributions which are completely different from each other, such as the N(0,I) with $\rho = 0$ against N(0,I) with $\rho = 0.9$, 30 points are enough to distinguish the difference between the two samples. This can be seen in Table 6.4 rows 22, 23 and 26 where very small ASL was given for 30 entries and consistently decreased as the number of entries increased.

Before embarking on the discussion of how many observed points can give reliable result of the energy test another issue needs to be discussed. When the number of observations was increased it was noticed that the GoF value ($\hat{\theta}$) was approaching zero (Figure 6.12).

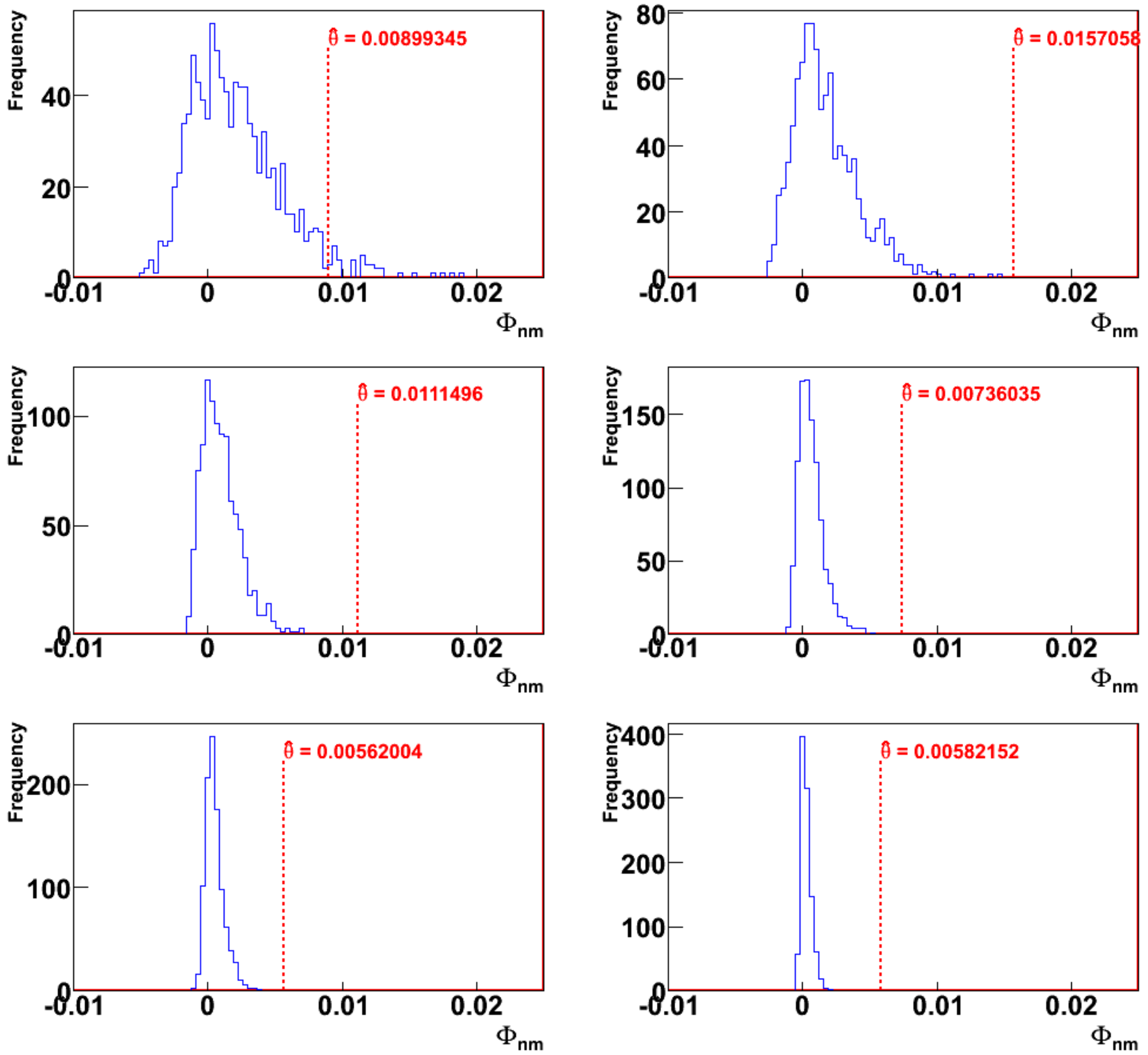

Figure 6.12: The number of points increases from left to right, top to bottom 500, 1000, 3000, 5000, 8000, 10000. The $\hat{\theta}$ can be seen approaching zero as the number of entries increases.

Although the separation between the H_0 distribution and the $\hat{\theta}$ became more defined, it seemed that both were converging to zero. To make sure that this was not the case further tests were performed to study the behaviour of the $\hat{\theta}$ as the number of observation approaches ∞. The results as can be seen from 6.13 showed this is not the case. As the number of observation approaches ∞ the energy increases exponentially. This is expected as the number of entries per sample approaches ∞ the gap between the ASL value and the H_0 distribution should get bigger (see Appendix of [1]).

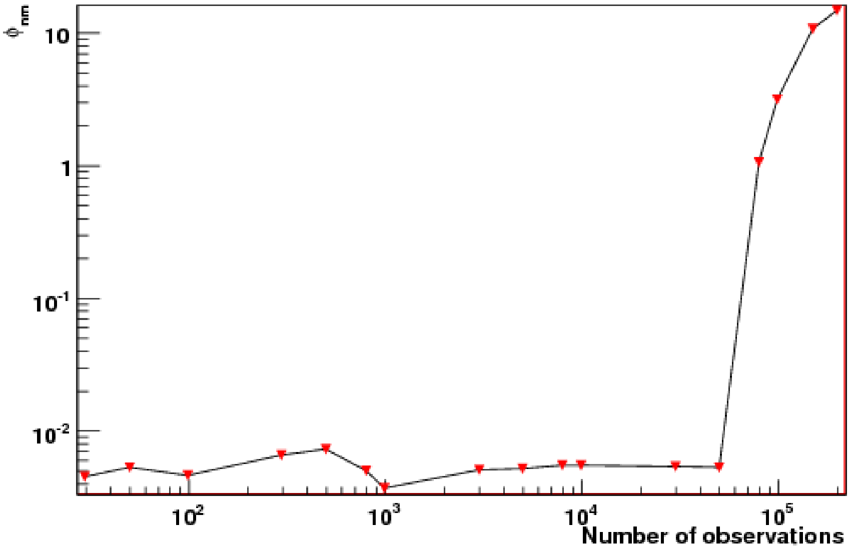

Figure 6.13: The values of $\hat{\theta}$ are plotted against the increasing number entries for the test to see if it converges to zero. but it can be clearly seen as the number of observation increases the $\hat{\theta}$ value increases dramatically

It has been shown above that the energy test gives reliable results as long as the the sample data sets have enough entries to reduce the uncertainties. But what number of entries in a sample data set is enough? To set up this test, a method for defining the alternative (H_1) distribution was deemed necessary. This is not a problem for the null hypothesis as the permutation bootstrap generates this.

A similar idea used to generate the H_0 distribution by [68] can be used to generate the H_1 distribution. A closer study of the method of permutation (see 6.3.1) tells us that the $\hat{\theta}$ is a statistical test that was picked at random from the $\hat{\theta}$ distribution given by the test. Using the fact that it is known that the two distributions are different, by randomly selecting data from the given distribution and calculating the $\hat{\theta}$ the H_1 distribution can be achieved. By studying how well the H_0 and H_1 distribution are separated, the power of the test can be calculated.

The two distributions used for this test are the Normal against the Cauchy. The procedure was done as follows, first the whole data sample was selected. Then samples were taken at random to generate the H_0 distribution using the permutation test. The GoF test is then repeated 1000 times by randomly picking data from the whole data sample to generate the H_1 distribution by collecting the $\hat{\theta}$ values. For example, a Gaussian and a Cauchy distributions are generated containing 5000 entries. Then a sample with 30 entries is selected which defines the H_0 distribution. Then 1000 samples of 30 are taken to calculate the probability of them falling in the H_0 distribution. The two distributions are then plotted against each other to calculate the power of the test. This test was performed for data samples with entries 30, 50, 100, 200, 500, 800, 1000 and 2000 to generate the plot showing H_0 and H_1 distribution for the for the sample data sets with high ASL values. The results can be seen in Figure 6.14. Here α is the cut off point where the H_0 is rejected. β is the probability that the statistical data will fall under the H_0 distribution and the Power of the test is calculated using $(1-\beta)$.

From the tests we conducted it can be seen that the energy test will give reliable results as long as the data samples used are representative of the true distribution of the sample data sets (having very small uncertainties). In this case the entry number of sample data sets for the test to give reliable result for comparison of Cauchy and Gaussian is around 200. We believe as with all GoF tests the energy test is dependant on the sample size and we have proved there are times when the test will give unreliable results for samples with entry points less than 100 in contradiction to [1].

6.5.2 Comparing Energy test with various other Goodness of Fit tests

To further check the usability of the implemented Energy test, Monte Carlo generated data (discussed in Chapter 4) was used. First the Energy test by itself was used to check if it would separate two distributions, one coming from aligned tracker geometry and a second one from misaligned geometry. The results are given in Table 6.6. The results form the χ^2

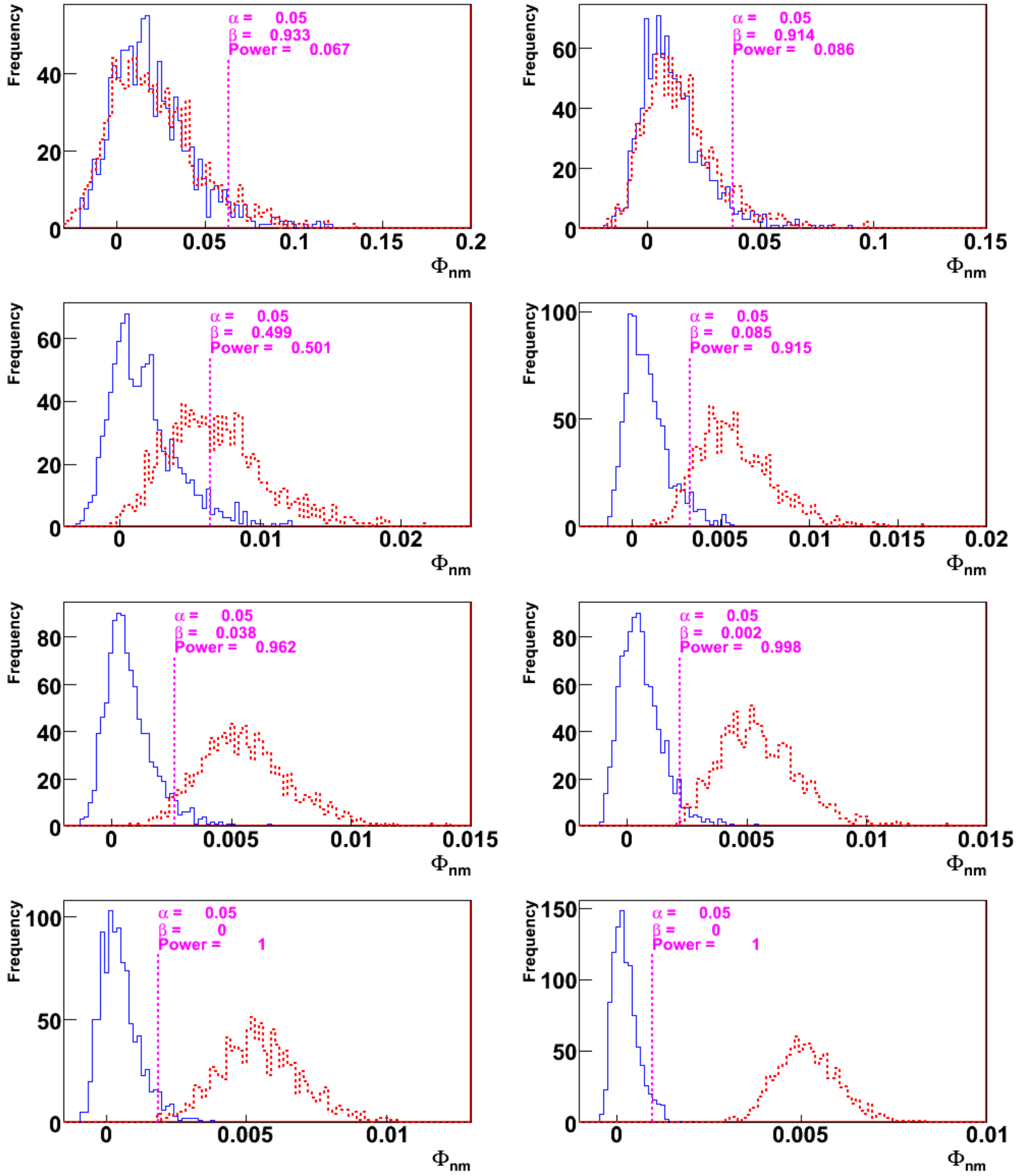

Figure 6.14: The null hypothesis distribution (blue) plotted against the alternative hypothesis (red) for the energy test between Gaussian ($mean = 0$, $\sigma = 1$ and $\rho = 0$) and Cauchy ($mean = 0$, $\sigma = 1$ and $\rho = 0$) distribution.

Files compared	$ASL_{(30,30)}$	$ASL_{(50,50)}$	$ASL_{(100,100)}$
χ^2 vs η	0.246	0.372	0.743
χ^2 vs ϕ	0.443	0.545	0.831
D_0 vs η	0.003	0.009	0.008
D_0 vs ϕ	0.021	0.015	0.006
Pt vs η	0.003	0.002	0.006
Pt vs η	0.01	0.007	0.01

Table 6.6: Results from the binning free energy test for data samples from the CMS tracker simulation with aligned against misaligned data sets. D_0 and P_t are the distance of closest approach and transverse momentum respectively.

parameter gave high ASL values which means the H_0 was accepted. The reason for this was discussed in Chapter 4, where the χ^2 parameter is affected by the alignment position error correction parameter (see Section 4.5.2). The values from D_0 and P_t gave very low ASL rejecting the H_0. This results showed the Energy test gives reliable test results on the MC generated data which is similar to the results in the bottom row of Table 6.4. This shows that if the data sample is representative (with small statistical uncertainties) of the underlying distribution the test will give reliable result even small data samples. The idea behind statistical uncertainties is discussed by G. Zech in [67].

The Energy test was then compared against two other goodness of fit test implementations, the binned Energy test discussed in section 6.4.6 and the 2D-KS test discussed in section 6.4.3. The binned Energy test implementation can take unbinned data, but the process will bin the data before calculating the Energy difference between the two distributions. The two data sets chosen for the test were the bivariate Gaussian compared against the bivariate Cauchy and data from the ideal Tracker geometry against the misaligned Tracker geometry. For the MC data the D_0 vs. ϕ track parameter data was taken. Each samples of data contained entries of around 50000. Out of the 50000 entries, samples were taken randomly to make new sets of data samples with their quantities varying from 30 up to 2000. The test was performed using high amount of reference data (2000) against small sample data (ranging $30 - 1000$). This was done to reflect the case of real usage of the test. High amount of simulated data is possible to achieve in most cases given there is enough computing power. The reason for using high amount of simulation data is to lower the statistical uncertainties of the simulated distribution [67].

The first set of test is a control test where data are randomly picked from the same distribution and compared against each other. The tests for the control data samples should accept the H_0 hence rejecting the hypothesis that the two distributions are different. Table 6.7 gives the results achieved for tests on sample sizes starting from 30 going to 1000. The results were as expected, giving high ASL values that exceeds the significance level of 10% showing that the H_0 can not be rejected on all tests.

On the second row of Table 6.7 where two Gaussian distributions, one with 2000 and the other with 50 entries, are compared there appears to be anomaly from the rest of the table. It seems to have the lowest ASL values in comparison to the rest and it is not consistent. This behaviour has been experienced in the tests done in Section 6.5.1. After a few more tests it was seen this behaviour is not only seen with 50 points but also with 100. Looking at Table 6.4 this behaviour is not seen when testing Normal distribution with $\bar{x} = 0$ and $\rho = 0.9$. A further test where 1000 samples of 50 were taken and tested against the H_0 was then done to see if this behaviour can be explained. It is clear from the top rightmost graph of Figure 6.14 where the H_0 is not separable from the alternative hypothesis. The results from Table 6.4 and Figure 6.14 bring us to the conclusion that the test does not give reliable results when using samples below 100 entries for Gaussian against Cauchy comparison. In fact the results form Figure 6.14 show that for the test to separate distributions of Gaussian and Cauchy, they would have to have entries of more than 200. Therefore the effects seen in Table 6.7 and 6.8 are arising from statistical fluctuation when very low samples are used. The reason for including data entries with as low as 30 is to be consistent with previous tests.

When the test was used to compare two different distributions, the Gaussian and the Cauchy, the results were again as expected giving values below the 10% significance level for samples with entries more than 200. The results can be seen in Table 6.8. The unbinned Energy test seems to be giving better results in comparison to the K-S test and the binned Energy test, as it gave smaller probability of accepting the H_0. It was also noticed that

Files compared	Energy test$_{Unbinned}$	Energy Test$_{Binned}$	2D K-S test
Gaussian Vs Gaussian $_{(2000,30)}$	0.53	0.117	0.634
Gaussian Vs Gaussian $_{(2000,50)}$	0.175	0.151	0.36
Gaussian Vs Gaussian $_{(2000,100)}$	0.653	0.84	0.529
Gaussian Vs Gaussian $_{(2000,200)}$	0.297	0.742	0.524
Gaussian Vs Gaussian $_{(2000,500)}$	0.632	0.849	0.926
Gaussian Vs Gaussian $_{(2000,1000)}$	0.699	0.998	1

Table 6.7: Comparison of two different Gaussian samples generated using the R statistics package. Three different goodness of fit tests were used to test the same data sample. Both the unbinned tests gave similar results.

Files compared	Energy test$_{Unbinned}$	Energy Test$_{Binned}$	2D K-S test
Gaussian Vs Cauchy $_{(2000,30)}$	0.638	0.402	0.658
Gaussian Vs Cauchy $_{(2000,50)}$	0.09	0.211	0.021
Gaussian Vs Cauchy $_{(2000,100)}$	0.152	0.515	0.313
Gaussian Vs Cauchy $_{(2000,200)}$	0	0.005	0.043
Gaussian Vs Cauchy $_{(2000,300)}$	0	0.011	0.028
Gaussian Vs Cauchy $_{(2000,400)}$	0	0.001	0.015
Gaussian Vs Cauchy $_{(2000,500)}$	0	0	0.003
Gaussian Vs Cauchy $_{(2000,700)}$	0	0.001	0.001
Gaussian Vs Cauchy $_{(2000,1000)}$	0	0	0

Table 6.8: Comparison Gaussian and Cauchy samples generated using the R statistics package.

the test was giving inconsistent values for samples between 30 and 100 which is due to statistics uncertainty discussed above.

A further test using the Monte Carlo simulation data was used for testing the goodness of fit tests. The data samples were taken using the ideal and misaligned Tracker geometry. For the test results listed in Table 6.9 the Monte Carlo simulation data discussed in Section 4.5 was used to reconstruct tracks using both the ideal and misaligned Tracker

Files compared	Energy test$_{Unbinned}$	Energy Test$_{Binned}$	2D K-S test
D_0 vs $\phi_{(2000,30)}$	0.051	0.29	0.294
D_0 vs $\phi_{(2000,50)}$	0.008	0.057	0.356
D_0 vs $\phi_{(2000,100)}$	0	0.005	0.008
D_0 vs $\phi_{(2000,200)}$	0	0.001	0
D_0 vs $\phi_{(2000,300)}$	0.001	0.026	0.009
D_0 vs $\phi_{(2000,500)}$	0.005	0.017	0.006
D_0 vs $\phi_{(2000,1000)}$	0	0.006	0

Table 6.9: Ideal vs Misaligned data test The data used were on Track parameters the D_0 and Pt which are the distance of closest approach and transverse momentum respectively.

geometry. Track parameter D_0 plotted against ϕ from the ideal geometry is then compared against the misaligned. The results are consistent with the results achieved from comparing Gaussian distribution against the Cauchy. The results for samples with small entries are inconsistent but as the number of entries increase the results give values that decreases consistently. Again in these results the Energy test gives smaller probability of accepting H_0 compared to the KS test.

6.6 Conclusion

The reason for studying the goodness of fit tests discussed in this chapter is the lack of suitable test in the DQM sub-system. As the results show the Energy test gave reliable results which can be used to identify anomalies in data generated by the Tracker. This test was implemented in a C++ code ready to be integrated with the DQM_Services package of the CMSSW. This has has been welcomed by the DQM team and is awaiting to be integrated and will be release with the next major CMSSW version release.

Overall the GoF tests used in this chapter were able to separate between two different distributions given the samples contain enough data to make the statistical fluctuations negligible. We followed the recommendations given by [1] and used data samples with as low as 30. It was not possible to find the data samples used by [1] therefore random generators were used from R statistics package. We found that tests for distributions such as Gaussian against Cauchy or Gaussian against Student$-t_4$ samples with entries below 200 did not give reliable results. For samples which are very different such as Gaussian with $\bar{x} = 0$ and $\rho = 0.9$ samples with entries as low as 30 gave reliable results.

Chapter 7

Conclusion and future work

The nominal data read-out rate of the CMS detector is 40 MHz. Typical *pp* collisions at this rate with a beam energy of 7 TeV and luminosity of $\mathcal{L} = 10^{34}$ cm$^{-2}s^{-1}$ will result in the creation of almost 1000 particles every 25 ns. This would mean nearly 650 MB of data will be produced at the rate of 40 MHz by the Tracker readout system alone. This data is far too large to be stored for later analysis and has to be reduced from 40 MHz to a 100 Hz readout rate. This reduction is achieved using two levels of triggering system. The Level 1 Trigger, which is made up of programable logic controls, will initially reduce the rate to 100 kHz. The next triggering system, the High Level Trigger (HLT), will then take the data that has been accepted by the Level 1 Trigger and reduce the rate further to 100 Hz.

The HLT uses very fast reconstruction algorithms to decide whether an event is interesting or not. Once the HLT has accepted a physics event as interesting then the quality of the reconstructed physics object is checked using the tools provided by the data quality monitoring (DQM) system. If the reconstructed object is rejected by the DQM module then the event will not be saved.

One DQM module that will be used by the CMS data acquisition system is the track monitoring package discussed in this thesis. It was developed to monitor physics objects known as tracks and produces data that can be used to monitor the quality of a reconstructed track. These data, known as Monitoring Elements, are then analysed to make a decision on whether a particle track is of acceptable quality. The Tracker_Monitor_Track package is designed to run alongside the HLT in the data acquisition computer farm so that it can help maintain the quality of tracks reconstructed by the fast reconstruction algorithms. The package has two components, the Monitor_Track_Global and the Monitor_Track_Residual.

The Monitor_Track_Global is built to monitor the global parameters of a track such as the track χ^2, track η, track P_t etc. These parameters provide important information on how good the quality of a reconstructed track is. The Monitor_Track_Residual module helps monitor the quality of reconstructed track with respect to the individual detector modules of the Tracker. In the case of Tracker misalignment or faulty detector modules the data generated by the Monitor_Track_Residual module will indicate these faults.

The Tracker_Monitor_Track package was successfully built and it has played an important role in the Magnet Test and Cosmic Challenge (MTCC) test and the commissioning of the Tracker in the Tracker Integration Facility at CERN. During the MTCC, where part of the Tracker was installed to be tested using cosmic rays, the Tracker_Monitor_Track package was used to monitor the quality of the reconstructed tracks and it was used to reject events that had poor quality tracks. It also provided residual information which was used in the alignment of individual detector modules with respect to each other. The success of the package during these tests has led to the package being integrated with the muon detector monitoring system. The tracker alignment group has also used it for analysis of their data.

The next step in the development of the Track quality monitoring system, after providing the Tracker_Monitor_Track package, was to provide statistical test tools which can be used to identify anomalies in the Tracker data. However, the lack of a two-dimensional statistical test in the DQM system meant that research was needed to find one. The physics community at CERN uses the ROOT statistical package for most analysis work, and interfacing the tools provided by ROOT was seen as an option, but ROOT's implementation of the two-dimensional Kolmogorov-Smirnov test was found to be unreliable as discussed in Chapter 6.

To resolve this issue of statistical test a so-called "Energy test" [1] which is a newly developed goodness of fit test was implemented. This is a powerful test for two reasons;

firstly it uses all the data points in a sample and secondly it does not depend on the ordering of the data. When testing the Energy test implementation, it was observed that the power of the goodness of fit test given in [1] could not be achieved for data samples between $30 - 100$ entries. The reason for this is explained in Chapter 6. The test has showed to be reliable, with data sample sizes greater than 200, giving a high power for the test. The implementation was also compared to two different goodness of fit tests and it proved to be as good and even better in some cases.

We recommended the integration of the Energy test in to DQM_Services sub-system for analysis of Monitoring Elements with two-dimensional data sets based on the fact that the test has been shown to be reliable and that the simplicity of its implementation means maintenance of the software package will be relatively easy. The software package which implements the Energy test has been accepted by the DQM group at CERN and it will be integrated with the next release of the CMSSW software.

7.1 Future work

At present the Tracker_Monitor_Track package and the Energy test implementation are two decoupled systems. The monitoring data generated by the Tracker_Monitor_Track package is saved in a file where the Energy test implementation can later be used to analyse the data. These two systems however can be combined together to create an "Expert system" which can run in the CMS data acquisition computer systems along with the HLT. An expert system in this context is defined as a system which consists of a set of rules that are devised by human experts to help make decisions.

Due to time constraints it was not possible to integrate these two software packages to create an Expert system, but this is something that can be done in the future to help process data quality monitoring of CMS. Figure 7.1 shows a flow chart of how the Expert system could be set up. The process starts by generating monitoring elements using the Tracker_Monitor_Track package. These monitoring elements are then accessed directly by the Expert system to check for anomalies. In the case of anomalies it would start working its way through the various levels of the data structure where the monitoring elements are stored until all the erroneous data has been found. After the erroneous data has been isolated there is a choice of making the Expert system fix the problem or send an alarm message.

The Expert system can be used to correct some errors. One such example would be in the case of misalignment, where the alignment position error (APE) variable can be adjusted to compensate for the detector misalignment. The APE has been discussed in Chapter 4 where it is useful when a misalignment has been noticed. This information is provided by the Monitor_Track_Residual of the Tracker_Monitor_Track package.

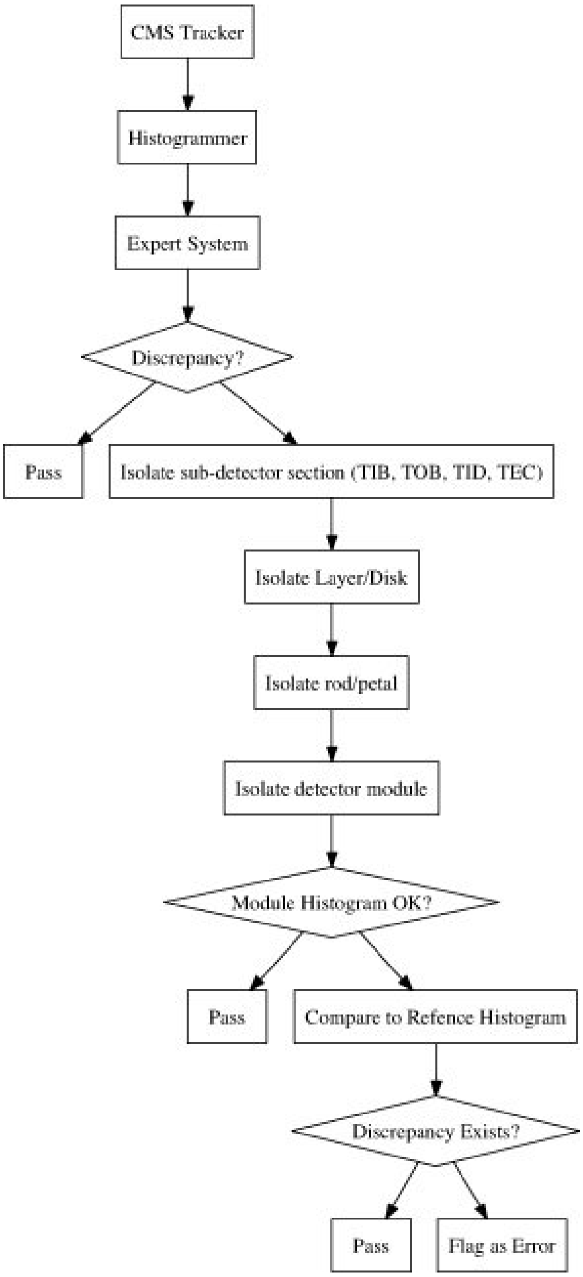

Figure 7.1: A flow chart of the Expert System showing the various steps needed to detect Error

Appendix A

Hit Residual distribution fit

The hit residuals of the Silicon Strip Tracker detector modules (one from each layer or disk) were fitted with single, double, triple and quadruple Gaussian pdfs. Each of the figures below have four histograms showing the four attempted fits to the distribution of the hit residual. It was observed that all the data fit to double Gaussian pdf well. It was also observed that although all of them have $mean \approx 0$, the σ of the detector modules from the different parts were varied. This is clearly seen when comparing residuals of detector modules from TIB and TID.

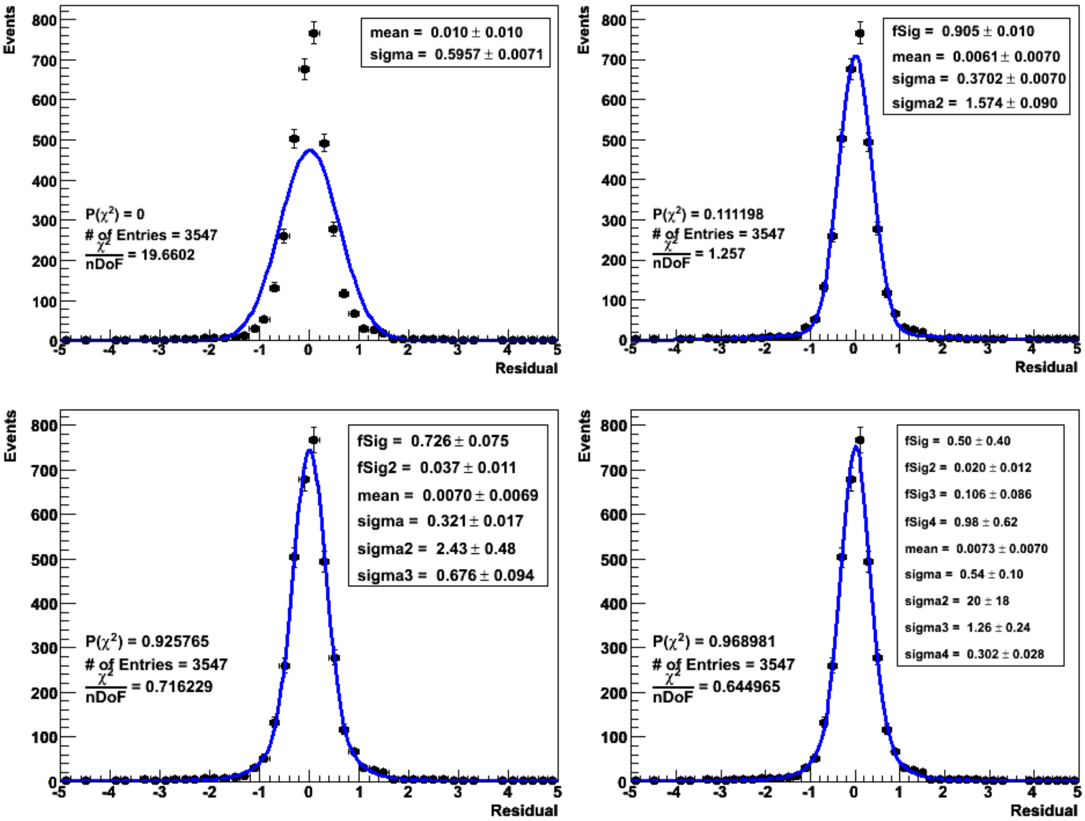

Figure A.1: Hit residual of TIB detector module from layer 1.

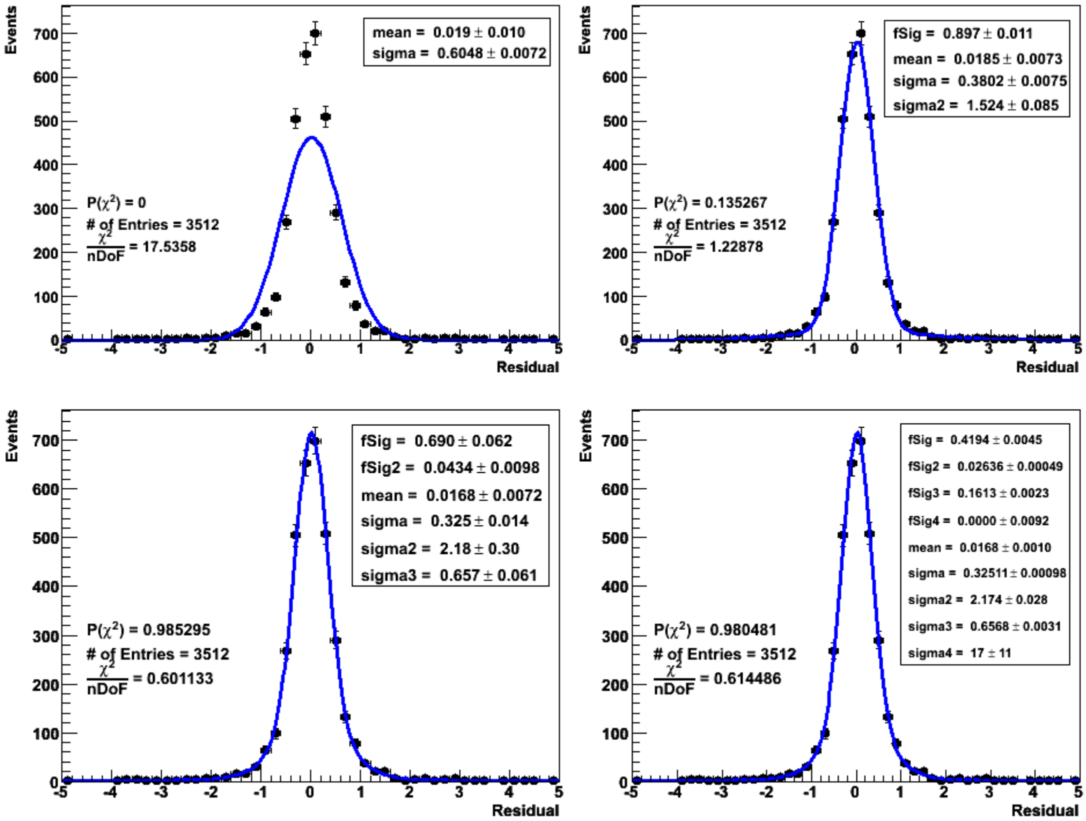

Figure A.2: Hit residual of TIB detector module from layer 2.

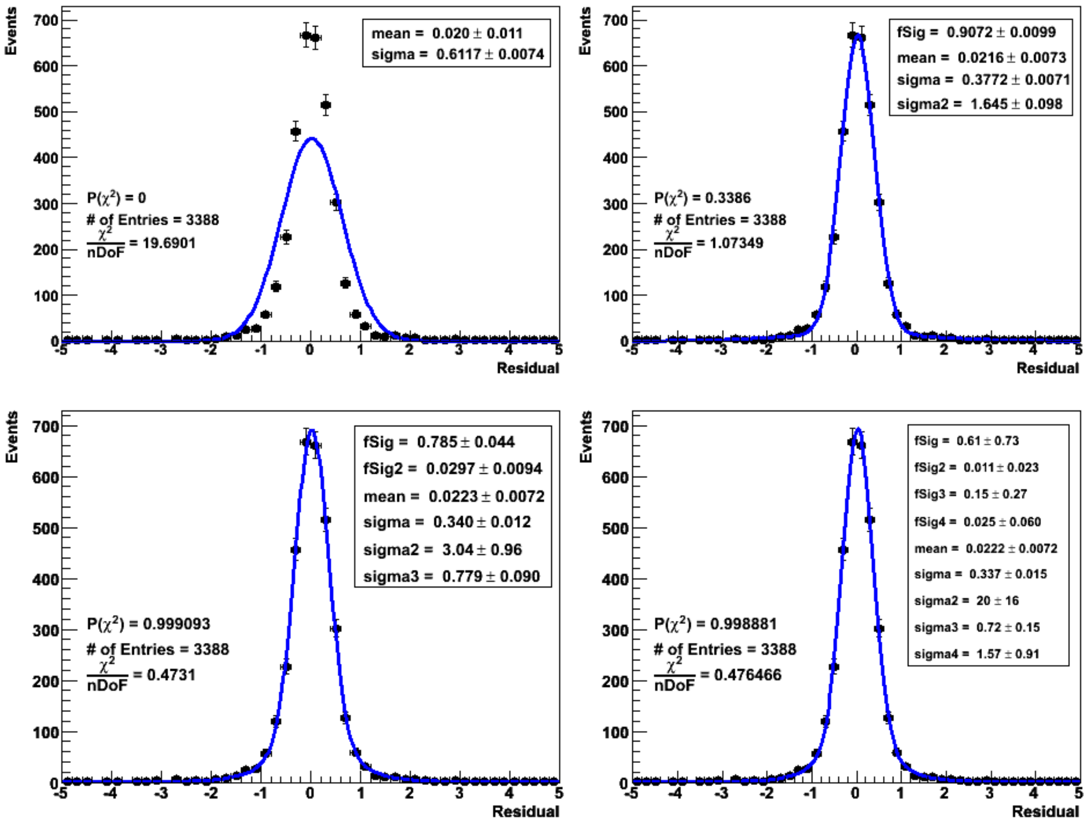

Figure A.3: Hit residual of TIB detector module from layer 3.

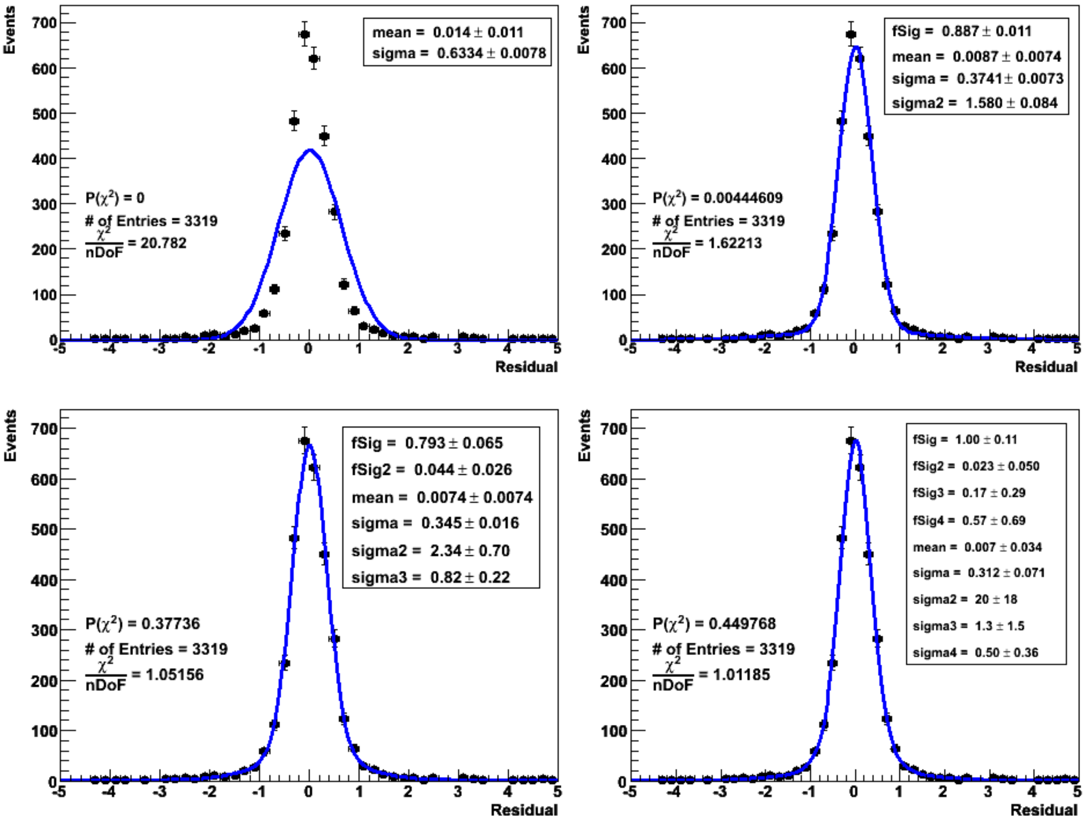

Figure A.4: Hit residual of TIB detector module from layer 4.

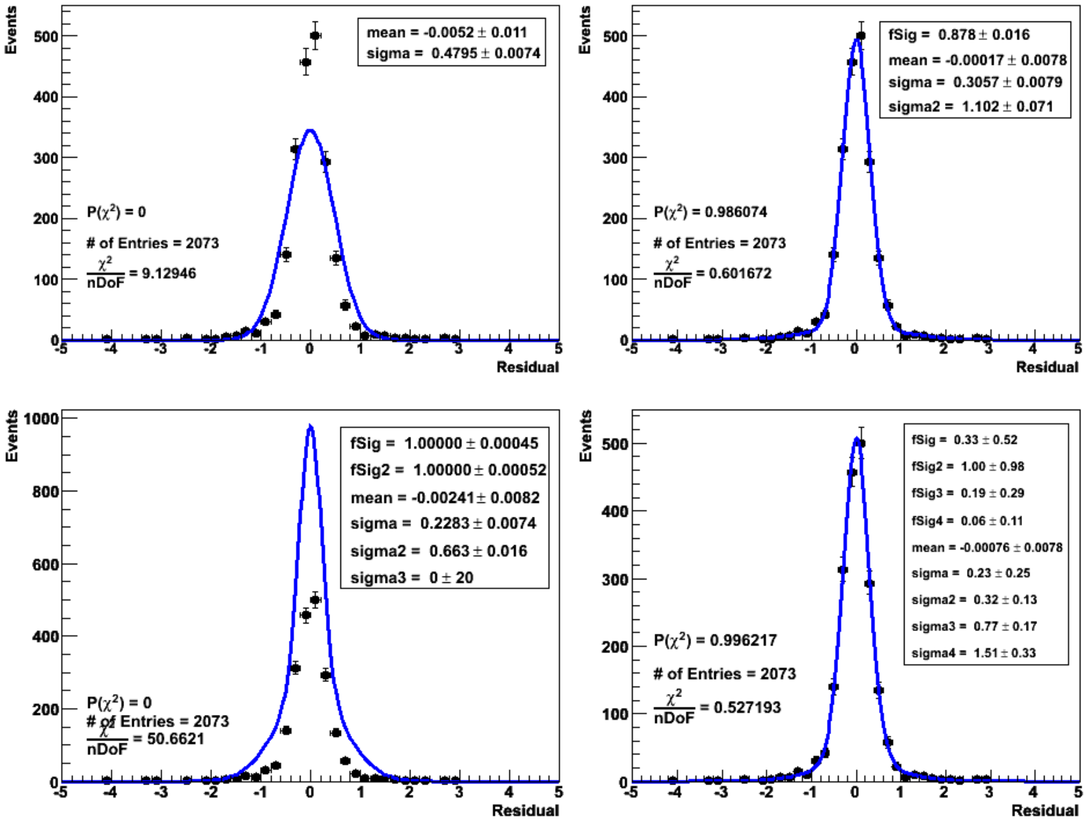

Figure A.5: Hit residual of TOB detector module from layer 1.

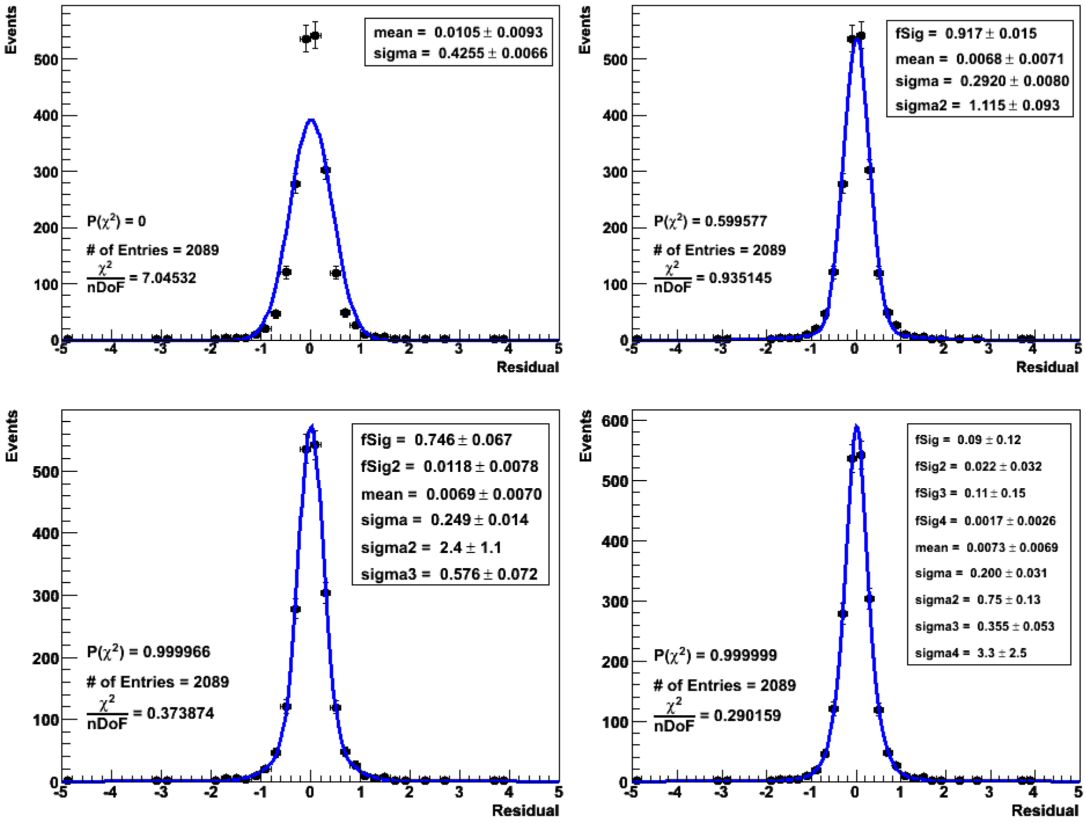

Figure A.6: Hit residual of TOB detector module from layer 2.

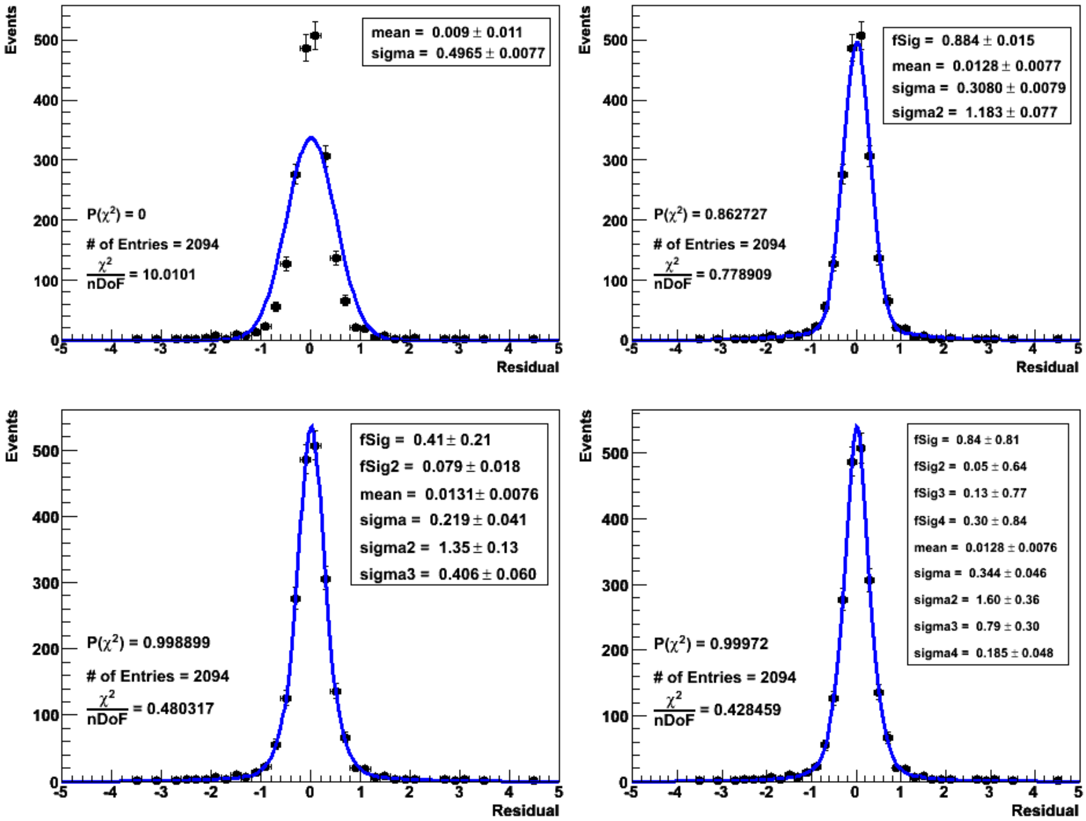

Figure A.7: Hit residual of TOB detector module from layer 3.

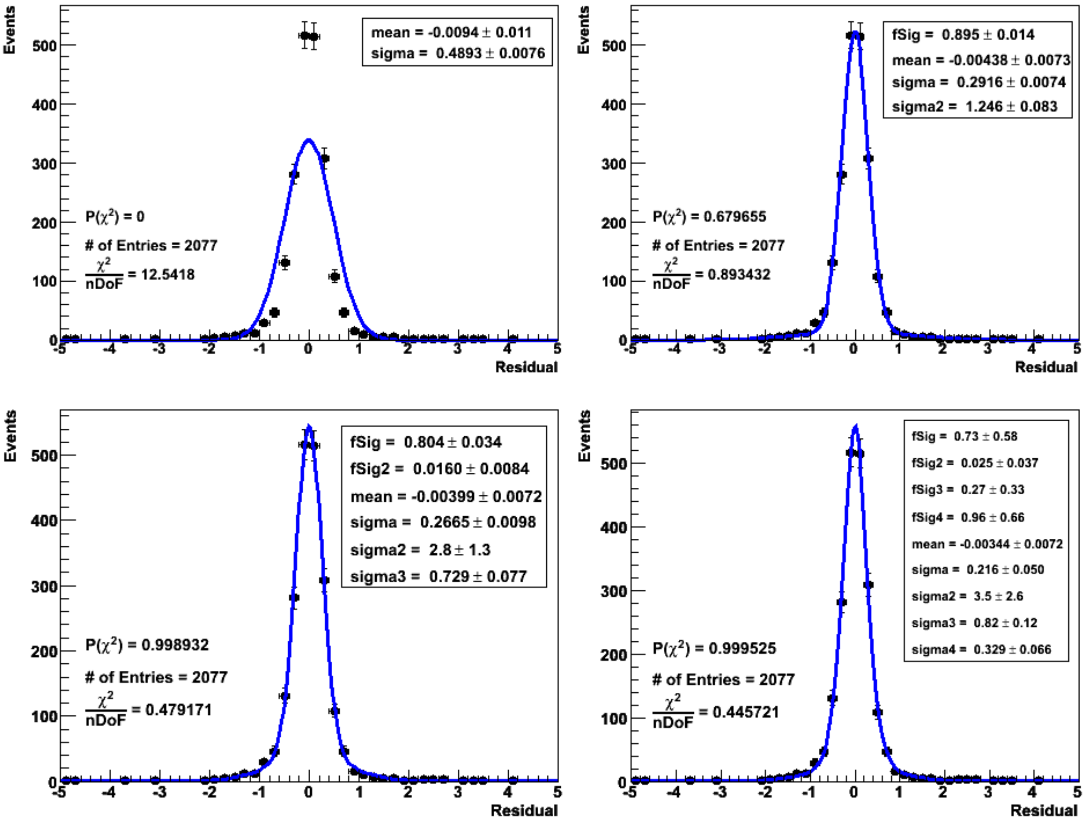

Figure A.8: Hit residual of TOB detector module from layer 4.

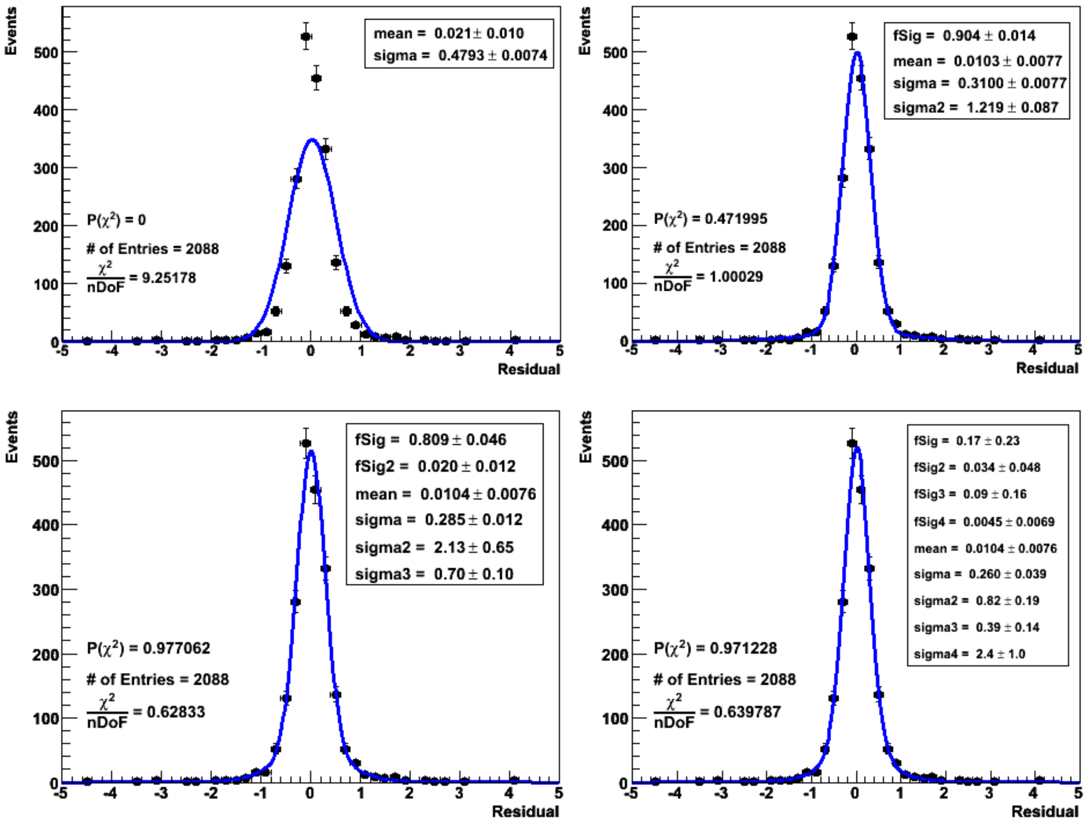

Figure A.9: Hit residual of TOB detector module from layer 5.

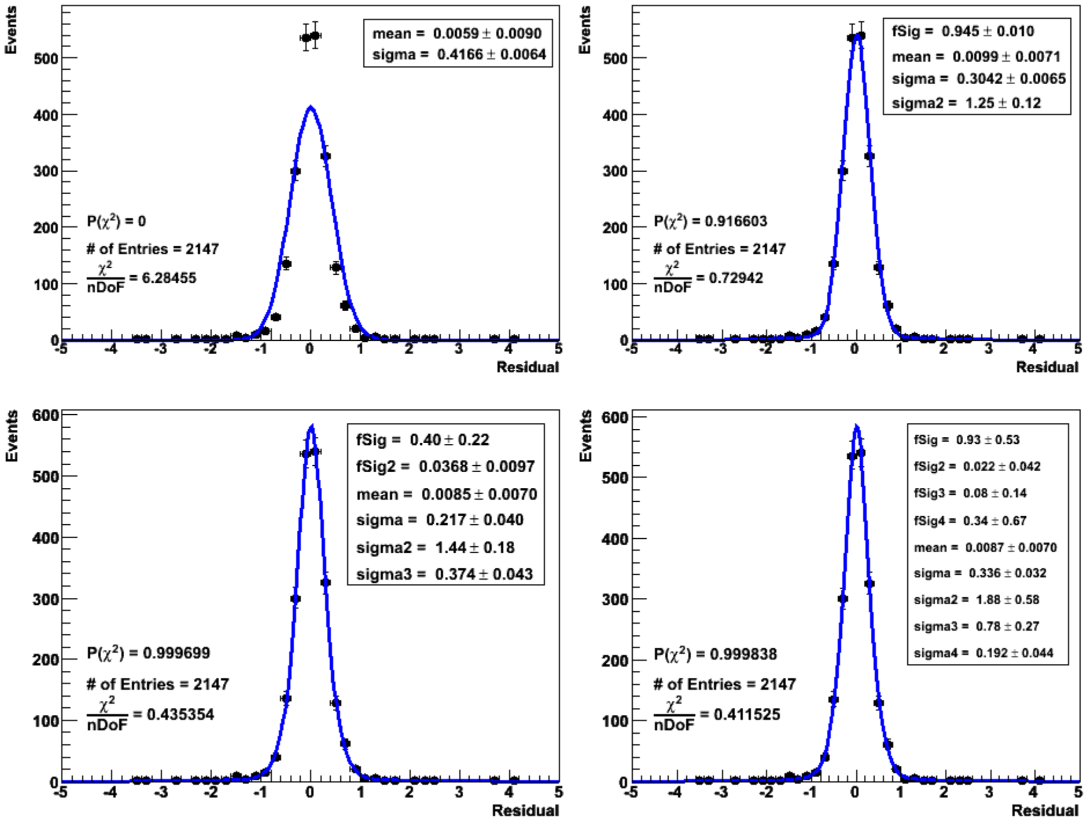

Figure A.10: Hit residual of TOB detector module from layer 6.

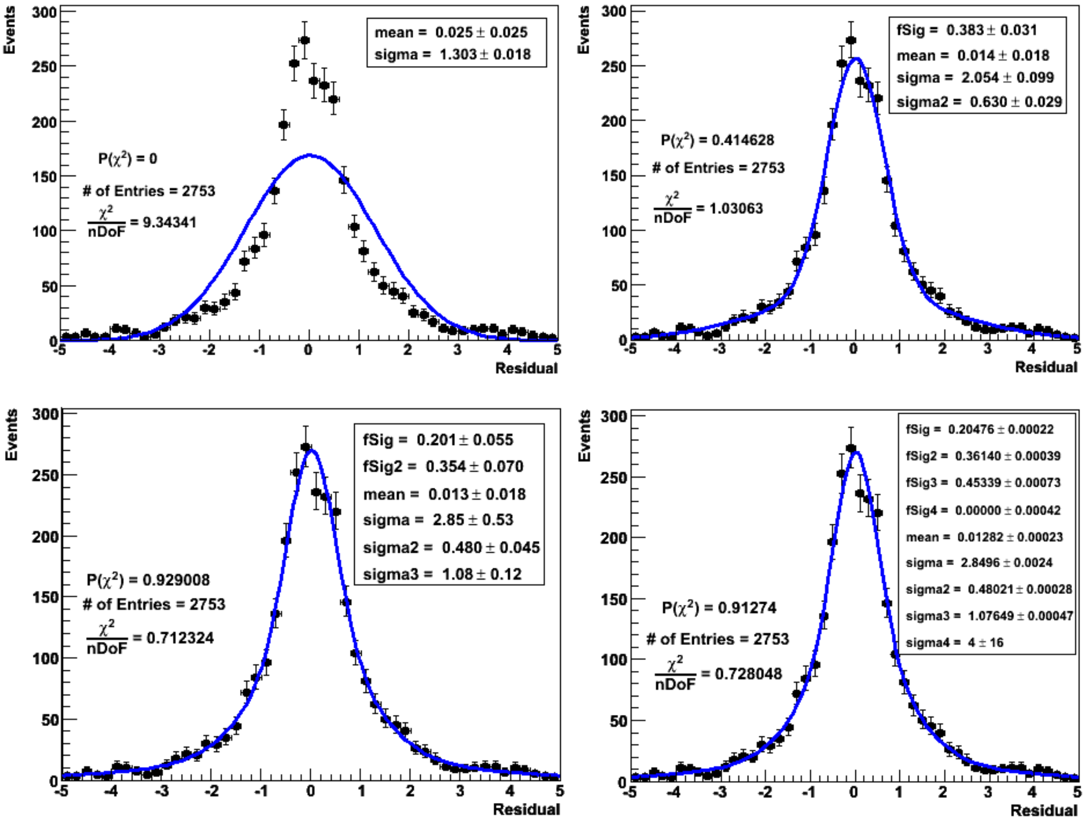

Figure A.11: Hit residual of TID detector module from disk 1.

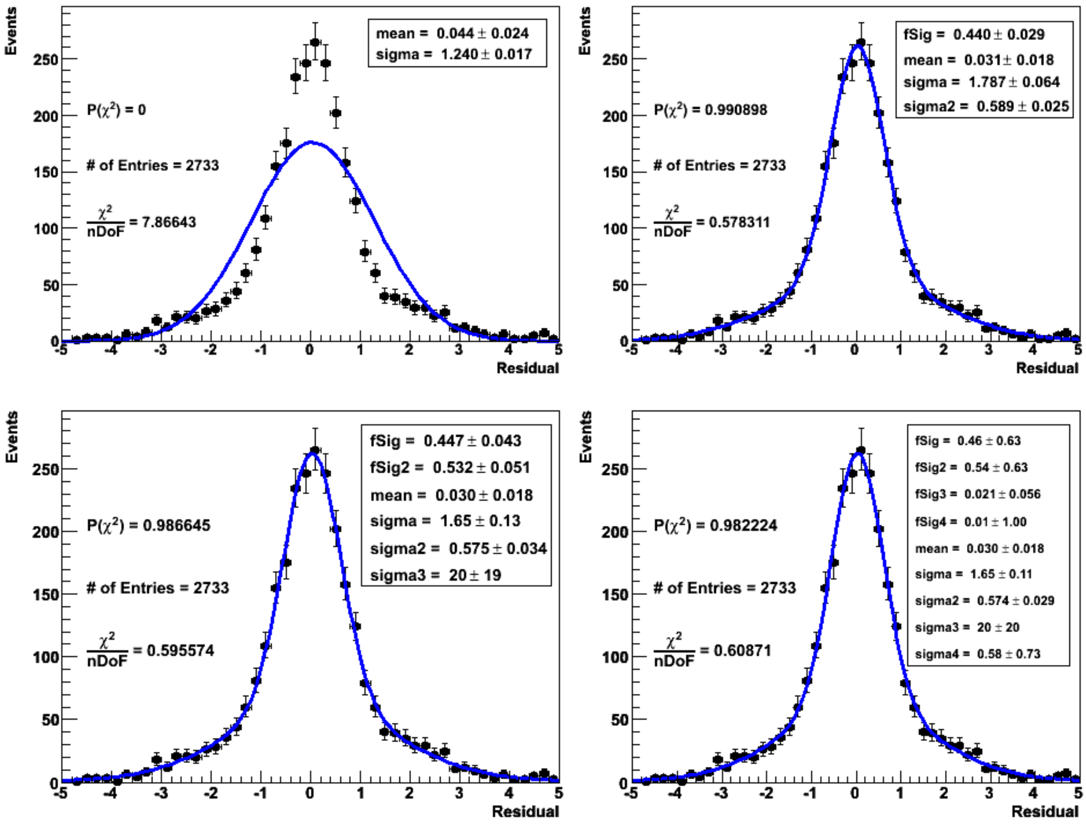

Figure A.12: Hit residual of TID detector module from disk 2.

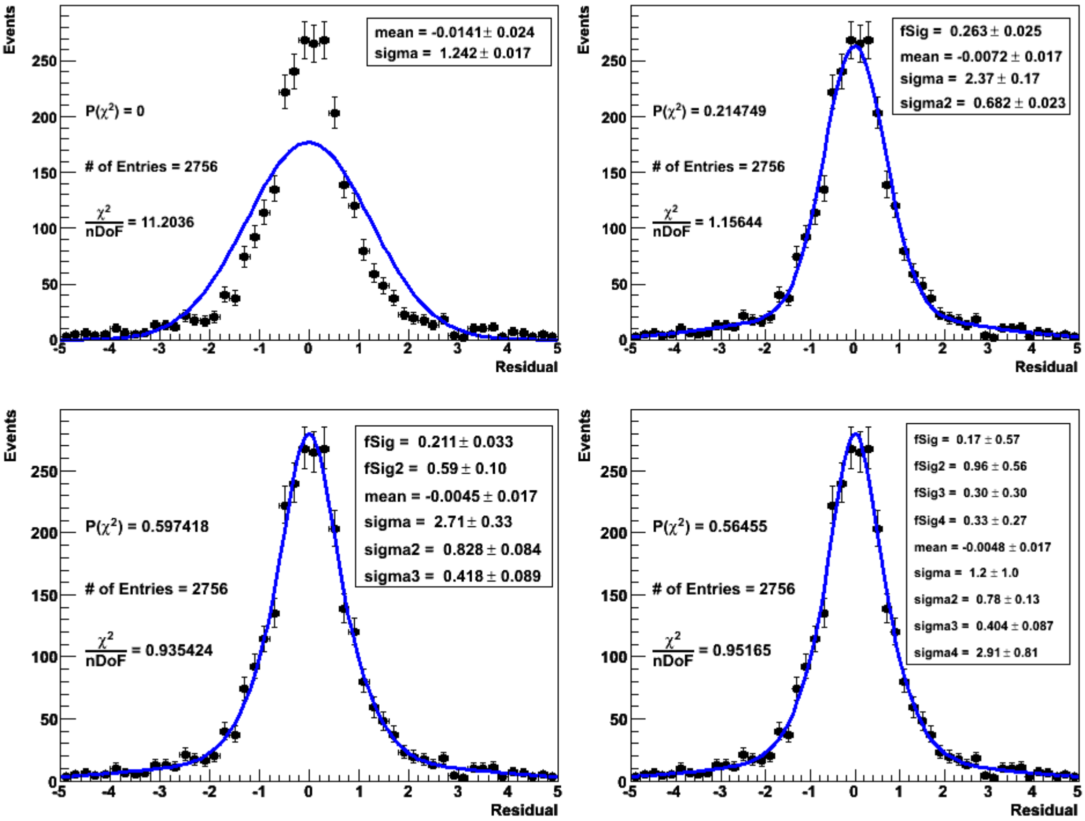

Figure A.13: Hit residual of TID detector module from disk 3.

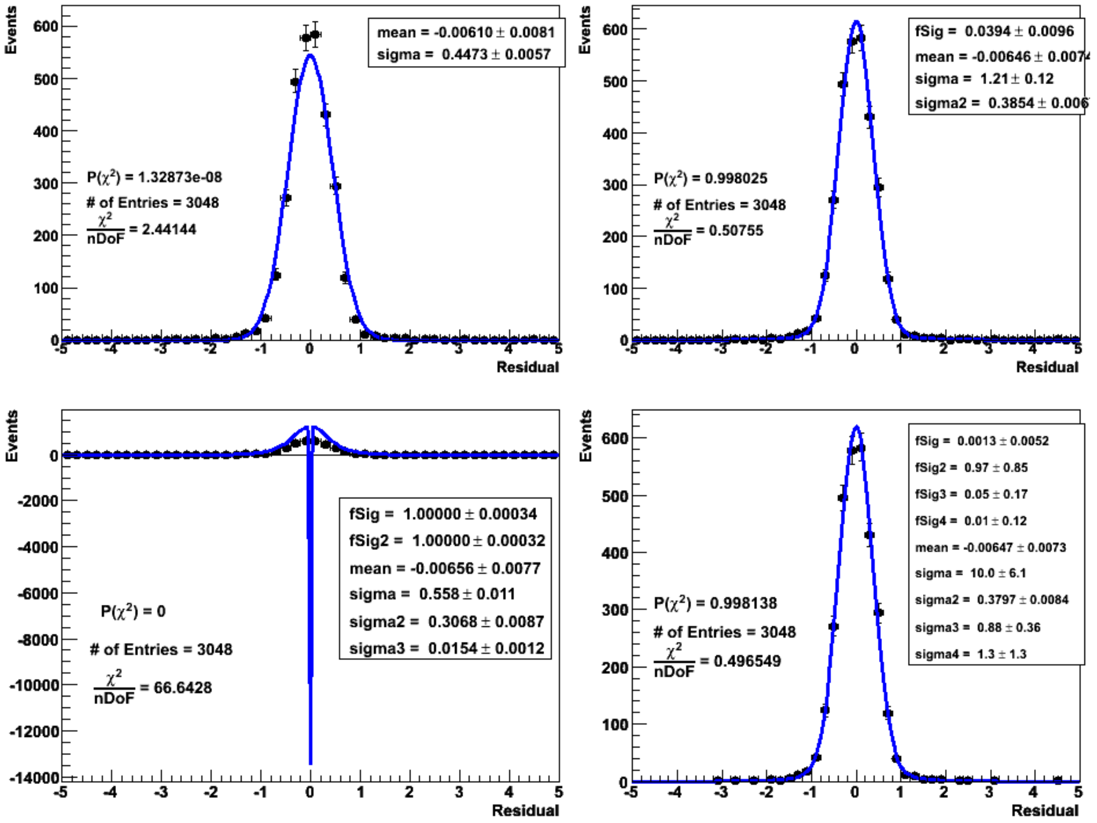

Figure A.14: Hit residual of TEC detector module from disk 1.

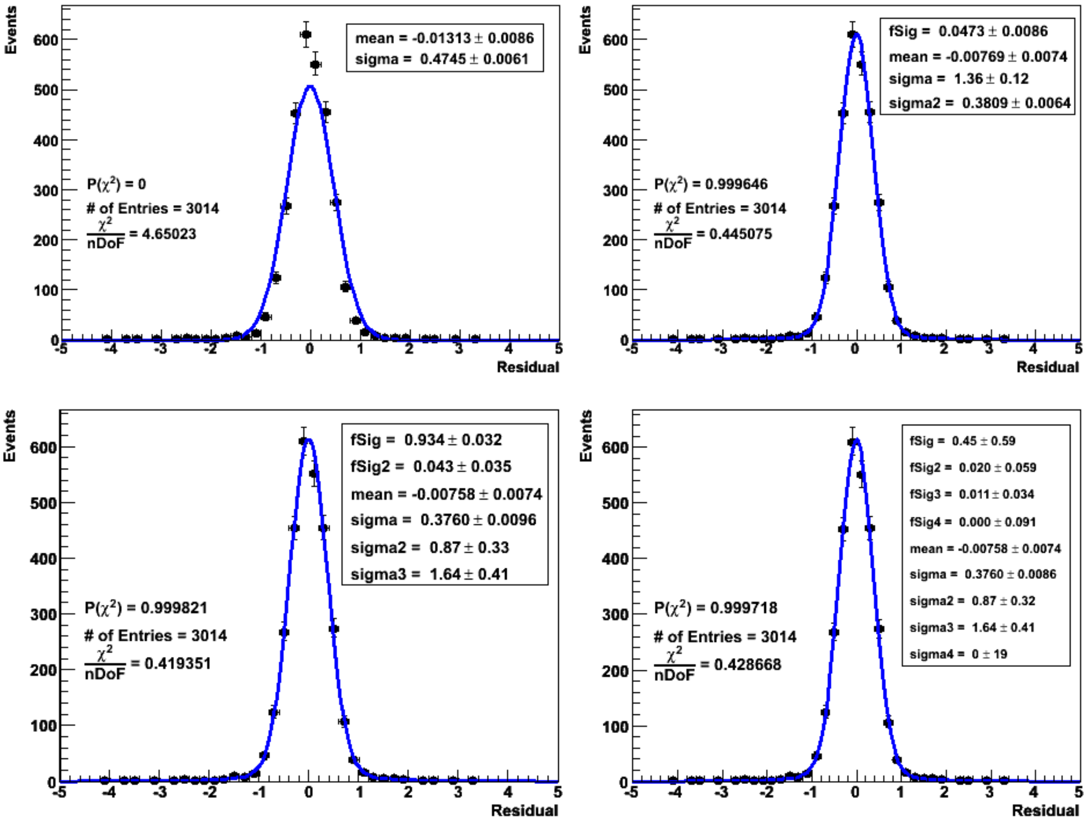

Figure A.15: Hit residual of TEC detector module from disk 2.

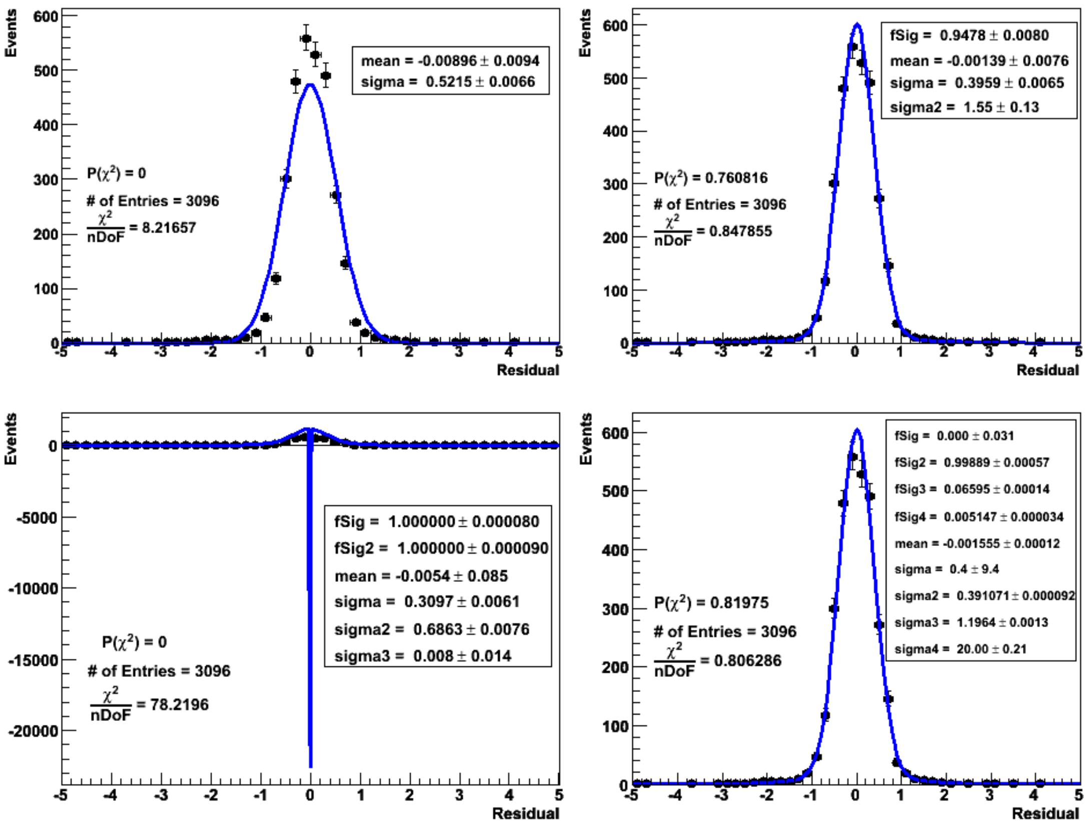

Figure A.16: Hit residual of TEC detector module from disk 3.

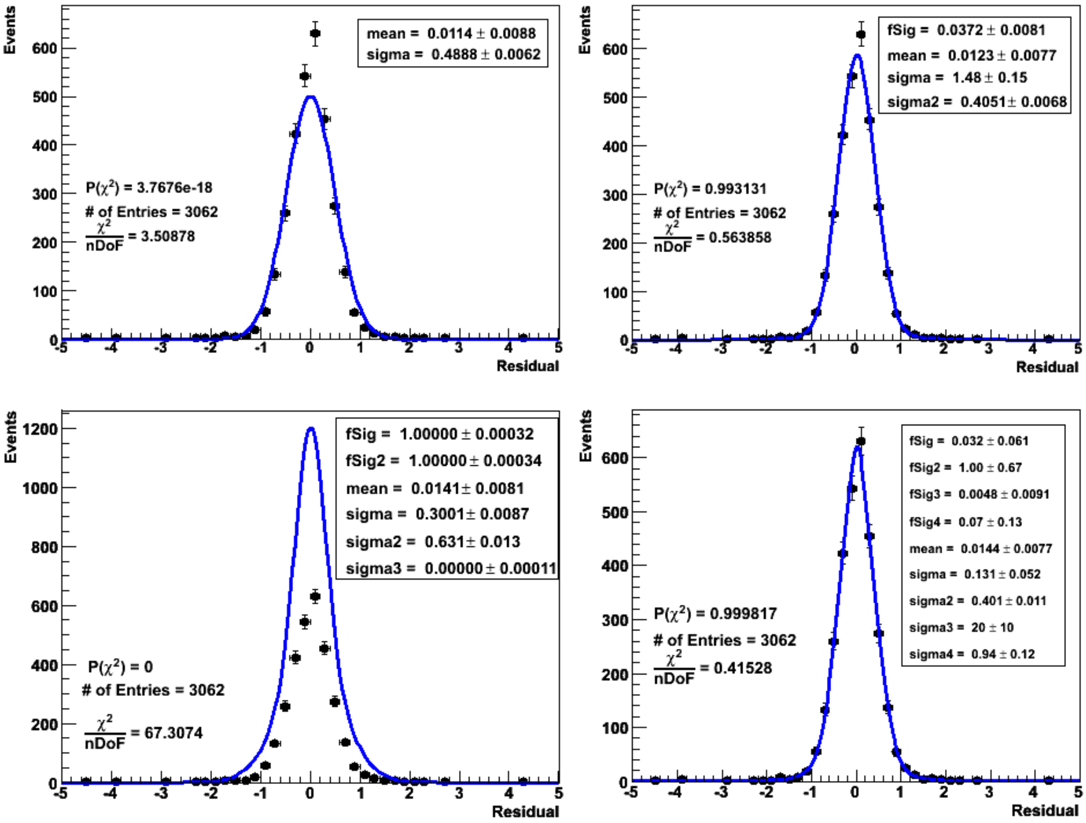

Figure A.17: Hit residual of TEC detector module from disk 4.

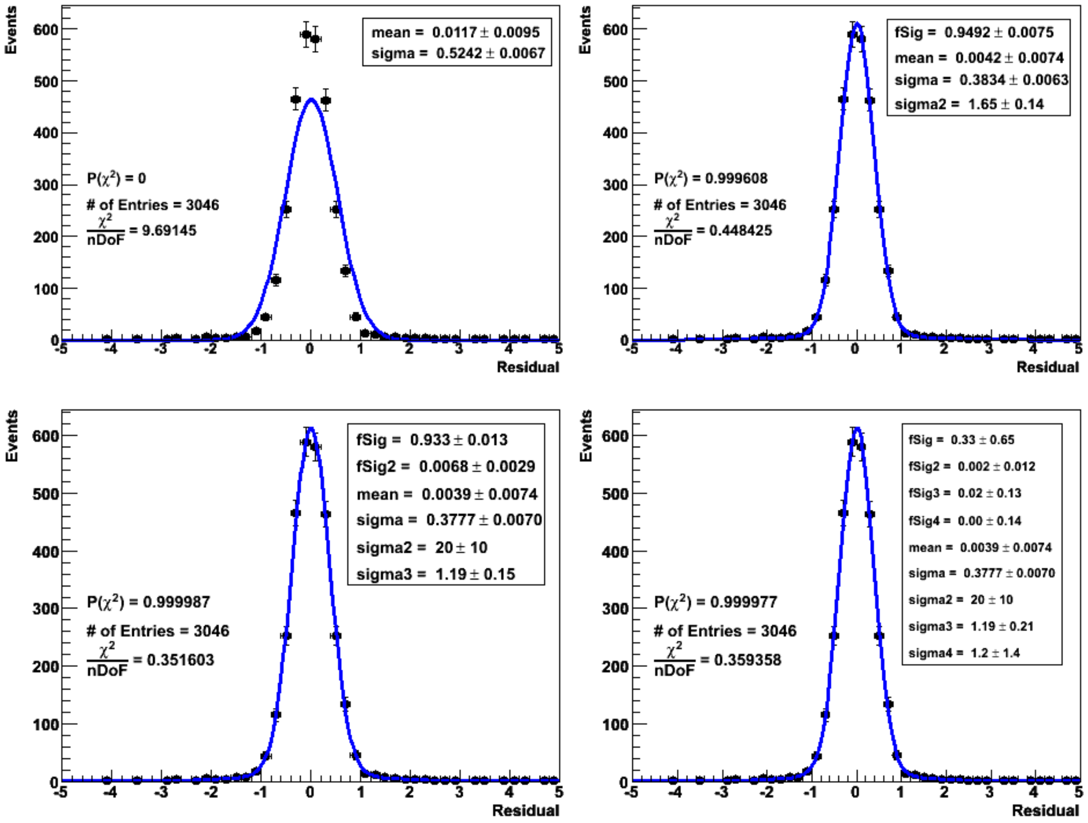

Figure A.18: Hit residual of TEC detector module from disk 5.

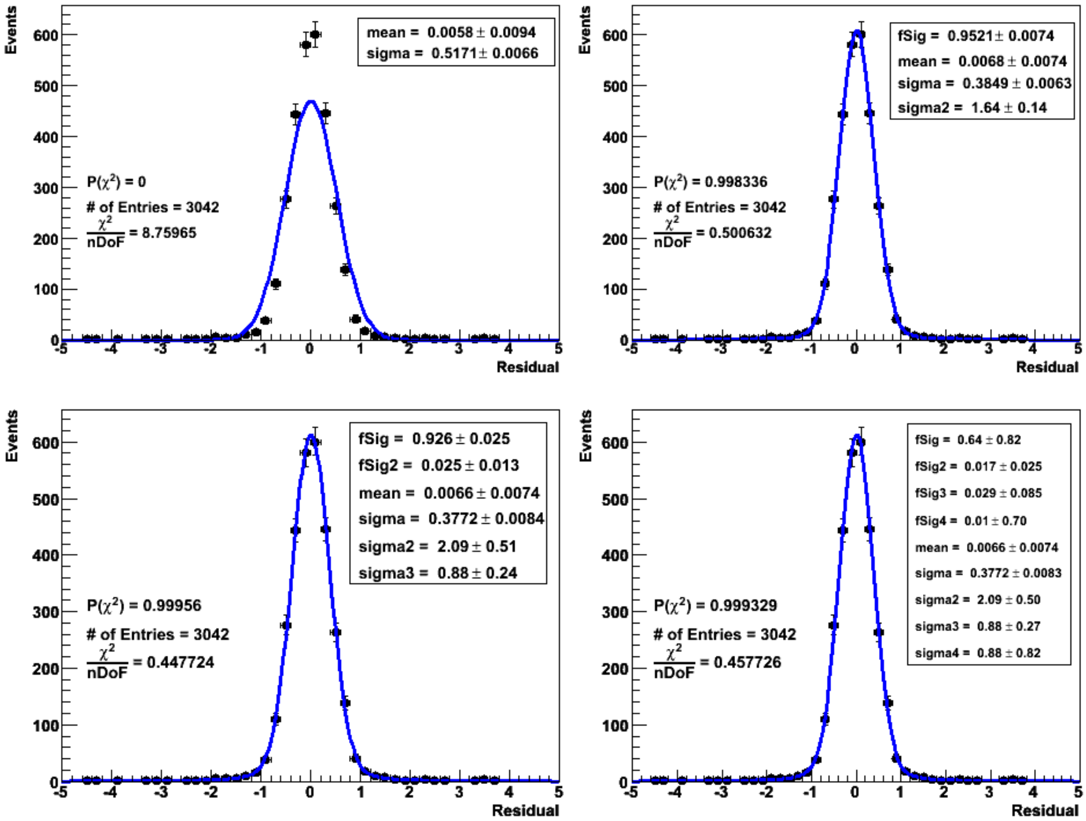

Figure A.19: Hit residual of TEC detector module from disk 6.

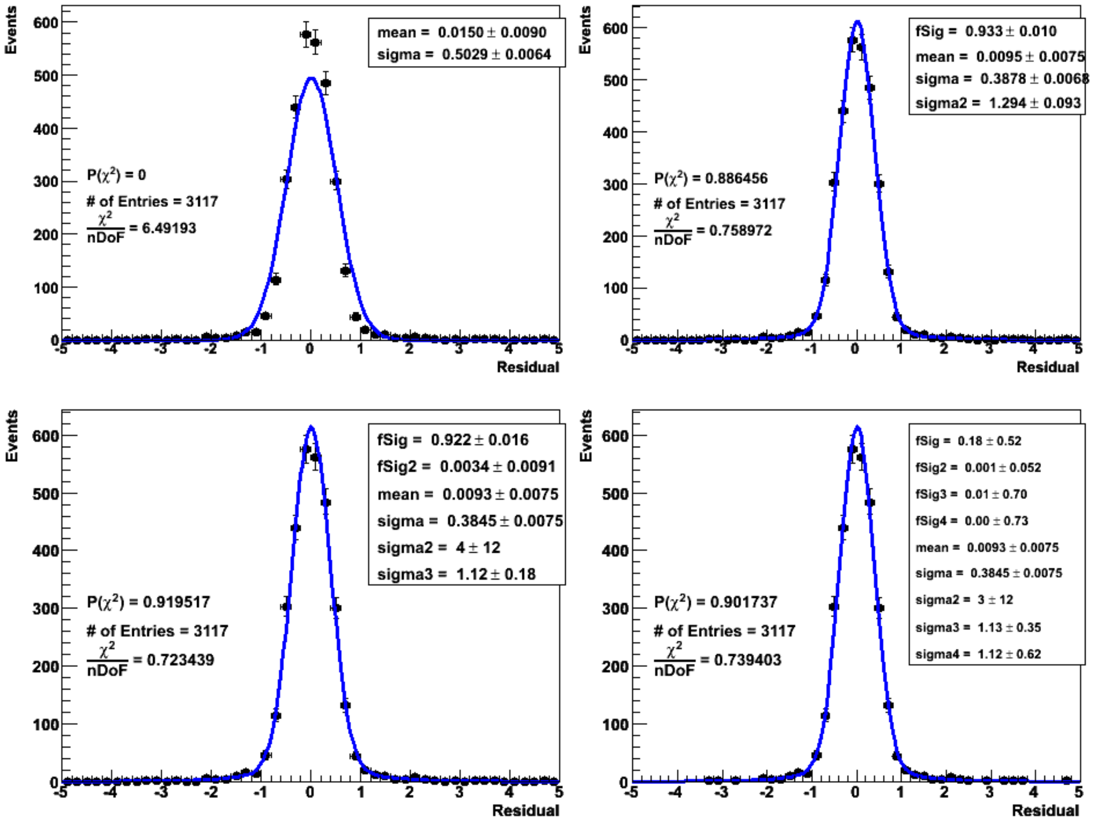

Figure A.20: Hit residual of TEC detector module from disk 7.

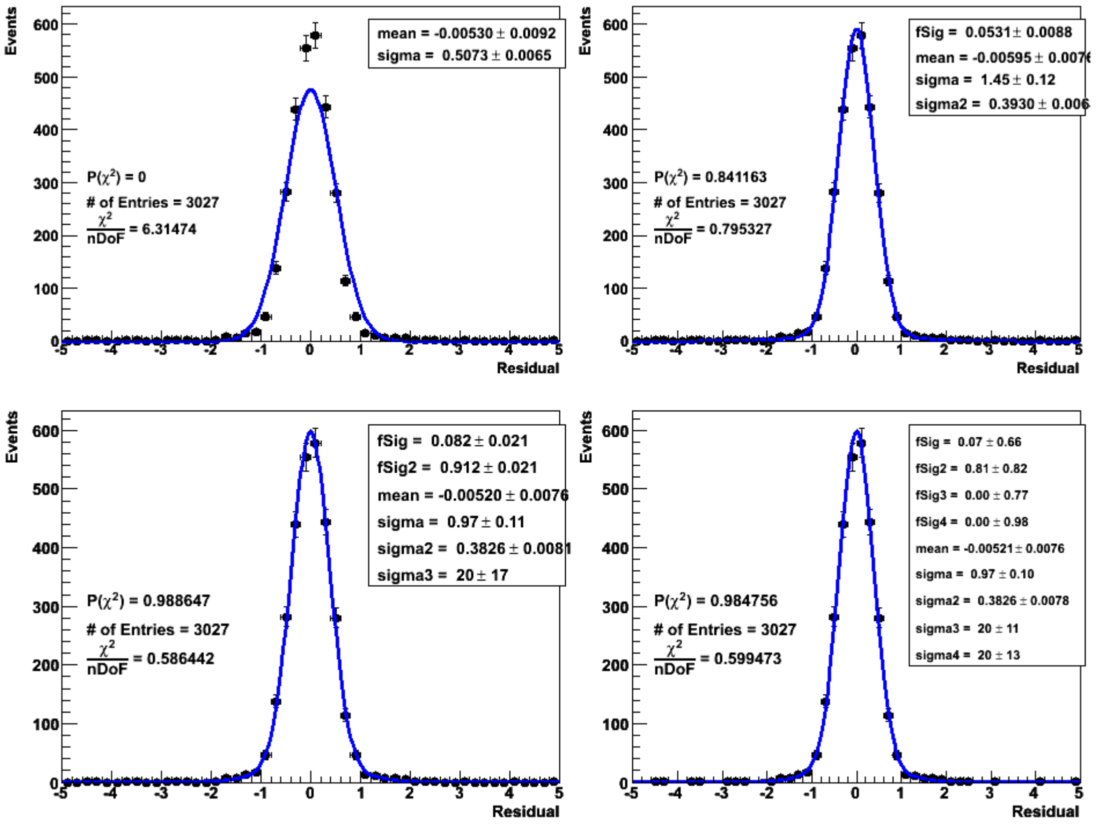

Figure A.21: Hit residual of TEC detector module from disk 8.

Bibliography

[1] G. Zech and B. Aslan. A new test for the multivariate two-sample problem based on the concept of minimum energy. *University of Siegen. arXiv:math/0309164v1 [math.PR]*, September 2003.

[2] *Review of particle physics (Particle Data Group)*. Journal of Physics G: Nuclear and Particle Physics, 2006.

[3] The CMS Collaboration. CMS physics, technical design report volume 1. Cern/lhcc 2006-001, CERN, 2 February 2006.

[4] CMS Collaboration. The CMS Experiment at the CERN LHC. *Journal of Instrumentation. 2008 JINST 3 S08004*, 2008.

[5] L. Silvestris. Performance of the silicon detectors for the CMS barrel tracker. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 392(1-3):161–164, 6/21 1997.

[6] http://livefromcern.web.cern.ch/livefromcern/antimatter/history/historypictures/LHC-drawing-half.jpg.

[7] http://cms-project-cmsinfo.web.cern.ch/cms-project-cmsinfo/index.html.

[8] The CMS Collaboration. The tracker project, technical design report. CERN/LHCC 98-6, CERN, European Laboratory for Particle Physics, 15 April 1998.

[9] The CMS Collaboration. Addendum to the CMS tracker TDR. CERN/LHCC 2000-016, CERN, 21 February 2000.

[10] Manfred Krammer. The silicon sensors for the inner tracker of the Compact Muon Solenoid experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 531(1-2):238–245, 2004/9/21.

[11] Wolfram Erdmann. The CMS pixel detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 447(1-2):178–183, 6/1 2000.

[12] C. Civinini, S. Albergo, M. Angarano, and et. al. CMS silicon tracker developments. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 477(1-3):440–445, 1/21 2002.

[13] Thomas Ferbel. *Experimental Tchniques in High Energy Physics.* Addison-Wesley Publishing Company, Inc., 1987.

[14] CMS Collaboration. *The hadron calorimeter technical design report.* CERN/LHCC 97-31, CMS TDR2, 1997.

[15] *CMS MUON Technical Design Report.* CMS Collaboration, CERN/LHCC 97-32, December, 1997.

[16] J. Fernandez. The CMS silicon strip tracker. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 573(1-2):257–259, 4/1 2007.

[17] Thomas Muller. The CMS tracker and its performance. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 408(1):119–127, 5/1 1998.

[18] Stefan Schael. The CMS silicon strip detector–mechanical structure and alignment system. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 511(1-2):52–57, 2003/9/21.

[19] Giacomo Sguazzoni. CMS inner tracker detector modules. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 552(1-2):212–215, 10/21 2005.

[20] Manfred Krammer. Experience with silicon sensor performance and quality control for a large-area detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 511(1-2):136–144, 9/21 2003.

[21] G. Segneri, L. Borrello, R. Dell'Orso, and et. al. Results with microstrip detectors produced by STMicroelectronics for the CMS tracker. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 476(3):729–733, 1/11 2002.

[22] A. Buffini. Studies on performances of wedge silicon microstrip detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 478(1-2):280–284, 2/1 2002.

[23] S. Assouak, E. Forton, and G. Gragoire. Irradiations of CMS silicon sensors with fast neutrons. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 514(1-3):156–162, 11/21 2003.

[24] M. J. French, L. L. Jones, Q. Morrissey, and et. al. Design and results from the APV25, a deep sub-micron cmos front-end chip for the cms tracker. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 466(2):359–365, 7/1 2001.

[25] L. L. Jones, M. J. French, Q. Morrissey, A. Neviani, M.Raymond, G. Hall, P. Moreira, and G. Cervelli. The APV25 deep submicron readout chip for CMS detectors, http://www.te.rl.ac.uk/med/.

[26] L. Borrello. Status of the integration of the tracker inner barrel of CMS. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 570(2):253–257, 1/11 2007.

[27] Giacomo Sguazzoni. CMS inner tracker detector modules. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 552(1-2):212–215, 2005/10/21.

[28] W. Adam, Th. Speer, B. Mangano, and T. Todorov. Track reconstruction in the CMS tracker. *CMS NOTE 2006/041*, 21 December 2005.

[29] Wolfgang Pils. Analyzing data from a real aleph event. http://teachers.web.cern.ch/teachers/archiv/HST2001/detectors/trackdata/intro.htm.

[30] V. Bartsch, W. deBoer, J. Bol, A. Dierlamm, E. Grigoriev, F. Hauler, S. Heising, and L. Jungermann. An algorithm for calculating the Lorentz angle in silicon detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 497(2-3):389–396, 2/1 2003.

[31] V. Bartsch, W. deBoer, J. Bol, A. Dierlamm, E. Grigoriev, F. Hauler, S. Heising, O. Herz, L. Jungermann, R. Keranen, M. Koppenhofer, F. Roderer, and T. Schneider. Lorentz angle measurements in silicon detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 478(1-2):330–332, 2/1 2002.

[32] V. Bartsch, W. deBoer, J. Bol, A. Dierlamm, E. Grigoriev, F. Hauler, S. Heising, O. Herz, L. Jungermann, R. Keranen, M. Koppenhofer, F. Roderer, and T. Schneider. Lorentz angle measurements in silicon detectors. *IEKP-KA/2001*, LC-DET-2001-028, 2001.

[33] A. Strandlie and R. Fruhwirth. Reconstruction of charged tracks in the presence of large amounts of background and noise. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 566(1):157–165, 10/1 2006.

[34] S. Cucciarelli, M.Konecki, D. Kolinski, and T. Todorov. Track reconstruction, primary vertex finding and seed generation with the pixel detector. *CMS NOTE*, 026, January 31 2006.

[35] Rainer Mankel. A concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 395(2):169–184, 8/11 1997.

[36] R. Fruhwirth and A. Strandlie. Application of adaptive filters to track finding. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(1):162–166, 4/1 2006.

[37] R. Fruhwirth. Application of Kalman filtering to track and vertex fitting. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 262(2-3):444–450, 12/15 1987.

[38] C. Roland. Track reconstruction in heavy ion collisions with the CMS silicon tracker. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 566(1):123–126, 10/1 2006.

[39] T. Speer, W. Adam, R. Fruhwirth, A. Strandlie, T. Todorov, and M. Winkler. Track reconstruction in the CMS tracker. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(1):143–147, 4/1 2006.

[40] Alexandre Khanov. The CMS tracker performance. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 446(1-2):338–345, 5/11 2000.

[41] Physics and data quality monitoring infrastructure. http://cmsevf.web.cern.ch/cmsevf/DQM_doc/DQM_instructions.html.

[42] S. Dutta, V. Chiochia, M. S. Mennea, and G. Zito. Data quality monitoring for the CMS Silicon Tracker. *CMS Conference Report CMS CR 2006/012*, 2006.

[43] A. Meyer. Oneline DQM baseline architecture. *Summary of DQM workshop 27 August 2007*, 2007.

[44] C. Lionidopulus, E. Meschi, I. Segoni, and et al. Physics and data quality monitoring at CMS. *Proceeding of CHEP06, 2006*, 2006.

[45] Carsten Noeding. Track reconstruction and experience with cosmic ray data in CMS. *CMS CR 2008/006*, 2008.

[46] D. Abbaneo, S. Albergo, and et. al. (with I. Goitom). Tracker operation and performance at the magnet test and cosmic challenge. *CMS NOTE 2007/029*, 2007.

[47] R. J. Barlow. *Statistics - A Guide to the Use of Statistical Methods in Physical Sciences*. John Wiley & Sons, 1989.

[48] L. Barbone, N. De Filippis, O. Buchmueller, F. P. Schilling, T. Speer, and P. Vanlaer. Impact of CMS silicon tracker misalignment on track and vertex reconstruction. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 566(1):45–49, 2006/10/1.

[49] I. Belotelov, O. Buchmuller, and et al. Simulation of misalignment scenarios for CMS tracking devices. *CMS Analysis Note CMS AN 2005/035*, 2005.

[50] Stefan Konig. *Deformation Studies on CMS Endcap Modules and Misalignment Studies on the CMS Tracker*. PhD thesis, 30.07.2003.

[51] T. Lampen. General alignment concept of the CMS. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 566(1):100–103, 10/1 2006.

[52] Running a grid job using CRAB. https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookRunningGrid.

[53] Craig Munro, Julia Andreeva, and Akram Khan. Asap distributed analysis. *IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 54, NO. 5, OCTOBER 2007*, 2007.

[54] Christophe Delare and et. al. CMS tracker commissioning and first operation experience. *Proceedings of Science "16th International Workshop on Vertex Detectors", PoS(Vertex 2007)002*, 2007.

[55] J. Rehn and et. al. PhEDEx high-throughput data transfer managment system. *Proceedings of Computing in High Energy and Nuclear Physics, (CHEP06), Mumbai, 2006.*

[56] D. Benedetti and et. al. Tracking and alignment with the silicon strip tracker at the CMS Magnet Test and Cosmic Challenge. *CERN CMS-NOTE 2007 (unpublished)*, 2007.

[57] V. Karimaki and et. al. The HIP algorithm for track based alignemnt and its application to the CMS pixel detector. *CERN CMS NOTE 2006/018*, 2007.

[58] TheGEANT4webpageishttp://geant4.web.cern.ch/geant4/.

[59] A. Aerts, M. Case, M. Liendl, and Asif Jan Muhamad. CMS DETECTOR DESCRIPTION: NEW DEVELOPMENTS. *http://doc.cern.ch//archive/cernrep/2005/2005-002/p498.pdf*.

[60] Riccardo Ranieri. The Tracker geometry validation procedure. *CERN CMS-IN 2007/055*, 2007.

[61] Torbjorn Sjostrand, Stephen Mrenna, and Peter Skands. A brief introduction to pythia 8.1. *CERN-LCGAPP-2007-04*, 2007.

[62] G. Corcella, I.G. Knowles, and et.al. Herwig 6. *[hep-ph/0011363]; hep-ph/0210213*, 2000.

[63] Tanju Gleisberg, Stefan Hoche, and et.al. Sherpa 1. alpha: A proof of concept version. *JHEP 0402:056,2004*, 2003.

[64] http://www.ba.infn.it/~zito/cms/tracker2.gif.

[65] Rene Brun, Nenad Bunci, and et. al. ROOT - An interactive object oriented framework and its application to na49 data analysis. http://root.cern.ch.

[66] Glen Cowan. *Statistical Data Analysis*. Oxford Science Publications, 1998.

[67] G. Zech. Comparing statistical data to Monte Carlo simulation - Parameter fitting and unfolding. *DESY 95-113*, June 1995.

[68] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall CRC, 1993.

[69] Berkan Aslan. The concept of energy in nonparametric statistics-goodness-of-fit problems and deconvolution. Thesis desertation, Universitat Siegen, 2004.

[70] Donald E. Knuth. *The art of computer programming*, volume 2. Addison-Wesley Publishing Company, Inc., 1998.

[71] N. D. Gagunashvili. Comparison of weighted and unweighted histograms. *XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research, PoS(ACAT)060*, 2007.

[72] CERN Program Library Long Writeups. *HBOOK Statistical Analysis and Histogramming.* CERN Geneva, Switzerland.

[73] Raul H. C. Lopes, Ivan D. Reid, and Peter R. Hobson. The two-dimensional Kolmogorov-Smirnov test. *Proceedings of science, XI international workshop on advanced computing and analysis techniques in physics research.*, 2007.

[74] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualisation.* John Wiley & Sons, 1992.

[75] R. Gnanadesikan. *Methods for Statistical Data Analysis of Multivariate Observations.* John Wiley & Sons, 1977.

[76] J. A. Peacock. Two-dimensional goodness-of-fit testing in astronomy. *Monthly Notices Royal Astronomy Society 202 (1983) 615–627*, 1983.

[77] G. Fasano and A. Franceschini. A multidimensional of the Kolmogorov-Smirnov test. *Monthly Notices Royal Astronomy Society 225 (1987) 155–170*, 1987.

[78] Hugh D. Young and Roger A. Freedman. *University Physics with modern physics.* Pearson Addison Wesley, 11 edition, 2004.

[79] Ivan D. Reid, Raul H. C. Lopes, and Peter R. Hobson. Comparison of two-dimensional binned data distributions using the energy test. *CMS NOTE - 2008/023*, 2008.

[80] http://www.r-project.org/.

[81] Diethelm Wuertz and et al. The fmultivar package. http://cran.r-project.org/doc/packages/fMultivar.pdf.

[82] http://www.gnu.org/software/gsl/.

[83] Bootstarp methods and permutation tests. (electronic book) http://www4.stat.ncsu.edu/~muse/Teaching/ST302/moore14.pdf.