DISSERTATION

Global Alignment of the CMS Tracker

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Univ.Doz. Dipl.-Ing. Dr.techn. Rudolf Frühwirth

E107
Institut für Statistik und Wahrscheinlichkeitstheorie

eingereicht an der Technischen Universität Wien
Fakultät für Physik

von

Dipl.-Ing. Edmund Erich Widl

Matrikelnummer: 9725850
Badgasse 1-7/7/12, 1090 Wien

Wien, am 29. Oktober 2008

Sehen Sie, die Beobachtung ist ja im allgemeinen ein sehr komplizierter Prozess.

Werner Heisenberg
aus "Quantentheorie und Philosophie"

Kurzfassung

Die vorliegende Arbeit wurde am Institut für Hochenergiephysik (HEPHY) der Österreichischen Akademie der Wissenschaften im Rahmen der CMS Kollaboration ausgeführt. Eine Hauptkomponente des CMS Experiments, eines von zwei Vielzweck-Experimenten am Large Hadron Collider (LHC) des CERN (Genf, Schweiz), ist der so genannte Tracker (von engl. Track, Spur). Dieses Gerät, entwickelt zur Vermessung der Flugbahn von geladenen Teilchen, besteht aus etwa 16.000 planaren ortsauflösenden Siliziumsensoren und ist somit das bei weitem größte Exemplar seiner Art. Systematische Messfehler, hervorgerufen durch unvermeidliche Ungenauigkeiten bei der Konstruktion, verringern die Messgenauigkeit allerdings drastisch. Die daher notwendigen geometrischen Korrekturen des experimentellen Versuchsaufbaus – das so genannte *Alignment* – sollten mit einer Genauigkeit unterhalb der Auflösung der Einzelsensoren bekannt sein. Zu diesem Zweck müssen mit Hilfe spezieller Algorithmen bereits rekonstruierte Teilchenspuren analysiert werden.

Der *Kalman Alignment Algorithmus* (KAA) ist eine neuartige Methode, diese geometrischen Korrekturen auch für derart große Systeme wie den CMS Tracker zu berechnen. Die vorliegende Arbeit stellt einen Querschnitt über die gesamte bisherige Entwicklung dar, angefangen vom zugrunde liegenden Konzept, über die Implementierung bis hin zur konkreten Anwendung in Simulationsstudien sowie der Verarbeitung echter experimenteller Daten. Zudem wird eine neuartige Methode zur Ausnutzung kinematischer Zwangsbedingungen von Zwei-Körper-Zerfällen vorgestellt, welche generell die Präzision von Alignment-Algorithmen verbessern kann.

Das erste Kapitel gibt einen kurzen Abriss über das CMS Experiment. In Kapitel 2 werden die einzelnen Strategien der CMS Kollaboration zur Erkennung und Beseitigung der oben beschriebenen systematischen Messfehler dargestellt. Das dritte Kapitel widmet sich der zu diesem Zweck erstellten Software, deren Mit- und Weiterentwicklung ein wesentlicher Bestandteil dieser Arbeit war. Kapitel 4 gibt einen vollständigen Überblick

zur prinzipiellen Funktionsweise des KAA und zeigt darüber hinaus auch grundlegende Studien über die wichtigen Eigenschaften. Das fünfte Kapitel behandelt die oben erwähnten kinematischen Zwangsbedingungen. Kapitel 6 präsentiert eine ausführliche Simulationsstudie, welche die tatsächliche Anwendbarkeit des KAA für ein derart großes System wie den CMS Tracker demonstriert. Das siebenten Kapitel widmet sich der Auswertung der ersten experimentellen Daten des CMS Trackers, der anschließend an die Montage teilweise in Betrieb genommen wurde. Kapitel 8 gibt schließlich noch einen kurzen Gesamtüberblick und Perspektiven für künftige Entwicklungen.

Abstract

The work at hand has been carried out at the Institute of High Energy Physics (HEPHY) of the Austrian Academy of Sciences within the framework of the CMS Collaboration. One of the main components of the CMS experiment, one of two multi-purpose experiments at the Large Hadron Collider (LHC) at CERN (Geneva, Switzerland), is the so called Tracker. This device, designed to measure the flight paths of charged particles (hence the name), is composed of approximately 16,000 planar silicon detector modules, which makes it the biggest of its kind. However, systematical measurement errors, caused by unavoidable inaccuracies in the construction phase, reduce the precision of the measurements drastically. The consequently required geometrical corrections of the experimental setup — the so called alignment — should be known with an accuracy better than the resolution of the detector modules. To this goal, special algorithms are utilized to analyze recorded particle tracks.

The *Kalman Alignment Algorithm* (KAA) is a novel approach to extract a set of alignment constants even for a system as big as the CMS Tracker. The work at hand gives an overview on the entire development, starting with the underlying concept, to the implementation and the concrete application in simulation studies and the processing of real experimental data. In addition, a novel method for utilizing kinematical constraints of two-body decays is presented, which can be used to improve the precision of alignment algorithms in general.

The first chapter gives a short overview of the CMS experiment. In Chapter 2 the strategies of the CMS collaboration to determine and eliminate misalignment is outlined. The third chapter deals with the software that has been designed for this purpose, the development and improvement of which was an essential part of this work. Chapter 4 gives a full overview of the principle functionality of the KAA and shows some basic studies about its most important properties. The fifth chapter covers the aforementioned kinematical constraints. Chapter 6 presents a detailed simulation study which demon-

strates the practicality of the KAA for a system as big as the CMS Tracker. The seventh chapter deals with the analysis of the first experimental data recorded with the CMS Tracker, which was partially operated after its construction. Finally, in Chapter 8 a short summary is given along with an outlook for future developments.

Acknowledgements

Many, many, many thanks to my supervisor and mentor Rudolf Frühwirth, for giving me the great opportunity to work on the subject presented in this thesis. Not only is he the father of the basic ideas of this work, but he helped me with great personal commitment from the first day on to deepen my knowledge in the needed scientific subjects and gain ground in the CMS collaboration. Also a great thank-you to Wolfgang Adam, who was never tired of answering my questions on the analysis software and helped me out more than once, when I still had to learn the craft of coding or simply when CERN bureaucracy hit me.

Further thanks go to all present and former members of the CMS Alignment Group that crossed my way during the last four years. You guys were great, even when my usual "I-am-staying-at-CERN-and-I-had-to-get-up-too-early" bad mood showed up. Here I have to mention with great emphasis, that I owe especially to Roberto Covarelli, Nhan Tran, Roberto Castello and Johannes Hauk for helping me with some of the more tricky and time-consuming plots for the analysis of the Tracker Integration Facility data.

Also all my colleagues at the HEPHY have earned themselves more than a big thank-you from my side, especially my poor room mate Thomas Bergauer, who has to bear my denial to speak when I'm concentrated. Also many thanks to Stephan Hänsel and Marko Dragicevic who travel the same road with me for more than ten years now – for me you make the HEPHY an especially amusing place to work at! Not to forget, a big thank-you to Wolfgang Waltenberger, who also helped me out more than once with this and that.

Last but not least, a thank-you so big it actually shouldn't be able to fit on this page – but it does since I'm a lazy person – to my family and friends. Without your support – and the acceptance of my before mentioned laziness and other quirks – I would not stand where I am today. And I actually like where I stand today ☺

Contents

1 Introduction **1**

1.1 The Large Hadron Collider 1

1.2 The CMS Experiment 3

1.3 The CMS Tracker 7

2 CMS Tracker Alignment Strategy **8**

2.1 The Structure of the CMS Tracker 8

2.2 Hardware Alignment 14

2.3 Track-Based Alignment 18

3 Alignment Software Framework **24**

3.1 Alignable Concept 25

3.2 TrackerAlignment 26

3.3 CommonAlignmentParametrization 28

3.4 CommonAlignmentAlgorithm 30

3.5 ReferenceTrajectories 33

3.6 CommonAlignmentProducer 37

3.7 Dedicated Data Streams 37

3.8 Validation Tools 39

4 The Kalman Alignment Algorithm 40

4.1 Sequential Update of Alignment Parameters 41

4.2 Implementation and Computational Complexity 43

4.3 Selective Updates of Alignable Objects 46

4.4 Results 51

5 Two-Body Decay Constraints 58

5.1 Prerequisites 58

5.2 Parametrization 59

5.3 Representation 61

5.4 Estimation 62

5.5 Two-Body Decay Constraints in Tracking 66

5.6 Two-Body Decay Constraints for Alignment 68

6 Full-scale CMS Tracker Alignment 72

6.1 Available Datasets 72

6.2 Available Computing and Software Resources 74

6.3 Full-Tracker Alignment Scenarios 75

6.4 Additional Selection Criteria for Alignable Objects 77

6.5 Results 78

6.6 Summary 80

7 Results from the CMS Tracker Integration Facility 85

7.1 Experimental Setup 85

7.2 Track Selection 87

7.3 Common Alignable Selection 89

7.4 Results of the Kalman Alignment Algorithm 91

7.5 Comparison to other Algorithms 103

7.6 Summary 107

8 Conclusion and Outlook 108

Bibliography 110

Chapter 1

Introduction

1.1 The Large Hadron Collider

The Large Hadron Collider (LHC) [1], presently the largest and most powerful particle accelerator of the world, is the latest addition to CERN's accelerator complex (Figure 1.1). Within its subterranean 27 km ring, two beams of either protons or lead ions are traveling in opposite directions at almost the speed of light in two separate tubes kept at ultrahigh vacuum. The beams are guided around the accelerator ring by a strong magnetic field, achieved using superconducting electromagnets, cooled down below 3 K. These include 1232 dipole magnets of 15 m length, used to bend the beams, and 392 quadrupole magnets, each 5–7 m long, to focus the beams. At four distinct points, the beams are brought to collision, where enormous detectors investigate the resulting particle interactions.

The LHC experiments cover a large range of physics topics. The most prominent ones are the following:

- What is the origin of mass? Within the *Standard Model* all particles acquire their masses by interacting with the *Higgs-field*. Thus a physically observable *Higgs-boson* should exist, but the latter has not yet been observed.

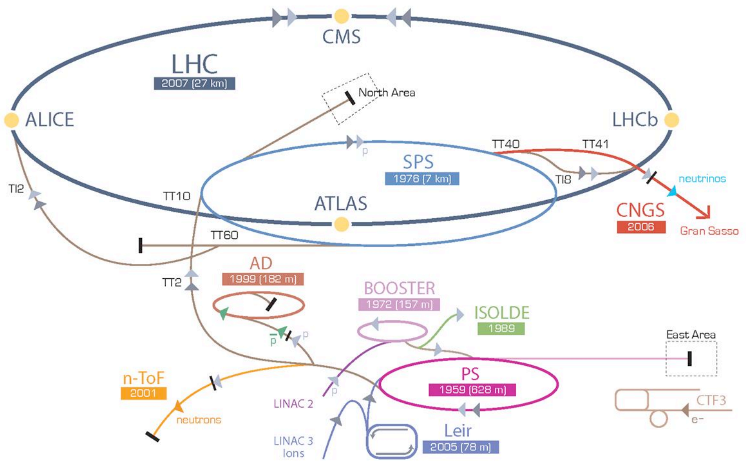


Figure 1.1: Overview of CERN's accelerator complex.

- Is a fundamental symmetry between bosonic and fermionic elementary particles, i.e. elementary particles with either integral or half-integral spin, the *supersymmetry*, realized in nature?
- Is there experimental evidence for a *Grand Unified Theory* (GUT) — a proposed unification of the electromagnetic, the weak, and the strong interaction?
- What is the origin of *dark matter*? Could the so-called neutralinos, particles proposed by the theory of supersymmetry, be a candidate for dark matter?
- Is there a new form of matter, the so-called quark-gluon plasma, as it might have existed in the early universe?
- Why is there an asymmetry in the quantity of matter as compared to antimatter?
- Why are there three families of quarks and leptons?
- Do quarks and leptons have a substructure?
- Is the world we live in really four-dimensional? Will evidence for further dimensions be found?

Hadron colliders are considered to be *exploration machines*. This is due to the hadronic substructure, where the energy is statistically spread between the constituting quarks and gluons, such that they can interact within a broad energy range. The disadvantage of course is the fact that the initial state of the colliding particles — be it their overall momentum, energy or flavor — is not a priori determined, which makes the reconstruction of the collision and its physical interpretation harder. Nevertheless, especially the first four of the questions above should be answered within the first few years of data taking, thanks to their expected significant experimental signatures and the high available energies at the LHC.

1.2 The CMS Experiment

On of the four experiments at the LHC will be the Compact Muon Solenoid [2], usually referred to as CMS. It is, together with the ATLAS experiment [3], one of the two so-called multi-purpose experiments. The whole design, as sketched in Figures 1.2 and 1.3, is centered around a large superconducting 4T solenoid, embedded within a massive iron return yoke. Four large detector systems are deployed either within or enclosing this solenoid:

- Closest to the interaction point the *Tracker* is located. Its aim is to reconstruct the trajectories of the charged traversing particles, to retrieve information about their momentum, flight direction and origin.
- The *Electromagnetic Calorimeter* (ECAL) encloses the tracking detector. Electrons and photons are stopped inside this detector to measure their energy.
- The *Hadronic Calorimeter* (HCAL) is placed between the ECAL and the solenoid. It stops particles that are subject the strong interaction, i.e. protons, neutrons, pions, etc., to measure their energy.
- The *Muon System* is the only detector system that is placed outside the solenoid, with its components inserted into the free space between the layers of the iron return yoke. It is a comparatively coarse tracking

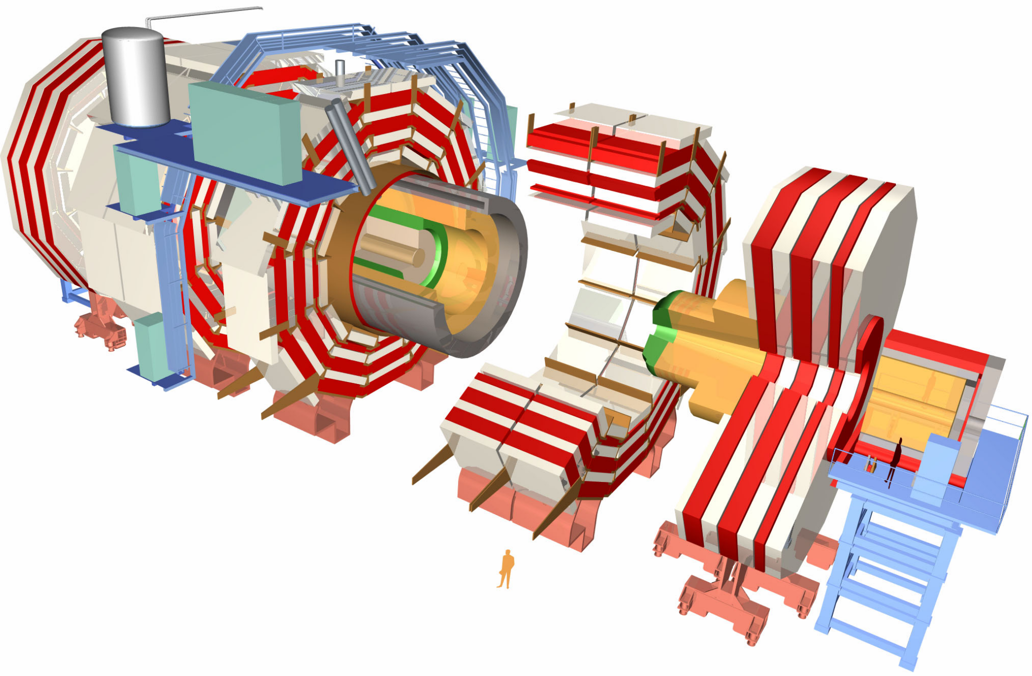


Figure 1.2: Schematic view of the CMS detector.

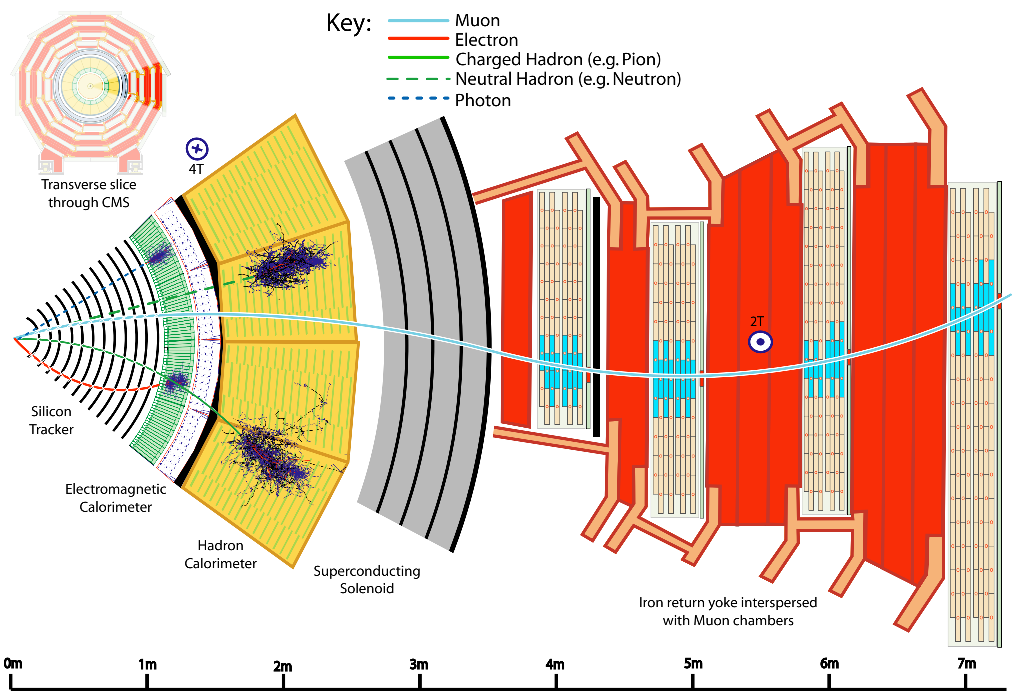


Figure 1.3: A slice through the barrel region of the CMS detector.

device for muons, which are supposed to be the only particles to pass through the inner detectors.

All four detector systems are split into a *barrel region* and two *endcap regions*, resulting in a onion-like structure to guarantee a complete as possible geometrical coverage. In the barrel region the detector components are arranged more or less in cylindrical layers, concentric around the LHC beam line. In the endcap regions the detector components are arranged in disks perpendicular to the LHC beam line, which cover the open ends of the detector systems in the barrel region.

Even though it is possible to address a large range of physics topics at the CMS Experiment, the investigation of the Higgs-mechanism – or any (not so) similar mechanism that is actually realized in nature – is its main purpose. Figure 1.4 shows the theoretically predicted production and decay rates for the Standard Model Higgs-boson in dependence on its mass. The smallest mass shown is approximately the currently known lower limit, determined from previous or still running experiments. The leading process for production over the whole mass spectrum is gluon fusion with no by-products. All other possible channels are, in a large part of the spectrum, orders of magnitude smaller. The resulting most prominent decay channels are therefore the following:

- For $M_H < 140$ GeV (small mass scenario): The decay channel with the highest branching ratio in this scenario is $H \to b\bar{b}$. Unfortunately, the large QCD background makes it unusable for physics analysis. Thus, one of the promising decay channels is $H \to \gamma\gamma$. The expected narrow width of the Higgs-boson should allow to discriminate the signal from an immense background, even though the production rate itself is very low. One possible alternative in this low mass region is to look at Higgs bosons with other particles as by-products, e.g. Higgs bosons with an associated $t\bar{t}$ pair. This would allow to trigger on e.g. the leptonic decay of one of the top quarks.

- For $130 < M_H < 500$ GeV (medium mass range): The most prominent channels are $H \to VV^* \to l^+l^-\nu\nu$, and $H \to VV^* \to l^+l^-l^+l^-$ ($l =$

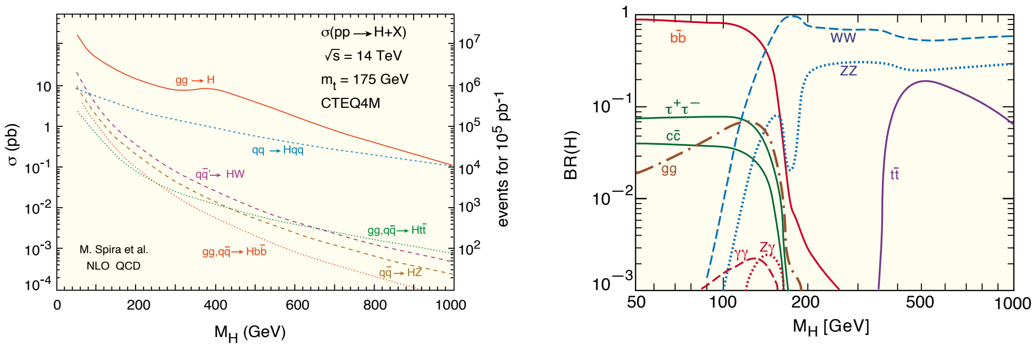


Figure 1.4: Expected production and decay rates for the Standard Model Higgs-boson.

$e, \mu; V = W, Z$). These channels are fully leptonic. Especially the four-lepton channel, sometimes referred to as the gold-plated channel, would yield a particularly clean signature.

- For $M_H > 500$ GeV (heavy mass range): In the case of such a massive Higgs-boson, the channel $H \rightarrow VV^* \rightarrow l^+l^- jet\, jet$ is one of the more promising channels.

The detector components are therefore optimized to cover all these decay channels. The layout provides good muon identification and momentum resolution over a wide range of momenta and angles, good dimuon mass resolution (≈1% at 100 GeV), and the ability to determine unambiguously the charge of muons with $p < 1$ TeV. Also a good momentum resolution and reconstruction efficiency for charged particles in the Tracker is achieved, enabling efficient triggering and offline tagging of τ's and b-jets. The design also allows for good electromagnetic energy resolution, good diphoton and dielectron mass resolution (≈1% at 100 GeV) with a wide geometric coverage, π^0 rejection, and efficient photon and lepton isolation at high luminosities. Finally, good missing-transverse-energy and dijet-mass resolution is achieved, due to the Hadronic Calorimeter's large hermetic geometric coverage and fine lateral segmentation.

distance from beamline	fluence [$n_{eq}\,cm^{-2}$]	technology
< 20 cm	10^{15}	n^+-type pixels on 270 μm thick n-type bulk, low resistivity ($\sim 2K\Omega\,cm$), oxygenated
20 - 50 cm	10^{14}	p-type strips on 320 μm thick n-type bulk, low resistivity ($\sim 2K\Omega\,cm$), pitch $\sim 80\mu m$
> 50 cm	10^{13}	p-type strips on 500 μm thick n-type bulk, high resistivity ($\sim 5K\Omega\,cm$), pitch $\sim 200\mu m$

Table 1.1: Technologies used in the CMS Tracker to match the specifications for radiation hardness and detector occupancy.

1.3 The CMS Tracker

At the design luminosity, a mean of about 20 inelastic collisions will be superimposed on the event of interest. This implies that around 1000 charged particles will emerge from the interaction region every 25 ns. The products of an interaction under study may be confused with those from other interactions in the same bunch crossing. To cope with this enormous pile-up effects as well as the associated high radiation dose, a multi-layer full-silicon tracking detector [4, 5] has been designed, using high-granularity sensors (see table 1.1) connected to custom-made read-out electronics. This design allows operation with a high time resolution, resulting in a reasonably low detector occupancy.

The innermost part is made of 1440 pixel detectors. Silicon pixel detectors are characterized by their two-dimensional spatial resolution and high granularity and are thus very suitable for being used as vertex detectors. A total of 15,148 silicon microstrip detectors is mounted around the pixel detector in order to be able to track the particles over a large volume with high accuracy. A detailed description of the geometry is given in Section 2. With a sensitive area of about $200\,m^2$ the CMS Tracker is the largest full-silicon tracking detector ever built.

Chapter 2

CMS Tracker Alignment Strategy

Alignment is the general term used in experimental high energy physics to refer to the process of obtaining and applying corrections to the nominal setup of a given experiment. These corrections are typically related to geometrical displacements of devices with a spatial resolution, in contrast to calibrations, where the corrections are usually extracted from pedestal or reference measurements to compensate for offsets in scalar measurements. Misalignment compromises tracking and vertex finding and thus directly affects physics measurements like momentum and invariant mass resolutions or the efficiency of b-tagging algorithms.

In the following chapter a detailed overview of the alignment strategy for the CMS Tracker [6] is given, including a review of its mechanical structure and the hardware and software alignment schemes.

2.1 The Structure of the CMS Tracker

The CMS Tracker is a highly complex structure. Its design aims to combine a layout capable of providing precise and efficient trajectory measurements on the one side and an adequate cooling and power supply on the other. The large quantity of read-out channels and the fast response time of the associated electronics, needed for the desired tracking performance, imply a high power density and a considerable amount of utility and service equipment inside the Tracker volume. Thus, a trade-off between the number of sensitive modules and the material budget for the passive elements had to be found.

The resulting mechanical structure of the Tracker, which provides the natural basis for all alignment related tasks, consists of a central barrel region and two endcap regions. Inside the barrel region, which itself is divided into the Tracker Pixel Barrel (TPB), the Tracker Inner Barrel (TIB) and the Tracker Outer Barrel (TOB), the modules form concentric cylindrical layers, centered around the nominal beam line. In the endcap regions, composed of the Pixel Endcaps (TPE), the Tracker Inner Disks (TID) and Tracker Endcaps (TEC), the modules are arranged on parallel disks. The subcomponents are usually labeled with either $+$ or $-$, indicating their global Z-position.

All coordinates refer to the global CMS frame, unless indicated otherwise. The origin of the CMS coordinate system is the nominal collision point, the coordinate system is defined as follows:

- The X-axis is horizontal, pointing south to the LHC center.
- The Y-axis is vertical, pointing upwards.
- The Z-axis is horizontal, pointing west along the beam line.
- The azimuthal angle ϕ is measured in the X-Y-plane, where the direction of $\phi = 0$ coincides with the $+X$-axis and $\phi = \pi/2$ with the $+Y$-axis.
- The polar angle θ is measured w.r.t. the Z-axis, where the direction of $\theta = 0$ coincides with the $+Z$-axis and $\theta = \pi$ with the $-Z$-axis.
- The sign of $\eta = -\ln\tan(\theta/2)$ is equal to the sign of Z.

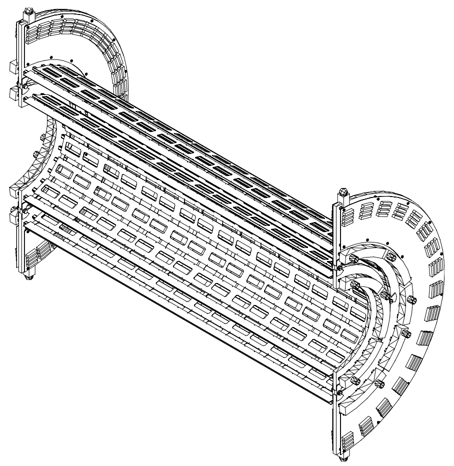


Figure 2.1: Schematic view of the pixel barrel support structure.

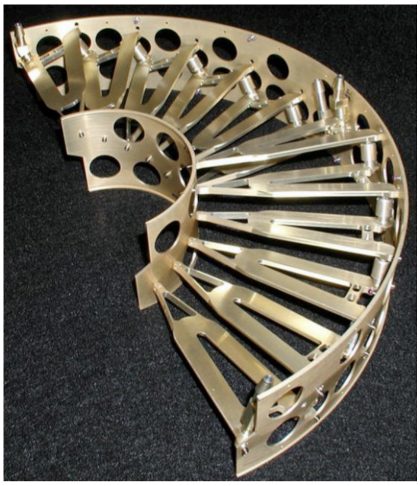


Figure 2.2: A forward pixel half-disk with the turbine-like geometry.

2.1.1 Pixel Tracker Geometry

The pixel detector is the innermost tracking detector of the CMS detector, mounted directly to the beam pipe.

Pixel Barrel Structure

In the barrel region, the pixel modules are arranged in three concentric cylindrical layers with radii of 4 cm, 7 cm and 11 cm. The actual mounting structure (see Figure 2.1) has a length of 570 mm ranging from -285 mm to $+285$ mm with respect to the interaction point. Its backbone are aluminum cooling tubes with a wall thickness of 0.3 mm. Carbon fiber blades with a thickness of 0.24 mm are glued at either the top or the bottom of two adjacent cooling tubes, such that their surfaces point alternating to the beam or away from it. The cross sections of the tubes have a trapezoidal shape to allow for the easy and stable mounting of the attached blades. These rigid composite structures of carbon fiber blades and cooling pipes are referred to as ladders. Support frames on both ends, which connect the single segments build a complete pixel barrel half shell. These flanges consist of thin fiber glass frames that are filled with foam and covered by carbon fiber blades.

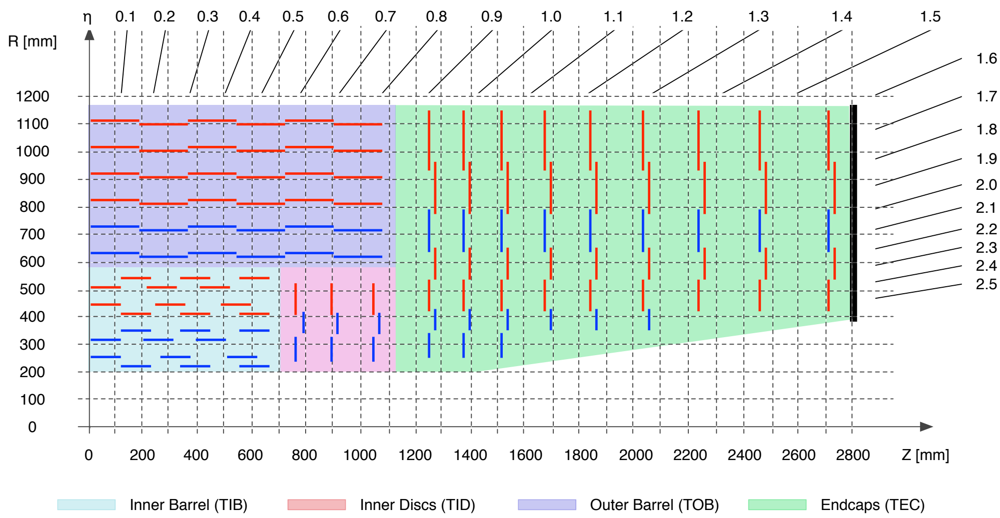


Figure 2.3: Schematic R-Z view of the silicon micro-strip Tracker. Single-sided modules are shown in red, double-sided in blue.

Pixel Endcaps Structure

In the endcap region, the pixel modules are arranged in two parallel disks at each side, positioned 35 cm and 47 cm away from the interaction point. The pixel modules are attached to panels, made of 0.5 mm thick beryllium, providing a strong, rigid and relatively low-mass support structure. A single cooling channel with panels mounted on both sides forms a subassembly called a blade. A total of 24 panels, forming 12 individual blades, comprise a half-disk (see Figure 2.2). Two half-disks make up a half-cylinder.

2.1.2 Silicon Micro-Strip Tracker Geometry

The silicon micro-strip tracker is composed of 15,148 individual detector modules [7, 8]. In the TIB and TID as well as in rings 1 to 4 of the TEC they are equipped with only one sensor, whereas the remaining modules in the TOB and rings 5 to 7 in the TEC have two sensors. Their mechanical stability is guaranteed by the carbon frames they are mounted on, which

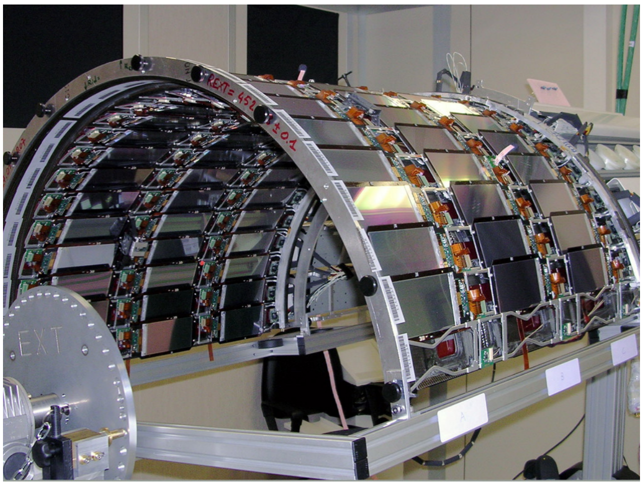


Figure 2.4: An inner barrel half-shell.

also carry the read-out electronics. Different types of aluminum inserts and precision bushings in the module frames are used to position and attach the modules to the larger support structures with high precision. Modules in the TIB, the TID and the TEC are mounted using four points, two being high precision bushings that allow for a mounting precision of better than 20 μm. For TOB modules four screws and two springs are used for the precision positioning. See Figure 2.3 for an conceptual overview of the silicon micro-strip Tracker.

Tracker Inner Barrel Structure

The four concentric cylindrical layers in the Tracker Inner Barrel have radii of 26 cm, 34 cm, 42 cm, and 50 cm, respectively, and extend from -70 cm to $+70$ cm along the Z-axis. The two innermost layers host double-sided modules, while the outer two layers host single-sided modules. Each cylinder is subdivided into four half-shells, i.e. sub-assemblies dividing the cylinders in $\pm Z$ and $\pm Y$, that are the actual rigid mounting structures. They are made of carbon fiber, onto which the modules are directly attached. Two service cylinders are coupled to the ends of TIB$\pm$, which end in a service distribution disk called the *margherita*, used to route the cooling pipes, power cables and read-out lines outside.

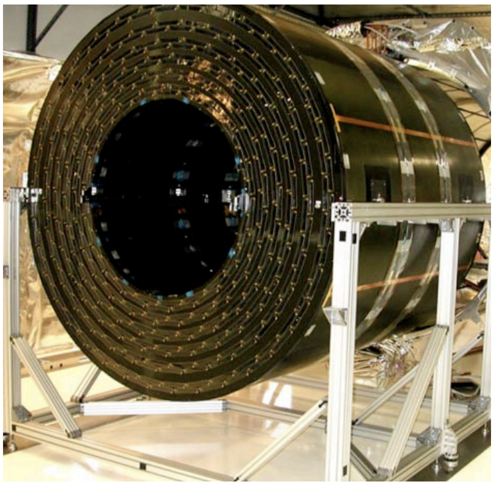


Figure 2.5: The outer barrel's un-equipped wheel.

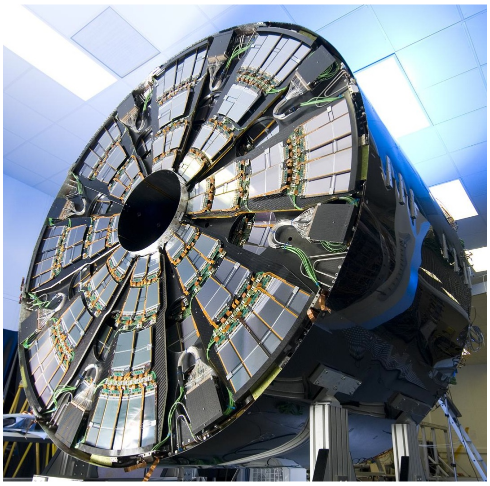


Figure 2.6: A fully assembled end-cap.

Tracker Inner Disks Structure

The service cylinders of the Tracker Inner Barrel also support the Tracker Inner Disks, which are placed inside them. The TIDs are composed of three parallel disks placed along the Z-axis between ± 80 cm and ± 90 cm. The disks are identical and consists of three rings, spanning radially from 20 cm to 50 cm. The two innermost rings host double-sided modules while the outer one hosts single-sided modules. The disks and service cylinders are made of carbon fiber, onto which the modules are directly attached.

Tracker Outer Barrel Structure

The Tracker Outer Barrel consists of a single mechanical structure called *wheel* (see Figure 2.5), into which a total of 688 self-contained sub-assemblies, called *rods*, are inserted. The wheel is composed of four identical disks joined by three outer and three inner cylinders. The disks, cylinders and rods are made of carbon fiber, whereas the cylinders have an additional honeycomb-structured core. Each disk contains 344 openings into which the rods are inserted, such that each rod is supported by two disks, and two rods always cover the whole length of the TOB along the Z-axis. The wheel has a length of 218 cm, and inner and outer radii of 56 cm and 116 cm, respectively. The modules, which are directly attached to the rods in groups of 6 or 12 pieces, form six layers with average radii of 61 cm, 69 cm, 78 cm, 87 cm, 97 cm and 108 cm, respectively.

Tracker Endcaps Structure

The Tracker Endcaps extend radially from 22 cm to 114 cm and from ±124 cm to ±280 cm along the Z-direction. Each endcap consists of nine disks that carry wedge-shaped sub-assemblies called *petals* (see Figure 2.6). The disks and petals are rigid carbon fiber honeycomb structures. The individual detector modules are directly attached to the petals. Disks 1 to 3 carry seven rings of modules, ring 1 is missing on disks 4 to 6, rings 1 and 2 are missing on disks 7 and 8, and disk 9 carries rings 4 to 7 only. Rings 1, 2 and 5 are built up of double sided modules. Eight U-shaped service profiles, used as service channels, join the disks along their outer periphery, while at its inner diameter each disk is attached at four points to an inner support tube. Back-plates at both ends of each endcap provide additional mechanical support.

2.2 Hardware Alignment

The CMS experiment uses several independent strategies for the alignment of the Tracker. For testing the long-term stability or the alignment of subdetectors among each other, so-called *hardware alignment* is utilized, where special reference markers are measured directly via optical systems or photogrammetry.

2.2.1 Tracker Survey

The first step for a good alignment is the monitoring of the assembly precision. To this end a large number of different surveying measurements, using photogrammetry, theodolite triangulation and 3D coordinate measurement systems, have been performed to verify the desired mechanical accuracy for most components. In cases where a sufficient amount of data has been collected, this information can also be included into the geometry description. Otherwise the measurements can be used to obtain an estimate of the position uncertainties.

The survey measurements were carried out on various hierarchical levels, starting with the positions of the sensors on the module frames, showing an overall accuracy below 10 μm in the sensitive coordinate. On top of that the mounting precision of the mechanical support structure was examined. For instance, the TOB wheel mechanics has been thoroughly measured before starting rod integration, and the relative positioning of the precision elements has been found to be typically within 100 μm of nominal values, with maximum deviations observed around 200 μm. Another example is the survey of the TEC, showing a mechanical accuracy of the endcap discs of 100 μm in the Rϕ-plane.

Some of the photogrammetry targets attached to the components remained visible after the insertion into the tracker support tube. This allowed to perform additional measurements of their positioning in the Tracker's reference frame or to monitor possible deformations of the loaded structure.

2.2.2 Laser Alignment System

The Laser Alignment System [9], usually referred to as LAS, was mainly designed to keep the stability of the Tracker's mechanical structure under permanent surveillance. To do so, it aims to provide alignment information on a continuous basis, observing the position of substructures at the level of 100 μm, which is mandatory for pattern recognition and for the High Level Trigger (HLT). In addition, possible Tracker structure movements can be measured at the level of 10 μm, providing additional input for the track-based alignment.

The LAS uses infrared laser beams to determine the positions of a selected

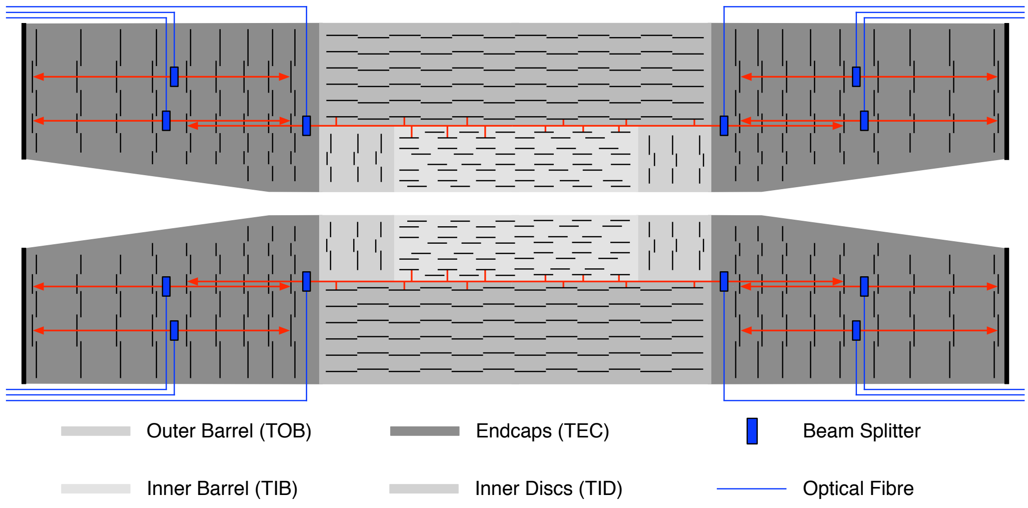


Figure 2.7: Outline of the LAS design.

set of detector modules located in the TIB, the TOB and the TEC. Since the number of included modules is small in comparison to the absolute number of modules within the Tracker, the retrieved information is used to monitor the positions of the large composite structures only, instead of the individual modules. The modules are assembled with special silicon sensors, having a 10 mm hole in the backside metallization and an anti-reflective coating. The LAS design is illustrated in Figure 2.7. Each Tracker endcap is equipped with 16 beams in total, distributed uniformly in global ϕ and crossing all 9 disks in rings 4 and 6, used for the internal alignment of the disks. The other 8 beams are foreseen to align the TIB, the TOB and both Tracker endcaps with respect to each other.

As the laser beams are consecutively attenuated by every traversed layer, the lasers fire a sequence of pulses with increasing intensities, each optimized for a given silicon layer. In addition, these measurements are repeatedly triggered for every intensity and the signals are averaged. A few hundred triggers are needed in total to get a full picture of the alignment of the Tracker structure. Since the trigger rate for the alignment system is around 100 Hz, this takes only a few seconds, such that these snapshots will be taken at regular intervals. The LAS is foreseen to operate both in dedicated runs and during physics data taking.

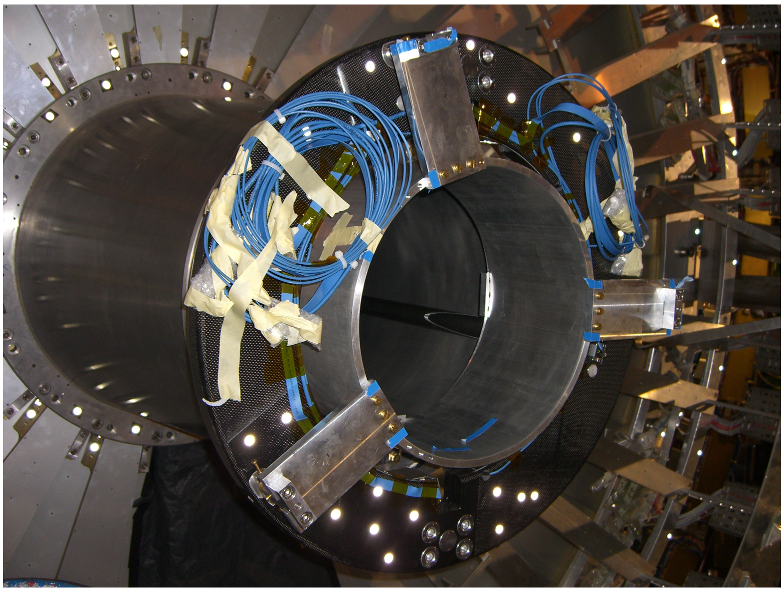


Figure 2.8: A mounted alignment ring for the Link Alignment System.

In the analysis, all events belonging to one snapshot are accumulated. Then for each silicon module the events with the optimal signal are identified, which are then averaged. Next, a Gaussian is fitted to the laser profiles, from which the actual positions and uncertainties for all considered planes are calculated. Accumulating these positions delivers the input for the alignment procedure, which finally gives the geometrical corrections for the different Tracker substructures with respect to each other.

2.2.3 Link Alignment System

The purpose of the Link Alignment System is to measure the relative position of the Muon System and the Tracker in a common CMS coordinate system. A distributed network of optoelectronic position sensors is connected by laser lines to provide high-precision measurements over long distances. A total of 12 laser paths (6 on each Z-side) use rigid carbon fiber annular structures placed at both ends of the Tracker (Alignment Rings, AR, see Figure 2.8) as mechanical references. Three pillars, acting as support holders, connect the last instrumented disc of each TEC with the corresponding AR. The position and orientation of the ARs with respect to TEC discs 9 and 10 were measured with a coordinate-measurement machine using the external survey fiducials, prior to TEC assembly and instrumentation. The link measurement network is complemented by electrolytic tiltmeters, proximity sensors in contact with aluminium tubes of calibrated length, magnetic probes, and temperature sensors.

2.3 Track-Based Alignment

The hardware alignment techniques used for the CMS Tracker are not able to retrieve information on the level of modules, which is naturally the preferred granularity for alignment corrections. To this end, the concept of *track-based alignment* has to be utilized, where the information from recorded particle tracks is directly used to obtain the alignment parameters. There are various possibilities for the treatment of alignment corrections, ranging from simple translations and rotations, equivalent to those of a rigid body, to more complex deformations, like sags or twists.

2.3.1 Overview

The fundamental basis for all track-based alignment algorithms is an adapted track model $\boldsymbol{f}$, where the measurements $\boldsymbol{m}$ depend not only on the true track-parameters $\boldsymbol{q}_t$ but also on a set of alignment parameters $\boldsymbol{p}_t$ that describe the effects of sufficiently small deviations from the ideal geometry:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{q}_t, \boldsymbol{p}_t) + \boldsymbol{\varepsilon}, \quad \mathsf{cov}(\boldsymbol{\varepsilon}) = \boldsymbol{V} \tag{2.1}$$

The stochastic term $\boldsymbol{\varepsilon}$, which describes the intrinsic resolution of the tracking devices and the effects of multiple scattering, is dealt with via its covariance matrix $\boldsymbol{V}$. Since typically high momentum particles are used, any energy-loss effects can be assumed to be deterministic and therefore directly taken care of in the track model $\boldsymbol{f}$ itself.

With an initial guess $\check{\boldsymbol{q}}$ for the track parameters and $\check{\boldsymbol{p}}$ for the alignment parameters, this model allows to define *residuals*, that are functions of the unknowns $\boldsymbol{q}$ and $\boldsymbol{p}$:

$$\boldsymbol{\Delta}(\boldsymbol{q},\boldsymbol{p}) = \boldsymbol{m} - \boldsymbol{f}(\boldsymbol{q},\boldsymbol{p}) \approx \boldsymbol{m} - \check{\boldsymbol{f}} - \boldsymbol{D}_q\boldsymbol{\Delta q} - \boldsymbol{D}_p\boldsymbol{\Delta p} \tag{2.2}$$

with

$$\check{\boldsymbol{f}} = \boldsymbol{f}(\check{\boldsymbol{q}},\check{\boldsymbol{p}}), \quad \boldsymbol{\Delta q} = \boldsymbol{q} - \check{\boldsymbol{q}}, \quad \boldsymbol{\Delta p} = \boldsymbol{p} - \check{\boldsymbol{p}}$$

$$\boldsymbol{D}_q = \partial\boldsymbol{f}/\partial\boldsymbol{q}\Big|_{\check{\boldsymbol{p}},\check{\boldsymbol{q}}}, \quad \boldsymbol{D}_p = \partial\boldsymbol{f}/\partial\boldsymbol{p}\Big|_{\check{\boldsymbol{p}},\check{\boldsymbol{q}}}.$$

The goal of every track-based alignment algorithm is to determine $\boldsymbol{p}$ from the residuals $\boldsymbol{\Delta}$, or more often their normalized square $\chi^2 = \boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{\Delta}$, from a sufficiently large set of recorded tracks. The methods used are rather diverse, but nevertheless they can be grouped into two different categories: biased and unbiased algorithms.

Biased algorithms ignore at first glance the fact that the initial guess for the track parameters $\check{\boldsymbol{q}}$ is in general biased from the factual misalignment. In other words, by setting $\boldsymbol{q} = \check{\boldsymbol{q}}$ for every track, the residuals become all but a function of $\boldsymbol{p}$ alone, i.e. $\boldsymbol{\Delta}(\boldsymbol{q},\boldsymbol{p}) \to \boldsymbol{\Delta}(\boldsymbol{p})$. In general, the influence of the biased track information has to be compensated by iterating several times over the track sample, where at each iteration step the previously determined parameters are applied to the track reconstruction.

Unbiased algorithms, on the other hand, try to minimize the residuals, or the normalized residuals respectively, by taking also the track parameters into account as degrees of freedom. The problem related to such an approach is the huge resulting number of parameters. In the presence of N alignment parameters and a sample of M tracks with n track parameters each, a total of $N + n \times M$ parameters have to be dealt with. While the value of N depends a lot on the experimental setup and $n = 5$ with a helical track-model, the number of tracks M has always to be of considerable size to acquire reasonable statistics. However, unbiased algorithms usually do not require iterations, with the possible exception of topics such as non-linearities or outlier-rejection.

Besides the differences between various algorithms it should be noted that the final result of any track-based alignment is always limited by the number of tracks used. Basic quality cuts, such as the selection of high momentum tracks to minimize the influence of multiple scattering, or cuts on the minimum number of hits, have apparently a strong influence on the convergence. More subtle is the effect of an unbalanced mixture of tracks or even complete absence of certain types of tracks, such as tracks from collisions and cosmic events or tracks taken with and without a magnetic field. This is due to the fact that any kind of tracks has several degrees of freedom it cannot constrain, usually referred to as *weak modes*, *weakly defined modes* or *χ^2-invariant modes*. For example, typical weak modes for straight tracks are shearings but not bendings, and vice versa for curved tracks. Combining the information of both kinds of tracks is therefore a reasonable strategy to avoid such deformations in the final result. The most obvious weak mode is a translation or rotation of the entire tracking device, which can be only fixed with some kind of reference frame, either an external system or by definition. This, however, is less severe and sometimes even not taken into account at all, since it does not affect the internal alignment of the tracking device.

Once a set of alignment parameters is calculated, it should always be validated. Apart from checking the improvement of the residuals, several physics measurements can be utilized, especially to probe for remaining weak modes. Known charge, forward-backward or ϕ-symmetries of suitable physics processes can be used. Distributions of the signed curvature, the signed transverse impact parameter or invariant masses are sensitive observables as well.

2.3.2 Algorithms

In the case of the CMS Tracker, with its approximately 16,000 individual modules, an amount of roughly 10^5 alignment parameters are needed for a complete description of the misalignment. In this case the computation of the parameters using straightforward recipes becomes unreasonably slow and even causes numerical problems. The three algorithms presented in the following section, which are rather diverse examples of how to cope with such challenging circumstances, have therefore been implemented within the

CMS software framework. Since the outputs from these three algorithms are independent from each other, this allows to conduct an additional validation.

The HIP Algorithm

The HIP algorithm [10] (for either Hits and Impact Points, or Helsinki Institute of Physics) is a straight forward and easy to implement biased alignment algorithm, that computes the alignment parameters for each alignable object separately. Only when iterating over the track sample a certain kind of indirect feedback between the alignable objects is established due to the track refit.

Since only individual alignable objects are regarded, equation 2.2 can be partitioned. This is simply done by evaluating the corresponding expressions for each alignable object i together with its associated parameters $\boldsymbol{p}_i$:

$$\boldsymbol{\Delta}_i(\boldsymbol{p}_i) = \boldsymbol{m}_i - \boldsymbol{f}_i(\check{\boldsymbol{q}}, \boldsymbol{p}_i) \approx \boldsymbol{m}_i - \check{\boldsymbol{f}}_i - \boldsymbol{D}_{p,i}\boldsymbol{\Delta}\boldsymbol{p}_i \tag{2.3}$$

with

$$\check{\boldsymbol{f}}_i = \boldsymbol{f}_i(\check{\boldsymbol{q}}, \check{\boldsymbol{p}}_i), \quad \boldsymbol{\Delta}\boldsymbol{p}_i = \boldsymbol{p}_i - \check{\boldsymbol{p}}_i, \quad \boldsymbol{D}_{p,i} = \partial \boldsymbol{f}_i / \partial \boldsymbol{p}_i \Big|_{\check{\boldsymbol{q}}, \check{\boldsymbol{p}}_i}.$$

The result is then determined by minimizing the normalized squared residuals from a given set of tracks, again for each alignable object separately. The formal solution is then given by:

$$\boldsymbol{\Delta}\boldsymbol{p}_i = \left(\sum_{\text{tracks}} \boldsymbol{D}_{p,i}^{\mathrm{T}} \boldsymbol{V}_i \, \boldsymbol{D}_{p,i} \right)^{-1} \left(\sum_{\text{tracks}} \boldsymbol{D}_{p,i}^{\mathrm{T}} \boldsymbol{V}_i \, \boldsymbol{\Delta}_i \left(\check{\boldsymbol{p}}_i\right) \right) \tag{2.4}$$

The MillePede Algorithm

The Millepede algorithm [11, 12] is an unbiased algorithm that minimizes the sum of the squared residuals of all tracks at once. To do so, a system of equations, equivalent to the formal solution of an ordinary least-sqares fit, is solved. However, to achieve this in a reasonable amount of time, only the solution for the alignment parameters is computed, while the computation of the improved track parameters is skipped. This is possible because of the special structure of the equations. Firstly, the coefficient matrix is symmetric and, mostly due to the independence of the single tracks, relatively sparse. Secondly, only the alignment parameters are common parameters for all track measurements, while the specific track parameters are only relevant for each corresponding track. Because of this the solutions for the alignment and track parameters are only coupled via coefficient matrices of the form

$$\boldsymbol{G} = \boldsymbol{D}_p^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_q \,.$$

To set up the reduced system of equations, for each track the following information has to be extracted:

$$\boldsymbol{\Gamma} = \boldsymbol{D}_q^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_q \,, \quad \boldsymbol{\beta} = \boldsymbol{D}_q^{\mathrm{T}} \boldsymbol{V}^{-1} \left(\boldsymbol{m} - \check{\boldsymbol{f}} - \boldsymbol{D}_p \, \boldsymbol{\Delta p}' \right) .$$

Here $\boldsymbol{\Delta p}' = \boldsymbol{p}' - \check{\boldsymbol{p}}$ may already include an estimate $\boldsymbol{p}'$ on the actual alignment. Then compute

$$\boldsymbol{\Delta C} = \boldsymbol{D}_p^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_p - \boldsymbol{G} \, \boldsymbol{\Gamma}^{-1} \boldsymbol{G}^{\mathrm{T}},$$
$$\boldsymbol{\Delta g} = \boldsymbol{D}_p^{\mathrm{T}} \boldsymbol{V}^{-1} \left(\boldsymbol{m} - \check{\boldsymbol{f}} - \boldsymbol{D}_p \, \boldsymbol{\Delta p}' + \boldsymbol{D}_q \, \boldsymbol{\Gamma}^{-1} \boldsymbol{\beta} \right) .$$

Note the expression $-\boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}$ instead of $\boldsymbol{\Delta q}$. These are all necessary terms, including implicitly the full information from all track parameters. The complete system of equations to determine the alignment parameters then reads:

$$\boldsymbol{C} \, \boldsymbol{\Delta p} = -\boldsymbol{g} \tag{2.5}$$

with

$$\boldsymbol{C} = \sum_{\text{tracks}} \boldsymbol{\Delta C}, \quad \boldsymbol{g} = \sum_{\text{tracks}} \boldsymbol{\Delta g}.$$

To solve this, matrix inversion is feasible only if the number of parameters is rather small ($N \leq 10^3$). However, usually the matrix $\boldsymbol{C}$ is also relatively sparse, such that less time consuming and more reliable methods can be used, for example the GMRES method [13].

The method also offers the possibility to introduce constraints into the solution, which allows to align on various hierarchical levels at once. When aligning for instance on module and layer level at the same time, these constraints can remove redundant degrees of freedom by forcing the mean shift of all modules within one layer to zero.

Annotation: So far only implementations for diagonal covariance matrices $\boldsymbol{V}$ exist, such that correlations due to multiple scattering are not taken into account.

The Kalman Alignment Algorithm

The Kalman alignment algorithm is an unbiased, sequential, global method, derived from the Kalman filter. It is sequential in the sense that the alignment parameters are updated after each processed track. Its global nature stems from the fact that these updates are not restricted to the alignable objects that were crossed by the track. In case the number of alignable objects is very large, it is possible to limit the update to those alignable objects that have significant correlations with the ones traversed by the current track trajectory. This is solely done by keeping track of which alignable objects were hit by which tracks. The algorithm as well as its performance are discussed in detail in the following chapters.

Chapter 3

Alignment Software Framework

All alignment related software has been merged within CMSSW, the CMS Software Framework [14]. Starting from a rather small software package intended only for track-based alignment for the Tracker [15], it was further developed to include also all necessary tools for the inclusion of survey [16] and laser alignment [17] related tasks, the validation of alignment results [18] as well as Muon System alignment [19]. Its contents can be roughly divided into the following categories:

- A hierarchical representation of the relevant mechanical structures of the Tracker and the Muon System together with a well defined parameterization of their possible movements and rotations.
- Common interfaces for the alignment algorithms and their concrete implementations.
- Common interfaces defining the input for the alignment algorithms together with concrete implementations and factories that provide this input for the algorithms.
- Tools for applying and reading misalignment from a database.
- Tools for the simulation of alignment scenarios for Monte Carlo studies.
- An instance that manages all of the above within the software framework.

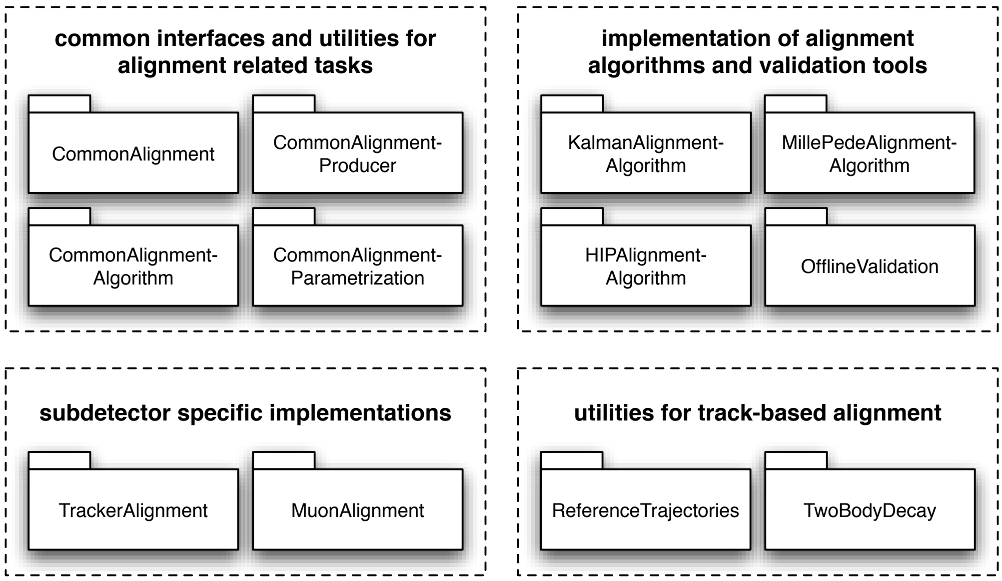


Figure 3.1: Conceptional view of the components of the Alignment Software Framework that are relevant for track-based alignment.

- Well defined tools for validating the output from alignment algorithms.
- Finally, so-called producers that select specific events from the general data stream for alignment purposes.

Figure 3.1 shows an conceptional view of the software components that are most relevant for track-based alignment. The ones that are used for the alignment of the Tracker are described in the following sections.

3.1 Alignable Concept

The fundamental object for alignment within the software framework is the so-called Alignable. Every relevant mechanical structure, be it a single module, an endcap or barrel layer, or even the entire Tracker, is mapped to a specialized representation of this abstract object. It allows for example access to the associated alignment parameters or provides a link to the corresponding counterpart in the tracking geometry and the means to move it.

In order to be usable by an alignment algorithm, an Alignable has to be equipped with an instance of an AlignmentParameters object. The object encapsulates, as its name suggests, the alignment parameters and the corresponding covariance matrix. When handed to an algorithm, the current information can be retrieved for processing and be updated afterwards. To avoid overhead when being copied, the data itself is stored using a reference counting smart pointer. This is especially important for the CompositeAlignmentParameters that hold the joint information for several Alignables, which are only used temporarily and tend to hold large vectors and matrices (see Section 3.4).

3.2 TrackerAlignment

Figure 3.2 sketches the hierarchy of the alignable objects in the Tracker. In contrast to the geometry hierarchy used for tracking, which is designed to facilitate the task of pattern recognition, the hierarchy of the alignable objects follows mostly the actual mechanical mounting structures. For convenience, a few additional hierarchical levels have been added, for instance the *TECRings* which enable the selection of the 2D-modules within the *TECPetals*.

Defining the alignment hierarchy this way has two advantages:

- Obviously, it is of advantage when the alignables used by the algorithms reflect the *hierarchical behavior* of the misalignment, stemming from the given mechanical structure of the Tracker.
- The knowledge of mounting precisions is usually available only hierarchically, i.e. sensor vs. module, module vs. rod, rod vs. layer and so on. Having an alignment hierarchy closely related to the mounting structure allows to provide realistic misalignment scenarios for Monte Carlo studies.

In addition to the definition of the alignment hierarchy itself, a link to the tracking geometry is defined. In the latter, the modules are represented by

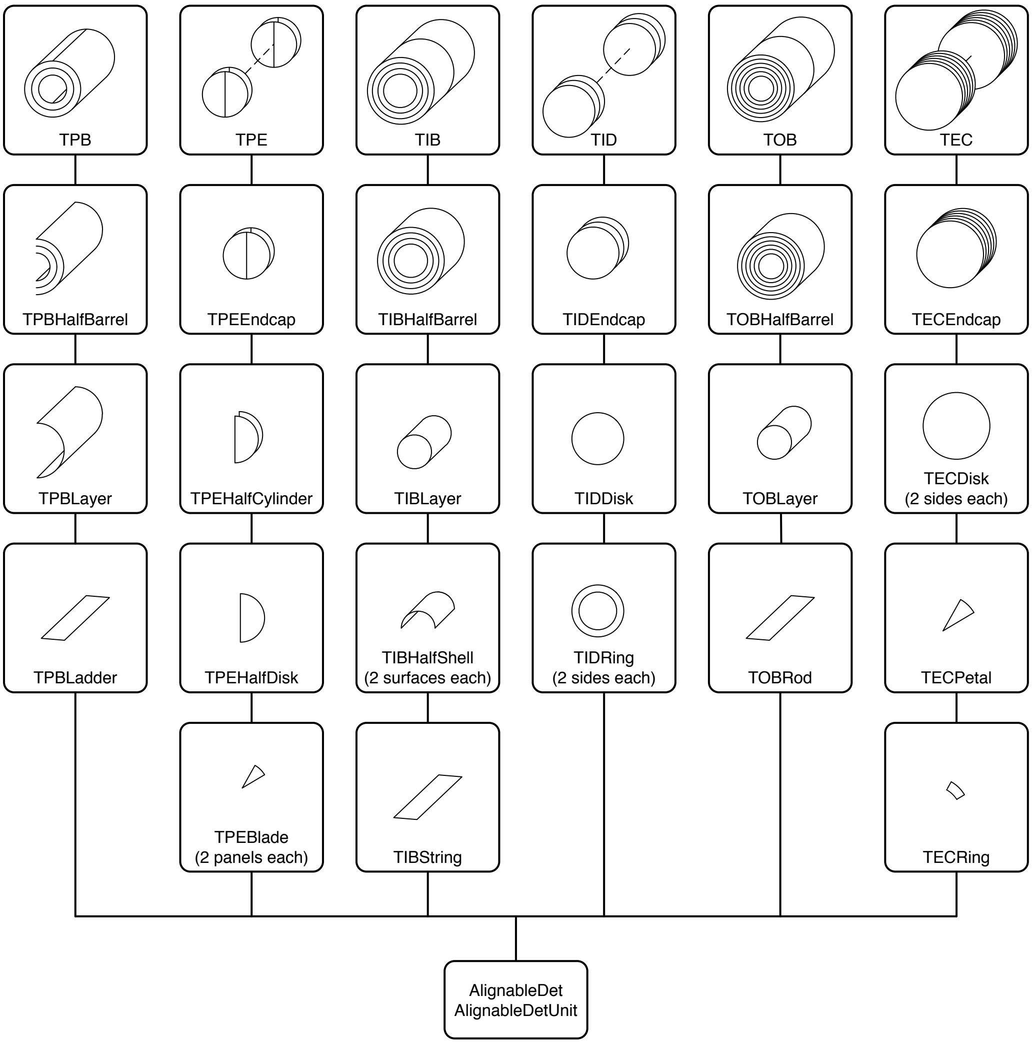


Figure 3.2: Representation of the alignable object hierarchy of the CMS Tracker.

objects of type GeomDet, holding the full information about their position, dimensions and resolution. Actual measurements always correspond to a certain GeomDet. When the measurements are handed to an algorithm, this link can be used to retrieve the corresponding alignable object.

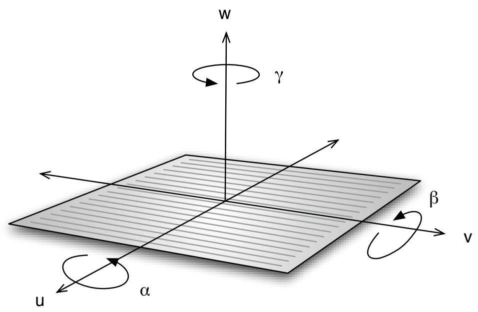


Figure 3.3: Definition of the three local coordinates and rotations.

3.3 CommonAlignmentParametrization

A properly defined coordinate system is extremely beneficial for every alignment task. For instance, the global coordinate system (X, Y, Z) is not very useful for the alignment of individual modules, since the sensitive coordinate perpendicular to the silicon strips does in general not coincide with one of the global axes. Therefore a local system (u, v, w) is defined for every individual module: u is the coordinate perpendicular to the strips, v is parallel to the strips, and w is perpendicular to u and v, pointing away from the surface (see Figure 3.3). Together with the three rotations α, β and γ, defined as the rotations around u, v and w, respectively, every possible rigid body movement of a module can be described.

The transformation from the global frame $\boldsymbol{r} = (X, Y, Z)^{\mathrm{T}}$ to the local frame $\boldsymbol{\rho} = (u, v, w)^{\mathrm{T}}$ is given by:

$$\boldsymbol{\rho} = \boldsymbol{R}\,(\boldsymbol{r} - \boldsymbol{r}_0)$$

where $\boldsymbol{R}$ is a rotation and $\boldsymbol{r}_0$ is the position of the module center in global coordinates. In the presence of misalignment, this equation has to be modified:

$$\boldsymbol{R} \longrightarrow \boldsymbol{\Delta R} \cdot \boldsymbol{R}, \quad \boldsymbol{r}_0 \longrightarrow \boldsymbol{r}_0 + \boldsymbol{\Delta r}.$$

The additional rotation matrix can be expressed as $\boldsymbol{\Delta R} = \boldsymbol{R}_\gamma \boldsymbol{R}_\beta \boldsymbol{R}_\alpha$, where $\boldsymbol{R}_\gamma$, $\boldsymbol{R}_\beta$ and $\boldsymbol{R}_\alpha$ represent small rotations by $\Delta\alpha$, $\Delta\beta$ and $\Delta\gamma$ around the local axes. The global position correction $\boldsymbol{\Delta r}$ transforms to the local frame as $\boldsymbol{\Delta\rho} = \boldsymbol{\Delta R}\cdot\boldsymbol{R}\cdot\boldsymbol{\Delta r} = (\Delta u, \Delta v, \Delta w)^{\mathrm{T}}$. Thus, the corrected transformation from the global to the local misaligned frame $\boldsymbol{\rho}^c$ reads:

$$\boldsymbol{\rho}^c = \boldsymbol{\Delta R}\cdot\boldsymbol{R}\,(\boldsymbol{r}-\boldsymbol{r}_0) + \boldsymbol{\Delta\rho}.$$

Finally, this representation can be used to determine by how much the position of given track prediction (f_u, f_v) on the surface of a module (labeled with i) changes if a small correction $\boldsymbol{\Delta p}_i = (\Delta u, \Delta v, \Delta w, \Delta\alpha, \Delta\beta, \Delta\gamma)^{\mathrm{T}}$ is applied. This dependence can be mathematically represented as a derivative matrix $\boldsymbol{D}_q$ (cf. Section 2.3.1). If the rotations are sufficiently small they can be linearized, and the resulting derivatives are:

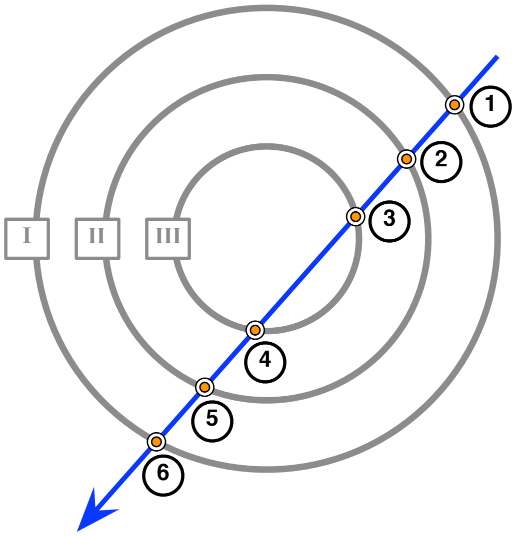
$$\boldsymbol{D}_{p,i} = \frac{\partial(f_u, f_v)}{\partial\boldsymbol{p}_i} = \begin{pmatrix} -1 & 0 & \tan\psi & f_v\tan\psi & f_u\tan\psi & f_v \\ 0 & -1 & \tan\theta & f_v\tan\theta & f_u\tan\theta & -f_u \end{pmatrix} \tag{3.1}$$

The quantity ψ is the angle between the track and the vw-plane, and θ is the angle between the track and the uw-plane.

In the software framework the individual alignable objects can be associated with RigidBodyAlignmentParameters, that are designed to hold a set of local alignment parameters as introduced above. For CompositeAlignable objects, i.e. for alignables representing hierarchical structures above single modules, the computation of the associated derivatives can be based on the equations above. First, the derivatives for the local module $\boldsymbol{D}_{p,\mathrm{mod}}$ are calculated, using the composite's alignment parameters translated into the module frame. Then, the transformation matrix $\boldsymbol{\Omega}_{\mathrm{mod}\to\mathrm{comp}}$ from the local module frame to the local frame of the composite alignable is computed. The final result is then calculated via the chain-rule, $\boldsymbol{D}_{p,\mathrm{comp}} = \boldsymbol{D}_{p,\mathrm{mod}}\,\boldsymbol{\Omega}_{\mathrm{mod}\to\mathrm{comp}}$. For composite alignables which are not planar or do not have an unambiguously defined sensitive direction[1], the definition of the axes has to be altered slightly.

Another very important feature are the so-called CompositeAlignmentPa-

[1]For illustration: A *TOBRod* is flat and the sensitive directions of all modules are collinear. A *TECLayer* is more or less flat, but the sensitive directions of the attached modules vary with azimuth. A *TIBHalfBarrel* cannot even be sensibly associated with a flat surface.



$$\boldsymbol{m} = \begin{pmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \\ \boldsymbol{m}_4 \\ \boldsymbol{m}_5 \\ \boldsymbol{m}_6 \end{pmatrix} \qquad \boldsymbol{p} = \begin{pmatrix} \boldsymbol{p}_{\mathrm{I}} \\ \boldsymbol{p}_{\mathrm{II}} \\ \boldsymbol{p}_{\mathrm{III}} \end{pmatrix}$$

$$\boldsymbol{D}_p = \begin{pmatrix} \boldsymbol{D}_{\mathrm{I},1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{\mathrm{II},2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}_{\mathrm{III},3} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D}_{\mathrm{III},4} \\ \boldsymbol{0} & \boldsymbol{D}_{\mathrm{II},5} & \boldsymbol{0} \\ \boldsymbol{D}_{\mathrm{I},6} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}$$

Figure 3.4: Example for a more advanced concatenation of derivatives, as it could happen for a cosmic particle traversing the pixel barrel layers. On the left, the track is represented as blue arrow, the pixel barrel layers (labelled from I to III) as grey concentric circles and the measurements (labelled from 1 to 6) as orange dots. On the right, a possible concatenation of the measurements, the alignment parameters and the derivatives is shown.

rameters. They hold the full alignment information of a given collection of alignables, i.e. the full set of parameters and their associated variance-covariance matrix. They are also capable of concatenating the alignment derivatives of the individual alignables $\boldsymbol{D}_{p,i}$ to the full matrix $\boldsymbol{D}_p$. This sounds rather trivial, but gets somewhat complicated in the general case where alignable objects can get hit several times in arbitrary order. This happens especially when higher level composite alignables are used — for an example see Figure 3.4.

3.4 CommonAlignmentAlgorithm

Even though the three algorithms implemented for track-based alignment are very different, they plug into the software framework via the same mecha-

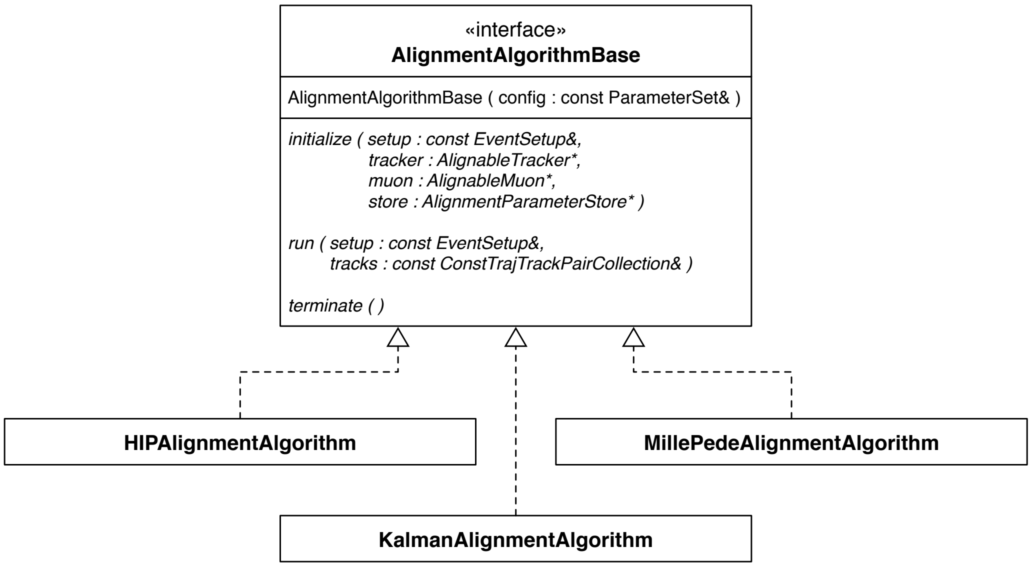


Figure 3.5: UML representation of the software implementations of the algorithms.

nism, which is defined in the package CommonAlignmentAlgorithm. An interface class, called AlignmentAlgorithmBase, defines the following features (see also Figure 3.5):

- Initialization: At this point usually several issues are addressed, from the simple initialization of alignment parameters to algorithm specific tasks, like opening files for I/O or preparing histograms or N-tuples.
- Processing tracks: The algorithms are called to process the reconstructed tracks.
- Termination: When all tracks have been processed, the algorithms have to write out their results and debugging information.

The actual implementations of the algorithms are housed in separate packages, called HIPAlignmentAlgorithm, MillePedeAlignmentAlgorithm and KalmanAlignmentAlgorithm.

In addition, several useful utilities are defined, such as I/O-interfaces or manipulators for alignment parameters and alignables. For the Kalman Alignment Algorithm, the AlignmentParameterStore utility is the most essential

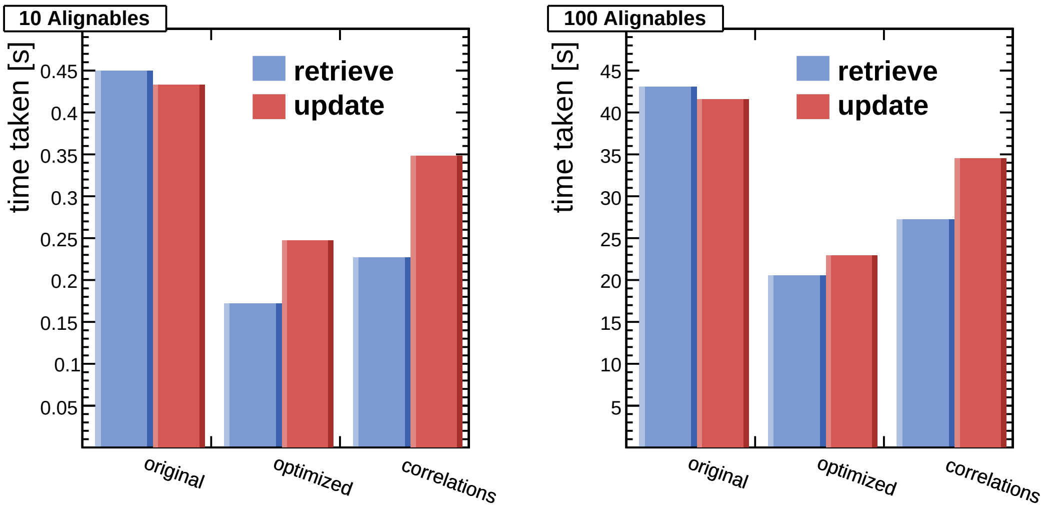


Figure 3.6: Time elapsed for updating and retrieving the concatenated alignment parameters 2000 times for 10 (left) and 100 (right) alignable objects with different versions of the alignment parameter store.

one. It is used to manage all issues regarding the retrieval, update and storage of alignment parameters and, even more important, the correlations *between* different alignable objects. The store utilizes CompositeAlignmentParameters objects (see Section 3.3) to hand data to and fetch data from an alignment algorithm. Internally the full information is stored in look-up tables. Since this functionality is of fundamental importance for the Kalman Alignment algorithm, the performance of this store has been carefully optimized with respect to its initial implementation. The original implementation was not designed to retrieve and update information for a rather large amount of alignables, i.e. $O(10^2)$, at once. In addition, the off-diagonal covariance matrix elements σ_{ij} were only stored as they were, while sometimes the storage of the correlations $R_{ij} = \sigma_{ij}/\sigma_i\sigma_j$ is preferable from the algorithmic point of view. The optimized version — and even the computationally much more intricate version storing the correlations — shows a highly improved computational performance (see Figure 3.6).

3.5 ReferenceTrajectories

All track-related information is handed to the Kalman Alignment Algorithm and the MillePede Algorithm in the form of so-called *reference trajectories*. For the HIP Algorithm this is not needed, as it deals only with the individual hits. These objects provide the following data:

- The vector $\boldsymbol{m}$ containing all N individual 2D measurements $(m_{u,i}, m_{v,i})$.
$$\boldsymbol{m} = (m_{u,1}, m_{v,1}, m_{u2}, m_{v,2}, \ldots, m_{u,N}, m_{v,N})^{\mathrm{T}}$$
- The measurement covariance matrix $\boldsymbol{V} = \mathsf{cov}(\boldsymbol{m})$, including not only the intrinsic measurement errors but also the material effects, i.e. multiple scattering. Energy loss is considered to be deterministic.
- The vector $\boldsymbol{q}$ that holds the n track parameters used to described the trajectory associated to the measurements $\boldsymbol{m}$:
$$\boldsymbol{q} = (q_1, q_2, \ldots, q_n).$$
- Optionally, the covariance matrix $\boldsymbol{C} = \mathsf{cov}(\boldsymbol{q})$. This is meaningful only if the track-parameters are not only a linearization point but result from an external measurement of the track.
- The vector $\boldsymbol{f}$ containing the N individual 2D trajectory positions $(f_{u,i}, f_{v,i})$ as predicted by the track parameters $\boldsymbol{q}$:
$$\boldsymbol{f} = (f_{u,1}, f_{v,1}, f_{u2}, f_{v,2}, \ldots, f_{u,N}, f_{v,N})^{\mathrm{T}}.$$
- The matrix $\boldsymbol{D}_q$, that holds the derivatives of the N predicted trajectory positions $\boldsymbol{f}_i$ w.r.t. the n track parameters $\boldsymbol{q}$:

$$\boldsymbol{D}_q = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} = \begin{pmatrix} \partial f_{u,1}/\partial q_1 & \partial f_{u,1}/\partial q_2 & \cdots & \partial f_{u,1}/\partial q_n \\ \partial f_{v,1}/\partial q_1 & \partial f_{v,1}/\partial q_2 & \cdots & \partial f_{v,1}/\partial q_n \\ \partial f_{u,2}/\partial q_1 & \partial f_{u,2}/\partial q_2 & \cdots & \partial f_{u,2}/\partial q_n \\ \partial f_{v,2}/\partial q_1 & \partial f_{v,2}/\partial q_2 & \cdots & \partial f_{v,2}/\partial q_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial f_{u,N}/\partial q_1 & \partial f_{u,N}/\partial q_2 & \cdots & \partial f_{u,N}/\partial q_n \\ \partial f_{v,N}/\partial q_1 & \partial f_{v,N}/\partial q_2 & \cdots & \partial f_{v,N}/\partial q_n \end{pmatrix}.$$

- Collections of the measurements and their associated trajectory states. They are handled inside the reconstruction framework as objects called TransientRecHit and TrajectoryStateOnSurface.

Several different types of reference trajectories have been implemented. The basic one is simply called ReferenceTrajectory. It uses a helix track model to describe a particle trajectory within a constant magnetic field. It can be considered as the standard one and is the basis for the other two kinds, which are either directly derived classes or at least make heavy use of it. The BzeroReferenceTrajectory does in principle the same, but a straight line track model for the case of no magnetic field is assumed. Since under these circumstances no momentum measurement is available, which is needed to estimate the effects of multiple scattering, the user has to provide a guess. Both implementations use the local track parameters defined on the surface of the first measurement for the parameterization of the reference trajectory. In contrast to that, the DualReferenceTrajectory and the DualBzeroReferenceTrajectory, which offer in principle the same functionalities, are parameterized by a set of local parameters defined on the surface of the $\lfloor N/2 \rfloor$-th measurement. Finally, the TwoBodyDecayTrajectory uses two tracks from a two-body decay to calculate a reference trajectory (see Chapter 5).

For all types of reference trajectories the design is defined via the ReferenceTrajectoryBase interface-class (see Figure 3.7), such that the handling for the algorithms is always the same. Even their creation, for which the set of input parameters differ considerably, is detached from the algorithms by using

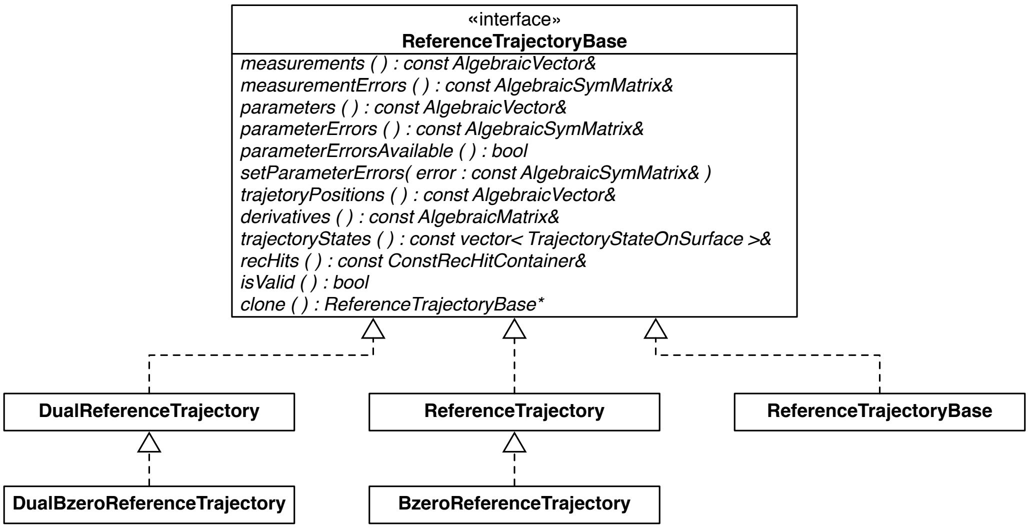


Figure 3.7: UML representation of the most important features of the implemented reference trajectories.

factories that are implemented as configurable plugins. The factories simply take a set of tracks, or if existing an additional set of external track estimates, and return the desired reference trajectory objects as smart pointers.

Since the reference trajectories are fundamental (physics) objects for track-based alignment, they have been tested thoroughly. The correct propagation through the magnetic field, together with the proper treatment of the material effects, was carefully examined. Especially the latter had to be validated with more care, as the tracking geometry description assumes that full material budget is concentrated only within the thin layers of the sensitive modules. This strategy works well for tracking purposes [20, 21], but had to be revised for this special case.

In order to check the quality of the reference trajectories, global least square fits were done, estimating the track parameters for a sample of fully MC-simulated tracks stemming from 50,000 $Z \to \mu^+\mu^-$ decays. As input for the fit, only data provided from standard reference trajectories, as described above, was used.

With the notation from above, the estimate of the track parameters $\tilde{\boldsymbol{q}}$ for a single trajectory is given by:

$$\tilde{\boldsymbol{q}} = (\boldsymbol{D}_q^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_q)^{-1} \boldsymbol{D}_q^{\mathrm{T}} \boldsymbol{V}^{-1} (\boldsymbol{m} - \boldsymbol{f} + \boldsymbol{D}_q \boldsymbol{q}).$$

It should be noted that this is conceptually very similar to what is done by the MillePede and Kalman alignment algorithms — except, of course, that no alignment parameters are estimated. Two different test statistics were analyzed. The first one was the resulting χ^2:

$$\chi^2 = \boldsymbol{\delta}^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{\delta},$$

with

$$\boldsymbol{\delta} = \boldsymbol{m} - \boldsymbol{f} - \boldsymbol{D}_q \boldsymbol{\Delta q}\,, \qquad \boldsymbol{\Delta q} = \tilde{\boldsymbol{q}} - \boldsymbol{q}.$$

The second one was the vector of the pull-quantities, defined for every hit i as:

$$\delta'_i = \frac{(\boldsymbol{\delta})_i}{\sqrt{(\boldsymbol{C})_{ii}}}\,, \qquad \boldsymbol{C} = \boldsymbol{V} - \boldsymbol{D}_q (\boldsymbol{D}_q^{\mathrm{T}} \boldsymbol{V}^{-1} \boldsymbol{D}_q)^{-1} \boldsymbol{D}_q^{\mathrm{T}}.$$

Figures 3.8 to 3.10 show the results. The overall χ^2 probability distribution (Figure 3.8, left) is obviously not a uniform distribution, as one would expect in the case of Gaussian errors. The reason for that can be elucidated when the distributions are plotted against the number of degrees of freedom (Figure 3.8, right), where one can see clearly that the non-uniform trend starts mostly only for tracks with more than 21 degrees of freedom. For the helical track model, this number refers to tracks with more than 13 hits[2], which are typically tracks from the endcap regions — there are only 13 layers in the barrel. It seems therefore that the measurement errors are somewhat overestimated for this region. Nevertheless, the shape of the overall χ^2 distribution function is still reasonable enough to work with the assumption of Gaussian errors.

The second test statistics, the pull quantities, show the expected mean value of zero and variance of one, but the shape of the distribution is clearly not

[2]number of degrees of freedom = 13 hits $\times$ 2 hits per measurement $-$ 5 estimated track parameters

Gaussian. This is true for the overall distributions, see Figure 3.9, as well as differentiated by the hit position, i.e. the pull-quantities for the first, second, ..., N-th hit of a track separately, see Figure 3.10. These results show that the current implementation of the reference trajectories is working sufficiently well.

3.6 CommonAlignmentProducer

A single instance object, called AlignmentProducer, handles and steers almost all alignment related tasks for the Tracker and the Muon System within the software framework. It applies misalignment scenarios, reads geometries from a database, saves them back to the database, calls the alignment algorithms etc.

3.7 Dedicated Data Streams

A dedicated data stream, called *AlCaReco*, for alignment and calibration of all detector parts will be present during the operation of CMS. The associated producers provide essentially a skim, leading to a much reduced data size both by event selection and event content selection. They are able to select events according to their physics event content and separate signal from background even in the presence of misalignment. Furthermore, only the analysis objects explicitly needed for the specific tasks will be recorded. For the track-based alignment of the Tracker, mostly collisions data from minimum bias events, isolated muons and muons from W-, Z-, J/Ψ- and Υ-decays will be selected. In addition, tracks from beam halo events and cosmic particles will be picked out.

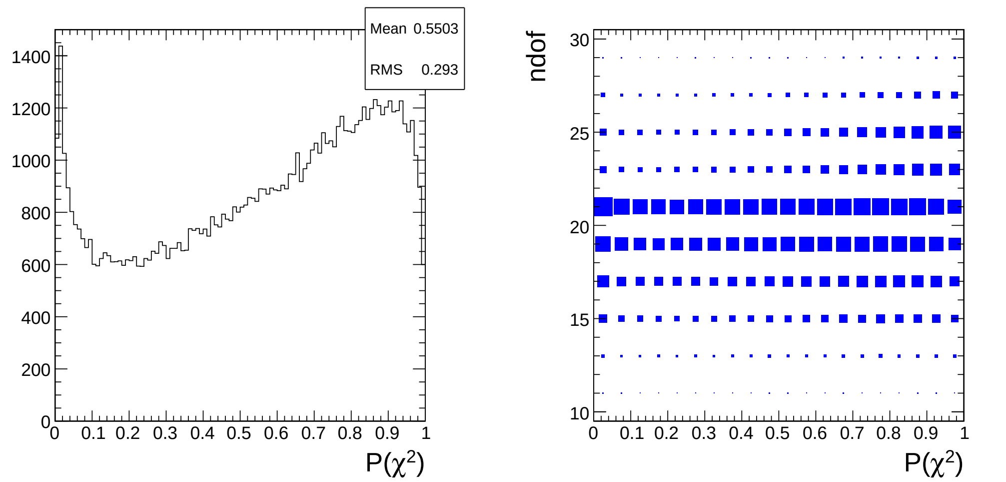


Figure 3.8: The χ^2-probabilities.

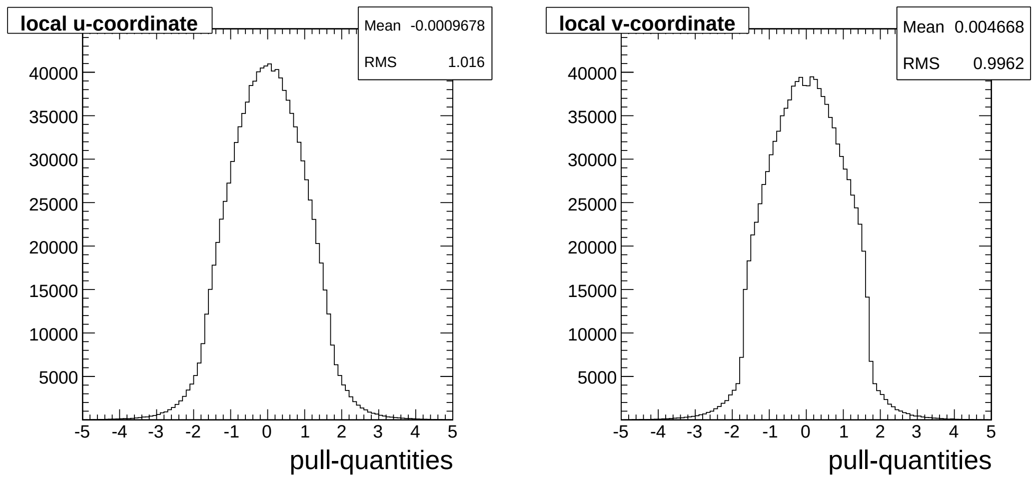


Figure 3.9: The pull-quantities.

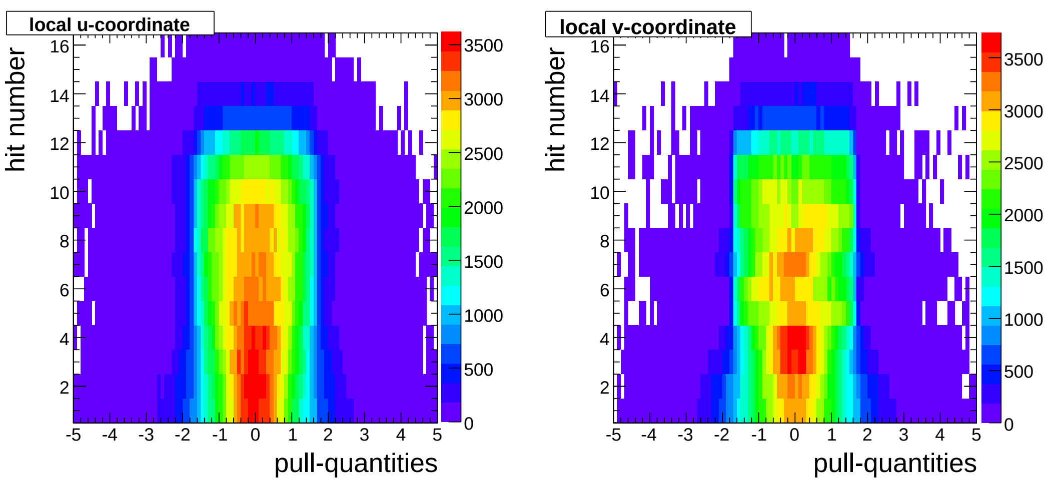


Figure 3.10: The pull-quantities for the n-th hit of the reference trajectory.

3.8 Validation Tools

Various user-friendly tools have been developed to validate the output of the alignment algorithms. They can be used for simulation studies as well as for real data. Distributions of (normalized) track residuals as well as global track parameters can be extracted and visualized. In addition, a tool for the comparison between different geometries is available, taking into account and eliminating effects of trivial global rotations or translations. These utilities allow the user not only to judge the quality of the output of a single alignment run, but also to compare the results from different algorithms in a well-defined and consistent way.

Chapter 4

The Kalman Alignment Algorithm

The Kalman alignment algorithm [22] is an unbiased, sequential, global method, derived from the Kalman filter [23, 24, 25]. It is sequential in the sense that the alignment parameters are updated after each processed track. Its global nature stems from the fact that these updates are not restricted to the alignable objects that were crossed by the track. If the number of alignable objects is very large, it is desirable to limit the update to those alignable objects that have significant correlations with the ones traversed by the current trajectory. This is achieved by keeping track of which alignable objects were crossed by which trajectories. In addition, the sequential approach allows to use always the up-to-date alignment parameters for track reconstruction.

In the formalism that is proposed here it is possible to use prior information about the alignment obtained from mechanical and/or laser measurements. It is also possible to fix the position of certain detector units by giving them a large prior weight (small prior uncertainty). A requirement that several detectors move along with each other can be enforced by large prior correlations.

4.1 Sequential Update of Alignment Parameters

In this section the formulas for sequentially updating a set of alignment parameters and their corresponding variance-covariance matrix are derived. The notation used follows the one in Section 2.3.1. Only an additional index k is introduced, to point out the fact that both the alignment parameters and their variance-covariance matrix are updated for each track. For the sake of legibility — and since these objects are unambiguously related to a single track — the index is omitted for the measurement vector $\boldsymbol{m}$ and its variance-covariance matrix $\boldsymbol{V}$ as well as the derivatives matrices $\boldsymbol{D}_q$ and $\boldsymbol{D}_p$.

Starting from the track model $\boldsymbol{f}$ given in Equation (2.1) and the associated residuals $\boldsymbol{\Delta}$ defined in Equation (2.2), the residuals at step k can be written as:

$$\boldsymbol{\Delta}(\boldsymbol{q}_k, \boldsymbol{p}_k) = \boldsymbol{m} - \boldsymbol{f}(\boldsymbol{q}_k, \boldsymbol{p}_k) \approx \boldsymbol{m}' - \begin{pmatrix} \boldsymbol{D}_q & \boldsymbol{D}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_k \\ \boldsymbol{p}_k \end{pmatrix},$$

with

$$\boldsymbol{m}' = \boldsymbol{m} - \boldsymbol{f}(\check{\boldsymbol{q}}_k, \check{\boldsymbol{p}}_k) + \boldsymbol{D}_q\, \check{\boldsymbol{q}}_k + \boldsymbol{D}_p\, \check{\boldsymbol{p}}_k.$$

where $\check{\boldsymbol{q}}_k$ comes from a preliminary track fit and $\check{\boldsymbol{p}}_k$ are typically the nominal positions of the considered alignable objects.

If the prediction of the track parameters $\check{\boldsymbol{q}}_k$ is statistically independent from the observations $\boldsymbol{m}$, the Kalman filter formalism can be directly applied to this relation. Thus, update equations for the track parameters $\boldsymbol{q}_{k+1}$ and the alignment parameters $\boldsymbol{p}_{k+1}$ can be derived:

$$\begin{pmatrix} \boldsymbol{q}_{k+1} \\ \boldsymbol{p}_{k+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_k \\ \boldsymbol{p}_k \end{pmatrix} + \boldsymbol{K} \left[\boldsymbol{m}'_k - \begin{pmatrix} \boldsymbol{D}_q & \boldsymbol{D}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_k \\ \boldsymbol{p}_k \end{pmatrix} \right],$$

with the gain matrix of the filter:

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{C}_{q,k} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{p,k} \end{pmatrix} \begin{pmatrix} \boldsymbol{D}_q^{\mathrm{T}} \\ \boldsymbol{D}_p^{\mathrm{T}} \end{pmatrix} \underbrace{\left(\boldsymbol{V} + \boldsymbol{D}_q\, \boldsymbol{C}_{q,k}\, \boldsymbol{D}_q^{\mathrm{T}} + \boldsymbol{D}_p\, \boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}} \right)^{-1}}_{\boldsymbol{G}_k}.$$

This can be simplified to

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{C}_{q,k}\,\boldsymbol{D}_q^{\mathrm{T}}\,\boldsymbol{G}_k \\ \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\,\boldsymbol{G}_k \end{pmatrix}.$$

Since they are of no further interest, the computation of the improved track parameters can be skipped. This way, the estimate of the alignment parameters can be updated with reduced effort, without neglecting the influence of the track parameters. The update of the variance-covariance matrix can be calculated by linear error propagation. The resulting update equations then read:

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\boldsymbol{G}_k\left(\boldsymbol{m}' - \boldsymbol{D}_q\,\boldsymbol{q}_k - \boldsymbol{D}_p\,\boldsymbol{p}_k\right), \tag{4.1}$$

$$\begin{aligned}\boldsymbol{C}_{p,k+1} = {} & \left(\boldsymbol{I} - \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\boldsymbol{G}_k\,\boldsymbol{D}_p\right)\boldsymbol{C}_{p,k}\left(\boldsymbol{I} - \boldsymbol{D}_p^{\mathrm{T}}\boldsymbol{G}_k\,\boldsymbol{D}_p\,\boldsymbol{C}_{p,k}\right) \\ & + \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\boldsymbol{G}_k\,(\boldsymbol{V} + \boldsymbol{D}_q\,\boldsymbol{C}_{q,k}\,\boldsymbol{D}_q^{\mathrm{T}})\,\boldsymbol{G}_k\,\boldsymbol{D}_p\,\boldsymbol{C}_{p,k}.\end{aligned} \tag{4.2}$$

As both terms on the right hand side of Equation (4.2) are positive definite the left hand side is guaranteed to be positive definite as well.

In general, however, there is no independent prediction of the track parameters. In this case, the preliminary track parameters $\check{\boldsymbol{q}}_k$ are used with zero weight in order not to bias the estimation. This is accomplished by multiplying $\boldsymbol{C}_{q,k}$ with a scale factor α and letting α tend to infinity:

$$\begin{aligned}\overline{\boldsymbol{G}}_k &= \lim_{\alpha\longrightarrow\infty}\left(\boldsymbol{V} + \boldsymbol{D}_p\,\boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}} + \alpha\,\boldsymbol{D}_q\,\boldsymbol{C}_{q,k}\,\boldsymbol{D}_q^{\mathrm{T}}\right)^{-1} \\ &= \boldsymbol{V}_k^{-1} - \boldsymbol{V}_k^{-1}\boldsymbol{D}_q\,(\boldsymbol{D}_q^{\mathrm{T}}\boldsymbol{V}_k^{-1}\boldsymbol{D}_q)^{-1}\boldsymbol{D}_q^{\mathrm{T}}\boldsymbol{V}_k^{-1},\end{aligned}$$

with

$$\boldsymbol{V}_k = \boldsymbol{V} + \boldsymbol{D}_p\,\boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}.$$

Here the Sherman-Morrison inversion formula

$$(\boldsymbol{X} + \boldsymbol{H}\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}})^{-1} = \boldsymbol{X}^{-1} - \boldsymbol{X}^{-1}\boldsymbol{H}\,(\boldsymbol{Y}^{-1} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{X}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{X}^{-1}$$

has been used (see e.g. [26]). It can easily be verified that $\overline{\boldsymbol{G}}_k\boldsymbol{D}_q = \boldsymbol{0}$. The update equation of the alignment parameters can therefore be simplified to:

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\overline{\boldsymbol{G}}_k\left(\boldsymbol{m}' - \boldsymbol{D}_p\,\boldsymbol{p}_k\right), \tag{4.3}$$

$$\begin{aligned}\boldsymbol{C}_{p,k+1} = {} & \left(\boldsymbol{I} - \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\overline{\boldsymbol{G}}_k\,\boldsymbol{D}_p\right)\boldsymbol{C}_{p,k}\left(\boldsymbol{I} - \boldsymbol{D}_p^{\mathrm{T}}\overline{\boldsymbol{G}}_k\,\boldsymbol{D}_p\,\boldsymbol{C}_{p,k}\right) \\ & + \boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\overline{\boldsymbol{G}}_k\,\boldsymbol{V}\,\overline{\boldsymbol{G}}_k\,\boldsymbol{D}_p\,\boldsymbol{C}_{p,k}.\end{aligned} \tag{4.4}$$

Note that no track parameters are present in the update equations anymore. And still, since both terms on the right hand side of Equation 4.4 are positive definite the left hand side is guaranteed to be positive definite as well.

The up-to-date alignment parameters as computed above can be used directly for the reconstruction of the next track.

4.2 Implementation and Computational Complexity

This section gives an estimate of the complexity of the algorithm. All equations are for the case where no independent prediction of the track parameters are present. The two cases are however very similar from the algorithmic point of view, so that the conclusions are valid for both — even the equations look very similar.

The total number of alignable objects is denoted by N. The current track crosses a certain number of alignables, denoted by k. If each of them gives a two-dimensional measurement, the dimension $n = 2\,k$ of the observation vector $\boldsymbol{m}$ is small for high-energy tracks, usually not larger than 30. The matrix $\boldsymbol{D}_q$ is — in case of a helix track model — of size $n \times 5$ and is therefore small. The matrix $\boldsymbol{D}_p$ is a row of N blocks $(\boldsymbol{D}_p)_i$ of size $n \times m$, where m is the number of alignment parameters per detector unit (usually equal or less than 6). For each track, only k out of these N blocks are different from zero. The set of detector units crossed by the current track is denoted by

$I = \{i_1, \ldots, i_k\}$. Then the matrix $\boldsymbol{D}_p$ has the following form:

$$\boldsymbol{D}_p = \Big(\mathbf{0} \ldots \mathbf{0}\ (\boldsymbol{D}_p)_{i_1}\ \mathbf{0} \ldots \mathbf{0}\ (\boldsymbol{D}_p)_{i_2}\ \mathbf{0} \ldots\ldots \mathbf{0}\ (\boldsymbol{D}_p)_{i_k}\ \mathbf{0} \ldots \mathbf{0}\Big).$$

Note that the matrices $(\boldsymbol{D}_p)_i$ differ somewhat from the derivatives matrices $\boldsymbol{D}_{p,i}$ in Section 3.3, as the latter are only identical to the $2 \times m$ non-zero blocks of the full $n \times m$ matrix (compare also to Figure 3.4).

4.2.1 Update of the Alignment Parameters

The only large matrix in the parameter update is the product $\boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}$. It is a column of N blocks each of which has size $m \times n$. However, only those blocks need to be computed that correspond to the alignable objects that have a significant correlation with the ones in the current track. In order to keep track of the necessary updates, a list L_i is attached to each alignable object i, containing the alignable objects that have significant correlations with i. This list may contain only i itself in the beginning and grows as more tracks are processed. If there is prior knowledge about correlations, for instance because of mechanical constraints, it can be incorporated in the list and in the initial variance-covariance matrix. The length of the list can hopefully be restricted to a fairly small number, as the correlations of alignable objects that are far from each other tend to be small. This leads to the following procedure for computing the updated alignment parameters:

1. Update the list L_i for every $i \in I$ (see Section 4.3).
2. Form the list L of all alignable objects that are correlated with the ones crossed by the current track: $L = \bigcup_{i \in I} L_i$. The size of L should be much smaller than N.
3. For all $j \in L$ compute:

$$\left(\boldsymbol{C}_{p,k}\,\boldsymbol{D}_p^{\mathrm{T}}\right)_j = \sum_{i \in I} (\boldsymbol{C}_{p,k})_{ji}\ (\boldsymbol{D}_p)_i^{\mathrm{T}}\,.$$

   Each block $(\boldsymbol{C}_{p,k})_{ji}$ is of size $m \times m$.

4. Compute:
$$\boldsymbol{D}_p \boldsymbol{C}_{p,k} \boldsymbol{D}_p^{\mathrm{T}} = \sum_{i \in I} \left(\boldsymbol{D}_p\right)_i \left(\boldsymbol{C}_{p,k} \boldsymbol{D}_p^{\mathrm{T}}\right)_i .$$

5. Compute: $\boldsymbol{V}_k$ and $\overline{\boldsymbol{G}}_k$. All matrices involved are of size $n \times n$.

6. Compute:
$$\overline{\boldsymbol{m}} = \overline{\boldsymbol{G}}_k \left(\boldsymbol{m}' - \sum_{i \in I} (\boldsymbol{D}_p)_i \left(\boldsymbol{p}_k\right)_i\right).$$

7. For all $j \in L$ compute:
$$\left(\boldsymbol{p}_{k+1}\right)_j = \left(\boldsymbol{p}_k\right)_j + \left(\boldsymbol{C}_{p,k} \boldsymbol{D}_p^{\mathrm{T}}\right)_j \overline{\boldsymbol{m}}. \tag{4.5}$$

The computational complexity of the parameter update is of the order $|L| \cdot |I|$. For the complexity of the list update see Section 4.3.

4.2.2 Update of the Variance-Covariance Matrix

In the beginning the variance-covariance matrix $\boldsymbol{C}_p$ is block-diagonal and contains the prior uncertainty of the alignment parameters, derived from laser alignment and mechanical measurements. If required, it may also contain prior correlations between different detector units. After each track, only the blocks in the list $L = \bigcup_{i \in I} L_i$ need to be updated. This is done in the following way:

1. For all $i, j \in I$ compute:
$$\left(\boldsymbol{I} - \boldsymbol{C}_{p,k} \boldsymbol{D}_p^{\mathrm{T}} \overline{\boldsymbol{G}}_k \boldsymbol{D}_p\right)_{ij} = \boldsymbol{I}_{ij} - (\boldsymbol{C}_{p,k} \boldsymbol{D}_p^{\mathrm{T}})_i \overline{\boldsymbol{G}}_k (\boldsymbol{D}_p)_j .$$

   Each block is of size $m \times m$, with $\boldsymbol{I}_{ij} = \delta_{ij} \boldsymbol{I}_{m \times m}$ (and δ_{ij} being the Kronecker delta).

2. Compute for all $i, j \in I$ the updated variance-covariance matrix for the

alignable objects that have been directly hit :

$$(\boldsymbol{C}_{p,k+1})_{ij} = \sum_{l,m \in I} \left(\boldsymbol{I} - \boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}} \overline{\boldsymbol{G}}_k\, \boldsymbol{D}_p\right)_{il} \left(\boldsymbol{C}_{p,k}\right)_{lm} \left(\boldsymbol{I} - \boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}} \overline{\boldsymbol{G}}_k\, \boldsymbol{D}_p\right)^{\mathrm{T}}_{jm} + \left(\boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}}\right)_i \overline{\boldsymbol{G}}_k \boldsymbol{V} \overline{\boldsymbol{G}}_k \left(\boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}}\right)^{\mathrm{T}}_j$$

3. Compute for $i \in L \setminus I$ and $j \in L$ the updated variance-covariance matrix for the alignable objects that were not directly hit and their correlations with the directly hit alignable objects:

$$(\boldsymbol{C}_{p,k+1})_{ij} = (\boldsymbol{C}_{p,k})_{ij} + \sum_{l \in I} \left(\boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}}\right)_{il} \overline{\boldsymbol{G}}_k \left(\boldsymbol{V}_k - 2\overline{\boldsymbol{G}}_k^{-1}\right) \overline{\boldsymbol{G}}_k \left(\boldsymbol{C}_{p,k}\, \boldsymbol{D}_p^{\mathrm{T}}\right)^{\mathrm{T}}_{jl}$$

The computational complexity of the update of the variance-covariance matrix is of the order of $|L|^2$. Restricting the size of the lists L_i is therefore of crucial importance. An algorithm is proposed in the following section.

4.3 Selective Updates of Alignable Objects

The goal is to develop a mechanism to select which alignable objects should be included into an update. Here, an computationally rather light-weight algorithm is proposed that allows such a selection only by recording which alignable objects were hit by which tracks.

4.3.1 Definition of a Track-based Metrics

First, a relation "$\sim$" between two different alignable objects i and j is defined:

$$i \sim j \iff i \text{ and } j \text{ have been crossed by the same track.}$$

The relation is symmetric, but not transitive. On the basis of this relation a distance between different alignable objects i and j can be defined:

If $i \sim i_1 \sim i_2 \sim \cdots \sim i_n \sim j$ is the shortest chain connecting i

to j, the distance is $d(i,j) = n+1$. In particular, if $i \sim j$, then $d(i,j) = 1$.

With the additional definition $d(i,i) = 0$ it is easy to see that the distance d is a proper metrics, i.e. that

1. $d(i,j) = 0$ if and only if $i = j$,
2. $d(i,j) = d(j,i)$, and
3. $d(i,j) \leq d(i,k) + d(k,j)$ for all k.

Figure 4.1 shows an schematic example of how this metrics works.

4.3.2 Definition of a Selection Algorithm

Using this distance, the following algorithm for updating the lists $L_i, i \in I$ is proposed:

For all $i \in I$ do:

1. For all $j \in I \setminus \{i\}$ do:
   (a) For $i \neq j$ set $d(i,j) = 1$.
   (b) For all $k \in L_j \cap L_i$: If $d(j,k) < d_{\max}$ store $d(i,k) = d(k,i) = d(j,k) + 1$.
   (c) For all $k \in L_j \setminus L_i$:
      i. If $d(j,k) < d_{\max}$ add k to L_i, add i to L_k and store $d(i,k) = d(k,i) = d(j,k) + 1$.
      ii. For all $l \in L_i \setminus L_j$, using the new value $d(i,k)$: If $d(i,l) + d(i,k) \leq d_{\max}$ add l to L_k and store $d(k,l) = d(i,l) + d(i,k)$.
2. If for any combination of two alignable objects i and j more than one value for $d(i,j)$ was stored, keep only the occurence with the smallest value.

In step 1a the distances of all alignable objects listed in I are simply set to 1, i.e. the relation $i \sim j$ is expressed in terms of the metrics. In step 1b this information is propagated for each $i \in I$ to those alignable objects $j \in L_i \setminus I$ that were not hit by this track but were already related before. In case a track hits alignable objects i and j that were not related before, the already existing relations within L_i have to be propagated to the alignables contained in L_j and vice versa. This is taken care of in step 1c. Finally, since new tracks can introduce several possible chains $i \sim i_1 \sim i_2 \sim \cdots \sim i_n \sim j$ between two alignable objects i and j, the shortest one has to be selected in step 2. It should also be noted that it is ensured throughout the entire algorithm that the symmetric nature of the metrics is conserved.

The computational complexity of the list update is of the order $|L| \cdot |I|^2$. It is assumed that the distance $d(k,i)$ is stored along with k in the list L_i. $d_{\max}$ is the largest distance for which correlations are deemed to be significant.

Figure 4.2 shows an example for modules in the Tracker Pixel Barrel. The metrical distances for a module in the third layer are visualized using a color code. The computation was done with a sample of approximately 20,000 high-momentum muons, emerging from the interaction point. If tracks with lower momenta would have been used, a bigger amount of possible "chains" connecting more distant modules would be present, resulting in larger regions of constant metrical value.

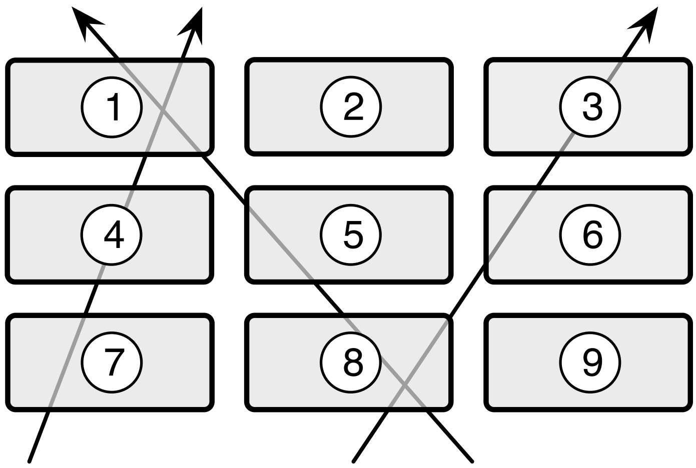


Figure 4.1: Schematic example of the track-based metrical distances: $d(1, 8) = 1$, $d(5, 3) = 2$, $d(7, 6) = 3$, etc.

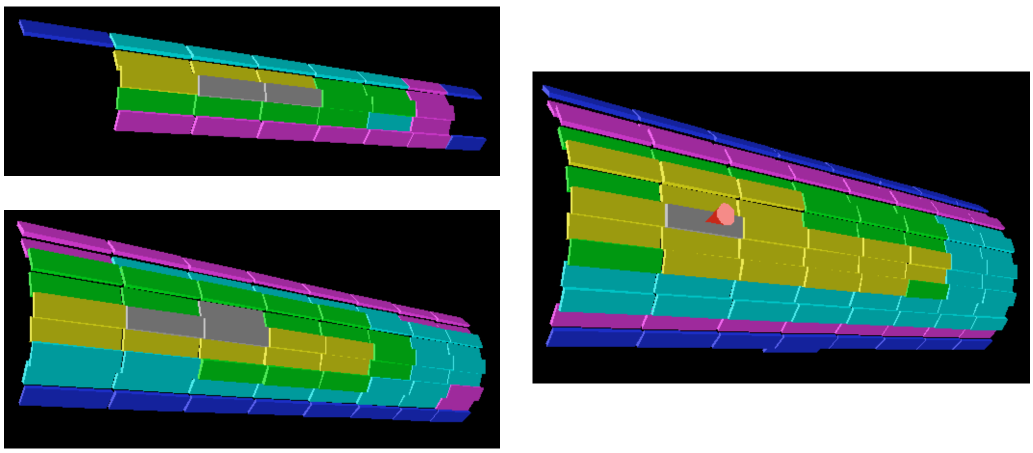


Figure 4.2: Visualization of the track-based metrical distances $d(i, j)$ for a distinct module (marked by the red cone) and the remaining modules in the three layers of the Tracker Pixel Barrel. Only modules with $d \leq 6$ are shown (white: $d = 1$, yellow: $d = 2$, green: $d = 3$, light blue: $d = 4$, magenta: $d = 5$, dark blue: $d = 6$).

4.3.3 Selective Updates of the Variance-Covariance Matrix

The selective update of only a subsample of all alignable objects comes with a big problem: The updated variance-covariance matrix is not guaranteed to be statistically meaningful and consistent anymore. Consider the following example: For a given update, the parameters $\boldsymbol{p}_i$ of only the alignable objects $i \in L = \{i_1, \ldots, i_N\}$ are being updated, while the parameters $\boldsymbol{p}_j$ of the remaining alignable objects $j \in \bar{L} = \{j_1, \ldots, j_M\}$ stay the same.

$$\begin{pmatrix} \boldsymbol{p}_{i_1} \\ \vdots \\ \boldsymbol{p}_{i_N} \\ \boldsymbol{p}_{j_1} \\ \vdots \\ \boldsymbol{p}_{j_M} \end{pmatrix} \longrightarrow \begin{pmatrix} \widetilde{\boldsymbol{p}}_{i_1} \\ \vdots \\ \widetilde{\boldsymbol{p}}_{i_N} \\ \boldsymbol{p}_{j_1} \\ \vdots \\ \boldsymbol{p}_{j_M} \end{pmatrix}$$

Accordingly, also the variance-covariance matrix of only the alignable objects contained in L is being updated:

$$\begin{pmatrix} \boldsymbol{C}_{i_1 i_1} & \cdots & \boldsymbol{C}_{i_1 i_N} & \boldsymbol{C}_{i_1 j_1} & \cdots & \boldsymbol{C}_{i_1 j_M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{i_N i_1} & \cdots & \boldsymbol{C}_{i_N i_N} & \boldsymbol{C}_{i_N j_1} & \cdots & \boldsymbol{C}_{i_N j_M} \\ \boldsymbol{C}_{j_1 i_1} & \cdots & \boldsymbol{C}_{j_1 i_N} & \boldsymbol{C}_{j_1 j_1} & \cdots & \boldsymbol{C}_{j_1 j_M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{j_M i_1} & \cdots & \boldsymbol{C}_{j_M i_N} & \boldsymbol{C}_{j_M j_1} & \cdots & \boldsymbol{C}_{j_M j_M} \end{pmatrix} \longrightarrow \begin{pmatrix} \widetilde{\boldsymbol{C}}_{i_1 i_1} & \cdots & \widetilde{\boldsymbol{C}}_{i_1 i_N} & \boldsymbol{C}_{i_1 j_1} & \cdots & \boldsymbol{C}_{i_1 j_M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{\boldsymbol{C}}_{i_N i_1} & \cdots & \widetilde{\boldsymbol{C}}_{i_N i_N} & \boldsymbol{C}_{i_N j_1} & \cdots & \boldsymbol{C}_{i_N j_M} \\ \boldsymbol{C}_{j_1 i_1} & \cdots & \boldsymbol{C}_{j_1 i_N} & \boldsymbol{C}_{j_1 j_1} & \cdots & \boldsymbol{C}_{j_1 j_M} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{j_M i_1} & \cdots & \boldsymbol{C}_{j_M i_N} & \boldsymbol{C}_{j_M j_1} & \cdots & \boldsymbol{C}_{j_M j_M} \end{pmatrix}$$

Given that the initial variance-covariance matrix was valid, the relation

$$-1 \;\leq\; (\boldsymbol{R}_{ij})_{kl} = \frac{(\boldsymbol{C}_{ij})_{kl}}{\sqrt{(\boldsymbol{C}_{ii})_{kk}\,(\boldsymbol{C}_{jj})_{ll}}} \;\leq\; 1, \qquad \forall\, i, j \in \{L \cup \bar{L}\}$$

has to be always fulfilled. Since $\left(\widetilde{\boldsymbol{C}}_{ii}\right)_{kk} \leq \left(\boldsymbol{C}_{ii}\right)_{kk}$ this cannot be guaranteed any more after the update for the case where $\{i \in L\} \wedge \{j \in \bar{L}\}$. This is a fundamental problem of the selective update of the variance-covariance matrix. As long as the list L is solely defined by the alignable objects contained in I, it cannot be ruled out a priori that one of the alignables included in L is significantly correlated to one of the alignables that were excluded from the update. However, if all alignable objects with significant correlation were added recursively, i.e. composing lists like $I \mapsto L$, $L \mapsto L'$, $L' \mapsto L''$, etc., one would end up with a list of all alignable objects again and nothing would be gained.

To overcome this problem, at least for the practical purpose presented here, a rather pragmatical approach is used. The idea behind it is the fact that the correlations between the various alignables reflect in principle only some geometrical properties of the system, i.e. the (corrected) relative positions and rotations with respect to each other. Since the geometry is fixed and the alignment corrections are considered to be relatively small, the correlations $(\boldsymbol{R}_{ij})_{kl}$ should be — in contrast to the $(\boldsymbol{C}_{ij})_{kl}$ — more or less constant, once they have built up after a few updates. Therefore it can be assumed that the crucial values are the $(\boldsymbol{R}_{ij})_{kl}$, used by the Kalman filter in the scaled and weighted form of $(\boldsymbol{C}_{ij})_{kl}$. By computing and storing the correlations rather than the covariances it should therefore be safe to update only a subsample of the full variance-covariance matrix. In addition it allows to check for consistency, as it can be checked at each update whether the absolute values of the updated correlations are indeed smaller than one.

4.4 Results

To prove the basic operability of the Kalman alignment algorithm, various tests on simplified setups have been carried out. All of these setups were chosen only to demonstrate the functionality of certain features of the algorithm, and are by no means realistic scenarios. Nevertheless, since all tests were carried out using the actual CMS Tracker geometry, it is demonstrated that the algorithm is able to cope with such a complex system, even considering realistic material effects.

Precision and Speed of Convergence

A simple setup was chosen to demonstrate the convergence of the algorithm. The TIB was misaligned on the level of *TIBStrings* (see Section 3.2), giving a total of 652 alignable objects. Misalignment was applied as shifts in the u-direction and rotations around the w-axis of the local coordinate system of the individual strings. The actual values for the shifts and rotations were drawn from Gaussian distributions with standard deviations of $\sigma_u = 100\ \mu$m and $\sigma_\gamma = 0.1$ mrad, respectively. The TPB and the TOB were not misaligned to serve as an ideal reference frame. Tracks from a sample of 50,000 fully simulated $Z \rightarrow \mu^+\mu^-$ events were used.

Scenario A In the first scenario, the outermost TPB layer and the innermost TOB layer are directly included as reference. Only tracks with at least six hits in the parts of the Tracker considered in this scenario are selected. All alignables are updated at once, i.e. update equations (4.3) and (4.4) are utilized and no track-based alignable selection is applied. Figure 4.3 shows the evolution of the r.m.s of the difference between the estimated and true parameters for the local u-coordinate and the rotation angle γ around the local w-axis. The solid line shows the case where all starting parameters are set to zero. For the dashed line they are drawn from a Gaussian distribution with standard deviations according to the simulated misalignment.

The precision of the estimated alignment parameters becomes better than the typical resolution of a silicon sensor after a few thousand processed tracks. For a further — but considerably slower — improvement more tracks have to be processed, ending up with a remaining misalignment of typically $\Delta u \approx 10\ \mu$m and $\Delta\gamma \approx 0.02$ mrad.

Scenario B In the second scenario, the measurements in TPB and TOB are used to calculate an independent track estimate that is then used for the alignment of the TIB. Only tracks with at least six hits in the combined TPB and TOB region (for the external track estimate) as well as four hits in the TIB (for the alignment) are selected. All alignables are updated at once, i.e. update equations (4.1) and (4.2) are utilized, and no track-based alignable selection is applied. Figure 4.4 shows the evolution of the r.m.s of the difference between the estimated and true parameters for the local u-coordinate and the rotation angle γ around the local w-axis. The solid line shows the case where all starting parameters are set to zero. For the dashed line they are drawn from a Gaussian distribution with standard deviations according to the simulated misalignment.

Even though less tracks are selected, the final alignment is better than in the case of Scenario A, ending up with a remaining misalignment of typically $\Delta u \approx 5$ μm and $\Delta\gamma \approx 0.02$ mrad. This is due to the fact that the external tracks contain very good positional information (due to the high precision of the TPB) and momentum resolution (due to the TOB's lever arm).

Scenario C The setup of Scenario C is identical to the setup of Scenario B. The only difference is the fact that at every alignment step the updated alignables are chosen accordingly to the selection algorithm described in Section 4.3. The metrical threshold value was set to $d_{\max} = 3$.

The resulting precision is the same as for the previous case, only that the computing time is just a small fraction. Whereas the computation for Scenarios A and B takes around three hours, for Scenario C only six minutes are needed. Figure 4.6 shows the timing for the case of Scenario C. As can be seen, the update of the parameters and the corresponding variance-covariance matrix takes about half of the time. The computation of the involved vectors and matrices ($\boldsymbol{f}$, $\boldsymbol{D}_p$, $\overline{\boldsymbol{G}}$, etc.) as well as retrieving and writing the parameters and their corresponding variance-covariance matrix from and to the memory (referred to as "IO") take both about a quarter of the time. The calculation of the track-based metrics is rather negligible.

Evolution of the Metrical Distances

The state of the track-based metrics is by definition completely dynamic, depending on the sample of tracks that is used. Figure 4.7 shows an example of such an evolution. In this case the metrical distances were computed for

Evolution of the RMS of the difference between the estimated and true parameters for the local u-coordinate and the rotation angle γ around the local w-axis for the three different Scenarios described in Section 4.4.

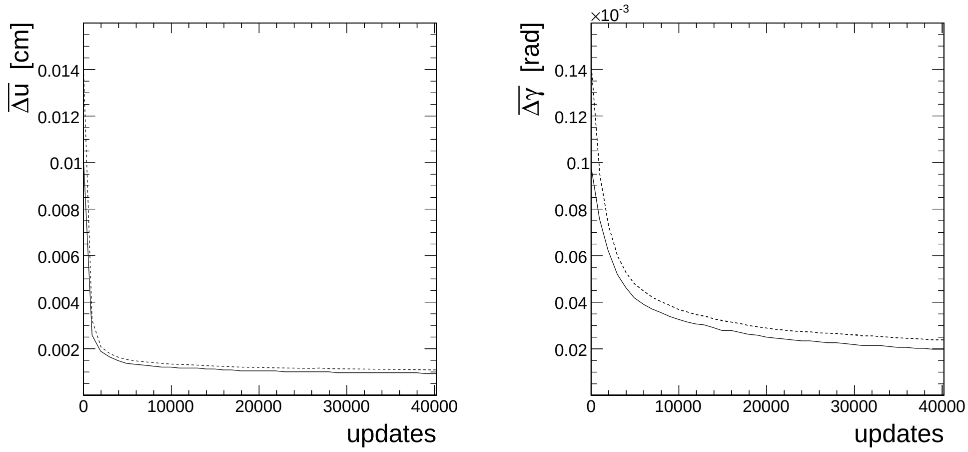


Figure 4.3: Scenario A.

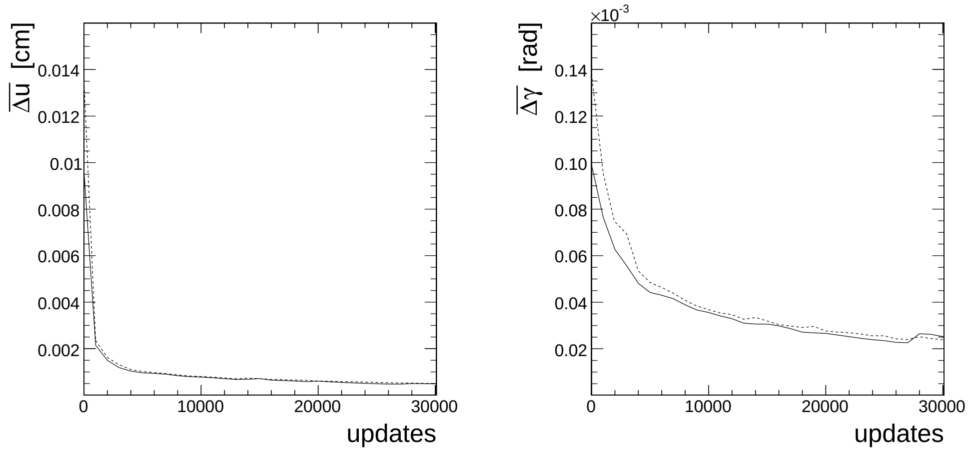


Figure 4.4: Scenario B.

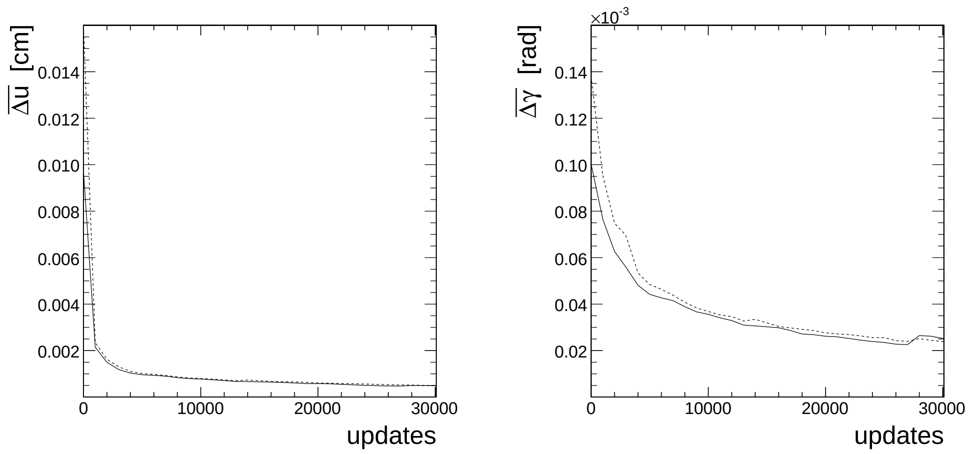


Figure 4.5: Scenario C.

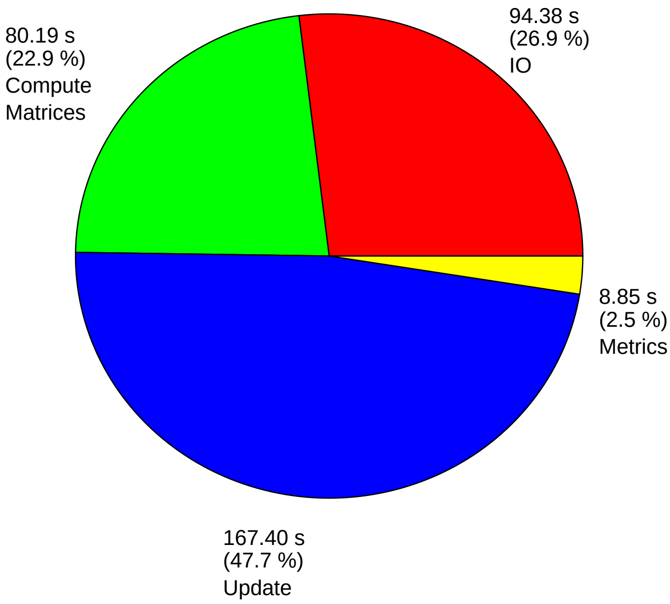


Figure 4.6: Timing for Scenario C.

the modules in the Tracker Pixel Barrel, using high-momentum tracks coming from the interaction point. The procedure was repeated several times, using different values for $d_{\max}$. By plotting the total number of entries versus the number of processed tracks for all these cases, the evolution of the populations of the individual metrical values can be extracted.

The left hand plot shows the situation during the first few updates. As expected, the bigger metrical values "develop" on top of the smaller ones, i.e. longer chains only show up once short chains have already been established. From the right hand plot one can see immediately that the process "saturates" as soon as a significant number of tracks is processed. This is due to the fact that at one point all important chains have been established, such that the processing of further tracks does not give any significant new information. Of course the distances also do not grow to infinity, in the sense that there exists a maximum metrical value d_∞ for a given setup. For the given example this value is about $d_\infty = 17$, even though before saturation sets in there are populations with even higher values, which then become

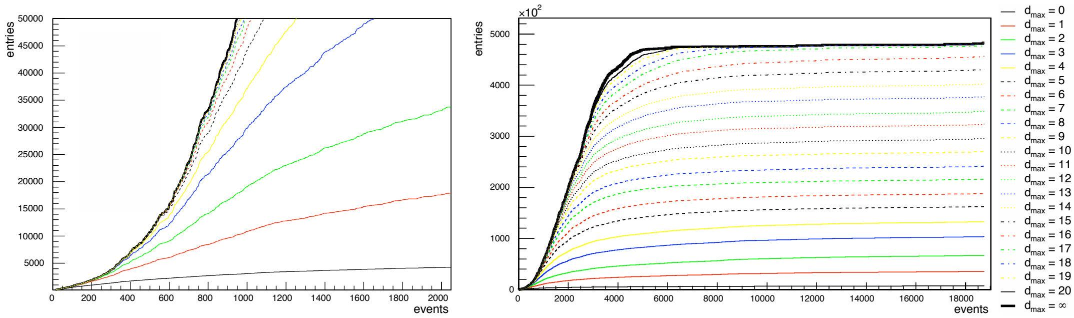


Figure 4.7: Evolution of the number of entries $d(i, j)$ of the track-based metrics for the modules of the Tracker Pixel Barrel computed for a sample of high-momentum collision tracks with different values of d_{max}.

obsolete when shorter chains come up.

Effects of Selective Updates

An important question is whether it is in fact possible to select all alignables with significant correlations by using only a track-based metrics as defined above. On the one hand, if the selection is too loose, i.e. if much more alignables are chosen than is required, the algorithm is slowed down needlessly. On the other hand, disregarding significant correlations between alignable objects would at least deteriorate the convergence, and maybe cause inconsistencies. Therefore, to check the validity of the approach, the evolution of the correlations in dependence on the metrical distances has been investigated in more detail.

A rather small setup of only 500 modules, contained within a narrow Z-slice of the TIB, was misaligned along the local u-coordinate ($\sigma_u = 100\ \mu$m). The system was aligned against the fixed TPB, serving as an ideal reference frame. The setup was chosen to be that small to allow for updating all modules at once at every step within a reasonable amount of time. The track-based metrics was calculated without using a threshold.

The absolute value of the correlations in dependence on the metrical distances between the two associated alignables after processing 10,000 tracks is shown in Figure 4.8. In this case even for rather large metrical distances such as $d =$

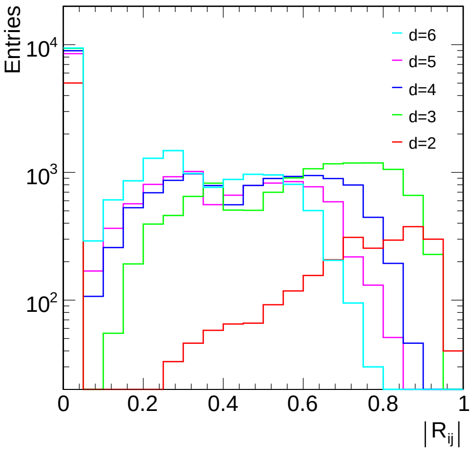


Figure 4.8: $\sqrt{\Delta V} = 0\,\mu$m

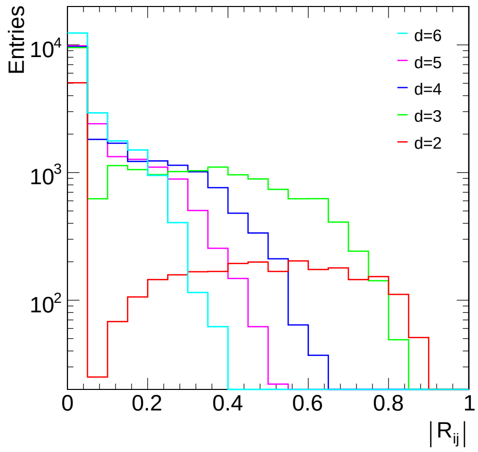


Figure 4.9: $\sqrt{\Delta V} = 50\,\mu$m

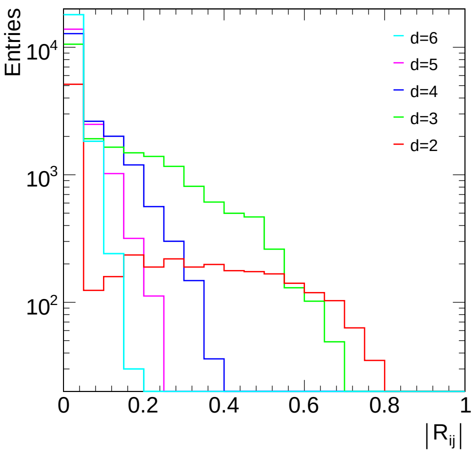


Figure 4.10: $\sqrt{\Delta V} = 100\,\mu$m

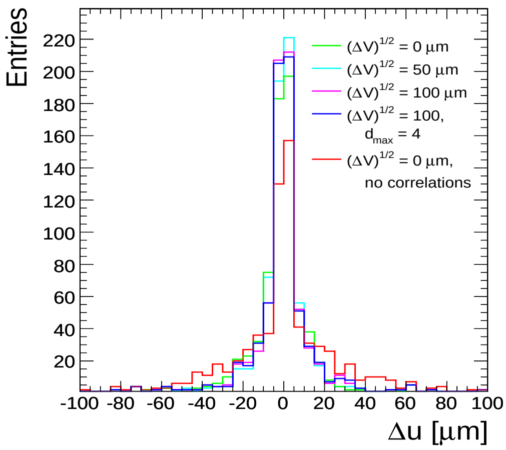


Figure 4.11: Final resolution.

6 significant correlations are present. However, by inflating the measurement variance-covariance matrix $\boldsymbol{V}$, i.e. by adding an artificial error ΔV to $\boldsymbol{V}$ such as

$$\boldsymbol{V} \longrightarrow \boldsymbol{V} + \Delta V \cdot \boldsymbol{I}, \tag{4.6}$$

more distant alignables get decoupled, as can be seen in Figures 4.9 and 4.10. In addition, this inflation of the measurement errors does not deteriorate the resulting precision of the alignment, as can be seen in Figure 4.11. When using an extra error of $\sqrt{\Delta V} = 100\,\mu$m the correlations between more distant alignables are already so small that a metrical threshold of $d_{\mathrm{max}} = 4$ has no influence on the final result anymore. The final result gets significantly worse only if the correlations are neglected entirely.

Chapter 5

Two-Body Decay Constraints

The full information of reconstructed trajectories is usually concentrated in a set of five parameters for further analysis. Here, a novel parameterization of the trajectories stemming from a two-body decay is presented, based on the kinematics of such decays. This approach allows not only a compact representation, but also a direct estimation of the decay parameters. Furthermore, this representation is suitable for using the trajectories in a track-based alignment algorithm, due to the combined information content of the measurements and the superimposed physical constraints.

5.1 Prerequisites

The most fundamental prerequisite for track reconstruction is the *track model* $\boldsymbol{f}$, which allows the representation of a set of measurements $\boldsymbol{m}$, belonging to a single track, by a set of parameters $\boldsymbol{q}$:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{q}) + \boldsymbol{\varepsilon}_m, \quad \mathsf{cov}(\boldsymbol{\varepsilon}_m) = \boldsymbol{V}_m. \tag{5.1}$$

$\boldsymbol{\varepsilon}_m$ is the stochastic deviation from the exact track model, containing both multiple scattering and measurement errors, and $\boldsymbol{V}_m$ is its variance-covariance matrix.

There exists a broad variety of possible sets of parameters [27] that in principle all come into question for the method descibed below. However, to simplify matters, it is assumed that the parameters $\boldsymbol{q}$ describe the local state of the trajectory at the surface of the first sensitive detector layer the particles traverse (see Figure 5.1). If the local coordinate system is Cartesian with axes (u, v, w), and the detector surface is defined by $w = 0$, a possible representation is $\boldsymbol{q} = (u, v, \mathrm{d}u/\mathrm{d}w, \mathrm{d}v/\mathrm{d}w, \kappa)^{\mathrm{T}}$, where κ is the curvature or the transverse curvature. The track parameters at the other sensitive layers can be determined by propagating the initial state $\boldsymbol{q}$ through the magnetic field.

In the case of vertexing, another relation is of fundamental importance. Here, it is referred to as *measurement equation*, since it provides the connection between the observable and the primary physical and geometrical properties of a trajectory. The measurement equation expresses $\boldsymbol{q}$ as a function of the particle vertex $\boldsymbol{v}$ and its momentum at the vertex $\boldsymbol{p}_v$:

$$\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{v}, \boldsymbol{p}_v). \tag{5.2}$$

Of course also quantities such as the magnetic field in the tracking volume and the mass of the particle are a necessary input to the track model and the measurement equation. Nevertheless, it can be assumed that they are known and remain the same for all tracks (of the same kind). They are therefore not treated as free parameters.

5.2 Parametrization

Consider the decay of a primary particle with mass M and momentum $\boldsymbol{p}$ into a particle-antiparticle pair. The momenta $\boldsymbol{p}^{\pm}_{\mathrm{c.m.s.}}$ of the secondary particles in the center-of-mass system of this two-body decay, i.e. the rest frame of the primary particle, can be be obtained using the relativistic energy-momentum

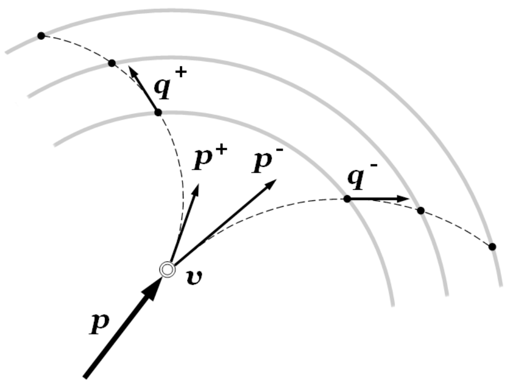


Figure 5.1: Schematic view of the decay. The dashed lines represent the trajectories, the detector layers are sketched by bold grey lines.

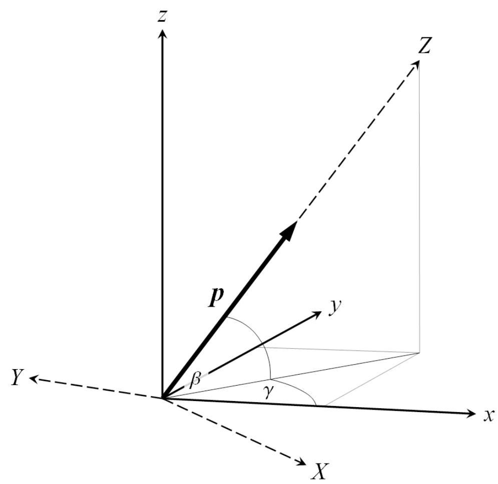


Figure 5.2: Relative position of the lab-frame (x, y, z) and the rest-frame (X, Y, Z) of the primary particle with momentum $\boldsymbol{p}$.

conservation law:

$$\boldsymbol{p}^{\pm}_{\text{c.m.s.}} = \pm m\sqrt{\alpha^2 - 1} \begin{pmatrix} \sin\theta\,\cos\phi \\ \sin\theta\,\sin\phi \\ \cos\theta \end{pmatrix}, \tag{5.3}$$

where m is the mass of the secondary particles and $\alpha = \frac{M}{2m}$.

At this point a convention is needed that defines the direction of the coordinate axes in the rest frame of the primary particle (X, Y, Z) with respect to the lab-frame (x, y, z), in order to provide a proper definition of the polar angle $\theta \in [0, \pi]$ and the azimuth angle $\phi \in [0, 2\pi)$. Here, the Z-axis was chosen to coincide with the direction of the primary particle momentum $\boldsymbol{p} = (p_x, p_y, p_z)^{\mathrm{T}}$ in the lab frame. The X- and Y-axis were chosen to coincide with the x- and y-axis after subsequent rotations around the z-axis by $\tan\gamma = p_y/p_x$ (clockwise) and the y-axis by $\tan\beta = p_z/p_{\mathrm{T}}$ (counter-clockwise, with $p_{\mathrm{T}}^2 = p_x^2 + p_y^2$). See Figure 5.2 for a schematic view of the relative positions of these frames.

By doing so, the momenta only have to be boosted along the Z-axis and properly rotated to obtain the lab-frame representation. The resulting *decay model* correlates the momenta of both secondary particles to a single set of *kinematical parameters* ($p = |\boldsymbol{p}|$):

$$\boldsymbol{p}^{\pm}(p_x, p_y, p_z, \theta, \phi, M) = \begin{pmatrix} \dfrac{p_x p_z}{p_\mathrm{T} p} & -\dfrac{p_y}{p_\mathrm{T}} & \dfrac{p_x}{p} \\ \dfrac{p_y p_z}{p_\mathrm{T} p} & \dfrac{p_x}{p_\mathrm{T}} & \dfrac{p_y}{p} \\ -\dfrac{p_\mathrm{T}}{p} & 0 & \dfrac{p_z}{p} \end{pmatrix} \begin{pmatrix} \pm m\sqrt{\alpha^2 - 1}\,\sin\theta\,\cos\phi \\ \pm m\sqrt{\alpha^2 - 1}\,\sin\theta\,\sin\phi \\ \frac{p}{2} \pm \frac{1}{2}\sqrt{\frac{\alpha^2-1}{\alpha^2}\,(p^2 + M^2)}\,\cos\theta \end{pmatrix}. \tag{5.4}$$

Note that the superscripts + and − refer to the signs in Equation (5.3) and not to the charge of the daughter particles.

Considering what was said above about the track model and the measurement equation it follows that in the case of two-body decays of the form $\mathrm{X}^0 \rightarrow \mathrm{Y}^+\mathrm{Y}^-$, the use of the decay model allows the representation of the corresponding trajectories by the following 9 parameters:

1. the position of the decay vertex $\boldsymbol{v} = (v_x, v_y, v_z)^\mathrm{T}$,
2. the momentum of the primary particle $\boldsymbol{p} = (p_x, p_y, p_z)^\mathrm{T}$ in the lab-frame,
3. the polar angle θ and the azimuth angle ϕ defining the direction of the secondary particles in the rest-frame of the primary particle, and
4. the mass M of the primary particle.

5.3 Representation

The full set of parameters defining the kinematic properties are from now on denoted by $\boldsymbol{z} = (p_x, p_y, p_z, \theta, \phi, M)$. The information passed for instance to an alignment algorithm then takes the following form:

$$\boldsymbol{m} = \begin{pmatrix} \boldsymbol{m}^+ \\ \boldsymbol{m}^- \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}^+(\boldsymbol{v},\boldsymbol{z}) + \boldsymbol{\varepsilon}_m^+ \\ \boldsymbol{f}^-(\boldsymbol{v},\boldsymbol{z}) + \boldsymbol{\varepsilon}_m^- \end{pmatrix}, \qquad \boldsymbol{V}_m = \begin{pmatrix} \boldsymbol{V}_m^+ & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{V}_m^- \end{pmatrix},$$

$$\boldsymbol{D} = \frac{\partial \boldsymbol{f}}{\partial(\boldsymbol{v},\boldsymbol{z})} = \begin{pmatrix} \dfrac{\partial \boldsymbol{f}^+}{\partial \boldsymbol{q}^+}\cdot\dfrac{\partial \boldsymbol{q}^+}{\partial \boldsymbol{v}} & \dfrac{\partial \boldsymbol{f}^+}{\partial \boldsymbol{q}^+}\cdot\dfrac{\partial \boldsymbol{q}^+}{\partial \boldsymbol{p}^+}\cdot\dfrac{\partial \boldsymbol{p}^+}{\partial \boldsymbol{z}} \\ & \\ \dfrac{\partial \boldsymbol{f}^-}{\partial \boldsymbol{q}^-}\cdot\dfrac{\partial \boldsymbol{q}^-}{\partial \boldsymbol{v}} & \dfrac{\partial \boldsymbol{f}^+}{\partial \boldsymbol{q}^-}\cdot\dfrac{\partial \boldsymbol{q}^-}{\partial \boldsymbol{p}^-}\cdot\dfrac{\partial \boldsymbol{p}^-}{\partial \boldsymbol{z}} \end{pmatrix}.$$

Here $\boldsymbol{m}^\pm$ and $\boldsymbol{V}^\pm$ are the measurements and the corresponding covariance matrices of the single trajectories. In the derivative matrix $\boldsymbol{D}$ the chain rule is used, combining the Jacobians $\partial \boldsymbol{f}^\pm/\partial \boldsymbol{q}$ with the Jacobians of the measurement equation $\boldsymbol{q}(\boldsymbol{v},\boldsymbol{p}_v)$ and the decay model $\boldsymbol{p}^\pm(\boldsymbol{z})$.

5.4 Estimation

5.4.1 General Least-Squares Solution

The *estimated* single-track parameters $\boldsymbol{q}_{est}^\pm$ are related to the true vertex position $\boldsymbol{v}$ and the true momentum of the particle at the vertex $\boldsymbol{p}^\pm$ via the measurement equation $\boldsymbol{q}(\boldsymbol{v},\boldsymbol{p}_v)$:

$$\begin{pmatrix} \boldsymbol{q}_{est}^+ \\ \boldsymbol{q}_{est}^- \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}(\boldsymbol{v},\boldsymbol{p}^+) \\ \boldsymbol{q}(\boldsymbol{v},\boldsymbol{p}^-) \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}^+ \\ \boldsymbol{\varepsilon}^- \end{pmatrix}, \quad \mathsf{cov}(\boldsymbol{\varepsilon}^\pm) = \boldsymbol{V}^\pm = (\boldsymbol{W}^\pm)^{-1}. \tag{5.5}$$

Using a first-order Taylor expansion at the expansion point $\boldsymbol{v}_0 = (v_{0x}, v_{0y}, v_{0z})^{\mathrm{T}}$ and $\boldsymbol{z}_0 = (p_{0x}, p_{0y}, p_{0z}, \theta_0, \phi_0, M_0)^{\mathrm{T}}$ this relation can be linearized:

$$\begin{pmatrix} \boldsymbol{q}_{est}^+ \\ \boldsymbol{q}_{est}^- \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}^+ \\ \boldsymbol{c}^- \end{pmatrix} + \begin{pmatrix} \boldsymbol{A}^+ & \boldsymbol{B}^+ & \boldsymbol{0} \\ \boldsymbol{A}^- & \boldsymbol{0} & \boldsymbol{B}^- \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{p}^+ \\ \boldsymbol{p}^- \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}^+ \\ \boldsymbol{\varepsilon}^- \end{pmatrix}, \tag{5.6}$$

where

$$\boldsymbol{A}^{\pm} = \partial \boldsymbol{q}/\partial \boldsymbol{v}\Big|_{\boldsymbol{v}_0, \boldsymbol{p}_0^{\pm}}, \qquad \boldsymbol{B}^{\pm} = \partial \boldsymbol{q}/\partial \boldsymbol{p}\Big|_{\boldsymbol{v}_0, \boldsymbol{p}_0^{\pm}},$$
$$\boldsymbol{c}^{\pm} = \boldsymbol{q}(\boldsymbol{v}_0, \boldsymbol{p}_0^{\pm}) - \boldsymbol{A}^{\pm}\boldsymbol{v}_0 - \boldsymbol{B}^{\pm}\boldsymbol{p}_0^{\pm}, \qquad \boldsymbol{p}_0^{\pm} = \boldsymbol{p}^{\pm}(\boldsymbol{z}_0).$$

It should be noted that this linearization is the standard approach for vertex fitting. For this reason, the computation of the matrices $\boldsymbol{A}^{\pm}$ and $\boldsymbol{B}^{\pm}$ is always provided in an actual implementation.

Since the momenta of the secondary particles depend on the same set of parameters, the linearization of the decay model (5.4),

$$\boldsymbol{p}^{\pm}(\boldsymbol{z}) = \boldsymbol{d}^{\pm} + \boldsymbol{C}^{\pm}\boldsymbol{z}, \tag{5.7}$$

with

$$\boldsymbol{C}^{\pm} = \partial \boldsymbol{p}^{\pm}/\partial \boldsymbol{z}\Big|_{\boldsymbol{z}_0} \quad \text{and} \quad \boldsymbol{d}^{\pm} = \boldsymbol{p}_0^{\pm} - \boldsymbol{C}^{\pm}\boldsymbol{z}_0,$$

can be used to rewrite the equations as:

$$\begin{pmatrix} \boldsymbol{q}_{est}^{+} - \boldsymbol{c}^{+} - \boldsymbol{B}^{+}\boldsymbol{d}^{+} \\ \boldsymbol{q}_{est}^{-} - \boldsymbol{c}^{-} - \boldsymbol{B}^{-}\boldsymbol{d}^{-} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^{+} & \boldsymbol{B}^{+}\boldsymbol{C}^{+} \\ \boldsymbol{A}^{-} & \boldsymbol{B}^{-}\boldsymbol{C}^{-} \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{z} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}^{+} \\ \boldsymbol{\varepsilon}^{-} \end{pmatrix}. \tag{5.8}$$

These 10 linear equations allow to perform an estimation of the vertex position (3 parameters) and of the kinematical quantities (6 parameters). It should be noted that this approach avoids the use of Lagrange multipliers; instead the kinematical constraints are enforced by incorporating the kinematics into the decay model. The estimates are obtained by a straightforward minimization of the objective function

$$\chi^2(\boldsymbol{v}, \boldsymbol{z}) = \sum_{i=\pm} \Big(\boldsymbol{r}^i - \boldsymbol{s}^i(\boldsymbol{v}, \boldsymbol{z})\Big)^{\mathrm{T}} \boldsymbol{W}^i \Big(\boldsymbol{r}^i - \boldsymbol{s}^i(\boldsymbol{v}, \boldsymbol{z})\Big) \longrightarrow \min, \tag{5.9}$$

with

$$\boldsymbol{r}^{\pm} = \boldsymbol{q}_{est}^{\pm} - \boldsymbol{c}^{+} - \boldsymbol{B}^{\pm}\boldsymbol{d}^{\pm}, \quad \boldsymbol{s}^{\pm}(\boldsymbol{v}, \boldsymbol{z}) = \boldsymbol{A}^{\pm}\boldsymbol{v} + \boldsymbol{B}^{\pm}\boldsymbol{C}^{\pm}\boldsymbol{z}.$$

The solution of the minimization problem is given by [28]:

$$\begin{pmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\boldsymbol{z}} \end{pmatrix} = \operatorname{argmin} \chi^2(\boldsymbol{v}, \boldsymbol{z}) = (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{r}, \tag{5.10}$$

with

$$\boldsymbol{r} = \begin{pmatrix} \boldsymbol{r}^+ \\ \boldsymbol{r}^- \end{pmatrix}, \quad \boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}^+ & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}^- \end{pmatrix}, \quad \boldsymbol{J} = \begin{pmatrix} \boldsymbol{A}^+ & \boldsymbol{B}^+\boldsymbol{C}^+ \\ \boldsymbol{A}^- & \boldsymbol{B}^-\boldsymbol{C}^- \end{pmatrix}.$$

5.4.2 Provision for the Beam Profile and the Mass Constraint

The beam profile $\boldsymbol{x}_{\mathrm{bp}}$ can — as far as it is known and the primary particle is sufficiently short-lived — be included as a virtual measurement:

$$\boldsymbol{x}_{\mathrm{bp}} = \boldsymbol{v} + \boldsymbol{\varepsilon}_{\mathrm{bp}} = \boldsymbol{J}_{\mathrm{bp}} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{z} \end{pmatrix} + \boldsymbol{\varepsilon}_{\mathrm{bp}}, \quad \mathsf{cov}(\boldsymbol{\varepsilon}_{\mathrm{bp}}) = \boldsymbol{V}_{\mathrm{bp}} = (\boldsymbol{W}_{\mathrm{bp}})^{-1}, \tag{5.11}$$

with

$$\boldsymbol{J}_{\mathrm{bp}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

This inclusion of the beam profile to constrain the result of the estimate within a certain region is especially important if the primary particle is at rest or if the directions of the secondary particles happen to be (anti-)parallel to the primary momentum. In this cases, the secondary particles move into exactly opposite directions, such that the geometrical intersection of their trajectories is not well defined.

In addition, if a hypothesis for the mass of the primary particle $\widehat{M} \pm \sigma_{\widehat{M}}$ is available, a virtual measurement can be introduced to enforce a mass constraint:

$$\widehat{M} = M + \epsilon_{\widehat{M}} = \boldsymbol{J}_M \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{z} \end{pmatrix} + \epsilon_{\widehat{M}}, \quad \mathsf{var}(\epsilon_{\widehat{M}}) = \sigma^2_{\widehat{M}}, \tag{5.12}$$

with

$$\boldsymbol{J}_M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Analogous to the previous subsection the estimate turns out to be:

$$\begin{pmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\boldsymbol{z}} \end{pmatrix} = \left(\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{J} + \boldsymbol{J}_{\mathrm{bp}}^{\mathrm{T}} \boldsymbol{W}_{\mathrm{bp}} \boldsymbol{J}_{\mathrm{bp}} + \frac{1}{\sigma_{\widehat{M}}^2} \boldsymbol{J}_M^{\mathrm{T}} \boldsymbol{J}_M \right)^{-1} \times \left(\boldsymbol{J}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{r} + \boldsymbol{J}_{\mathrm{bp}}^{\mathrm{T}} \boldsymbol{W}_{\mathrm{bp}} \boldsymbol{x}_{\mathrm{bp}} + \boldsymbol{J}_M^{\mathrm{T}} \frac{\widehat{M}}{\sigma_{\widehat{M}}^2} \right). \tag{5.13}$$

5.4.3 Robustification

Due to the block-diagonal form of the full weight matrix $\boldsymbol{W}$, the least-squares approach can be robustified by applying an iterative M-estimator procedure [29, 30].

Technically this is done by rewriting the objective function by a diagonalization of the weight matrices:

$$\chi^2(\boldsymbol{v}, \boldsymbol{z}) = \sum_{i=\pm} \left(\boldsymbol{\rho}^i - \boldsymbol{\sigma}^i(\boldsymbol{v}, \boldsymbol{z}) \right)^{\mathrm{T}} \boldsymbol{\Omega}^i \left(\boldsymbol{\rho}^i - \boldsymbol{\sigma}^i(\boldsymbol{v}, \boldsymbol{z}) \right), \tag{5.14}$$

with

$$\boldsymbol{\rho}^{\pm} = \boldsymbol{U}^{\pm} \boldsymbol{r}^{\pm}, \quad \boldsymbol{\sigma}^{\pm}(\boldsymbol{v}, \boldsymbol{z}) = \boldsymbol{U}^{\pm} \boldsymbol{s}^{\pm}(\boldsymbol{v}, \boldsymbol{z}),$$
$$\boldsymbol{W}^{\pm} = \boldsymbol{U}^{\pm} \boldsymbol{\Omega}^{\pm} \boldsymbol{U}^{\pm\mathrm{T}}, \quad \boldsymbol{\Omega}^{\pm} = \mathrm{diag}\left(\Omega_1^{\pm}, \ldots, \Omega_5^{\pm} \right).$$

After an initial estimation step (analogous to Equation (5.10)), yielding the estimates $\tilde{\boldsymbol{v}}_0$ and $\tilde{\boldsymbol{z}}_0$, the entries of the diagonalized weight matrices $\boldsymbol{\Omega}^{\pm}$ are downscaled and the estimation is repeated:

$$\Omega_k^{\pm} \longrightarrow \omega_{N,k}^{\pm} = \begin{cases} \Omega_k^{\pm} & , \text{ if } |\delta_{N,k}^{\pm}| \leq \dfrac{R}{\sqrt{\Omega_k^{\pm}}}. \\[2ex] \dfrac{R\sqrt{\Omega_k^{\pm}}}{|\delta_{N,k}^{\pm}|}, & \text{ if } |\delta_{N,k}^{\pm}| > \dfrac{R}{\sqrt{\Omega_k^{\pm}}}. \end{cases} \tag{5.15}$$

Here, N is the iteration cycle, R is the robustification constant (with a typical value between 1 and 2) and $\delta_{N,k}^{\pm} = \rho_k^{\pm} - \sigma_k^{\pm}(\tilde{\boldsymbol{v}}_{N-1}, \tilde{\boldsymbol{z}}_{N-1})$, where $\tilde{\boldsymbol{v}}_{N-1}$ and $\tilde{\boldsymbol{z}}_{N-1}$ are the estimates from the previous iteration step. The iteration stops if the improvement of the objective function becomes negligible, i.e. if $\Delta\chi^2 =$

$|\chi^2(\tilde{\boldsymbol{v}}_N, \tilde{\boldsymbol{z}}_N) - \chi^2(\tilde{\boldsymbol{v}}_{N-1}, \tilde{\boldsymbol{z}}_{N-1})| < \epsilon_{\chi^2}$.

This procedure is also possible in the presence of the mass constraint and the inclusion of the beam profile, since the corresponding virtual measurements are independent from each other and the residuals $\boldsymbol{r}^{\pm}$.

5.5 Two-Body Decay Constraints in Tracking

The concept described above has been implemented in the CMS reconstruction and analysis framework CMSSW [14]. The decays of primary particles into muon-antimuon pairs and the corresponding signal in the Tracker [4, 5] have been fully simulated. Two samples, one with 50,000 Z-bosons and another one with 25,000 J/Ψ-mesons as primary particles, have been produced to allow for a comparison between the two cases of loose and tight mass constraint. The primary momenta were uniformly distributed between 50 GeV and 100 GeV in the Z-sample and between 100 GeV and 150 GeV in the J/Ψ-sample, resulting in a rather broad spectrum of secondary momenta from the region of a few GeV up to the kinematic limit in both cases. The positions of the decay vertices were generated randomly by drawing from Gaussians with variances according to the expected beam profile ($\sigma_X = \sigma_Y = 15\,\mu$m, $\sigma_Z = 5.3$ cm). The resulting tracks covered the whole sensitive η-range of the Tracker.

For the reconstruction of the tracks the standard algorithms were utilized [20, 21]. The resulting estimates on the track parameters were used for a geometrical vertex fit to obtain $\boldsymbol{v}_0$ and to compute the expansion point $\boldsymbol{z}_0 = (\boldsymbol{p}_0, \theta_0, \phi_0, M_0)^{\mathrm{T}}$ for the susbsequent kinematical fit. The provisional momentum of the primary particle $\boldsymbol{p}_0$, estimated as the sum of the track momenta at their point of closest approach to $\boldsymbol{v}_0$, was also used to obtain a first guess of the parameters defining the Lorentz transformation from the lab-frame to the rest-frame of the two-body decay. By transfoming (at least one of) the single-track momenta into the rest-frame, the parameters θ_0 and ϕ_0 were computed. The invariant mass of the tracks was used for the parameter M_0.

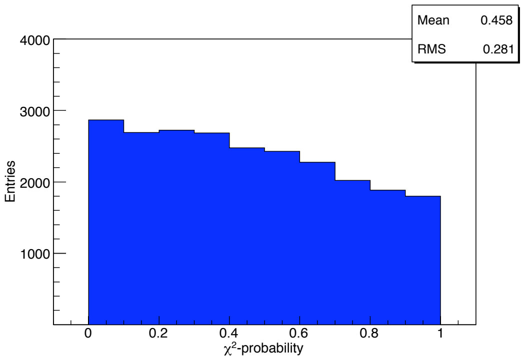


Figure 5.3: χ^2-probability of the kinematical fit (including the beam pofile and mass constraint).

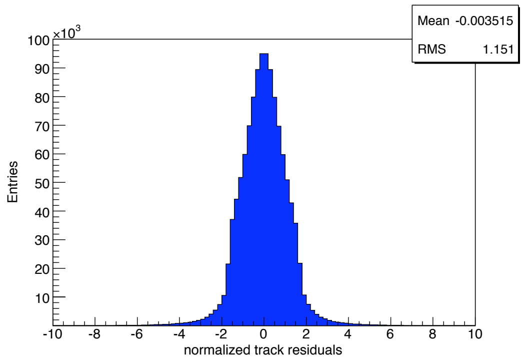


Figure 5.4: Normalized residuals of the re-propagated trajectories.

In all cases the robustified procedure (with $R = 1$ and $\epsilon_{\chi^2} = 10^{-2}$), including the proper mass constraint and the beam profile, was used. The results from the kinematical fit as well as the results from the standard reconstruction (a least-squares vertex fitter implemented in the Kalman filter formalism using the beam profile as external prediction) have been compared to the simulated truth.

5.5.1 Test of functionality

The basic functionality has been tested on the $Z \to \mu^+\mu^-$ sample. Figure 5.3 shows the χ^2-probability distribution of the estimates, or more strictly speaking, the distribution of $1 - F_{n=5}\left(\chi^2(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{z}})\right)$. $F_n(x)$ is the cumulative χ^2-distribution function with n degrees of freedom, and $\tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{z}}$ are the final estimates of the iterative M-estimator procedure. The resulting flat, almost uniform distribution is a strong indication that, despite the high non-linearity of the decay-model and the need of a twofold linearization step, the linear model is appropriate to parameterize and estimate the properties of two-body decays.

Furthermore, the proper representation of both single-tracks through the proposed set of parameters is demonstrated in Figure 5.4, where the standardized residuals between the observed measurements $\boldsymbol{m}$ and the reference trajectory $\boldsymbol{f}(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{z}})$ are shown, i.e. the distribution of $(m_i - f_i)/\sqrt{V_{ii}}$. The highly non-gaussian shape of the distribution is mostly due to the fact that the measurements of the local v-coordinate, i.e. along the direction of the strip of the sensitive modules, are uniformly distributed.

5.5.2 Impact of the Mass Constraint

Tables 5.1 and 5.2 show the results for the decays of J/Ψ and Z. These two examples clearly demonstrate the influence of the particle width on the precision of the estimated parameters. Whereas the mass of the J/Ψ is precisely defined ($\sigma_{J/\Psi} = 91.0$ keV), the width of the Z is much larger ($\sigma_Z = 2.5$ GeV). As can be seen, this distinct constraint restricts not only the resulting value of the fitted mass, but also improves the estimate as a whole, since the mass is a crucial parameter for the kinematics.

5.5.3 Performance in the Presence of Misalignment

In the case of a misaligned Tracker, the benefit from exploiting the kinematics becomes greater, even in the case of weak mass constraints. Table 5.3 shows the gain of precision compared to the standard reconstruction in the presence of misalignment as expected at the start-up phase of the CMS experiment [31, 32]. Though the expected misalignment is accounted for in the standard track reconstruction by increased position errors of the measurements, the kinematical fit improves the estimate still by approximately 20%.

5.6 Two-Body Decay Constraints for Alignment

To prove that two-body decay constraints are indeed useful for alignment purposes, a very simple scenario has been studied. The information is passed to the alignment algorithm in the form described in Section 5.3. In this way, only the interpretation of the measurement vector $\boldsymbol{m}$, the variance-covariance

	σ_{SR} [μm]	σ_{KF} [μm]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δv_x	25.89	13.36	0.52
Δv_y	26.91	13.33	0.50
Δv_z	119.0	139.0	1.17

	σ_{SR} [GeV/c]	σ_{KF} [GeV/c]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δp_x	1.034	0.481	0.47
Δp_y	1.024	0.516	0.50
Δp_z	2.206	1.025	0.46

	σ_{SR} [GeV/c^2]	σ_{KF} [GeV/c^2]
Δm	0.066	$9.074 \cdot 10^{-5}$

Table 5.1: Comparison of the results from the kinematical fit (KF) and the standard reconstruction (SR) for a sample of 25,000 simulated $J/\Psi \rightarrow \mu^+\mu^-$ decays (no misalignment applied). The table shows the RMS errors of the estimated quantities w.r.t. to their true values.

	σ_{SR} [μm]	σ_{KF} [μm]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δv_x	12.15	11.04	0.91
Δv_y	12.41	11.22	0.90
Δv_z	33.06	35.65	1.08

	σ_{SR} [GeV/c]	σ_{KF} [GeV/c]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δp_x	0.762	0.701	0.92
Δp_y	0.737	0.679	0.92
Δp_z	1.003	0.934	0.93

	σ_{SR} [GeV/c^2]	σ_{KF} [GeV/c^2]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δm	0.940	0.858	0.91

Table 5.2: Comparison of the results from the kinematical fit (KF) and the standard reconstruction (SR) for a sample of 50,000 simulated $Z \rightarrow \mu^+\mu^-$ decays (no misalignment applied). The table shows the RMS errors of the estimated quantities w.r.t. to their true values.

	σ_{SR} [μm]	σ_{KF} [μm]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δv_x	22.76	16.92	0.74
Δv_y	23.66	17.34	0.73
Δv_z	76.93	82.93	1.08

	σ_{SR} [GeV/c]	σ_{KF} [GeV/c]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δp_x	3.021	2.426	0.80
Δp_y	2.859	2.280	0.80
Δp_z	2.795	2.158	0.77

	σ_{SR} [GeV/c^2]	σ_{KF} [GeV/c^2]	$\sigma_{\mathrm{KF}}/\sigma_{\mathrm{SR}}$
Δm	2.860	2.265	0.79

Table 5.3: Comparison of the results from the kinematical fit (KF) and the standard reconstruction (SR) for a sample of 50,000 simulated $Z \rightarrow \mu^+\mu^-$ decays (with applied misalignment). The table shows the RMS errors of the estimated quantities w.r.t. to their true values.

matrix $\boldsymbol{V}$, the derivative matrix $\boldsymbol{D}_q$ and the track-parameters $\boldsymbol{q}$ is changed, such that the same formalism as for usual track-based alignment can be used.

The scenario used here is by no means realistic. The setup is only meant to provide a framework for a proof of concept. The TPB was misaligned on the level of TPBLadders (see Section 3.2), resulting in a total of 96 individual ladders to align. Misalignment was applied as shifts in the u-direction of the individual ladders. The actual values for the shifts were drawn from a Gaussian distribution with a standard deviation of $\sigma_u = 100\,\mu$m. The remaining Tracker was not misaligned, but was used to compute external track estimates. Tracks from a sample of 5,000 fully simulated $Z \rightarrow \mu^+\mu^-$ events were used.

For the sake of comparability and since the number of parameters is small, a straightforward KAA approach according to Equations (4.1) and (4.2) was chosen, without selective updates or any weighting mechanisms. Figure 5.5 shows the evolution of the RMS of the difference between the estimated and true parameters for the local u-coordinate, using ordinary single tracks (left) and the two-body decay constraint (right). For the latter, the external track estimates were used to compute an independent estimate for the vertex

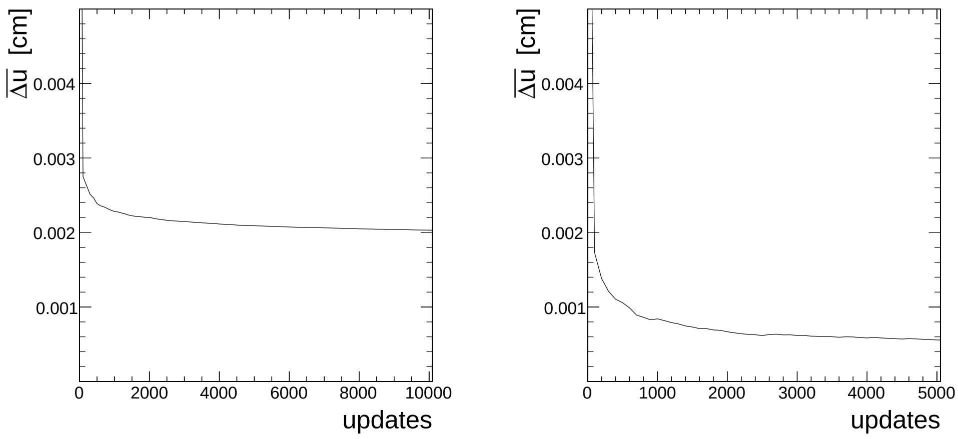


Figure 5.5: Evolution of the RMS of the difference between the estimated and true parameters for the local u-coordinate, using ordinary single tracks (left) and the two-body decay constraint (right).

position and the decay parameters, including already the beam profile and mass constraints. In the case of single tracks the number of updates is of course twice as large, as the alignment algorithm is called for every single track.

When processing single tracks, the algorithm seems to get stuck in a local χ^2-minimum. The information coming from the single tracks is not conclusive enough to obtain an alignment precision below the resolution of the pixel modules, resulting in a final precision of approximately 20 μm. Using the two-body decay constraints, this value is substantially reduced to approximately 6 μm.

Chapter 6

Full-scale CMS Tracker Alignment

The final objective is to prove the applicability of the Kalman Alignment Algorithm to the full-scale Tracker under start-up conditions, given the available resources in terms of computing power and virtual memory. This has been done in the course of the *CMS Computing Software and Analysis challenge* (CSA08) [33] carried out around May 2008. Its goal has been to test — in the manner of a dress rehearsal — the full scope of offline data handling and analysis activities which will be needed for the first months of real data-taking. It constitutes the first full-scale challenge with large statistics under the conditions expected at LHC start-up.

6.1 Available Datasets

Realistic start-up scenarios also need data samples containing event signatures and rates typical for the conditions during early data-taking, defined primarily by the LHC commissioning sequence as shown in Table 6.1. Each line is expected to last approximately one week, with up to three days allocated to data-taking by the experiments after machine setup. Two representative operations scenarios were chosen for the purposes of CSA08: The 43×43

bunches	$\beta^\star$ [m]	luminosity [$\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$]	events per bunch crossing	collision rate [MHz]
1×1	11	1.6×10^{27}	low	0.00009
43×43	11	7×10^{28}	low	0.004
43×43	11	1.2×10^{30}	0.14	0.06
43×43	2	6.1×10^{30}	0.76	0.34
156×156	2	2.2×10^{31}	0.76	1.2
156×156	2	1.2×10^{32}	3.9	6

Table 6.1: Planned LHC commissioning sequences and approximate luminosities, the average number of events per bunch crossing and the inelastic collision rates.

configuration with a luminosity of $\mathcal{L} = 2\times10^{30}$ cm^{-2} s^{-1} and the 156×156 configuration with a luminosity of $\mathcal{L} = 2\times10^{31}$ cm^{-2} s^{-1}. Data samples for the challenge were created for both scenarios, called S43 for the first scenario and S156 for the second. It takes six days at 100% efficiency to accumulate 1 pb^{-1} at $\mathcal{L} = 2\times10^{30}$ cm^{-2} s^{-1} and 10 pb^{-1} at $\mathcal{L} = 2\times10^{31}$ cm^{-2} s^{-1} (or a month at 20% duty cycle), resulting in a number of approximately 1.5×10^{8} recorded events in both cases. This may be more than the actual time available to accumulate data at these configurations, but the scenarios are meant to represent the average over several machine configurations.

The actual data samples were produced using a full detector simulation, with a center-of-mass energy of 10 TeV and a magnetic field strength of 4 T. Zero suppression was applied to the readout of all detectors. No pile-up was included because pile-up plays no significant role in either the S43 or S156 scenarios. A realistic event selection based on trigger bits was used. On top of this, the following realistic AlCaReco producers were used:

- Skims for the physics signals $Z \to \mu^+\mu^-$, $J/\Psi \to \mu^+\mu^-$, and $\Upsilon \to \mu^+\mu^-$ have been produced.
- Two additional selectors were used for collision data. The first selects isolated tracks in the tracker and is targeted mainly for selection from minimum bias events which are abundant at early LHC start-up. The

second selects isolated muons regardless of other activity in the tracker and is targeted for increasing luminosity.

- Special producers are used for non-collision data, working only in special runs. One selects muons from the beam halo when LHC circulates beam(s) without collisions, another one selects cosmic muons during cosmic runs, and the last one selects events from the laser alignment system.

6.2 Available Computing and Software Resources

Alignment and calibration related computational tasks will be carried out at a dedicated facility, the CERN Analysis Facility (CAF) [34, 35, 36]. A particular advantage of the CAF in this respect is the access to the full raw data recorded by the experiment, which is needed by the AlCaReco producers.

The alignment studies presented in this chapter were all performed at the CAF, as will be the case for the alignment runs with real data. This also means that the requirements of all alignment strategies have to comply with the available computing resources. This is especially relevant with respect to the available memory, since the CAF computing nodes are commercially obtainable cores fitted with a typical amount of RAM (usually about 2 GB per core), whereas the KAA is rather costly regarding memory, due to the storage of a potentially vast amount of correlation matrices.

In addition, the handling, submission and monitoring of all jobs has been done with the help of a production system. It is an adaption of the *MillePede Production System* (MPS) [37], a collection of tools dedicated for executing tracker alignment runs with MillePede in a production environment. It is based on the functionalities of the CMS Alignment Software and an underlying batch system, which is presently LSF [38]. The general workflow of a single alignment run for the KAA as realized with this production system

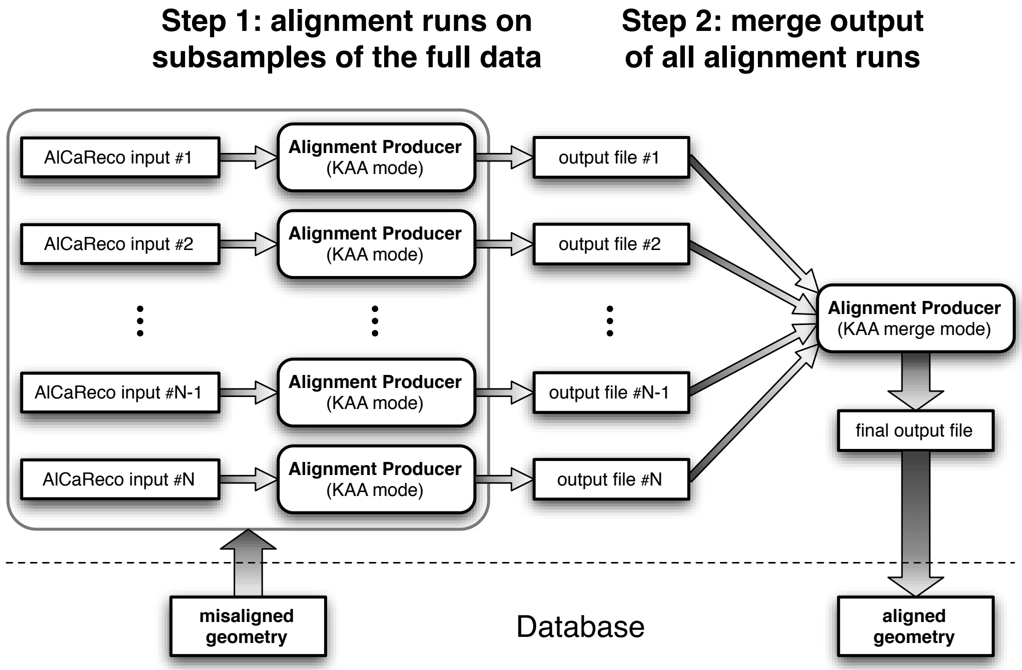


Figure 6.1: Workflow of the MPS adapted for the KAA.

is shown in Figure 6.1. Additional scripts for automatizing the execution of several consecutive alignment runs have been developed.

6.3 Full-Tracker Alignment Scenarios

For the purposes of the CSA08, misalignments and miscalibrations have been applied as expected before collisions. The so-called *TrackerSurveyLASCosmics* scenario [31, 32] was used to simulate the misalignment of the Tracker for both the S43 and S156 exercise. This scenario reflects the remaining tracker misalignment after inclusion of the data from survey measurements, from the Laser Alignment System and from track-based alignment with cosmic muons.

The goal is to carry out alignment on the level of individual modules to achieve the best possible results. However, the large amount of individual modules together with the available computing resources pose some serious problems for the KAA. The attempt to align the full Tracker on this level at once would fail for two reasons. First, due to the close proximity of the pixel modules to the interaction point and the therefore large geometric coverage a huge amount of alignables – especially in the outer layers – would be included into the updates of the alignment parameters. Thus the retrieval and update of the variance-covariance matrix as well as the multiplication of some of the involved matrices gets unreasonably time-comsuming. Secondly, the available amount of virtual memory would not be sufficient to store all covariance matrices. These two problems would even remain if the additional selection criteria described in Section 6.4 would be applied.

To overcome these issues, the Tracker is aligned in steps. First, the outer parts, i.e. the TOB and TEC, are aligned. Due to its rigid mechanical support structure, the outer barrel is not expected to exhibit large misalignment effects, so that it is an ideal starting point for the alignment procedure. After that, the inner parts, consisting of the TIB and TID, are aligned. At this stage, the previously aligned outer parts of the Tracker can serve as reference frame. Also, due to the larger misalignment expected on the higher hierarchical levels, prior to the alignment on the level of modules the barrel layers and endcap disks are aligned. Finally the pixel tracking system, i.e. the TPB and TPE, is aligned. As for the previous step, the barrel layers and endcap disks are aligned before the modules. All these steps are done using single tracks from either isolated muons or cosmic events, which are abundantly available. For a final precision alignment of the pixel barrel layers and endcaps the sparsely available tracks from two-body decays (decays of Z and J/Ψ) are processed, using the kinematic constraints presented in Chapter 5. This kinds of events make it possible to correlate especially the pixel half-barrels, which otherwise do not overlap, thus allowing a more accurate alignment of these structures with respect to each other.

All of these consecutive steps are carried out using the production system as explained in Section 6.2, utilizing the scheme for a single alignment run as shown in Figure 6.1 repeatedly. To speed up the computation, the full data sample is split and processed in parallel on individual cores. The final result is then obtained by merging the independent results, which is done by calculating the weighted mean of the obtained alignment constants.

6.4 Additional Selection Criteria for Alignable Objects

Due to the limitations regarding the memory of the computing nodes, additional selection criteria for including alignable objects into the update of the alignment parameters and their associated variance-covariance matrix had to be introduced, to keep the number of stored correlation matrices within practical bounds. This means that on top of the track-based metrical selection as presented in Chapter 4.3 further restrictions are applied to the alignable objects. In order to avoid an unnecessary consumption of computing time rather simple geometrical selection cuts have been chosen.

For all alignable objects $i \in I$, which were crossed by the current track, compute the lists L_i. Then, for all lists L_i apply the following selection criteria:

1. Select all alignable objects $j \in L_i$ for which the metrical distance is below a certain threshold d_{thr}, i.e. select j directly if $d(i,j) \leq d_{\text{thr}}$.

2. If the metrical distance is above the threshold value but still below a maximum value $d_{\max}$, i.e. $d_{\text{thr}} < d(i,j) \leq d_{\max}$, also the following additional geometrical selection criteria have to be fulfilled:

   (a) Only alignables in the same or the next few layers beyond shall be selected. This can be achieved by checking either the difference between the polar radii R or the Z-coordinate of the positions, depending on whether the alignable object i is placed inside the barrel region or within one of the endcaps.

   - $\Delta R_{\min} \leq (R_j - R_i) \leq \Delta R_{\max}$ if i is located inside the barrel region or
   - $\Delta Z_{\min} \leq \operatorname{sgn}(Z_i) \cdot (Z_j - Z_i) \leq \Delta Z_{\max}$ if i is located within one of the endcaps.

   (b) In addition, the alignable objects should not be too distant from each other. The requirement chosen here was that the alignable j at position $\boldsymbol{r}_j$ has to be placed within a cylindrical volume, having the radius r_c and an axis of symmetry going through the origin and the position $\boldsymbol{r}_i$ of alignable i. This prerequisite can be easily expressed as (with $r_i = |\boldsymbol{r}_i|$ and $r_j = |\boldsymbol{r}_j|$):

$$r_j^2 - \frac{(\boldsymbol{r}_i \boldsymbol{r}_j)^2}{r_i^2} \leq r_\mathrm{c}^2$$

The parameters d_thr, d_max, ΔR_min, ΔR_max, ΔZ_min, ΔZ_max as well as r_c have to be chosen appropriately. For the alignment simulations studies presented in this chapter, the actual values were solely chosen such that the restrictions due to available memory were (as closely as possible) met. This means that no feedback from the simulations' final alignment precision was included into the choice, in order to mimic the situation of real data-taking as closely as possible.

6.5 Results

6.5.1 The S43 Scenario

This scenario, where the full Tracker is aligned for the first time with a reasonable amount of collision tracks, is of course the biggest challenge. The alignment procedure has been carried out using two million events with isolated muons, having at least a transverse momentum of 5 GeV. In addition, two million events with cosmic muons and 10,000 events from Z-decays have been used for the alignment of layers and disks. The whole procedure took about 15.4 hours to complete, running on ten cores in parallel; for a more detailed compilation of the individual computing times see Table 6.2.

The obtained alignment object, referred to as KAA-S43, is visualized in Figures 6.3 and 6.4. Both figures show the displacements of the individual modules w.r.t. the ideal geometry in the precise coordinate for the KAA-S43 geometry (black) and the misaligned geometry (red).

The overall final precision of approximately 40 μm is sufficient to regain most of the nominal performancof the Tracker. Figure 6.5 compares the tracking residuals for the case of the ideal geometry (blue), the KAA-S43 geometry (black) and the misaligned geometry (red) for a sample of isolated muons, i.e. the same kind of tracks that were used for the alignment procedure. Figure 6.6 shows the respective distributions of the track χ^2 divided by its number of degrees of freedom. In both cases the improvement in performance is clearly visible.

An example of what this means in terms of physics observables can be seen in Figure 6.7, where the transverse momentum resolution for a sample of muons with $p_\mathrm{T} = 100\,\mathrm{GeV}$ is shown for the three different geometries. Whereas the misaligned geometry makes reasonable measurements of such high momenta impossible, the KAA-S43 geometry shows a drastic improvement, even if the resulting precision is still twice the value as for the case of the ideal geometry.

6.5.2 The S156 Scenario

Due to the increased luminosity an increased number of high momentum tracks in available for this scenario. Especially for the alignment of the Inner Tracker and the Pixel Tracker an amount of approximately 200,000 muons tracks from $W \rightarrow \mu\nu$ decays are available. For the Outer Tracker a sample of two million isolated muons, with a transverse momentum of at least 11 GeV, has been used. In addition, information from two million events with cosmic muons and 10,000 events from Z-decays has been used for the alignment on the level of layers and disks. The KAA-S43 alignment object was used as starting geometry. The whole procedure took about 6 hours to complete, running on ten cores in parallel for the alignment of the TOB/TEC and two cores for the remaining Tracker; for a more detailed compilation of the individual computing times see Table 6.2.

The resulting alignment object, referred to as KAA-S156, shows several improvements w.r.t. the KAA-S43 object. Especially in the TIB the precision

	$\sigma_{\text{misaligned}}$ [μm]	σ_{S43} [μm]	T_{S43} [h]	σ_{S156} [μm]	T_{S156} [h]
TOB	106	23	4.8	22	1.5
TEC	92	30		30	
TIB	482	56	3.9	36	1.7
TID	445	68		66	
TPB	105	18	6.7	16	3.0
TPE	120	26		22	
Total	242	40	15.4	34	6.2

Table 6.2: Alignment precisions for the precise coordinate $R\Delta\phi$ and computing times for the KAA-S43 and KAA-S156 alignment objects.

gets significantly better; see the compilation of results in Table 6.2. This refinement translates also into an improved tracking performance, as cen be seen from the distribution of the tracks' χ^2/ndof in Figure 6.8 and the transverse momentum resolution in Figure 6.9. For both figures the same track samples as for the corresponding plots for the S43 scenario have been used, in order to guarantee the comparability.

6.6 Summary

Using simulation data from the CSA08 computing challenge and exploiting the given resources of the CAF, it has been shown that the KAA can align the full Tracker under realistic start-up conditions within a reasonable time. Table 6.2 shows a compilation of the achieved alignment precisions and the time spent. For the case of high energy muons it has been demonstrated that the reconstruction efficiency and the transverse momentum resolution can be successfully recovered to a certain extent. However, to reach the nominal performance of the ideal geometry, larger samples of high-quality, high-momentum tracks than available during the start-up phase are needed.

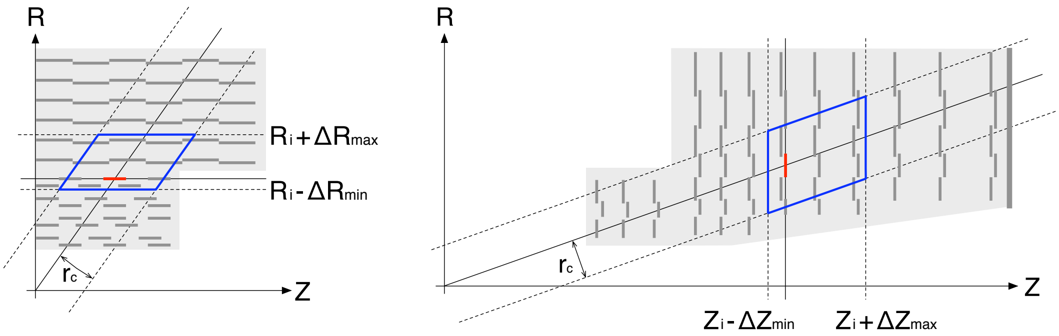


Figure 6.2: Sketch of the geometrical alignable selection cuts (R-Z-plane) for the barrel region (left) and the endcap region (right). Only modules within the highlighted area (blue), which is defined with respect to position of the reference module (red), are finally chosen from the reference module's update list.

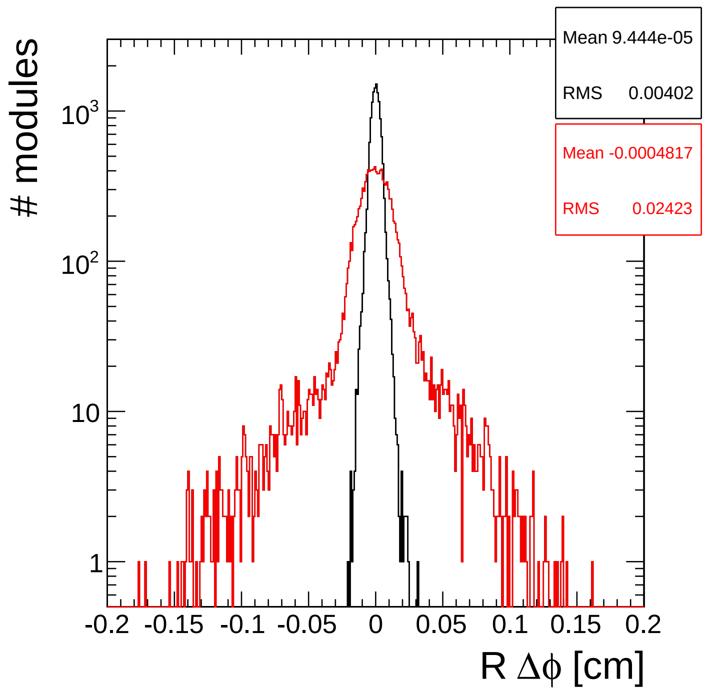


Figure 6.3: Distribution of $R\,\Delta\phi$ for all modules of the Tracker (Black: KAA-S43, Red: Misaligned)

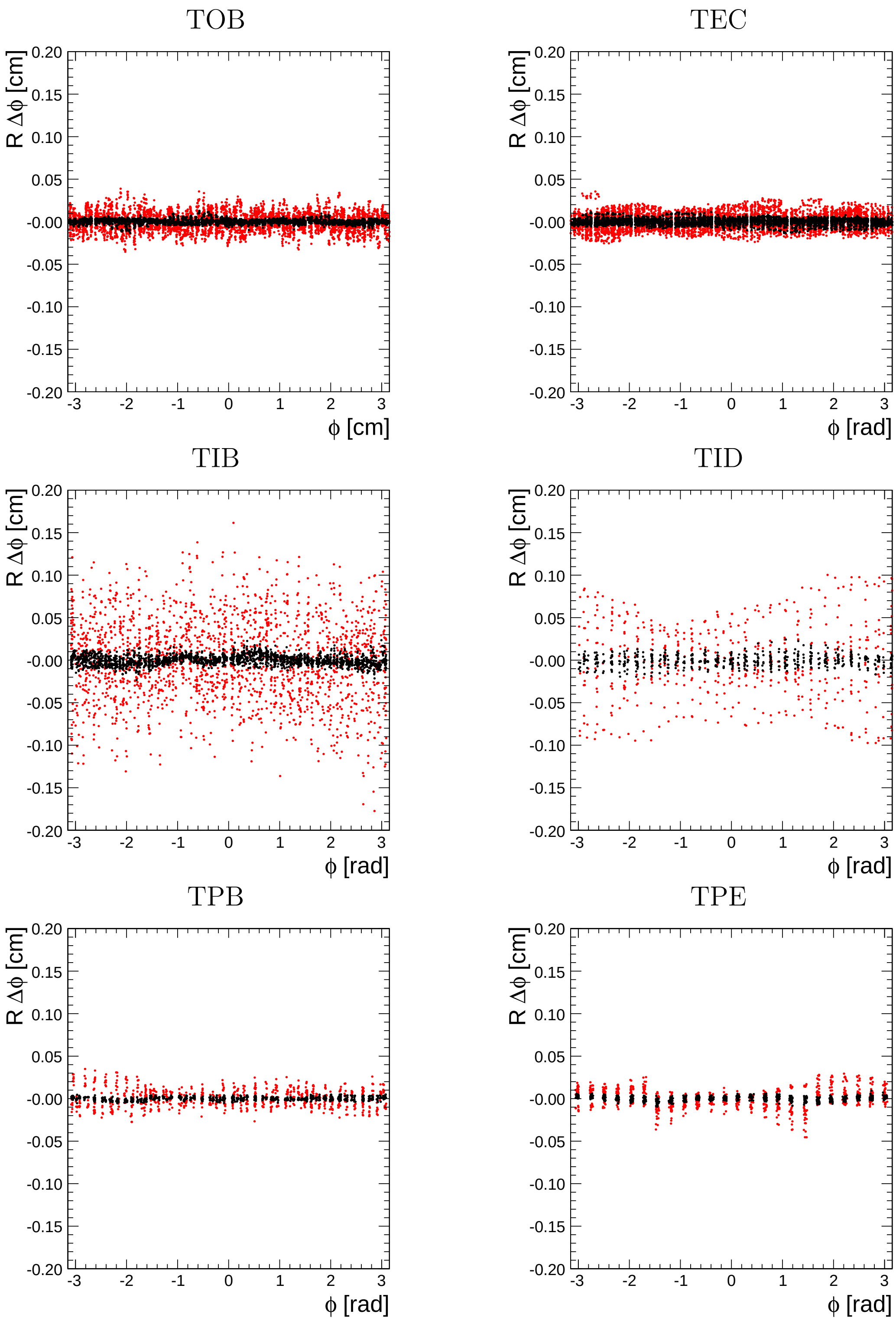


Figure 6.4: Misalignment in the precise coordinate before (red) and after (black) the alignment procedure for the S43 Scenario.

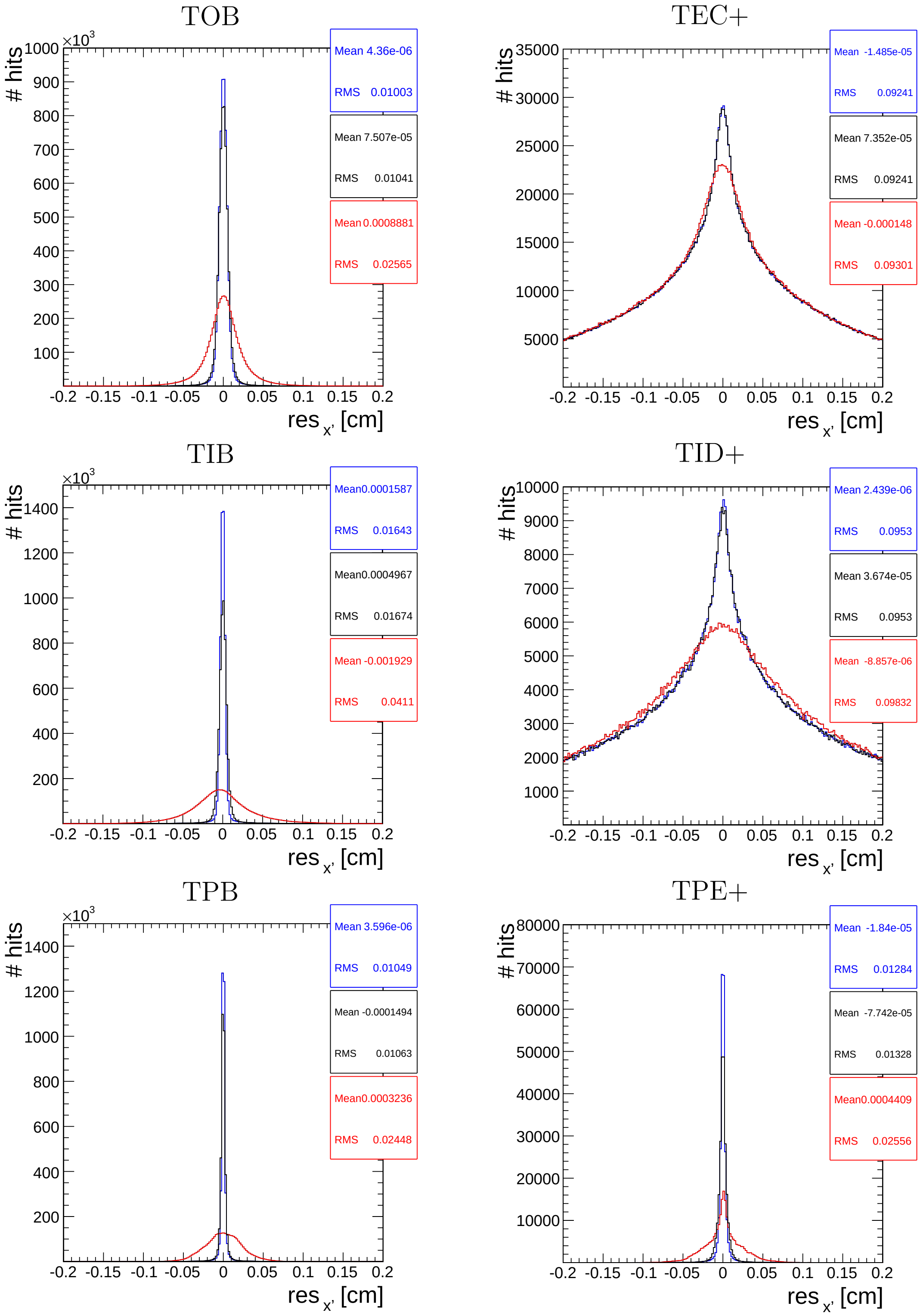


Figure 6.5: Tracking residuals for the ideal geometry (blue), the KAA-S43 geometry (black) and for the case of no applied alignment (red) for the S43 Scenario.

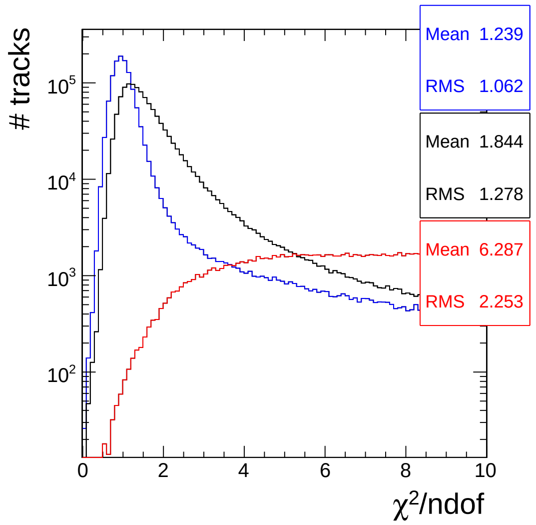


Figure 6.6: Distribution of χ^2/ndof for reconstructed tracks (Blue: Ideal, Black: KAA-S43, Red: Misaligned)

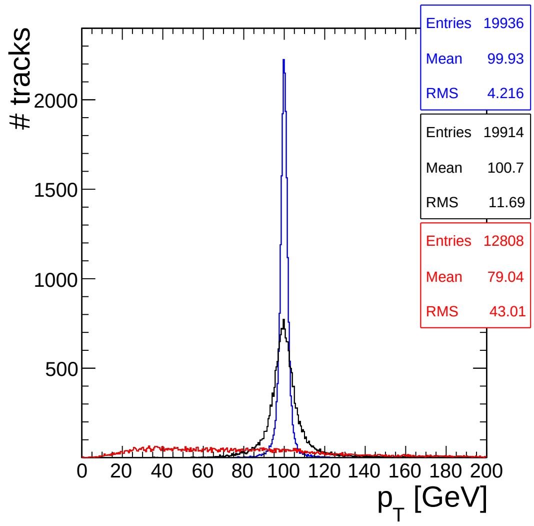


Figure 6.7: Distribution of the transverse momentum p_T for reconstructed tracks (Blue: Ideal, Black: KAA-S43, Red: Misaligned)

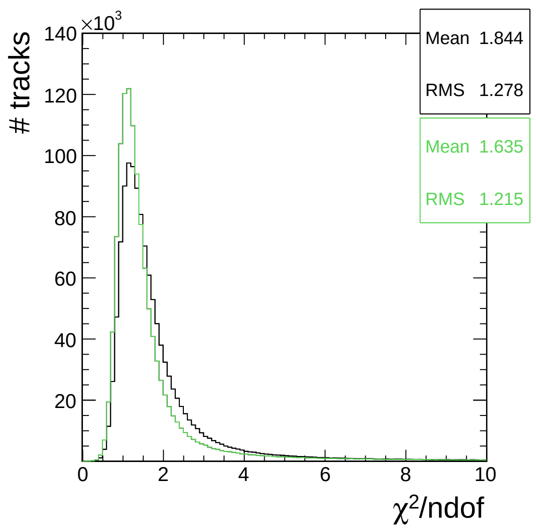


Figure 6.8: Distribution of χ^2/ndof for reconstructed tracks (Green: KAA-S156, Black: KAA-S43)

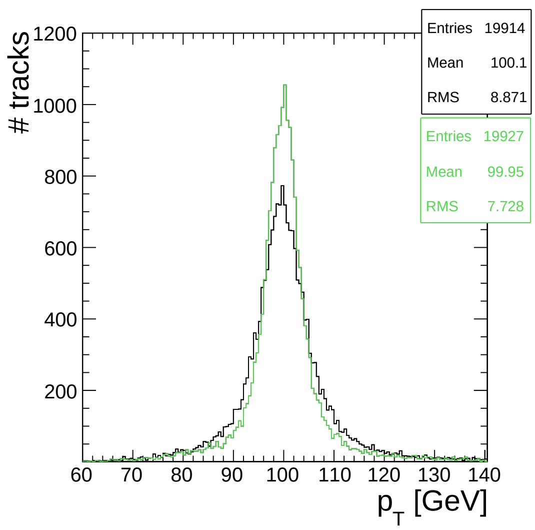


Figure 6.9: Distribution of p_T for reconstructed tracks (Green: KAA-S156, Black: KAA-S43)

Chapter 7

Results from the CMS Tracker Integration Facility

The final assembly of the CMS Silicon Strip Tracker has been carried out at a dedicated facility, referred to as the Tracker Integration Facility (TIF). Located on the surface at the CERN Meyrin site, this facility has been used not only for construction, but also for testing and debugging the power supplies and the read-out electronics before lowering them into the underground cavern. To this end, the Silicon Strip Tracker has been partially operated at different coolant temperatures ranging from $+15\,^{\circ}\mathrm{C}$ to $-15\,^{\circ}\mathrm{C}$. Up to 15% of the Silicon Strip Tracker has been powered and read-out simultaneously. An external trigger system has been used to trigger on cosmic events, which were subsequently used for detailed tracking and alignment performance studies [39, 40, 41].

The global coordinates at the TIF are defined as if the Tracker was at its nominal position in the underground cavern.

7.1 Experimental Setup

Three different trigger setups, consisting of scintillation detectors, as sketched in Figure 7.1, and an associated coincidence logic, have been used for data-

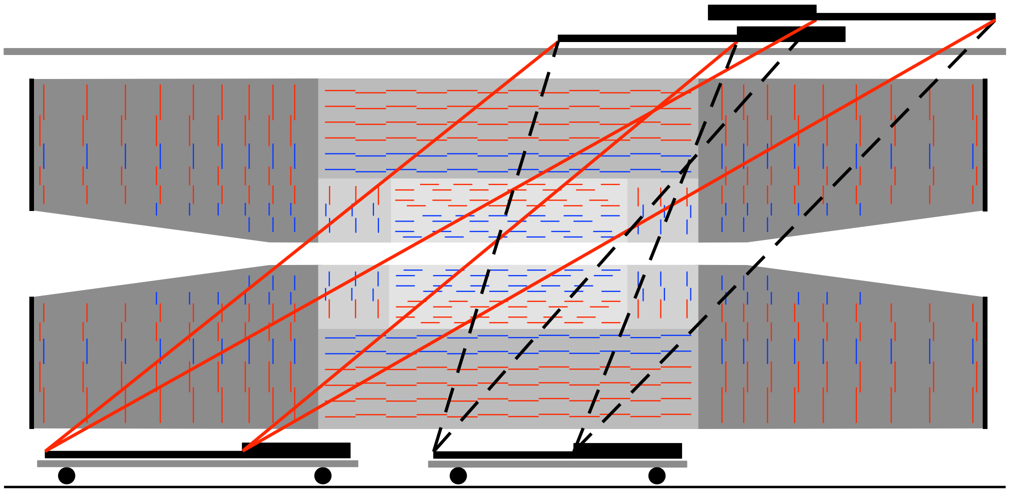


Figure 7.1: Schematic R-Z view of the cosmics trigger setup at the Tracker Integration Facility.

taking. The acceptance region is indicated by the straight lines connecting the active areas of the scintillators. *Configuration A* corresponds approximately to the acceptance region defined by the right bottom scintillator (black dashed lines); *Configuration B* corresponds to the left bottom scintillator (solid red lines); and *Configuration C* combines both of the above. Lead plates were placed on top of the lower trigger scintillators, ensuring an energy threshold for the cosmic particles of approximately 200 MeV.

About 15% of the detector modules, all located at $Z > 0$, were powered and read-out. This includes 444 modules in TIB (16%), 720 modules in TOB (14%), 204 modules in TID (25%), and 800 modules in TEC (13%). For trigger configuration A and B the data was collected at room temperature (+15 ℃). In addition to room temperature, configuration C was operated at +10 ℃, −1 ℃, −10 ℃ and −15 ℃. Due to cooling limitation, a large number of modules had to be turned off at −15 ℃. However, during the period of data taking with configuration A the insertion of the TEC− took place, such that the alignment stability is not guaranteed. The other configurations (B and C) already had all detector components integrated.

7.2 Track Selection

Charged track reconstruction includes three essential steps: seed finding, pattern recognition, and track fitting. Three tracking algorithms are deployed for the CMS Tracker, called *Combinatorial Track Finder* (CTF), *Road Search* and *Cosmic Track Finder*, the latter being specific for the reconstruction of cosmic tracks. All three tracking algorithms use a Kalman filter technique for the final track fitting, but differ in the first two steps. For the track-based alignment studies at the TIF only the CTF algorithm was utilized.

At the TIF, where no magnetic field is present, the momentum of tracks cannot be measured. Thus estimates of material effects, i.e. energy loss and multiple scattering, can be done only approximately. A track momentum of 1 GeV/c is always assumed, which is close to the expected average momentum of cosmic tracks. For alignment purposes, however, it is beneficial to reject low-momentum tracks, for which the single hit resolution is dominated by the contribution of the material effects. Therefore tight cuts must be applied, based on trajectory estimates and fiducial errors, to get indeed a cosmic track sample with sufficiently high momenta. To this end, the following selection criteria for tracks have been applied:

- The event contains only one track.
- The trajectories after the final track fit, which are represented by straight lines, are required to traverse the scintillation detectors. This can be approximately accomplished by applying the following cuts: $-1.5 < \eta_{\mathrm{track}} < 0.6$, -1.8 rad $< \phi_{\mathrm{track}} < -1.2$ rad.
- The track has
  - at least 5 hits associated in total and
  - at least 2 matched hits in double-sided modules.
- The normalized χ^2_{track}, i.e. χ^2_{track} divided by the number of degrees of freedom, is required to be lower than 4.0. In order to compensate the effect of misalignment on the performance of tracking, the measurement errors are increased for the pattern recognition and the final

track-fitting. The increase is chosen such that it reflects the expected shifts due to misalignment.

Additional quality checks have been introduced into the pattern recognition, to reject fake hits stemming from noise clusters or other sources of combinatorial background, while still maintaining high efficiency. To this end, the following selection criteria for hits have been applied:

- The hit has to be associated to a cluster with a total charge of 50 ADC counts at least. On a double-sided module, the hits on both sensors must satisfy this requirement.
- The hit has to be reasonably isolated. If any other reconstructed hits are found on the same sensor within a radius of 8.0 mm, the entire track is rejected. This cut helps in rejecting fake clusters generated by noisy strips and modules.
- The hit has to pass an outlier rejection criterion. The criterion is the quantity $\chi^2_{\mathrm{hit}} = \boldsymbol{r}^{\mathrm{T}}\boldsymbol{V}^{-1}\boldsymbol{r}$, where $\boldsymbol{r}$ is the one- or two-dimensional local residual vector and $\boldsymbol{V}$ its associated variance-covariance matrix. The hit is flagged as an outlier if χ^2_{hit} exceeds a given cut value ($\chi^2_{\mathrm{cut}} = 5$). If one or more outliers are found in a track fit, they are removed from the hit collection and the fit is repeated; this procedure is iterated until there are no more outliers or the number of surviving hits is less than 3.

For the track sample recorded with trigger configuration C at $-10\,^{\circ}\mathrm{C}$, which was mainly used for alignment purposes, the compound efficiency for all the cuts above is around 8.3%. For simulated data an efficiency of about 20.5% is observed.

For the visual validation of the tracking performance a set of pseudo-hit-maps for the triggering scintillators was compiled. Two reference planes were used, coplanar to the global X-Z-plane at $Y = 160$ cm and $Y = -160$ cm respectively (in the global coordinate-system). For every reconstructed track the trajectory state of the outermost hit was propagated to the upper reference plane using a straight-line track-model. Similarly, the trajectory states of the innermost hits were propagated the same way to the lower reference plane. The resulting intersection points were stored in two-dimensional histograms (see Figures 7.2 to 7.5).

The shapes of all nine scintillators can be clearly distinguished, and their roughly determined positions and dimensions are in good agreement with the experimental setup. Two of the scintillators show a drastically decreasing light-yield for positions further away from the photocathode (lowermost position in figure 7.2 and middle in figure 7.3). This is most likely due to damages of the devices themselves rather than from any tracking inefficiencies.

In addition, the artefact of a photomultiplier running at too high voltage and hence directly triggering a signal when being hit, can be recognized in Figure 7.4.

7.3 Common Alignable Selection

The recorded data were analyzed with all three track-based alignment algorithms implemented for the CMS Tracker. Although in the endcap region (TEC+, TID+) tracks were recorded and used for alignment [42], only in the barrel region (TIB+, TOB+) a sufficient amount of data were recorded to perform alignment at the level of single modules.

In order to allow a meaningful comparison of the results, a common selection of alignable objects was agreed upon. The selection criteria were chosen according to the following considerations. First, only modules which are approximately positioned between the upper and the lower scintillation detectors were included. Second, cosmic tracks traverse modules which are strongly tilted w.r.t. the X-Z-plane under a large inclination angle, which deteriorates the spatial resolution of the measurements drastically. These modules were excluded as well. Only the barrel modules with centers that lie within the following geometrical ranges have been taken into account (all coordinates refer to the global CMS frame):

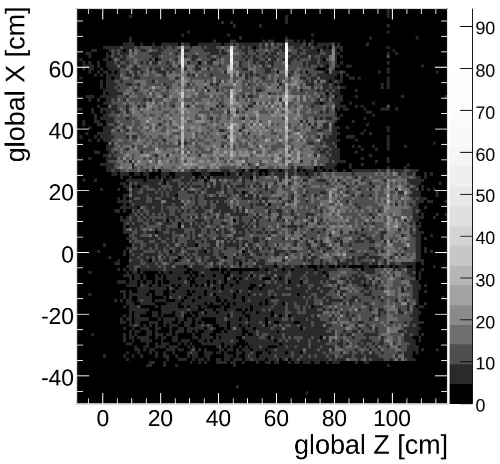


Figure 7.2: Upper scintillators triggering the readout of the barrel section.

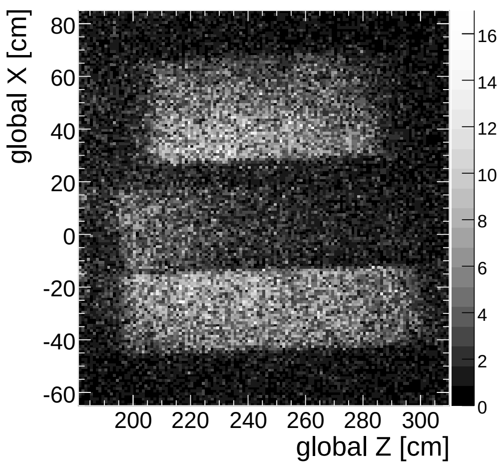


Figure 7.3: Upper scintillators triggering the readout of the endcap section.

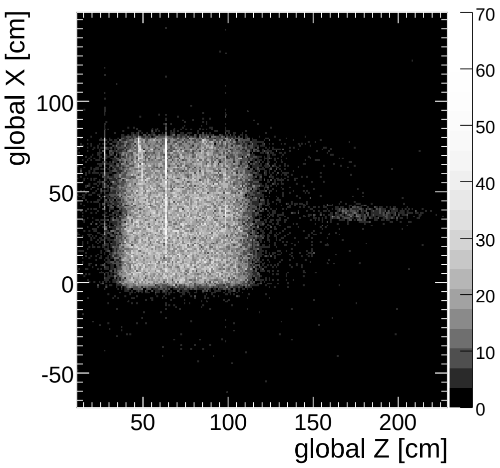


Figure 7.4: Lower scintillator triggering the readout of the barrel section.

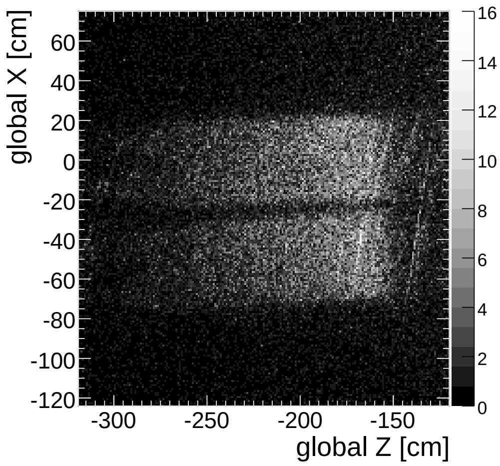


Figure 7.5: Lower scintillators triggering the readout of the endcap section.

- $Z > 0$ cm
- $X < 75$ cm
- 0.5 rad $< \phi <$ 1.7 rad

Furthermore, the selection of alignment parameters was standardized. Only degrees of freedom to which the algorithms are (believed to be) sensitive under the given conditions were taken into account. The local coordinates aligned for the several module types are:

- u, v, w, γ for TIB double-sided modules
- u, w, γ for TIB single-sided modules
- u, v, γ for TOB double-sided modules
- u, γ for TOB single-sided modules

7.4 Results of the Kalman Alignment Algorithm

Although only a small fraction of the full CMS Tracker was operated, and an even smaller fraction of the operated modules was selected for the alignment exercise at the TIF, the total number of alignment parameters taken into account here is comparable to the biggest silicon tracking devices employed up to now; see [6] for a detailed compilation.

Some specific features of the TIF setup have to be kept in mind when estimating alignment constants. The absence of a magnetic field causes the particles to traverse the detector in almost straight lines. This makes not only the determination of the particle momenta impossible, but also leaves some global degrees of freedom for the alignment unconstrained. As mentioned in Section 2.3.1, the usage of straight tracks can cause shearings; in the case of the TIF setup this corresponds to global layerwise deformations of the final geometry of the form $\Delta X \propto R$ and $\Delta Z \propto R$. In addition, the complete lack of symmetries in the setup or anything that could be utilized as reference frame increases the risk of global translations and rotations of the entire geometry. Although these effects can not be prevented in the KAA, they can however be compensated in the final result (see Section 7.4.1).

7.4.1 Results with Real Data

No further selection criteria for either tracks or alignables than the ones mentioned in Sections 7.2 and 7.3 were applied, except that tracks had to have at least six hits in the barrel region (TIB and TOB). Running on the data sample recorded with trigger configuration C at $-10\,^{\circ}\mathrm{C}$, this resulted in approximately 70,000 tracks hitting a total of 430 modules, out of which 159 are located in the TIB and 271 in the TOB. Due to this relatively small number of alignables involved, the Kalman Alignment Algorithm was used to align all modules at once, without employing the concept of selective updates. Since no external tracking information was available, update equations (4.3) and (4.4) were used.

Results for Module Alignment

For the alignment of the TIF setup at module level the so called *module survey geometry* rather than the ideal geometry was used as the starting point. This alternative geometry includes corrections from survey measurements of the module positions w.r.t. the shells in the TIB. Figure 7.6 visualizes the global shifts of the individual modules in X, Y and Z in dependence on their distance R from the global Z-axis.

Most significant are the large shifts observed in the TIB, especially the considerable shift of the entire second shell. Also the big spread within the individual shells is remarkable. However, the movements within a shell, or more precisely within the surface of a shell, turn out not to be random. On closer examination distinct correlations can be observed between the movements of the modules within a surface, indicating a deformation of the entire surfaces. Figures 7.7 and 7.8 show the situation for the inner surface of the second and

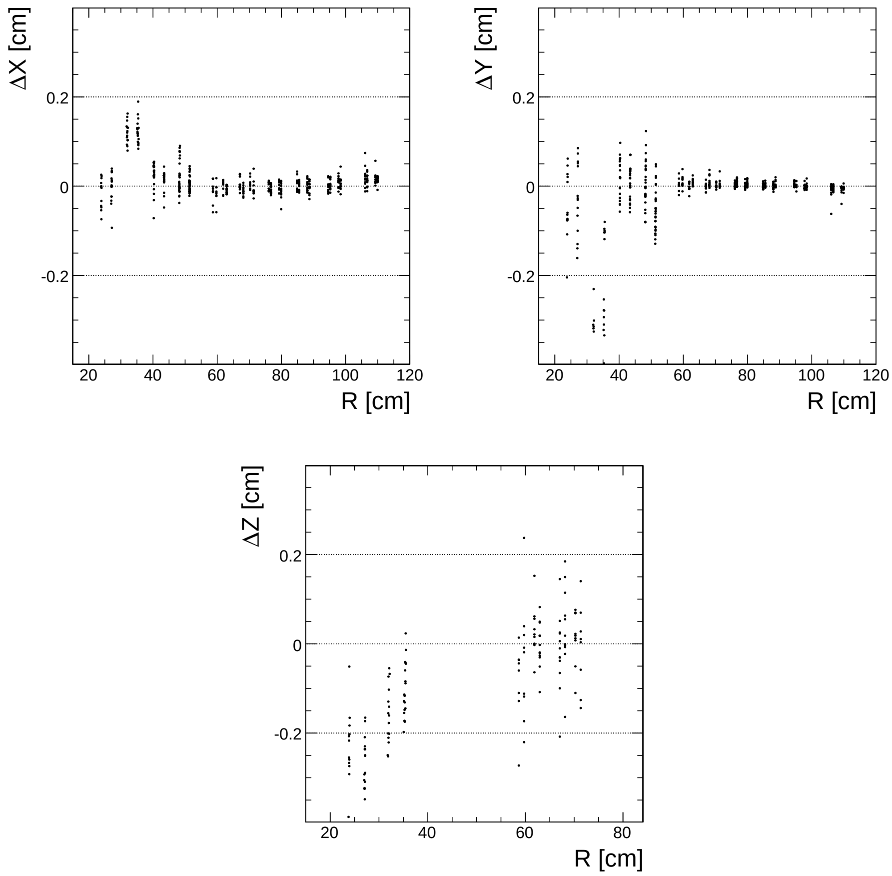


Figure 7.6: The computed global shifts for the modules in TIB and TOB. All shifts are with respect to the survey module geometry.

third shell, respectively, counting from inwards. The lower right plot in both figures visualize the deformation: The circles show the position of the center of the modules as described by the ideal geometry (in red) and as computed with the KAA (in black). The misalignment has been scaled as indicated in the figure captions. The green surfaces are merely a guide for the eye.

In the TOB the computed corrections along the local u-coordinate are at the scale of approximately $100\,\mu$m, which agrees very well with the expectation [31]. No correlated misalignment can be identified, as one would expect from the rigid structure of the TOB wheel. For the alignment corrections in Z-direction, which have been computed for double-sided modules only, it seems very likely that the result is superimposed with a ΔZ-R-shearing.

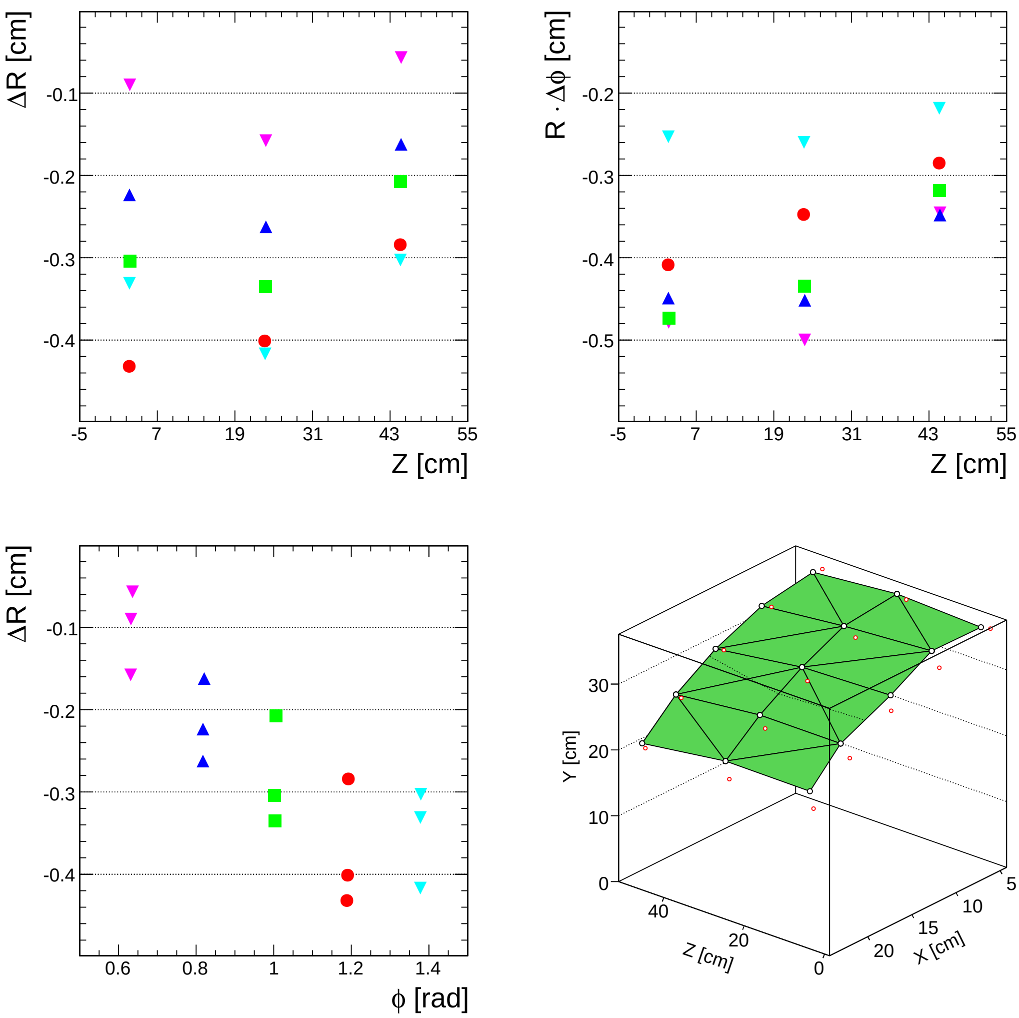


Figure 7.7: TIB, Shell 2, Inner Surface. Scaling factor for visualization: 10.

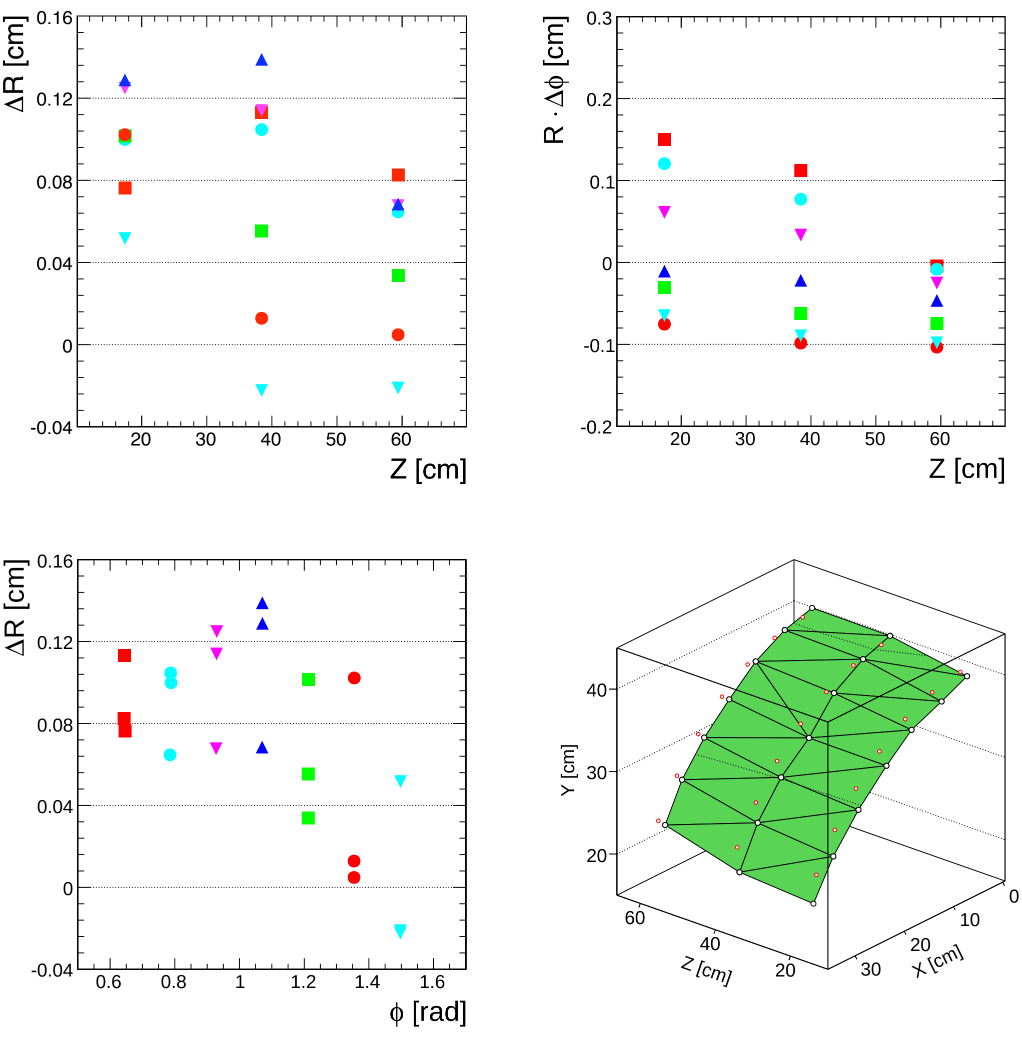


Figure 7.8: TIB, Shell 3, Inner Surface. Scaling factor for visualization: 20.

Consistency of the Module Alignment

To check the consistency of the results the alignment procedure has been repeated using random starting values as well as a different starting geometry. The resulting alignment constants are in general not directly comparable, as they are typically superimposed by ΔX-R- and ΔZ-R-shearings, respectively. However, these unconstrained degrees of freedom can be eliminated by fitting the data pairs (X_i, R_i) of all modules to a function of the form

$$\Delta X = aR + b.$$

The resulting function can then be evaluated for each data pair to retrieve a corrected set of data pairs $(X_i - aR_i - b, R_i)$, where all superimposed shearings and offsets are eliminated. To remove ΔZ-R-shearings an analogous method can be used. If the true geometry would indeed feature a shearing, this method would remove the associated correction from the final alignment constants. For a direct comparison of two results this would however not matter.

Figure 7.9 shows a comparison of the results presented in the previous section with the results obtained when using the ideal geometry as starting geometry. Figure 7.10 compares the results obtained when using the ideal geometry as starting geometry with either all starting values set to zero or randomly drawn from a Gaussian distribution ($\sigma_u = \sigma_v = \sigma_w = 200\,\mu$m, $\sigma_\gamma = 0.1$ mrad). In both cases the final the agreement for the X and Z shifts is good, verifying especially the very large displacements observed in the TIB.

For the Y-shifts the situation is similar, with the exception that the comparisons can be carried out only layer-wise. This means that the corrections ΔY, to which the alignment algorithm is the least sensitive in the given setup, are meaningful only relative between the modules in a layer. Therefore the significance of the apparent movement of an entire layer is not very high, as it is de facto an undefined degree of freedom. The alignment in ΔY for the modules within a layer is much better constrained due to hits in the overlapping regions.

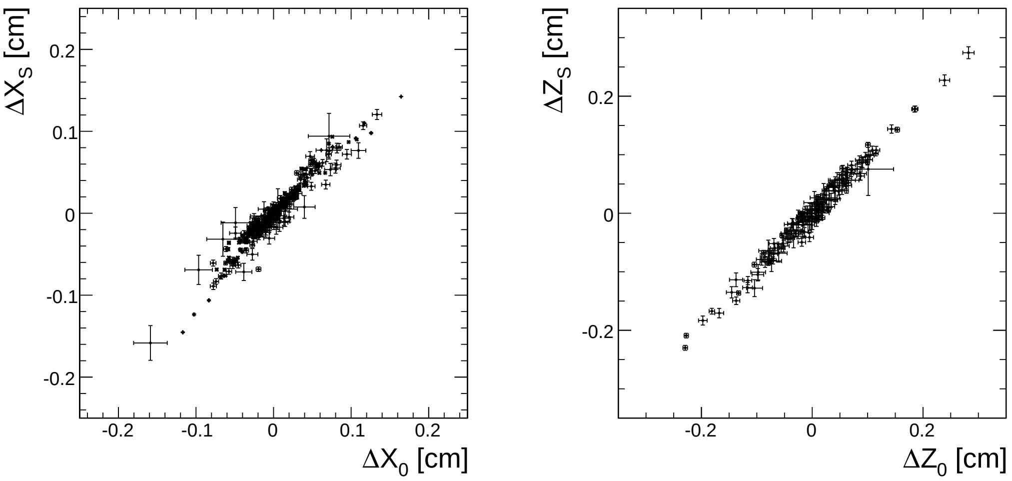


Figure 7.9: Comparison of the global shifts computed with different starting geometries. ΔX_0 and ΔZ_0 were computed using the ideal geometry, whereas for ΔX_S and ΔZ_S the module survey geometry was used. All shifts are with respect to the ideal geometry.

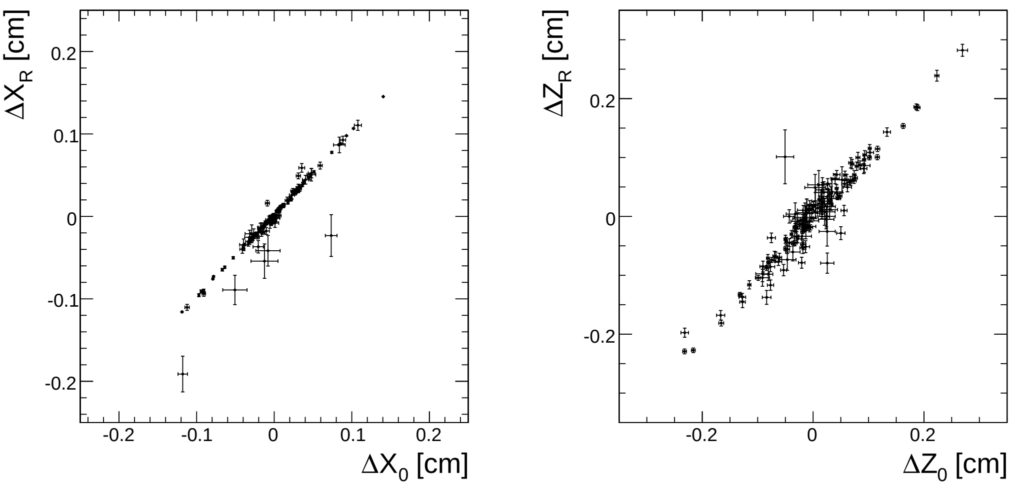


Figure 7.10: Comparison of the global shifts computed with different starting values. For ΔX_0 and ΔZ_0 the starting values for all local alignment parameters were set to 0, whereas for ΔX_R and ΔZ_S they were drawn randomly from a Gaussian distribution. All shifts are with respect to the ideal geometry.

Consistency between Module Alignment and String Alignment

An additional consistency check comes from the comparison to the alignment on the level of TIB strings and TOB rods. The same track selection and alignment procedure as for the modules was used, but all six degrees of freedom were aligned. For each string/rod the resulting alignment constants have been compared to the ones from the associated modules. For the X- and Y-positions the values from the modules simply had to be averaged. The comparison of the angle β_{str}, which is defined as the rotation of the string/rod around the axis parallel to the Y-axis and going through the center of the module, is somewhat more complicated: For the modules within a string/rod, the data pairs $(\Delta X_i, Z_i)$ were fitted to functions of the form

$$\Delta X_i = cZ_i + d.$$

For the resulting function the fitted slopes c can be related to the value $\overline{\Delta\beta}_{\mathrm{mod}} = \arctan(c)$ that is comparable to $\Delta\beta_{\mathrm{rod}}$.

The deformations of the TIB shells, which is seen as layer-wise correlated misalignment on the level of modules, causes also a correlated misalignment for the modules within an individual string. This should of course translate into distinct movements and rotations of the strings. The results for the TIB strings, compiled in Figure 7.11, clearly verify this situation. Due to the small and uncorrelated misalignment in the TOB the values for ΔX_{str}, ΔZ_{str} and $\Delta\beta_{\mathrm{str}}$ on the one side as well as $\overline{\Delta X}_{\mathrm{mod}}$, $\overline{\Delta Z}_{\mathrm{mod}}$ and $\overline{\Delta\beta}_{\mathrm{mod}}$ are close to zero, such that a comparison is not very conclusive. The TOB rods are therefore omitted from the plots.

Validation

In the case of the TIF only the track residual distributions can be used to judge the quality of a set of alignment constants. Figure 7.12 shows histograms of these distributions before (red) and after (black) alignment for the constants presented in Section 7.4.1. In particular, the residuals are shown for the direction along x', which is always parallel to the local u-axis but points

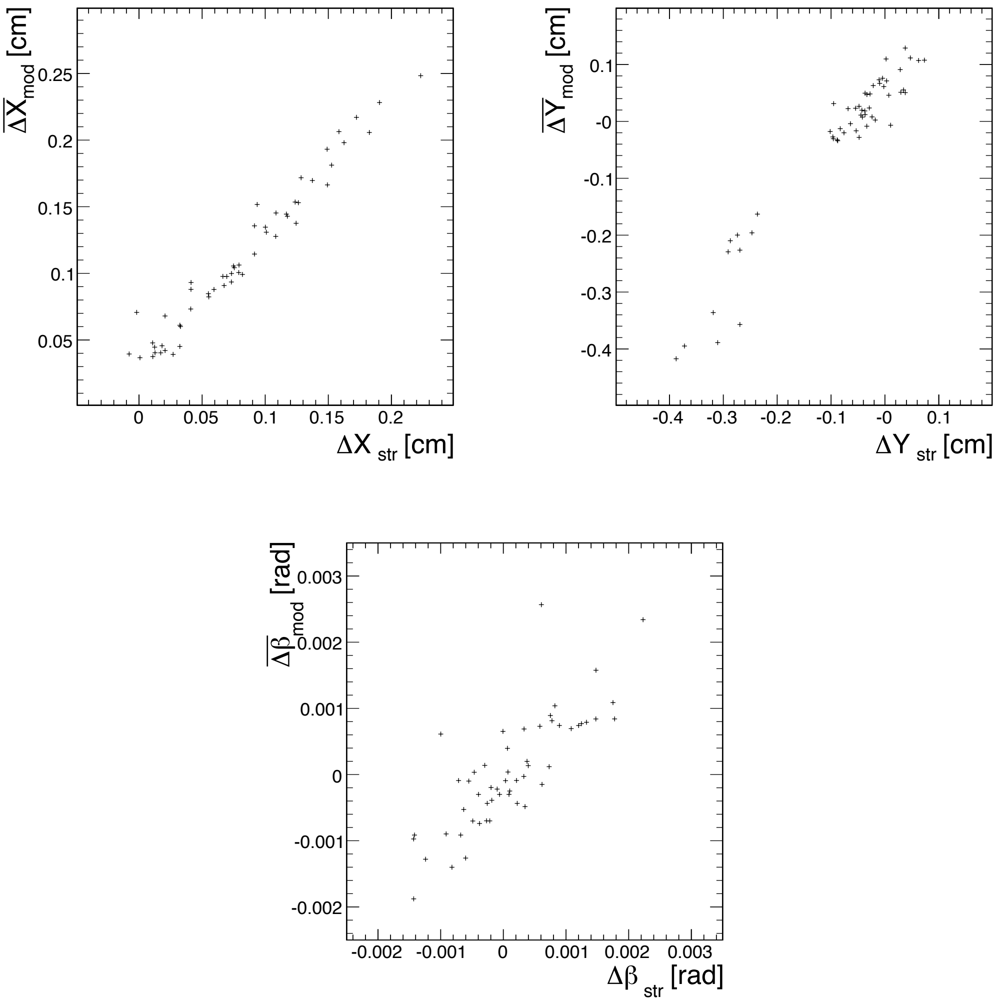


Figure 7.11: Comparison of alignment on module and string level.

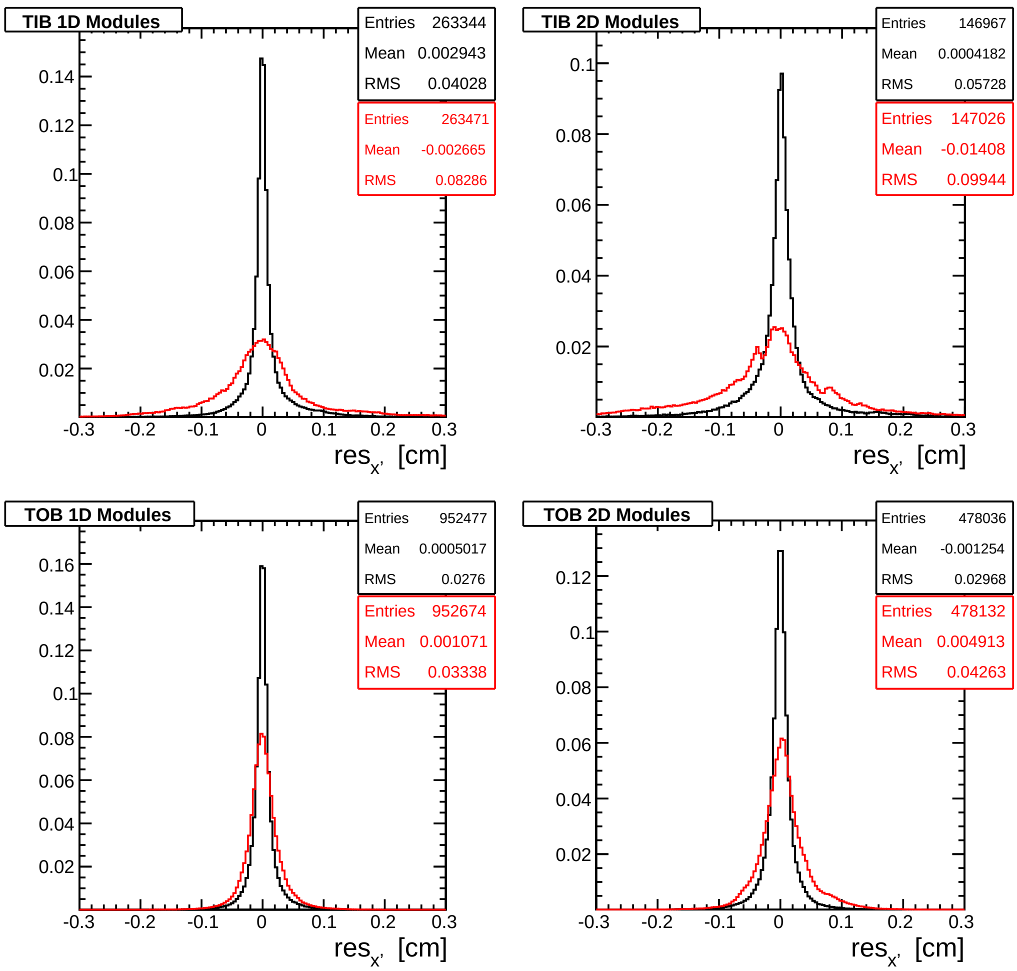


Figure 7.12: Track residuals before (red) and after (black) alignment.

into the same global R-ϕ-direction (see sketch in Figure 7.13). The plots are separated for single- and double-sided modules in the TIB and the TOB, and include also modules that were not selected for alignment because of the alignable selection described in Section 7.3. The ordinate of the plots shows the number of entries per bin, normalized to the absolute number of entries per histogram. Especially in the TIB, the improvement of the residuals is significant, as is expected.

To get an upper bound for the final precision of the alignment, these residuals have to be evaluated with more care. Especially the first and last hit of every track, which have to be computed using by extrapolation rather than

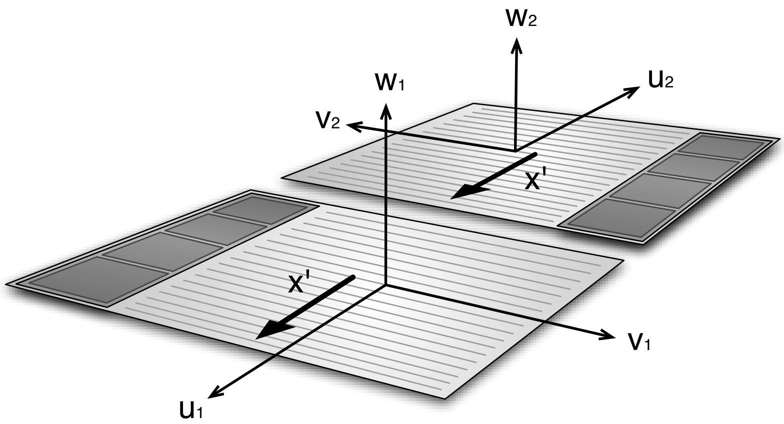


Figure 7.13: The local u-axes of the modules is defined relative to the read-out chips and point in general in different global directions. The direction of x' is per definition always parallel to the local u-axis but pointing into the same global R-ϕ-direction.

interpolation, are less accurately defined and hence deteriorate the track residual distribution. When these hits are excluded from the histograms this effect can be eliminated. This drastically decreases the number of hits in the innermost layer of the TIB and the outermost layer of the TOB, but due to overlap hits a reasonable statistics can still be maintained. In addition, by excluding modules that have not been aligned, the quality of the final track-fit can be improved. Figure 7.14 shows the widths of the initial and the corrected residual distributions for the four TIB layers (numbered 1 to 4) and the six TOB layers (numbered 5 to 10). For the latter, the widths are approximately the same for all layers at around 100μm. Bearing in mind that also the intrinsic detector resolution contributes to the width, this value can be regarded as upper limit for the precision of the alignment along the local u-coordinate.

The improved quality of tracking due the inclusion of the computed alignment constants can also seen from the values of χ^2_{track}. Figure 7.15 shows the values' distributions before and after alignment, displaying a drastic improvement in the latter case. Since in both cases no additional alignment errors have been used the absolute values are rather large, even for the final alignment which is believed to be accurate within a precision of 100μm. In addition, also the lack of reasonable estimates on the material effects for single tracks results in enlarged values for χ^2_{track}.

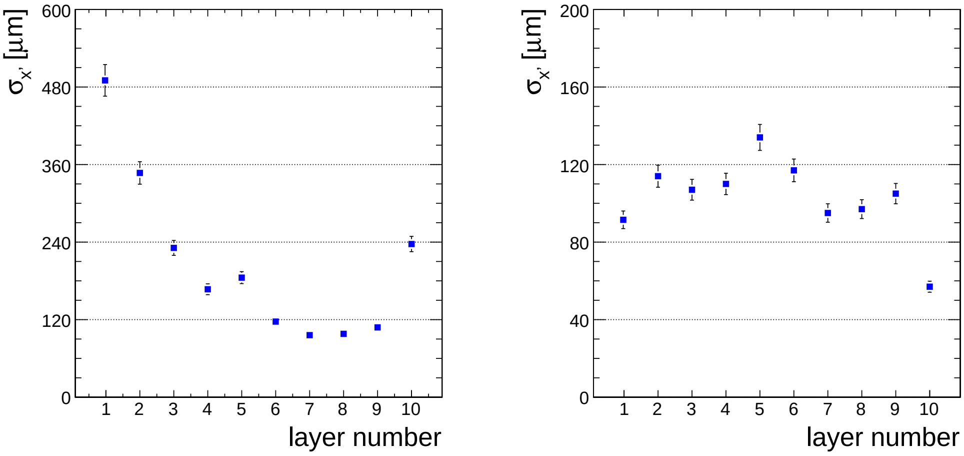


Figure 7.14: RMS of the raw (left) and corrected (right) residuals for each layer.

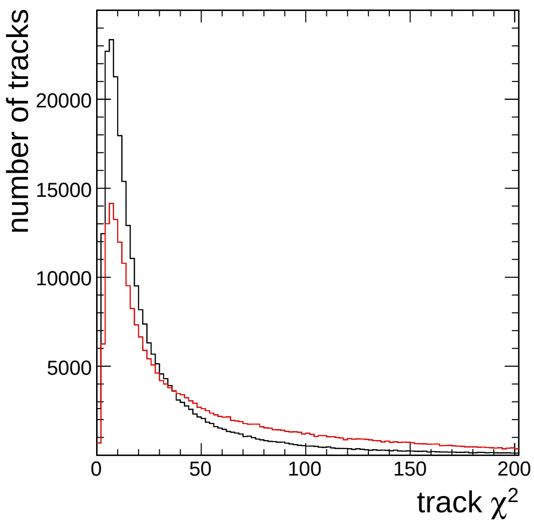


Figure 7.15: Track χ^2 before (red) and after (black) alignment.

7.4.2 Results for a Simulated Setup

To complement and further validate the alignment studies for TIF setup a simulation study has been carried out. An amount of 1,000,000 tracks was simulated, using an energy spectrum as expected for cosmic muons at the surface and an experimental setup close to the one described above. The alignment procedure was carried out in the same way as with the real data, whereas the misalignment was applied by drawing from Gaussian distributions with standard deviations according to the shifts observed at the TIF.

The remaining misalignment after processing all tracks is shown in Figure 7.16. The overall precision is — apart from a few outliers that were hit by only a few tracks — about 50 μm for the global X-coordinate, 100 μm for the global Y-coordinate (TIB modules only) and 150 μm for the global Z-coordinate (double-sided modules). The resulting track-residuals, shown in Figure 7.17, are of the same order as for the real data. These results suggest that the precision achieved in this simulation study can be seen as a lower limit for the precision of the constants computed from real data.

7.5 Comparison to other Algorithms

The results obtained with the Kalman Alignment Algorithm have been compared to the results from the other two alignment algorithms deployed for the CMS Tracker, the MillePede II Algorithm and the HIP Algorithm (see Section 2.3.2). The good agreement of the results is demonstrated in the correlation plots in Figure 7.18, where the alignment constants of the three algorithms are directly compared. Especially in the sensitive coordinates the compatibility is very satisfying, showing for instance an agreement of about 150 μm in the global X-coordinate. Furthermore, the comparison has also been carried out on the basis of track residuals as shown in Figure 7.19. The performance of the three algorithms is very similar, even though the KAA shows typically slightly better results for the sensitive coordinate.

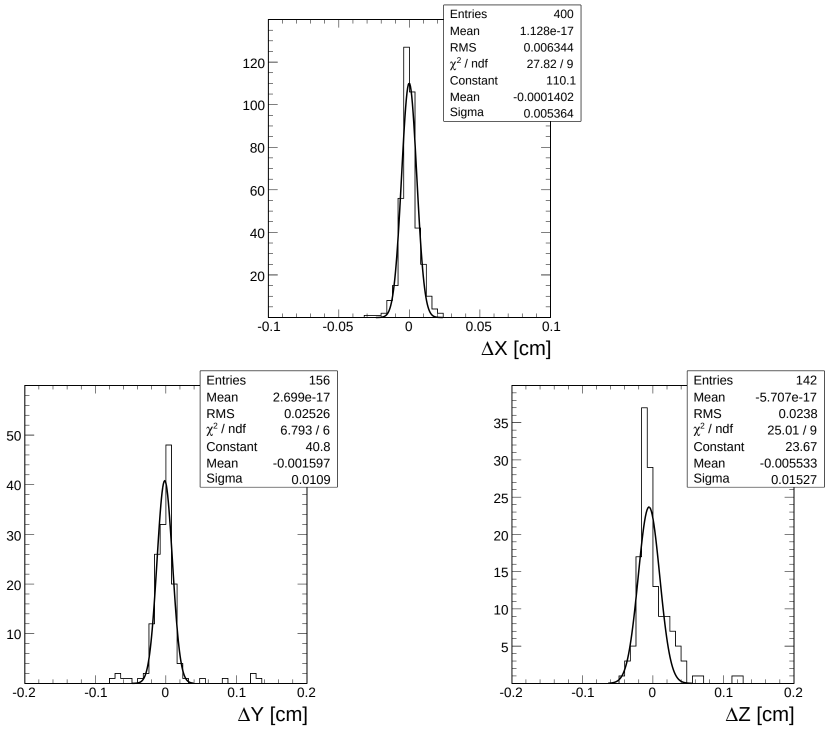


Figure 7.16: Remaining misalignment for the simulated setup.

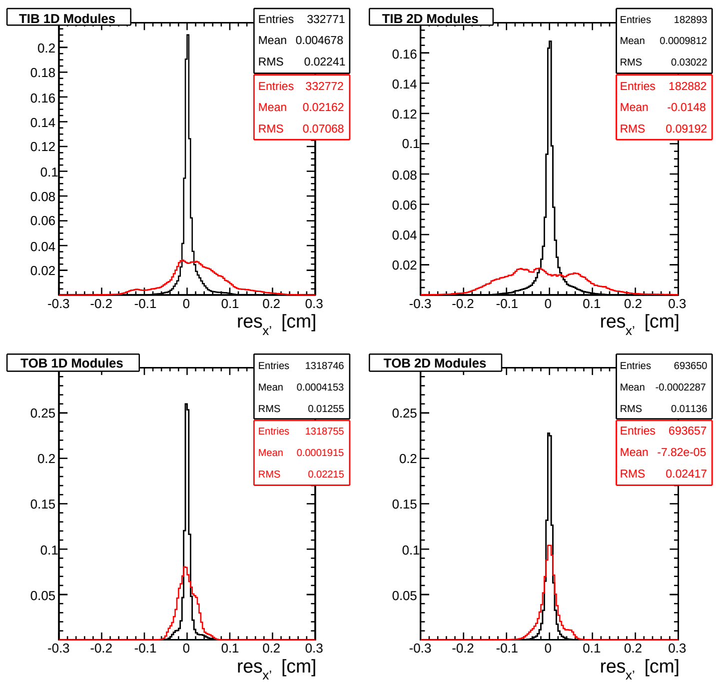


Figure 7.17: Track residuals for the simulated setup before (red) and after (black) alignment.

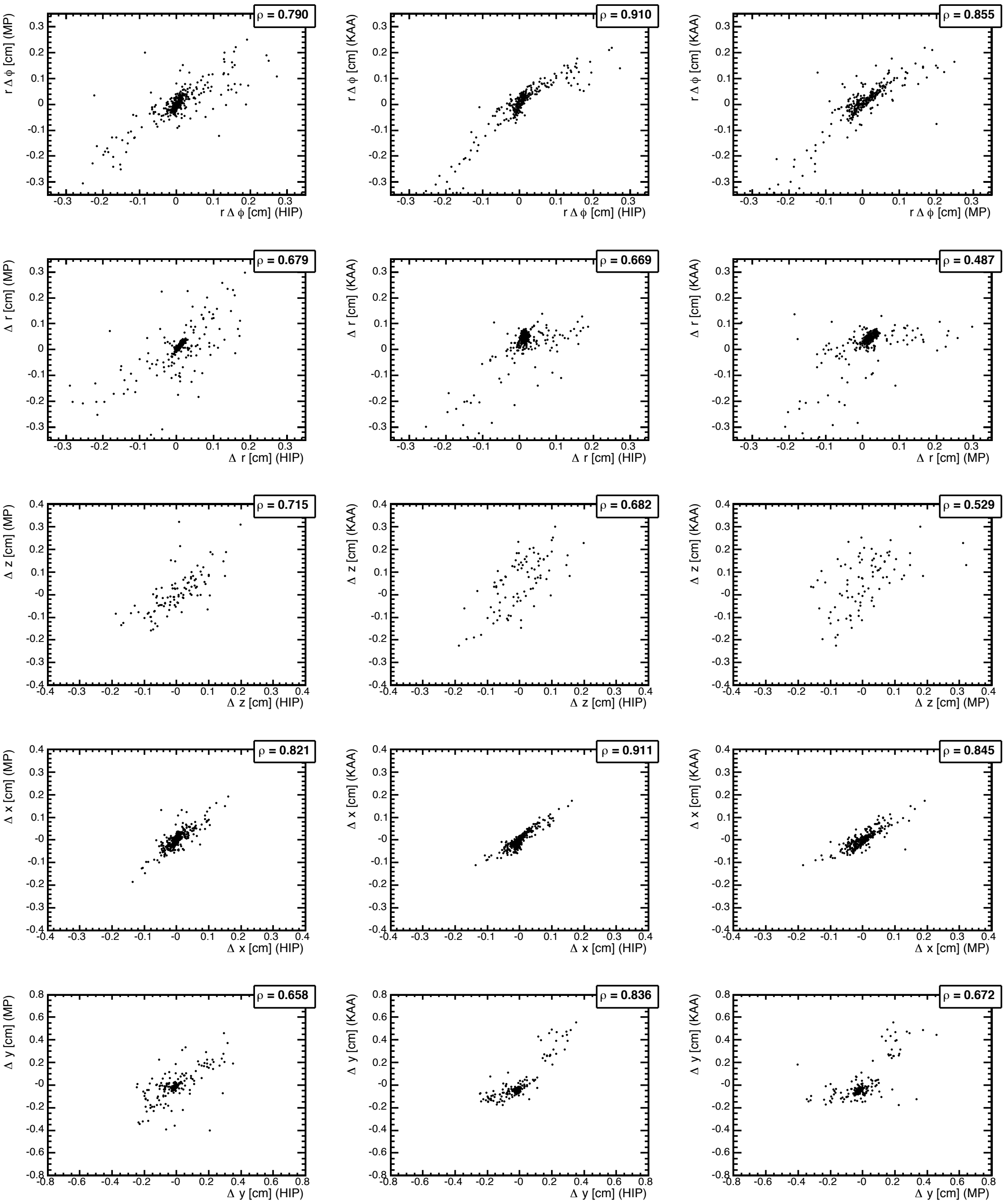


Figure 7.18: Comparison of the alignment constants for the three algorithms. Left: MillePede (y-axis) vs. HIP (x-axis); middle: KAA (y-axis) vs. HIP (x-axis); right: KAA (y-axis) vs. MillePede (x-axis).

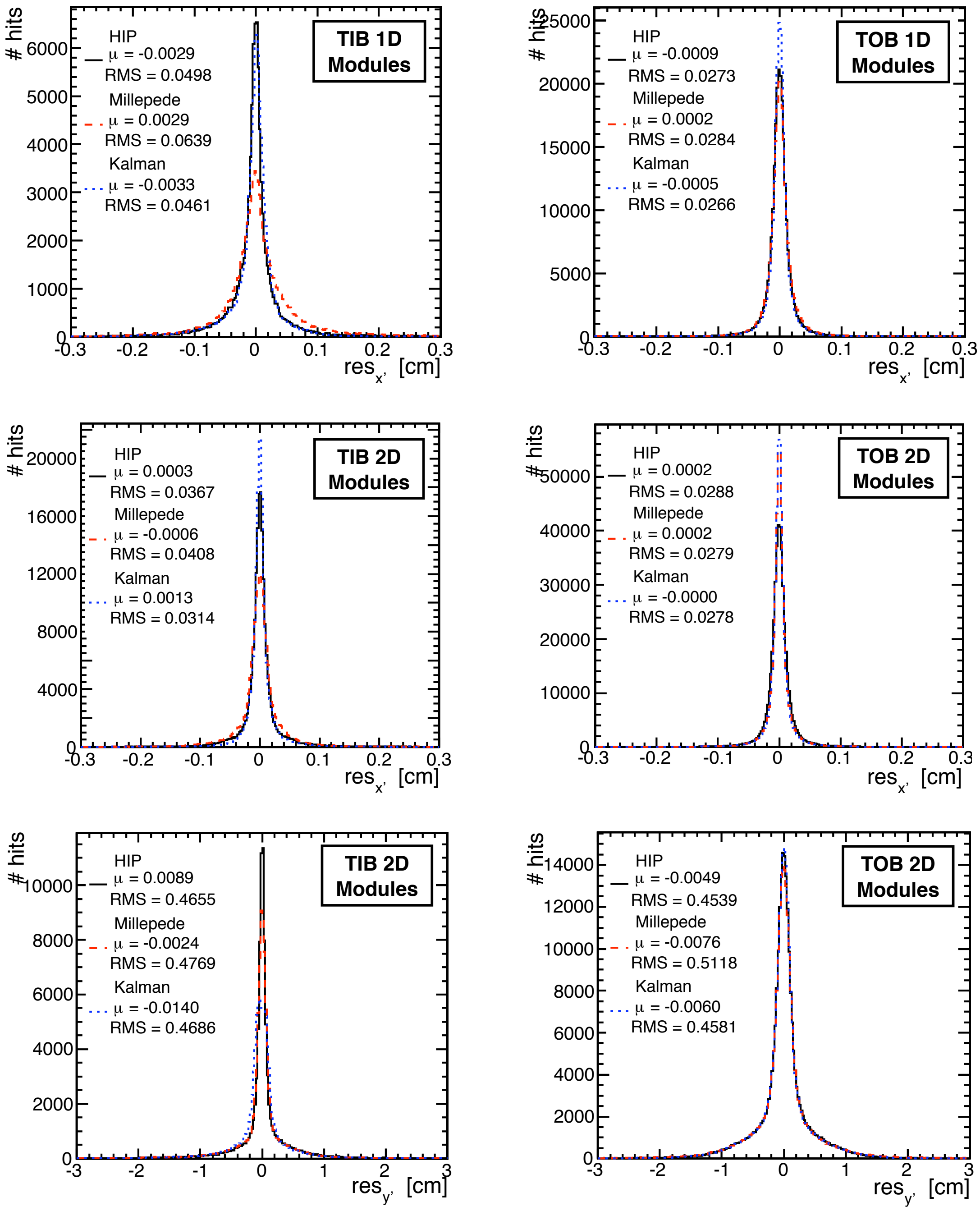


Figure 7.19: Comparison of the track residuals for HIP, MillePede II and KAA.

7.6 Summary

The Kalman Alignment Algorithm has been applied very successfully to real data from the CMS Tracker Integration Facility, where a large amount of cosmic events have been recorded. Even though only a relatively small fraction of all silicon modules of the CMS Tracker have been used for this study, the total amount of alignable objects is very well comparable to the biggest silicon tracking devices employed by today's running experiments. The validation studies indicate that the final precision of the computed alignment constants is below 150 μm. This corresponds to the expected precision, taking into account the relatively low momentum spectrum of the cosmic events and the fact that no magnetic field for momentum measurements was available.

Summing up, the performance of the KAA at the CMS Tracker Integration Facility proves that the algorithm is reliable and at least competitive with other existing alignment algorithms. This gives great confidence that the KAA will also be a benefit for the alignment efforts of the CMS Tracker Collaboration during future data-taking periods and beyond.

Chapter 8

Conclusion and Outlook

The Kalman Alignment Algorithm provides a new way to compute alignment constants with good precision even for large experimental setups. It is derived using a Kalman filter approach resulting in a sequential method, updating the alignment constants track-by-track. However, this concept comes with advantages and disadvantages. On the one hand it allows to use always the up-to-date estimate on the alignment parameters for tracking. Due to the design of the KAA no large matrices need to be inverted and no large systems of equations need to be solved. On the other hand, the size of the involved matrices may sometimes become rather large, depending on the number of alignable objects included into an update. The time spent at every update for retrieving and writing the alignment parameters and their variance-covariance matrix is a non-negligible factor that has to be dealt with[1]. In addition, if a huge amount of tracks is processed, the multiplication of several large matrices at each step cause considerable computational times in total. However, since it is possible to restrict the number of alignables at each step without degrading the final alignment precision, these issues can be overcome.

Successful examples of applying the KAA have been given in the scope of this work. Simulation studies for the case of a full-scale alignment of the CMS

[1]The restricted physical access speed to the data stored in memory is here actually not the bottle neck, but the insertion/extraction of the individually stored variance and covariance matrices into/from the full variance-covariance matrices.

Tracker have shown to give good results. The application to real data taken at the Tracker Integration Facility indicate that the KAA is also working with at least comparable performance as other, more renowned methods. All of these studies have been carried out within a reasonable amount of time and with typically available computing resources. This implies that the method is functional and well understood, and thus expedient for the data-taking period of the CMS experiment. Given that the expectations in the performance of the KAA will be met during future CMS operations, this means that existing and future experiments will have a new possibility at hand to retrieve a set of alignment constants independently from the methods in use up to now, thus allowing an additional input for validation and quality assessment of the alignment of their tracking devices.

Another possible area of application for the KAA is the alignment of the CMS Muon System. Since the software implementation of the KAA is not Tracker-specific but only expects to be fed with consistent input, i.e. meaningful derivative matrices, alignment parameters, etc., this would be a rather straightforward project.

A possible extension of the KAA would be a more sophisticated inclusion of survey measurements. By now, they are only accounted for by using a revised starting geometry. However, it should be possible to include the information coming from survey measurements in a similar way as is done currently with the information from track measurements, since alignment algorithms are in principle independent from the source of information.

Bibliography

[1] L. Evans, P. Bryant. *LHC Machine.* JINST **3**, S08001 (2008)

[2] The CMS Collaboration. *The CMS experiment at the CERN LHC.* JINST **3**, S08004 (2008)

[3] The ATLAS Collaboration. *The ATLAS Experiment at the CERN Large Hadron Collider.* JINST **3**, S08003 (2008)

[4] The CMS Collaboration. *CMS Tracker Technical Design Report.* Technical Report, CERN/LHCC 98-6, CERN (1998)

[5] The CMS Collaboration. *Addendum to the CMS Tracker TDR.* Technical Report, CERN/LHCC 2000-016, CERN (2000)

[6] S. Blusk et al. *Proceedings of the first LHC Detector Alignment Workshop.* Technical Report, CERN-2007-004, CERN (2007)

[7] J.L. Agram et al. *The silicon sensors for the Compact Muon Solenoid tracker – design and qualification procedure.* Nucl. Instrum. Meth. A **517**, pp. 77–93 (2004)

[8] L. Borrello et al. *Sensor design for the CMS Silicon Strip Tracker.* Technical Report, CMS NOTE 2003/020, CERN (2003)

[9] B. Wittmer et al. *The laser alignment system for the CMS silicon microstrip tracker.* Nucl. Instrum. Meth. A **581**, pp. 351–355 (2007)

[10] V. Karimäki, T. Lampén, F.-P. Schilling. *The HIP Algorithm for Track Based Alignment and its Application to the CMS Pixel Detector.* Technical Report, CMS NOTE 2006/018, CERN (2006)

[11] V. Blobel, C. Kleinwort. *A new method for high-precision alignment of track detectors*. In *Proceedings of the 2002 Conference on Advanced Statistical Techniques in Particle Physics*. University of Durham (2002)

[12] V. Blobel. *Software alignment for tracking detectors*. Nucl. Instrum. Meth. A **566**, pp. 5–13 (2006)

[13] Y. Saad, M. Schultz. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM J. Sci. Stat. Comp. **7**, p. 856 (1986)

[14] C. D. Jones et al. *The new CMS Event Data Model and Framework*. In *Proceedings of the CHEP06 Conference*. Tata Institute of Fundamental Research, Mumbai (2006)

[15] F. Ronga. *Track-based Alignment, Online Documentation*. https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideTrackAlignment

[16] R. Covarelli. *Tracker Survey Information, Online Documentation*. https://twiki.cern.ch/twiki/bin/view/CMS/TkSurveyInfo

[17] B. Wittmer. *Description of the LAS Software, Online Documentation*. https://twiki.cern.ch/twiki/bin/view/CMS/TkLasSoftware

[18] N. Tran. *Tracker Alignment Validation Tools, Online Documentation*. https://twiki.cern.ch/twiki/bin/view/CMS/TkAlignmentValidationTools

[19] P. Martinez. *Muon Alignment, Online Documentation*. https://twiki.cern.ch/twiki/bin/view/CMS/MuonAlignment

[20] N. De Filippis. *The performance of the CMS tracking*. Nucl. Phys. B (Proc. Suppl.) **150**, pp. 124–127 (2006)

[21] W. Adam. *Track and vertex reconstruction in CMS*. Nucl. Instrum. Meth. A **582**, pp. 781–784 (2007)

[22] E. Widl, R. Frühwirth, W. Adam. *A Kalman Filter for Track-based Alignment*. Technical Report, CMS NOTE 2006/022, CERN (2006)

[23] D. E. Catlin. *Estimation, control, and the discrete Kalman filter.* Springer, NewYork (1989)

[24] R. Frühwirth. *Application of Kalman filtering to track and vertex fitting.* Nucl. Instrum. Meth. A **262**, pp. 444–450 (1987)

[25] R. Frühwirth, T. Todorov, M. Winkler. *Estimation of detector alignment parameters using the Kalman filter with annealing.* J. Phys. G: Nucl. Part. Phys. **29**, pp. 561–574 (2003)

[26] R. Zurmühl, S. Falk. *Matrizen und ihre Anwendungen, 5th edition.* Springer, Berlin–Heidelberg (1984)

[27] A. Strandlie, W. Wittek. *Propagation of Covariance Matrices of Track Parameters in Homogeneous Magnetic Fields in CMS.* Technical Report, CMS NOTE 2006/001, CERN (2006)

[28] R. Frühwirth et al. *Data Analysis Techniques for High-Energy Physics.* Cambridge University Press, Cambridge (2000)

[29] Frank R. Hampel et al. *Robust Statistics.* Wiley, NewYork (1986)

[30] R. Frühwirth et al. *Vertex reconstruction and track bundling at the LEP collider using robust algorithms.* Comput. Phys. Commun. **96**, pp. 189–208 (1996)

[31] I. Belotelov et al. *Simulation of Misalignment Scenarios for CMS Tracking Devices.* Technical Report, CMS NOTE 2006/008, CERN (2006)

[32] T. Lampén et al. *Comprehensive Set of Misalignment Scenarios for the CMS Tracker.* Technical Report, CMS 2008/029, CERN (2008)

[33] The CMS Collaboration. *The 2008 CMS Computing Software and Analysis Challenge.* Technical Report, CMS NOTE in preparation.

[34] The CMS Collaboration. *The Computing Project Technical Design Report.* Technical Report, CERN/LHCC 2005-023, CERN (2005)

[35] D. Acosta et al. *A T0 Architecture for the CMS Experiment.* Technical Report, CMS NOTE 2006/095, CERN (2006)

[36] P. Kreuzer. *CMS CAF (CERN Analysis Farm), Online Documentation.* https://twiki.cern.ch/twiki/bin/view/CMS/CAF

[37] R. Mankel. *The MillePede Production System (MPS), Online Manual.* https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideMillepedeProductionSystem

[38] Platform LSF – Load Sharing Facility. http://www.platform.com/Products/platform-lsf

[39] The CMS Tracker Collaboration. *Track Reconstruction with Cosmic Ray Data at the Tracker Integration Facility.* Technical Report, CMS NOTE in preparation.

[40] The CMS Tracker Collaboration. *Silicon Strip Tracker Detector Performance with Cosmic Ray Data at the Tracker Integration Facility.* Technical Report, CMS NOTE in preparation.

[41] The CMS Tracker Collaboration. *CMS Tracker Alignment at Integration Facility.* Technical Report, CMS NOTE in preparation.

[42] D. Sprenger. *Track-based Alignment of a CMS Tracker Endcap.* Diploma thesis, CMS TS-2008/025, CERN (2008)

Curriculum Vitae

Persönliche Daten

	Dipl.-Ing. Edmund Erich Widl
	Badgasse 1–7/7/12 1090 Wien
	Geboren am 9. Februar 1979 in Wien

Ausbildung

1985 – 1989	Volksschule, Grünentorgasse, Wien	
1989 – 1997	Gymnasium	
	1989 – 1991	Bundeskonvikt Himmelhof, Wien
	1991 – 1997	BRG 9, Erich Fried Realgymnasium, Glasergasse, Wien
1997 – 2004	Diplomstudium der Technischen Physik, TU Wien	
	2001 – 2002	Tutor für *Thermodynamik – Rechenübungen*
	Sommer 2002	CERN Summer Student
	2002 – 2003	Tutor für *Physik – Rechenübungen*
2004 – 2008	Doktoratsstudium der technischen Wissenschaften, TU Wien	
	Sommer 2005	CERN School of Computing

Weitere Aktivitäten

1995 – 1997	Stellvertretender Schulsprecher am BRG 9

1997 – 2004	Ehrenamtliche Mitarbeit bei der *Studienvertretung Technische Physik* (Fachschaft Physik): gewählter Studienrichtungsvertreter (2001–2003), Kommissionsmitglied der *Studienkommission Technische Physik*, studentisches Mitglied zahlreicher Habilitationskommissionen.
2004 – 2008	Ehrenamtliche Mitarbeit bei der *Studienvertretung Doktorat* (Fachschaft Doktorat): zweimaliger gewählter Studienvertreter (2005–2008), Kommissionsmitglied der *Studienkommission Doktorat*, Mitarbeiter der Plattform *doktorat.at*