# Iterative local $\chi^2$ alignment algorithm for the ATLAS Pixel detector

Diploma Thesis
supervised by
Prof. Dr. E. Umbach
and
Prof. Dr. S. Bethke

Tobias Göttfert

Fakultät für Physik und Astronomie
Julius-Maximilians-Universität Würzburg
and
Max-Planck-Institut für Physik
Werner-Heisenberg-Institut

Munich, May 26, 2006

## Abstract

ATLAS is one of the particle detectors for the Large Hadron Collider currently under construction at the CERN laboratory. Its Inner Detector, which is responsible for accurate track reconstruction of charged particles, consists of three subdetectors: the Transition Radiation Tracker (TRT), the SemiConductor Tracker (SCT) and the Pixel detector.

The existing local $\chi^2$ alignment approach for the ATLAS SCT detector was extended to the alignment of the ATLAS Pixel detector. This approach is linear, aligns modules separately, and uses distance of closest approach residuals and iterations. The derivation and underlying concepts of the approach are presented. To show the feasibility of the approach for Pixel modules, a simplified, stand-alone track simulation, together with the alignment algorithm, was developed with the ROOT analysis software package.

The Pixel alignment software was integrated into Athena, the ATLAS software framework. First results and the achievable accuracy for this approach with a simulated dataset are presented.

# Contents

# Zusammenfassung

ATLAS ist eines der Teilchenkollisionsexperimente, die derzeit am Large Hadron Collider (LHC) am CERN in Genf entstehen. Der Innere Detektor, der für die Spurrekonstruktion geladener Teilchen veranwortlich zeichnet, besteht aus einem Driftröhrendetektor namens Transition Radiation Tracker (TRT), dem SCT-Halbleiterstreifendetektor (SemiConductor Tracker) und dem Pixeldetektor. Bei Teilchendetektoren bezeichnet Alignment die möglichst präzise Ausrichtung und Ortsbestimmung aller individuellen Detektorelemente, um die höchstmögliche Ortsauflösung erzielen zu können. Bei Spurdetektoren kommt dabei spurbasiertes Alignment als letztes Glied in der Kette zum Einsatz, um bei bereits präzise platzierten Modulen die Position mit der höchstmöglichen Genauigkeit zu bestimmen.

Der existierende lokale $\chi^2$ Alignmentansatz für den ATLAS SCT Detektor wurde erweitert, so dass er auch den Pixeldetektor ausrichtet.

Der hier verwendete Ansatz basiert auf einer linearisierten kleinste-Quadrate-Anpassung, richtet Module unabhängig voneinander aus und verwendet kürzeste Abstände im Raum zwischen Spuren und Treffern auf den Modulen, sogenannte Residuen. Die Herleitung des iterativen Verfahrens und zugrundeliegende Konzepte werden dargestellt. Eine Simulation mittels des Softwarepaketes ROOT wurde geschrieben und zeigt die Durchführbarkeit und Präzision des Verfahrens für Pixelmodule.

Diese Erweiterung auf Pixelmodule wurde dann in die ATLAS-Softwareumgebung Athena eingearbeitet. Erste Resultate und erreichbare Genauigkeiten, basierend auf simulierten Daten, werden präsentiert.

- Kapitel 1 geht zunächst auf den LHC und seine Experimente, insbesondere ATLAS, ein. Die Funktionsweise eines Pixeldetektors wird erläutert. Der Begriff des Alignment wird eingeführt sowie Motivation und benötigte Präzision angegeben.
- Kapitel 2 umfasst die mathematische Herleitung des verwendeten Minimierungsverfahrens für spurbasiertes Alignment. Die Annahmen, unter denen das Verfahren funktioniert, werden dargestellt.
- Kapitel 3 beschreibt die Simulation, die im Vorfeld im Programmpaket ROOT erstellt wurde, um den Ansatz auf seine Präzision zu testen. Die Berechnung aller eingehenden Größen wird dargestellt und die erreichbare Genauigkeit ermittelt.
- Kapitel 4 zeigt die Einarbeitung des Verfahrens in die ATLAS-Softwareumgebung Athena. Es wird ein Überblick über die Programmstruktur gegeben und Verteilungen aller in das Verfahren eingehenden Größen dargestellt.

- Kapitel 5 umfasst die Analyse und Ergebnisse der Methode. Die mit den vorhandenen Simulationsdaten erreichbare Genauigkeit wird ermittelt. Die Konvergenz des Algorithmus unter Iterationen und Studien mit verschobenen Detektorpositionen werden präsentiert.

# Chapter 1

# Introduction

## 1.1 Standard model of particle physics

The standard model of particle physics (SM) consists of quantum field theories which are found to describe the elementary particles and their interactions precisely up to an energy scale of $\mathcal{O}(200~\text{GeV})$ [1, 2]. The elementary particles are spin-1/2 fermions exchanging field quanta, which are spin-1 gauge bosons and which mediate the forces between fermions. The bosons arise from the requirement of local gauge invariance of the fermion fields and are manifestations of the symmetry group of the standard model, which is SU(3)×SU(2)×U(1).

The SM describes three generations of quarks and leptons, the fundamental fermions. Their interactions belong to two different sectors: the strong interaction, mediated by gluons and described by quantum chromodynamics (QCD), and the electroweak interaction, mediated by photons, $W^{\pm}$, and Z bosons. The electroweak interaction is a unification of the weak interaction and the electromagnetic interaction.

A key feature of the SM is the mechanism of spontaneous breaking of the electroweak symmetry group SU(2)×U(1). An attempt to describe this, the Higgs mechanism, leads to the predicted existence of a massive neutral scalar boson, the Higgs boson [3]. The Higgs boson is the only fundamental particle predicted by the SM which has not yet been discovered.

## 1.2 Large Hadron Collider

The Large Hadron Collider (LHC) is a ring collider, which is currently under construction in the 27 km circumference old LEP[1]-tunnel at the CERN laboratory near Geneva [4, 5]. Its main purpose is to collide two proton beams, but it can also be filled with lead ions for special heavy ion runs. The LHC is scheduled to begin operation in 2007. The main purpose of the LHC and its five experiments is to probe the standard model, especially the mechanism of electroweak symmetry breaking, and to explore possible new physics beyond the standard model in the TeV energy region.

---

[1] Large Electron-Positron collider

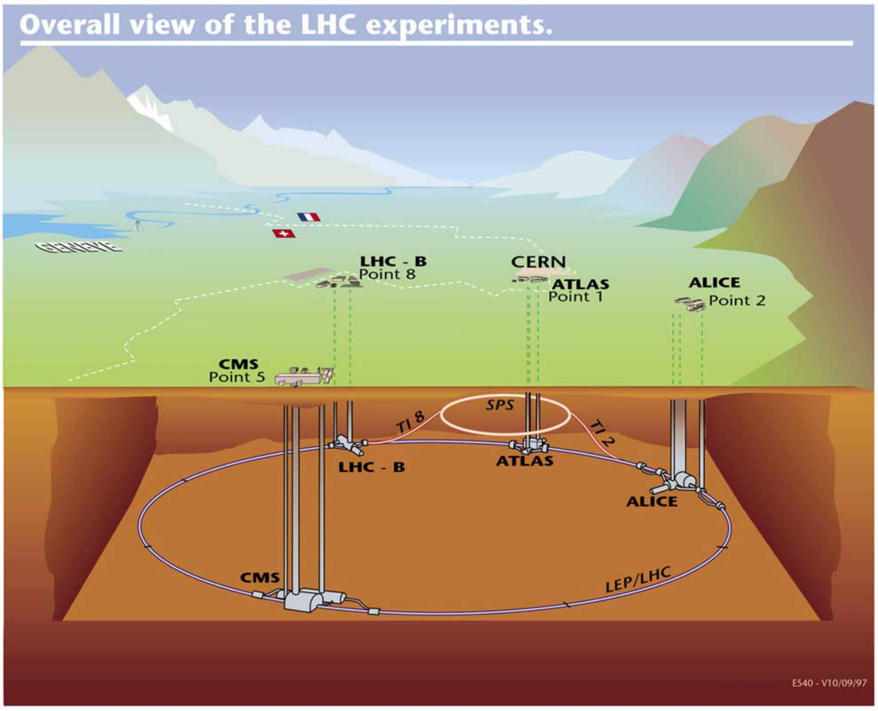

Figure 1.1: *Schematics of the LHC ring with the four experimental caverns [10].*

The LHC will have four interaction points. Two of them are for the multipurpose experiments ATLAS (A Toroidal LHC ApparatuS) [6] and CMS (Compact Muon Solenoid) [7], the other two for the LHCb experiment, which is dedicated to b-physics [8], and the heavy-ion experiment ALICE (A Large Ion Collider Experiment) [9]. The positions of the interaction points within the LHC ring are depicted in figure 1.1.

The energy of the protons within the bunches in the two storage rings will be 7 TeV, providing a nominal center-of-mass collision energy of 14 TeV. However, as the colliding particles are not elementary, only a fraction of this energy is carried by the individual constituents of the proton. It should be possible to probe energy regions up to 5 TeV.

The operation of the LHC will consist of two phases. The first phase will be a phase of low-luminosity running, where the expected instantaneous luminosity will be $10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. It is planned to attain high luminosity running after three to four years of operation, which will bring instantaneous luminosities up to $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ [5].

With this luminosity, one proton bunch consists of about $10^{11}$ protons. On average, one hard parton-parton interaction will happen within one bunch-crossing, together with about 20 so-called minimum bias events, producing up to 1000 tracks in the detectors [11].

A view into the tunnel, as shown in fig. 1.2 shows the already installed sections of the beampipe with the bending magnets integrated.

Figure 1.2: *View of the LHC tunnel with already installed parts of the accelerator ring [10].*

## 1.3 ATLAS detector

The ATLAS detector is designed to make full use of the available luminosities during LHC high-luminosity running [6].

Amongst the items on the physics program of ATLAS [12] are a number of points, which influenced the design of the detector. The most important points are:

- The search for the Higgs boson. If the Higgs mechanism is correct, the mass of the Higgs boson is already constrained from below by previous experiments and from above by theoretical arguments, suggesting an allowed range of 114.4 GeV/$c^2$ to about 1 TeV/$c^2$. In this case, the Higgs boson will be detectable by ATLAS.

- The tests of supersymmetric extensions of the SM, for example MSSM[2], and other new physics beyond the SM such as extra space dimensions or dark matter candidates. If the lightest supersymmetric particle (LSP) is stable this will lead to missing energy in supersymmetric events. The detection of more than one Higgs particle would also be a clear sign for supersymmetry.

- Precise measurements of standard model particles. The top quark is of particular interest; so far only few of its expected properties have been verified. Since the LHC will be a top-quark as well as a bottom-quark factory, ATLAS should be capable of measuring decays of t- and b-quarks with a high rate.

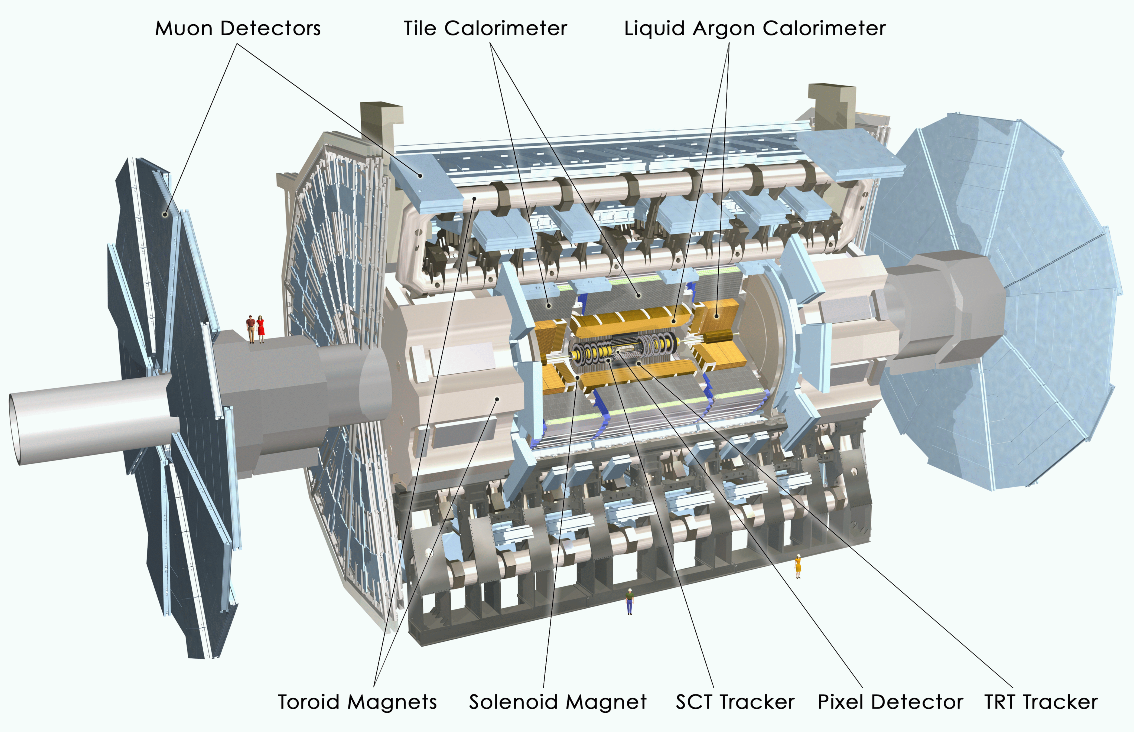

Figure 1.3: *Schematic view of the ATLAS detector [10].*

The ATLAS detector is shown in fig. 1.3 with its major components. It is divided into three main parts (from outside to inside):

- The Muon system
- The Calorimeter system
- The Inner Detector

The muon system [13] composes most of the actual volume of ATLAS. The muon chambers are positioned within a huge magnet system composed of air-core toroid magnets [14–16]. The barrel and the two endcap parts of the magnet system are shown separately in fig. 1.4. They produce a toroidal magnetic field of up to 4 T in the outer part of ATLAS. Two types of tracking detectors are used within the magnets: Monitored Drift Tubes (MDT) and Cathode Strip Chambers (CSC). For triggering, additional Resistive Plate Chambers (RPC) and Thin Gap Chambers (TGC) are used. The geometrical shape of the whole muon system is surveyed (monitored) by an optical laser system during running.

The barrel calorimeter system of ATLAS consists of the electromagnetic calorimeter (EMC), covering[3] $|\eta| < 3.2$, and the accordeon-shaped hadronic calorimeter (TileCal), covering $|\eta| < 1.7$. The hadronic calorimeter has an additional endcap (HEC) covering the range $1.5 < |\eta| < 3.2$. In the $\eta$-region $3.2 < |\eta| < 4.9$, the forward calorimeter (FCAL) is used. All types of ATLAS calorimeters are sampling calorimeters, but utilize different

---

[2] Minimal Supersymmetric extension of the Standard Model

[3] $\eta$ is the pseudorapidity, c.f. section 1.6

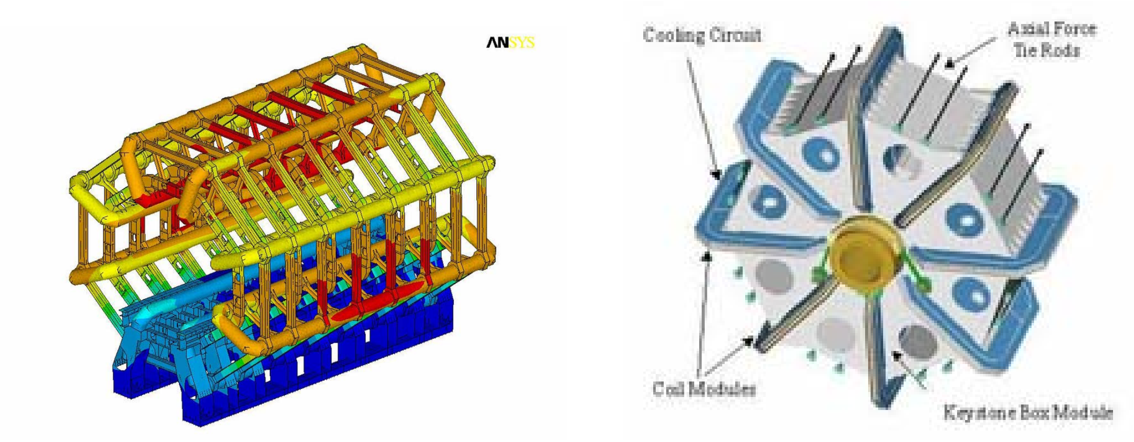

Figure 1.4: *Schematic view of the ATLAS air-core toroidal magnet system. Barrel part (left) and one endcap (right) are shown separately [10].*

types of absorbers and active materials. The TileCal uses scintillating plastics as active material, whereas all other calorimeters use liquid argon. The iron absorber of the TileCal also serves as flux return yoke for the solenoidal magnetic field of the inner detector. The EMC has lead as absorber, the HEC uses copper and the FCAL has copper in its electromagnetic and tungsten in its hadronic part [17–19].

The inner detector [20, 21] is responsible for accurate track reconstruction of charged particles and determination of secondary vertices. It is located in a 2 T magnetic field created by a solenoid magnet [22] and consists of three subdetectors:

- The Transition Radiation Tracker (TRT) comprises 600 000 kapton straw tubes, each with a gold wire to measure a particle trajectory with a high number of points. In addition, plastic foils are interspersed amongst the tubes to stimulate the production of transition radiation photons for high-$\beta$-particles. This makes electron identification possible. The TRT has a length of 6.80 m, an inner diameter of 1.12 m and an outer diameter of 2.14 m and therefore is responsible for the overall size of the inner detector.

- The SemiConductor Tracker (SCT) is a silicon strip detector with 4088 modules. Each of the modules has two pairs of wafers glued to a common support. Modules sitting in the barrel part have a different geometry than those on the endcaps. Barrel modules have rectangular wafers whereas endcap modules have trapezoidal wafers. In both cases the wafer pairs are rotated by an angle of 40 mrad with respect to each other to have sensitivity not only perpendicular to but also along the strips. Each wafer pair has 768 readout strips. In total, four barrel layers and nine endcap disks on both sides of the barrel result in about 6.3 million readout channels.

- The Pixel detector is the innermost part of the inner detector and is responsible for a precise measurement of spatial points as close as possible to the interaction point. The number of readout channels is about 80 million in total. The Pixel detector has to stand the hardest radiation conditions of all detectors while retaining good precision and efficiency.

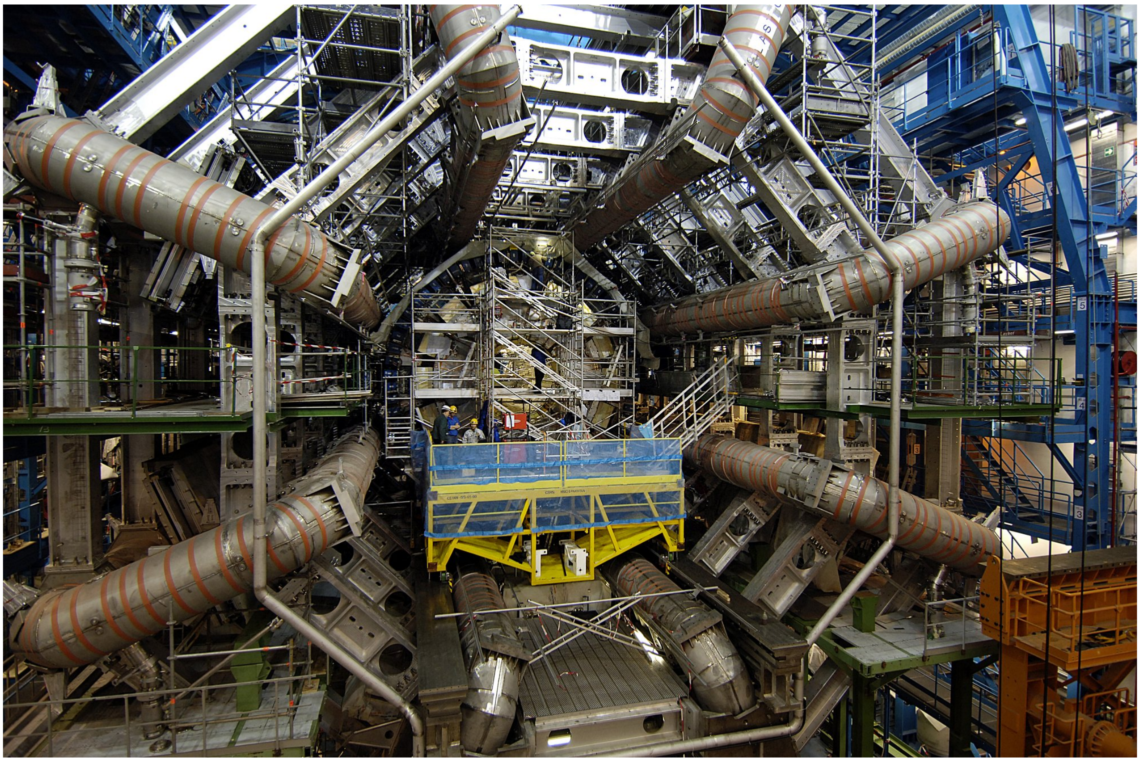

Figure 1.5: *A view of the cavern where the ATLAS detector is being constructed. The eight toroid magnets can be seen as well as the central part of the calorimeter, which is installed into its position within the toroids. The picture was taken in December 2005 [10].*

The status of the construction of the ATLAS detector at the beginning of 2006 can be seen from fig. 1.5.

## 1.4 Pixel detector

### 1.4.1 Layout

The Pixel detector [23] is a silicon detector consisting of 1744 rectangular modules, each of which covers a total sensitive area of $16.4 \times 60.8\,\mathrm{mm}^2$. Each module carries 47232 pixels of size $50 \times 400\ \mu\mathrm{m}^2$. This gives a 2-dimensional position measurement of charged particles within the plane of the module.

The layout of the Pixel detector is depicted in fig. 1.6. The detector is of cylindrical shape and is separated into a central barrel part and two endcaps. With dimensions of 130 cm in length and 38 cm in diameter, it covers a pseudorapidity range of $|\eta| < 2.5$.

The barrel consists of 3 cylindrical layers of modules with radii of 5.05, 8.85 and 12.25 cm. These layers are built of 22, 38 and 52 so-called staves, respectively, each holding 13 modules. Thus, a total of 1456 modules compose the barrel part of the detector. The staves are inclined by 20° with respect to the radial direction from the beamline to create an overlap between adjacent modules (turbine arrangement). Additionally, the modules on

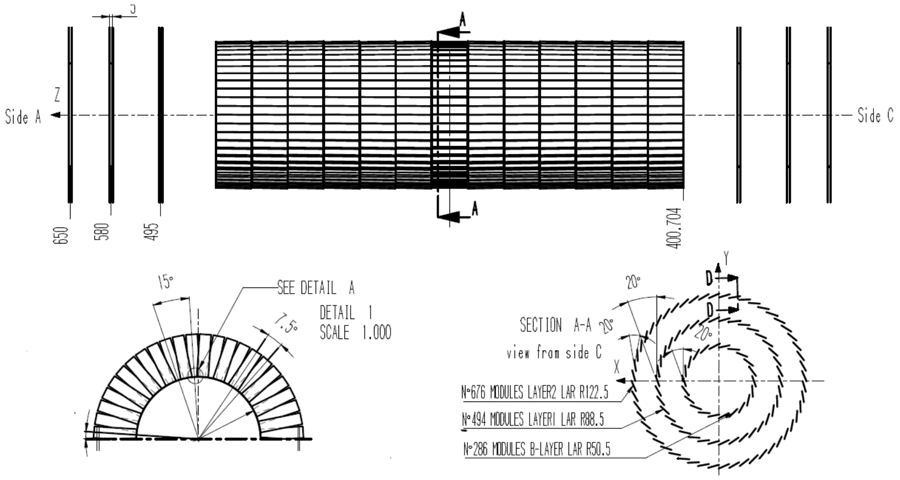

Figure 1.6: *Geometry of the ATLAS Pixel detector. The sketch shows the overlap in the modules on one endcap disk, as well as the turbine-arrangement of the barrel staves, which in turn have the individual modules mounted in a shingled-stave layout. Gaps between modules are thus avoided. Dimensions are given in mm [24].*

one stave are arranged in the so-called shingled stave layout to create an overlap in the global $\hat{z}$-direction[4]. The long sides of the modules are oriented parallel to the beamline as well as to the long side of each individual pixel.

Each of the two endcaps has three discs, which have the modules mounted radially to the beamline, with their normal axis parallel to the beam. A disc is comprised of 8 sectors with 6 pixel modules each. This adds up to 2 × 144 modules on the endcap discs. When the pixel detector is fully built up, it will look like portrayed in fig. 1.7.

### 1.4.2 Principle of operation

In semiconductor particle detectors, p- and n-doped semiconductors are brought together to form a p-n junction [25]. The mobile charge carriers diffuse into the junction and recombine, leaving the fixed donor and acceptor atoms behind as spatial charges. This creates a depletion region free of mobile charge carriers and an intrinsic electric field which counterbalances the diffusion of mobile charge carriers into the depletion region. In particle detectors, an additional external reverse bias voltage is applied to deplete the bulk as much as possible. Electron-hole pairs created by incident charged particles are separated by the electric field in this region and drift towards the electrodes. This charge is then collected and read out usually on one side of the detector. In pixel detectors, the two-dimensional separation of the readout pads makes it possible to have a two-dimensional position measurement.

The ATLAS Pixel modules are silicon sensors with $n^+$-pixel implants on an n-type bulk. The back contact is p-type (see fig. 1.8). The depletion region thus extends from the p-n junction at the backside. In the beginning, the reverse bias voltage must therefore be high

---

[4] for definition of coordinate systems see section 1.6

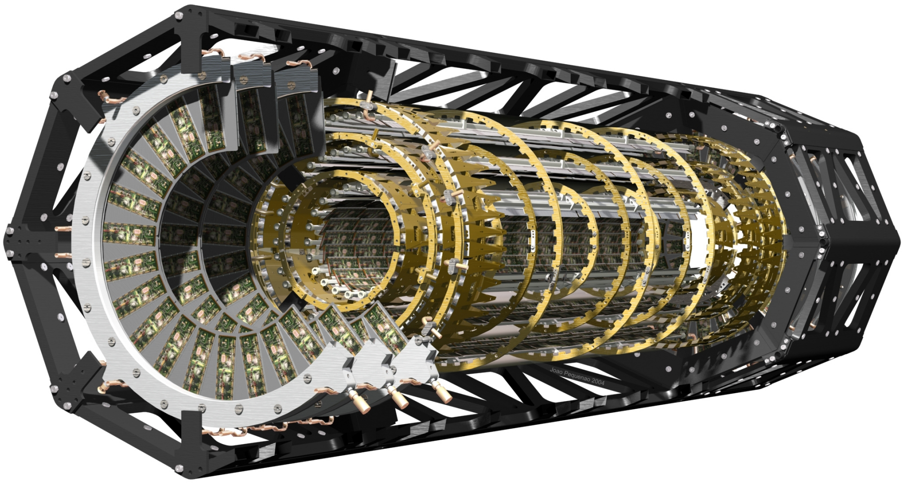

Figure 1.7: *Raytraced image of the Pixel detector with an intersection cut away. One can clearly see the three discs of one endcap and the three barrel layers residing in the support frame [10].*

enough so that the sensor is fully depleted. This is because the depletion region needs to reach the pixel implants so that any created charge can drift there to be read out.

The $n^+$-on-n-design is chosen deliberately, because it is known that the long period of operation in extremely intense radiation will lower the impurity concentration and finally even type-invert the bulk to be effectively p-type. After a few months in the ATLAS detector, radiation damage will turn the bulk into effective p-type and the depletion zone will extend from the $n^+$ pads, making the detector usable even when it is not fully depleted. However, with increasing radiation damage, more and more bias voltage will be needed to keep the depletion region large. The required bias voltage will range from about 150 V at the beginning up to 600 V after ten years of operation, which is the maximum depletion voltage possible for the sensor. Additionally, the silicon used is enriched with oxygen, which makes it more radiation hard.

The 250 $\mu$m thick silicon wafers are read out by 16 front-end (FE) chips. The chips provide 2880 channels each, which are bump-bonded with lead or indium solder bumps directly to the corresponding pixel on the silicon wafer (so-called flip-chip assembly). A Pixel module has a total of $144 \times 320 = 46080$ readout channels. Therefore, due to the necessary gap between adjacent FE chips, some of the 47232 pixels have to be grouped together on one readout channel. These form the so-called long pixels and ganged pixels. During track reconstruction, this irregular geometry due to fabrication necessities must be taken into account. A detailed sketch of this interchip region is shown in fig. 1.9.

The FE chips perform amplification, leakage current subtraction, signal shaping and threshold discrimination and finally output the time-over-threshold (ToT) together with a timestamp. On the other side of the detector substrate, the module control chip (MCC) performs signal collection, multiplexing and optical data transmission to the readout system.

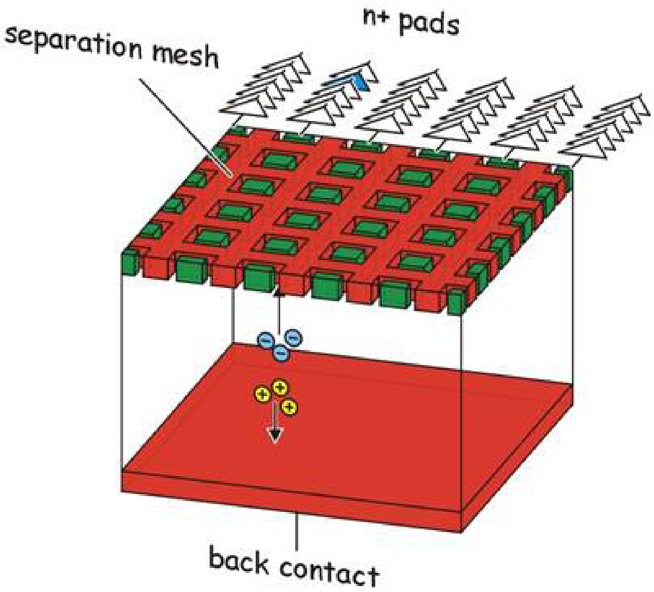

Figure 1.8: *Principle of operation of a pixel detector: electron/hole pairs generated by an ionizing particle are accelerated towards the electrodes. The signal is collected and read out at the* $n^+$ *pads. [26]*

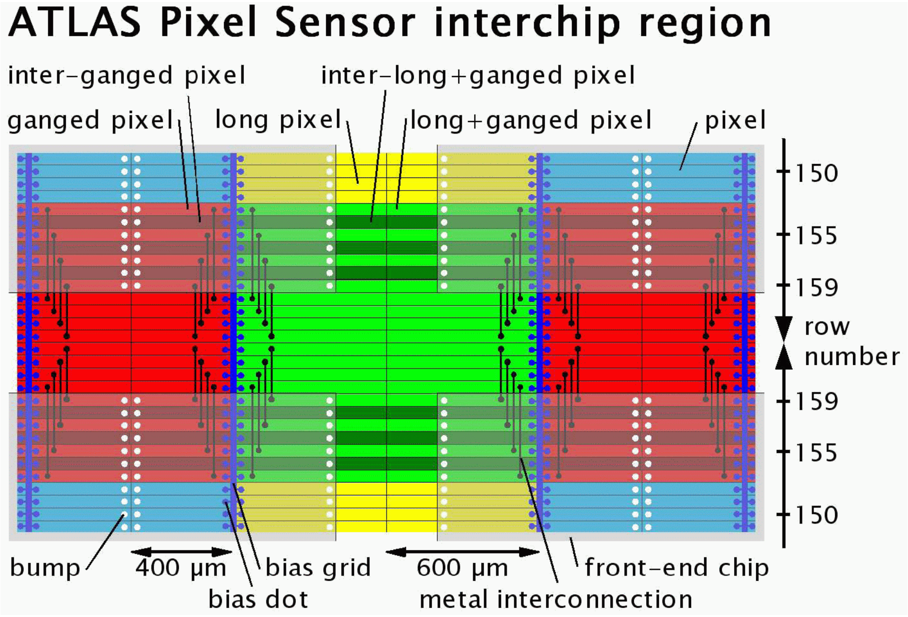

Figure 1.9: *The interchip region between adjacent FE chips on a Pixel module [27, 28]. The gap in the pixel columns is covered by enlarged pixels, the so-called long pixels. In the pixel rows, two pixels at a time are connected (ganged) to be read out together by one channel of the FE chip.*

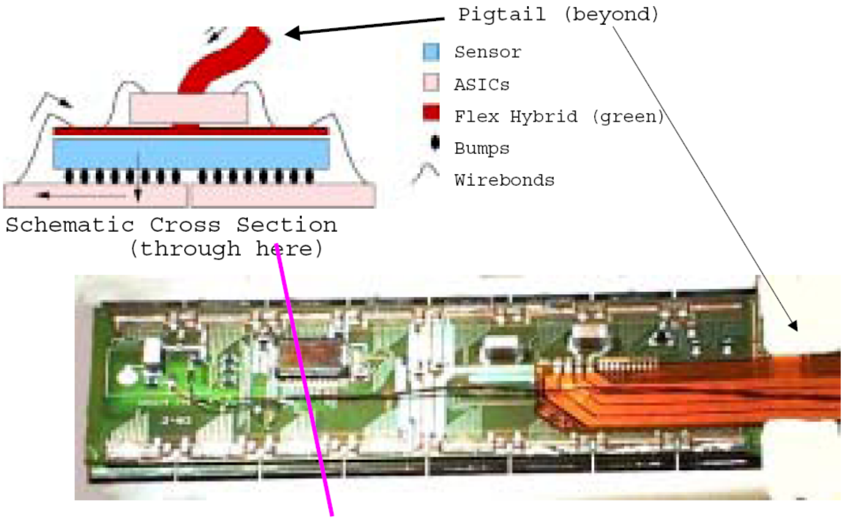

Figure 1.10: *Pixel module geometry. The cross-section of a module illustrates the attachment of the readout electronics on both sides of the sensor. Each module has a "pigtail" which connects to the readout cables. [29]*

## 1.5 Non-track-based alignment

For every position-sensitive detector in high energy physics it is essential that the position of each element be known very well to meet the detector performance requirement. The process of determining the true positions of the active elements of a detector is called alignment. For ATLAS, a three-stage strategy is foreseen: First, the detector parts will be built with as high as-built precision as possible and various geometrical quantities will subsequently be monitored (by optical means, mounting robot logfiles etc.). Second, some of the support structures will be monitored during running, for example with frequency scanning interferometry or other optical means, to detect movements and thermal distortions of bigger detector parts. Third, an accurate determination of geometrical positions will be performed using the particle tracks which traverse the detector themselves. The positions determined during alignment will be fed into the geometry database of the detector and will later be available for track reconstruction; no actual movement of the detector parts is done. Alignment based on tracks will be the topic of the next chapters of this thesis.

The expected precision of the various methods and the precision required to meet the standards given by the Inner Detector technical design report (TDR) [20] can be seen in table 1.1. The precision numbers are based on the requirement that any misalignment of the inner detector parts must not degrade the reconstructed track parameters by more than 20%. However, to exploit the full precision that ATLAS is capable of, for example in order to measure the W mass with a resolution of 15 MeV, much more ambitious requirements must be fulfilled. It is estimated that in this case, the alignment precision for the sensitive pixel coordinate x has to be as good as 1 $\mu$m, which is significantly better than the intrinsic resolution of the detector.

The difference in the initial positioning precision between barrel and endcap modules comes

| Pixel alignment precision | | | | |
|---|---|---|---|---|
| | Pixel barrel | | Pixel endcap | |
| local coordinate | required | as-built | required | as-built |
| x | 7 $\mu$m | 50 $\mu$m | 7 $\mu$m | 4.6 $\mu$m |
| y | 20 $\mu$m | 20 $\mu$m | 20 $\mu$m | 4.7 $\mu$m |
| z | 10 $\mu$m | 50 $\mu$m | 100 $\mu$m | 12.7 $\mu$m |

Table 1.1: *Comparison of required alignment precision for Pixel modules as given by [20] with the initial as-built precision from the survey [30].*

from the existence of a very precise optical survey for the endcap. However, the survey can only measure module-to-module positions and therefore the absolute position in space of the endcaps is not known to the precision given in the table. For the barrel, survey information is not completely available, so that the initial precision basically depends on the mounting precision of the modules on the support structures.

The only method capable of improving the present precision for each individual module, and thus of fulfilling the strict requirements, is track-based alignment, which will be the topic of the next chapter.

## 1.6 Coordinate conventions

Within this thesis, two of the ATLAS coordinate systems are used [31]:

The *global coordinate system* is called the tracking frame. It is a right-handed three-dimensional orthogonal coordinate system whose z-axis is parallel to the direction of the magnetic field in the inner detector. The origin lies at the nominal interaction point, and the x-axis points to the center of the LHC ring. We will denote coordinates within this frame with ($\hat{x}$, $\hat{y}$, $\hat{z}$). Directions are usually given using the angles $\theta$, which is the deflection with respect to the $\hat{z}$-axis, and $\phi$, which is the angle around the $\hat{z}$-axis starting from $\hat{x}$.

It is worthy to note that in collider experiments, the pseudorapidity $\eta$ is often used as a variable to describe kinematic quantities. For relativistic particles the pseudorapidity is a good approximation of the true rapidity; the two quantities are identical for massless particles. The pseudorapidity is defined as

$$\eta = -\ln \tan \frac{\theta}{2} \tag{1.1}$$

and often given instead of $\theta$, because the density of particles produced in hadron collisions is known to be approximately uniform in $\eta$.

The *local coordinate system* of each module (x, y, z) is a unique system for each detector module. For silicon modules, it lies in the corresponding plane of the detector with the origin at the center of the module. The x-axis runs in the direction of the short coordinate or across the strips, the y-axis runs in the direction of the long coordinate or along the strips and the z-axis is perpendicular to the surface and oriented away from the interaction point. In the following, rotations around these local axes will be denoted with ($\alpha$, $\beta$, $\gamma$).

# Chapter 2

# Track-based alignment

## 2.1 Alignment based on residuals

As outlined in the introduction, another method in addition to the survey of the detector system is needed to achieve the necessary alignment accuracy. To reach the desired goal, the direct use of tracks traversing the detector during operation achieves the highest precision of any method available. All track-based alignment methods are based on the minimization of residual distributions. The residual is hereby defined as “the distance” between a track and its associated hit on a module surface.

The reconstructed track is the estimate of the true particle trajectory going through the tracking detector and plays a crucial role in all track-based alignment approaches. The ATLAS tracking detector is designed such that in general a charged particle leaves more than enough hits on the detector to constrain the set of equations necessary to describe a particle trajectory. On average, this will be 3 hits in the Pixel detector, 8 hits in the SCT and about 30 hits in the TRT. A helical track model for a charged particle traveling through a magnetic field needs five track parameters. In this way, it is ensured that under normal circumstances a statistical fit with error estimates can be performed. In the tracking software of a detector, two steps must be performed to reconstruct tracks from the many detector hits in one event. First, pattern recognition flags certain hits from the sample as belonging to one common particle trajectory. Then the actual track fitting algorithm takes over and estimates the optimal parameters for the assumed tracking model to fit the observed hit coordinates.

Over many tracks, the distribution of hits on a detector module will be roughly uniform. The distribution of the residuals for a perfectly aligned module is then centered around zero with a gaussian shape. The shape is due to the acceptance for an individual detector element or a cluster to give a signal, which is dependent on the actual charge deposition position of the incident particle. A clustered hit arises if the path of a particle traversing the detector material is long enough so that the generated charge fires more than one readout channel. The position of the particle hit is then determined by taking into account position and charge information from these neighboring channels. Moreover, track fitting includes hits closer to the fitted particle trajectory more often than hits further away, which also contributes to the gaussian shape.

If the detector module is not centered at its nominal position, then the residual distributions will be shifted away from zero (c.f. fig. 2.1). In general, track based alignment relies on the minimization of residual distributions for some or all degrees of freedom of each detector module.

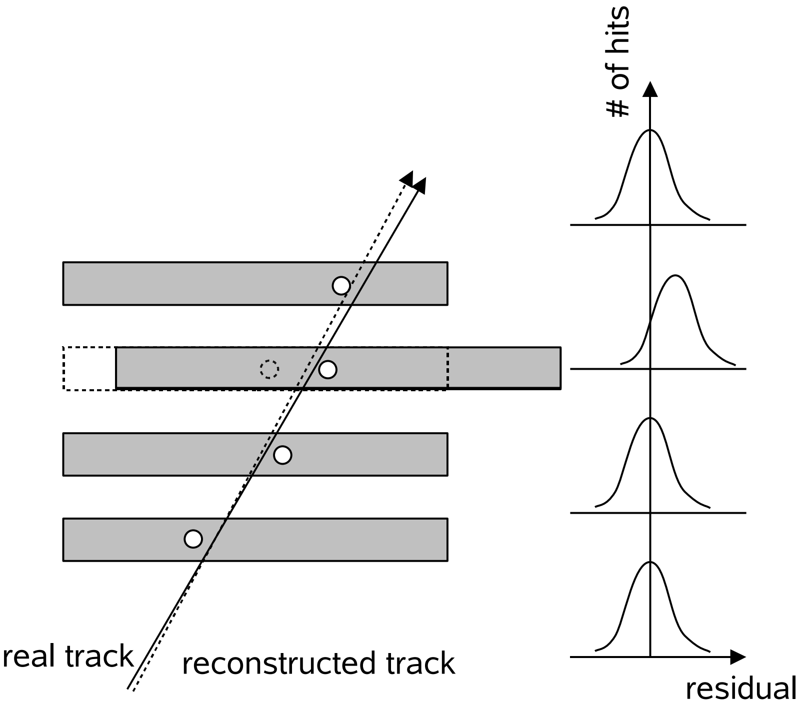

Figure 2.1: *Four layers of modules, one of which is shifted away from its nominal position. The real positions (shown in grey) are not known, thus the reconstructed track uses the nominal position (dashed) which degrades the fit quality. The residual distribution of this layer will be biased to one side. This is the basis for track-based alignment.*

In the next section, the mathematical background of the used alignment approach will be presented.

## 2.2 Algebraic derivation of the $\chi^2$-minimization algorithm

The proposed approach is a linearized $\chi^2$-minimization which treats every module independently (so-called local approach), as it was used e. g. for the BABAR SVT detector [32]. It is described in detail in [33]. Here the major formalism will be presented in brief outline.

Having the vector of all residuals for a given track $i$, $\vec{r}_i = \vec{r}_i(\vec{a}, \vec{\pi}_i)$, the following $\chi^2$-function is defined:

$$\chi^2(\vec{a},\vec{\pi}_1,\ldots,\vec{\pi}_m)=\sum_{i\,\in\,tracks}^{m}\vec{r_i}^{\,T}(\vec{a},\vec{\pi}_i)\cdot V_i^{-1}\cdot\vec{r}_i(\vec{a},\vec{\pi}_i). \tag{2.1}$$

Here, $\vec{r}_i(\vec{a},\vec{\pi}_i)$ depends on the alignment parameters $\vec{a}$ and the track parameters $\vec{\pi}_i$. $\vec{a}$ are all alignment parameters of all modules that have hits on one of the tracks. $\vec{\pi}_i$ is the vector of track parameters for track $i$. This can be e. g. 4 parameters for a straight line track model or 5 parameters for a helical track model. m is the total number of tracks used for the alignment. $V_i$ is the covariance matrix of the residuals for track $i$.

It is noteworthy that $r_i$ has two entries per traversed module for both SCT and Pixel modules: SCT modules have two wafer sides and therefore in general two hits per particle track, which are one-dimensional measurements. Pixel modules have one hit, which is a two-dimensional measurement, and therefore two residuals per hit on a module.

For a perfectly aligned detector, it is assumed that a minimal $\chi^2$ is reached:

$$\frac{d\chi^2(\vec{a})}{d\vec{a}}=\vec{0}, \tag{2.2}$$

with the derivative with respect to the alignment parameters defined as:

$$\frac{d}{d\vec{a}}\equiv\begin{pmatrix}\frac{d}{da_1}\\ \vdots\\ \frac{d}{da_n}\end{pmatrix}. \tag{2.3}$$

Here, $n$ is the total number of modules times the number of degrees of freedom for each module.

To solve equation 2.2, the left side is expanded in a linear Taylor expansion:

$$\frac{d\chi^2(\vec{a})}{d\vec{a}}\approx\left.\frac{d\chi^2(\vec{a})}{d\vec{a}}\right|_{\vec{a}=\vec{a}_0}+\left.\frac{d^2\chi^2(\vec{a})}{d\vec{a}^2}\right|_{\vec{a}=\vec{a}_0}\Delta\vec{a}. \tag{2.4}$$

The expansion point $a_0$ is the vector of initial alignment parameters and $(\vec{a}-\vec{a}_0)$ is abbreviated as $\Delta\vec{a}$. Inserting the definition of $\chi^2$ (2.1) yields:

$$\begin{aligned}\frac{d\chi^2(\vec{a})}{d\vec{a}}&=\vec{0}=\sum_{tracks}\frac{d}{d\vec{a}}\left.\left(\vec{r_i}^{\,T}(\vec{a})V_i^{-1}\vec{r}_i(\vec{a})\right)\right|_{\vec{a}=\vec{a}_0}+\sum_{tracks}\frac{d^2}{d\vec{a}^2}\left.\left(\vec{r_i}^{\,T}(\vec{a})V_i^{-1}\vec{r}_i(\vec{a})\right)\right|_{\vec{a}=\vec{a}_0}\Delta\vec{a}=\\&=\ldots=\sum_{tracks}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)2V_i^{-1}\vec{r}_i(\vec{a}_0)+\left(\sum_{tracks}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)2V_i^{-1}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)^T\right)\Delta\vec{a}\end{aligned} \tag{2.5}$$

Here, $\sum_{tracks}$ denotes the sum over all tracks $\sum_{i\,\in\,tracks}^{m}$, and $\frac{d}{d\vec{a}_0}$ is the derivative evaluated at $a_0$: $\left.\frac{d}{d\vec{a}}\right|_{\vec{a}=\vec{a}_0}$. Since a linear approximation is done, all terms of the form $\frac{d^2\vec{r}_i(\vec{a})}{d\vec{a}_0^2}$ were neglected during the derivation of (2.5).

The derivative of the vector of residuals with respect to the vector of alignment parameters is defined as:

$$\frac{d\vec{r}}{d\vec{a}} \equiv \begin{pmatrix} \frac{dr_1}{da_1} & \frac{dr_2}{da_1} & \cdots \\ \frac{dr_1}{da_2} & \frac{dr_2}{da_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}. \tag{2.6}$$

The formal solution to 2.5 is:

$$\Delta\vec{a} = -\left(\sum_{tracks}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot V_i^{-1}\cdot\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)^T\right)^{-1}\cdot\left(\sum_{tracks}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot V_i^{-1}\cdot\vec{r}_i(\vec{a}_0)\right). \tag{2.7}$$

Thus, the vector of all alignment parameters $\Delta\vec{a}$ can be calculated using the residuals $\vec{r}_i(\vec{a}_0)$ on all detector surfaces, their derivatives $\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}$ with respect to all alignment parameters and the covariance matrices of the measurements $V_i^{-1}$.

There are several approaches to calculate the solution:

One approach, called the *global* $\chi^2$*-approach* [34], tries to directly invert the large matrix appearing as the first factor in (2.7). This can become quite complicated, as the whole silicon part of the inner detector has 1744 Pixel modules plus 4088 SCT modules, giving a total number of degrees of freedom of $(1744 + 4088) \times 6 = 34992$. A single matrix of dimension $34992 \times 34992$ in double precision needs 4.5 Gigabytes in memory. Additionally, the inversion of this matrix is numerically challenging. How to do this is being investigated with special methods on a dedicated computer cluster.

The *local* $\chi^2$*-approach* tries to circumvent these problems by making some assumptions about the matrix elements. The big matrices are broken up into $6 \times 6$ block diagonal matrices, by [33]:

- assuming that the correlation between different modules is small, i. e. no contribution from the derivative of r with respect to track parameters:

  $$\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_k} = \frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{a}_k} + \underbrace{\frac{d\vec{\pi}_i}{d\vec{a}_k}\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{\pi}_i}}_{\approx 0} \tag{2.8}$$

  Here k is an index which counts the different modules.

  To have this, the track parameters must be unbiased, i. e. the hit which is treated at the moment has to be removed from the track fit to make the fit independent of it.

- ignoring the dependence of the residual on the alignment parameters of other modules, i. e.

  $$\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_l} \overset{!}{=} 0 \quad \forall \quad k \neq l. \tag{2.9}$$

  This can be justified if the error of the common track, which is the source of module-to-module correlations, is small compared to the error of a measurement on the module. Later on, we will test this assumption by determining if

  $$\left(\frac{\sigma_{Track}}{\sigma_{Hit}}\right)^2 < 1. \tag{2.10}$$

- assuming that the covariance matrix is diagonal, i. e. the measurements on different detector surfaces are uncorrelated. This is in general not true because of multiple coulomb scattering (MCS). To suppress these contributions, we seek to use only high-momentum tracks which have negligible MCS contributions. For example, a particle of 10 GeV/c momentum would have a RMS of the deflection angle due to MCS of 0.054 mrad after passing through 250 $\mu$m of silicon. For module layers which are 4 cm away from each other, this would mean an additional position error of 2.2 $\mu$m on the next layer, which is well below the position resolution of the silicon detectors. Thus, 10 GeV/c is reasonable as the minimum momentum when selecting appropriate tracks.

Making these assumptions, formula 2.5 reduces to a 6 × 6 block-diagonal form and can be solved independently for each module:

$$\Delta\vec{a}_k = -\left(\sum_{tracks}\frac{1}{\sigma_{i\,k}^2}\left(\frac{\partial r_{i\,k}(\vec{a}_k)}{\partial\vec{a}_{k\,0}}\right)\left(\frac{\partial r_{i\,k}(\vec{a}_k)}{\partial\vec{a}_{k\,0}}\right)^T\right)^{-1}\cdot\left(\sum_{tracks}\frac{1}{\sigma_{i\,k}^2}\left(\frac{\partial r_{i\,k}(\vec{a}_k)}{\partial\vec{a}_{k\,0}}\right)r_{i\,k}(\vec{a}_{k\,0})\right). \tag{2.11}$$

The covariance matrix for the vector of alignment parameters is then given by [35]:

$$\left[\sigma_a^2\right] = 2\left(\frac{\partial^2\chi^2}{\partial a_0^2}\right)^{-1} \approx \left(\sum_{tracks}\frac{1}{\sigma_i^2}\left(\frac{\partial r_i(\vec{a})}{d\vec{a}_0}\right)\left(\frac{\partial r_i(\vec{a})}{d\vec{a}_0}\right)^T\right)^{-1}, \tag{2.12}$$

and thus the error of the alignment parameters is:

$$\sigma_{a\,j} = \sqrt{\sigma_{a\,jj}^2}. \tag{2.13}$$

## 2.3 Iterations

It is clear that the solution to equation (2.11) will not initially be the best possible solution because of the neglecting of correlations. Therefore, the algorithm needs iterations to converge to a stable solution.

The first iteration leads to an alignment solution which already gives better track fits because of the improved positions of the modules. After redoing the track fitting with the better detector geometry, the alignment procedure is repeated to give a further improved alignment solution than in the first iteration. This procedure is performed again until a final convergence criterium is met.

Thus, by iterating over a track sample several times and repeating track reconstruction, the neglected correlations between different modules are taken into account implicitly.

Since the alignment algorithm just minimizes residuals, its solution is not necessarily unique: If the equation system (2.7) is degenerate, the algorithm will converge to a solution, which might not be the perfect alignment solution. Therefore another crucial point of the local approach is the appropriate selection of track samples and additional constraints – e. g. vertex or momentum constraints – to remove the multiple solutions to the alignment problem and make the solution unique.

# Chapter 3

# Prototype simulation with ROOT

## 3.1 ROOT software

ROOT is a $C^{++}$-based program package for data analysis in science, especially high energy physics. It was and still is developed at CERN, but may be used freely by the entire scientific community. Its capabilities range from histogramming to mathematical analysis, including powerful statistics and fit functionality, to geometry simulation and visualization. The full documentation and code can be found at the ROOT webpage at CERN [36, 37].

In this diploma thesis, the ROOT package was used to write a small simulation program to investigate the functioning and features of the local alignment approach for the Pixel detector.

## 3.2 Geometry and tracking

With ROOT it is possible to simulate a simple geometry independent of the ATLAS detector software framework and to study the performance of our approach in a clean and controllable environment. The geometry was reduced to one cuboid Pixel module having the dimensions of a real Pixel module, which are $16.4\,mm \times 60.8\,mm \times 0.3\,mm$. Then straight line tracks were created, originating from a common vertex and equally distributed across a certain solid angle, i.e. equally distributed in $\phi$ and $\cos\theta$ (c.f. fig. 3.1).

The coordinate system was chosen to be compliant with the ATLAS coordinate conventions, i. e. we use the global and local coordinate frames defined in section 1.6.

To mimic the setup of an ATLAS Pixel module, the positions were chosen as follows: All tracks originate from the global origin and the Pixel module was placed at the global coordinates (17.27 mm, 0, 47.45 mm). This position corresponds to a barrel module of the innermost Pixel layer sitting at $\eta = 0$.

Then tracks were propagated through the geometry by ROOT's tracking capabilities. The intersection point of the track with the surface of the module was taken to calculate the pixel which was hit by the track. Thus, neither an error from a track fit nor the effect of pixel clusters were simulated.

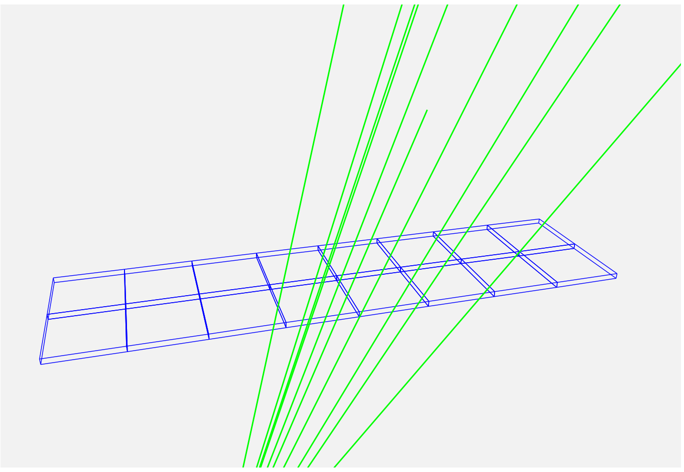

Figure 3.1: *Drawing of the simulated Pixel module. The division into 16 parts illustrates the sections read out by individual FE chips. Some straight line tracks originating from a common vertex cross the module.*

## 3.3 Implementation of the algorithm

From chapter 2 it can be seen that the simulation must calculate the following quantities: residuals, residual errors and residual derivatives with respect to the alignment parameters.

### 3.3.1 Choice of residuals

**In-plane residuals**

The in-plane residuals are defined as the distance between the track and its associated hits within the detector plane. Because a Pixel module performs two independent measurements of two coordinates, a single hit results in two residuals in the local x- and y-directions. Within the simulation, residuals are calculated by the signed distances of the pixel-center in x, y and the impact point of the track:

$$\begin{pmatrix} r_x \\ r_y \end{pmatrix}_{in-plane} = \begin{pmatrix} x_{impact\ point} - x_{pixel\ center} \\ y_{impact\ point} - y_{pixel\ center} \end{pmatrix} \tag{3.1}$$

The distributions of in-plane residuals are uniform with a total width equal to the corresponding pixel x- and y-size (not shown).

**DOCA-residuals**

The Distance Of Closest Approach-residuals (DOCA) are defined as the shortest distance in space between the straight line track and a straight line corresponding to one readout-coordinate of the detector (see [33]). Because a Pixel module has no readout-strips, a two-dimensional grid of readout-strips crossing each other in every pixelcenter is artificially formed, as shown in fig. 3.2. Thus the same formalism which is already in place for ATLAS SCT-modules can be used while retaining the two-dimensional readout for pixels.

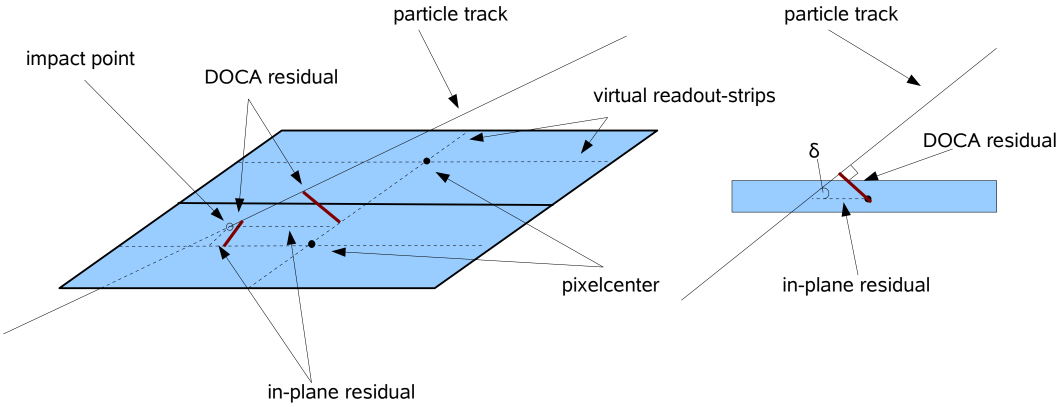

Figure 3.2: *The definition of distance of closest approach residuals for Pixel modules. The dashed lines symbolize the classical in-plane residuals, the red lines denote the DOCA residual. On the right hand side, the y-residuals are shown in a y-z-projection.*

The DOCA-residual is calculated as follows: from the two straight-line equations describing the track

$$\vec{x} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \lambda \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{3.2}$$

and one readout coordinate of a pixel

$$\vec{x}' = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} + \kappa \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \tag{3.3}$$

the signed shortest distance can be computed via determinants [38]:

$$r_{DOCA} = \frac{\begin{vmatrix} a_1 - c_1 & a_2 - c_2 & a_3 - c_3 \\ b_1 & b_2 & b_3 \\ d_1 & d_2 & d_3 \end{vmatrix}}{\sqrt{\begin{vmatrix} b_1 & b_2 \\ d_1 & d_2 \end{vmatrix}^2 + \begin{vmatrix} b_2 & b_3 \\ d_2 & d_3 \end{vmatrix}^2 + \begin{vmatrix} b_3 & b_1 \\ d_3 & d_1 \end{vmatrix}^2}} \tag{3.4}$$

If calculated in the local coordinate frame of the module, the two readout-strips receive direction vectors of $\vec{d} = (0, 1, 0)^T$ for x-strips (where the strip is parallel to the y-axis and the measurement is sensitive to the x-coordinate) and $\vec{d} = (1, 0, 0)^T$ for y-strips (vice-versa). In both cases, the starting point $\vec{c}$ will be the pixel center.

DOCA residuals have a uniform distribution with smeared edges because tracks crossing non-perpendicularly have a smaller DOCA residual compared to in-plane residual, as shown in fig. 3.3.

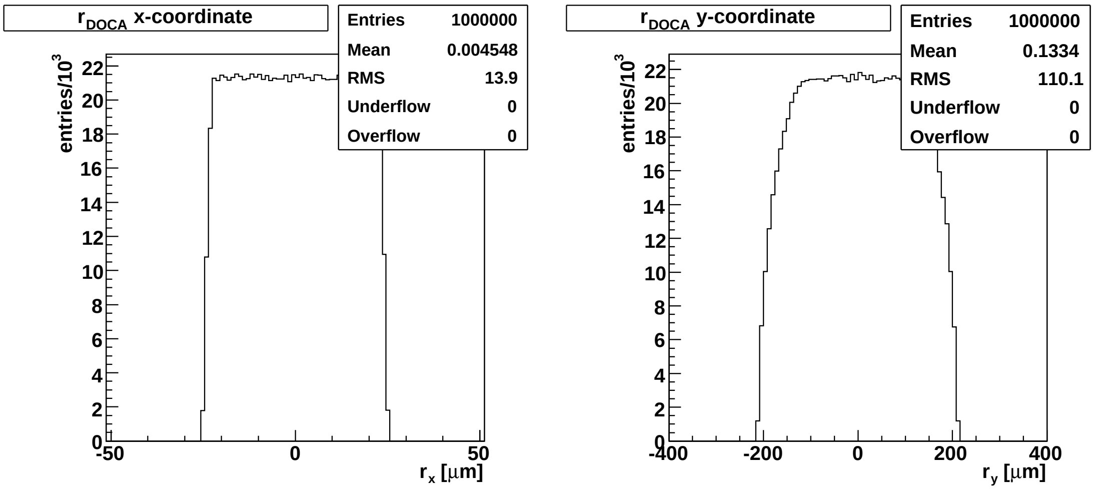

Figure 3.3: *Distributions of the DOCA residuals*

### "Single" residuals

If one takes the distance to the center of the pixel as the (single) quantity to minimise, the two residual definitions take the form:

$$r_{in-plane} = \sqrt{r_x^2 + r_y^2} \tag{3.5}$$

$$r_{DOCA} = \frac{\left\| \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \times \begin{pmatrix} r_x \\ r_y \\ 0 \end{pmatrix} \right\|}{\left\| \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right\|} \tag{3.6}$$

It was found that these residual definitions yield poor results compared to the traditional x- and y-residuals. This is because the two initially independent measurements are combined with one common accuracy. It is found that sensitivity is transferred from the sensitive x-coordinate to the less sensitive y-coordinate. This leads to the somewhat counterintuitive result that the x-coordinate is less sensitive than the y-coordinate.

This can be understood by looking at the error of the in-plane residuals:

$$\sigma_r = \sqrt{\left(\frac{\partial r}{\partial x}\sigma_x\right)^2 + \left(\frac{\partial r}{\partial y}\sigma_y\right)^2} = \sqrt{\left(\frac{r_x}{r}\sigma_x\right)^2 + \left(\frac{r_y}{r}\sigma_y\right)^2} \tag{3.7}$$

The errors for the x- and y-coordinate are only used in combination. Since large y-residuals are much more probable due to the large aspect ratio of the rectangular pixel, the y-coordinate dominates the error most of the time. Another result of this resolution loss are the peculiar derivatives of this single residual: the derivatives with respect to x are often close to zero, a clear sign that the x-coordinate is not sensitive to alignment corrections (see section 3.3.4).

Due to this shortcomings, “single” residuals are not discussed any further in this thesis.

### 3.3.2 Gaussian versus non-gaussian input

Due to track fitting uncertainties and clustering effects, which are missing in the ROOT simulation, the distributions of the residuals in the Athena-implementation of the alignment approach will not be uniform but instead will be roughly gaussian. The algorithm performs equally well under these conditions. This is due to the central limit theorem [38]: The sum of many random variables of the same distribution will have a gaussian distribution, regardless of the shape of the random variable’s distribution. Therefore the input distribution of the residuals becomes irrelevant since $\chi^2$ is computed as their sum. Of course the error estimate must be correct for both types of input. The validation of our simulation with gaussian-shaped input residuals is shown in [33].

### 3.3.3 Residual errors

The errors of the two in-plane residuals were taken to be the RMS of a uniform distribution, which is

$$\sigma = \frac{strip\ pitch}{\sqrt{12}}. \tag{3.8}$$

For the two errors of the DOCA-residuals, error propagation yields:

$$r_{DOCA} = r_{in-plane} \cdot sin\delta \tag{3.9}$$

$$\sigma\left(r_{DOCA}\right) = \sqrt{\left(\frac{\partial r_{DOCA}}{\partial x}\right)\sigma_x^2 + \left(\frac{\partial r_{DOCA}}{\partial y}\right)\sigma_y^2 + \left(\frac{\partial r_{DOCA}}{\partial \delta}\right)\sigma_\delta^2}, \tag{3.10}$$

where $\delta$ is the angle between the detector strip and the track projection on a plane perpendicular to the detector strip, as depicted in fig. 3.2.

The first term of (3.10) is zero for y-residuals, the second for x-residuals. The third term was taken to be zero, because in this simulation with perfect tracks, the error of $\delta$ is zero. Later on, the third term will denote the track fitting error.

It was verified that both the usage of partial derivatives and the direct calculation of $\sin\delta$ give the same results for the DOCA error.

### 3.3.4 Calculation of derivatives

The calculation of derivatives with respect to the six alignment parameters was performed numerically using the two-sided difference quotient:

$$\frac{\partial r}{\partial a_i} = \frac{r(a_i + h_i) - r(a_i - h_i)}{2h_i} \tag{3.11}$$

In the simulation, the module was displaced by a small distance $\pm h_i$ and the two residuals were calculated for each coordinate and each of the six alignment parameters. Distributions of the residual derivatives can be seen in figs. 3.4 and 3.5. Additionally, figs. A.1 and A.2 show these distributions in the case of in-plane residuals.

Many features of these derivative distributions for a barrel module have simple explanations:

- $\frac{\partial r_x}{\partial x}$ would be exactly one for in-plane residuals. Since the tracks cross at an angle, the DOCA residuals change less than in-plane residuals when moving in x. Due to the 20° tilt angle, $\frac{\partial r_x}{\partial x}$ scatters around $\cos -20^\circ = 0.94$ (fig. 3.4a).
- $\frac{\partial r_x}{\partial y}$ must be zero, since x-strips run parallel to the local y-axis (fig. 3.4b).
- $\frac{\partial r_x}{\partial z}$ would be zero for tracks crossing the module at right angles. Since this is not the case, the module acquires some sensitivity in z, i. e. has non-zero derivatives. The incident angle in the x-z-projection scatters around 20° and thus the derivative distribution is symmetric around $\sin -20^\circ = -0.34$ (fig. 3.4c).
- For the angles, large derivatives occur when the lever arm for rotations around the axis is large, i. e. the module has its long coordinate there. Here one has to keep in mind that the rotation is always with respect to the center of the Pixel module. Large derivatives occur e. g. for the long strips parallel to y when rotated around x, i.e. $\frac{\partial r_x}{\partial \alpha}$ (fig. 3.4d).
- The rotations around y, such as $\frac{\partial r_x}{\partial \beta}$, contribute smaller values than rotations around x since the module is not as wide in the x-direction than it is in the y-direction (fig. 3.4e).
- Large derivatives occur also for rotations around z, i.e. $\frac{\partial r_x}{\partial \gamma}$. The crossing angle in the y-z-plane is symmetric around zero, and so is $\frac{\partial r_x}{\partial \gamma}$ (fig. 3.4f).
- $\frac{\partial r_y}{\partial x}$ is zero, similarly to $\frac{\partial r_x}{\partial y}$ (fig. 3.5a).
- $\frac{\partial r_y}{\partial y}$ would be exactly one for in-plane residuals. For DOCA-residuals, $\frac{\partial r_y}{\partial y}$ is determined by the length of the module in y and receives contributions from $\cos 0^\circ = 1$ down to $\cos 59.0^\circ = 0.52$ (fig. 3.5b).
- $\frac{\partial r_y}{\partial z}$ would be zero for tracks crossing the module at right angles. Again, the symmetric distribution of crossing angles in the y-z-plane makes this distribution symmetric around zero, with a maximum of $\cos 32.6^\circ = 0.84$ (fig. 3.5c).
- $\frac{\partial r_y}{\partial \alpha}$ is large, since strips in x-direction on the outer sides of the module contribute large values when rotated around x (fig. 3.5d).
- $\frac{\partial r_y}{\partial \gamma}$ is not symmetric – in contrast to $\frac{\partial r_x}{\partial \gamma}$ – due to the tilt angle in the x-z-plane (fig. 3.5f).

As a consequence one can say that the observed distributions lie well within the expectations. Later on, they will be compared with the results obtained from the Athena implementation.

The absolute value of the derivatives is a direct measure of the sensitivity track-based alignment has with respect to this degree of freedom, since it indicates the change of the residual distribution when moving along this degree of freedom. It can be seen from the derivative distributions that x-residuals constrain the x-coordinate and the rotations fairly well. The largest derivatives and therefore the highest sensitivity are with respect to $\gamma$, since a rotation of the module within its plane always changes the residual distribution considerably. The z-coordinate acquires some sensitivity due to the tilt angle of the module, which leads to a change of x-residuals when moving along the local z-direction.

For residuals in the y-direction, the converse is true: These residuals are sensitive to shifts in y, but not in x. Sensitivity in z comes into play via the varying incident angle over the whole length of the module. The rotations are also well constrained, due to the large lever arms when rotating around the module center.

The advantage of Pixel modules from the point of view of the alignment is that the use of both residuals constrains both x and y well, with x being more sensitive than y due to the smaller error of the x-residual. From the simulation it is concluded that the combination of x- and y-residuals and the special barrel geometry chosen (tilt angle in x and central position in y) should constrain all six degrees of freedom well.

### 3.3.5 $\chi^2$ minimization

The calculated residuals, residual errors and derivatives were fed into equation 2.11 and the alignment parameters and their errors were calculated. Figure 3.6 shows the alignment results using 1000 hits on the module and repeating the whole procedure 1000 times. In Appendix A, the corresponding plot is shown using in-plane residuals (fig. A.3)

The pull distributions, i. e. the difference between the true and calculated values normalized by the error

$$pull_{a_i} = \frac{a_{i,true} - a_{i,calculated}}{\sigma_{a_i}} \tag{3.12}$$

are also shown (figs. 3.7, A.4). Pull distributions allow one to check whether the calculated values are correct (pulls should be centered around zero) and whether the error estimates are consistent (pulls should have a width of 1).

## 3.4 Performance of the algorithm and results

The algorithm was tested as shown above with no misalignment (nominal alignment).

In addition, runs with a misaligned module were performed. For this, alignment parameters and the corresponding pulls can be seen in figs. 3.8 and 3.9. For comparison, results with in-plane residuals are shown in appendix A (figs. A.5 and A.6). In both cases, the module was shifted by $\Delta x = 12\,\mu m, \Delta y = 25\,\mu m, \Delta z = -30\,\mu m$ and rotated around $\gamma$ by -2 mrad, then around $\beta$ by 5 mrad and finally around $\alpha$ by 5 mrad.

As one can see, in both the nominal and the misaligned case, all six degrees of freedom are recovered within small limits. The mean and widths of the alignment parameter distributions are summed up in table 3.1.

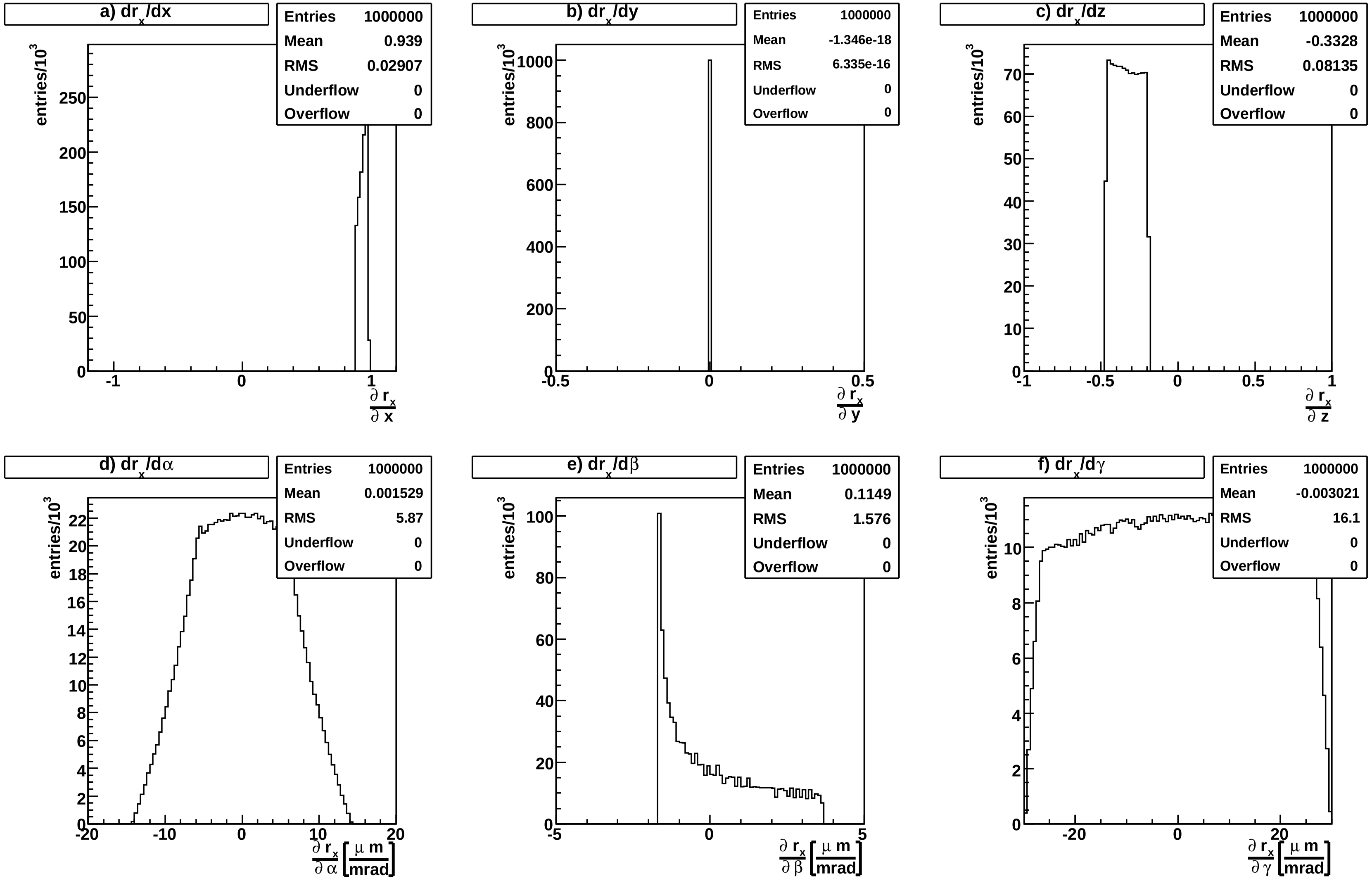

Figure 3.4: *Distribution of the derivatives of DOCA x-residuals with respect to the six alignment parameters*

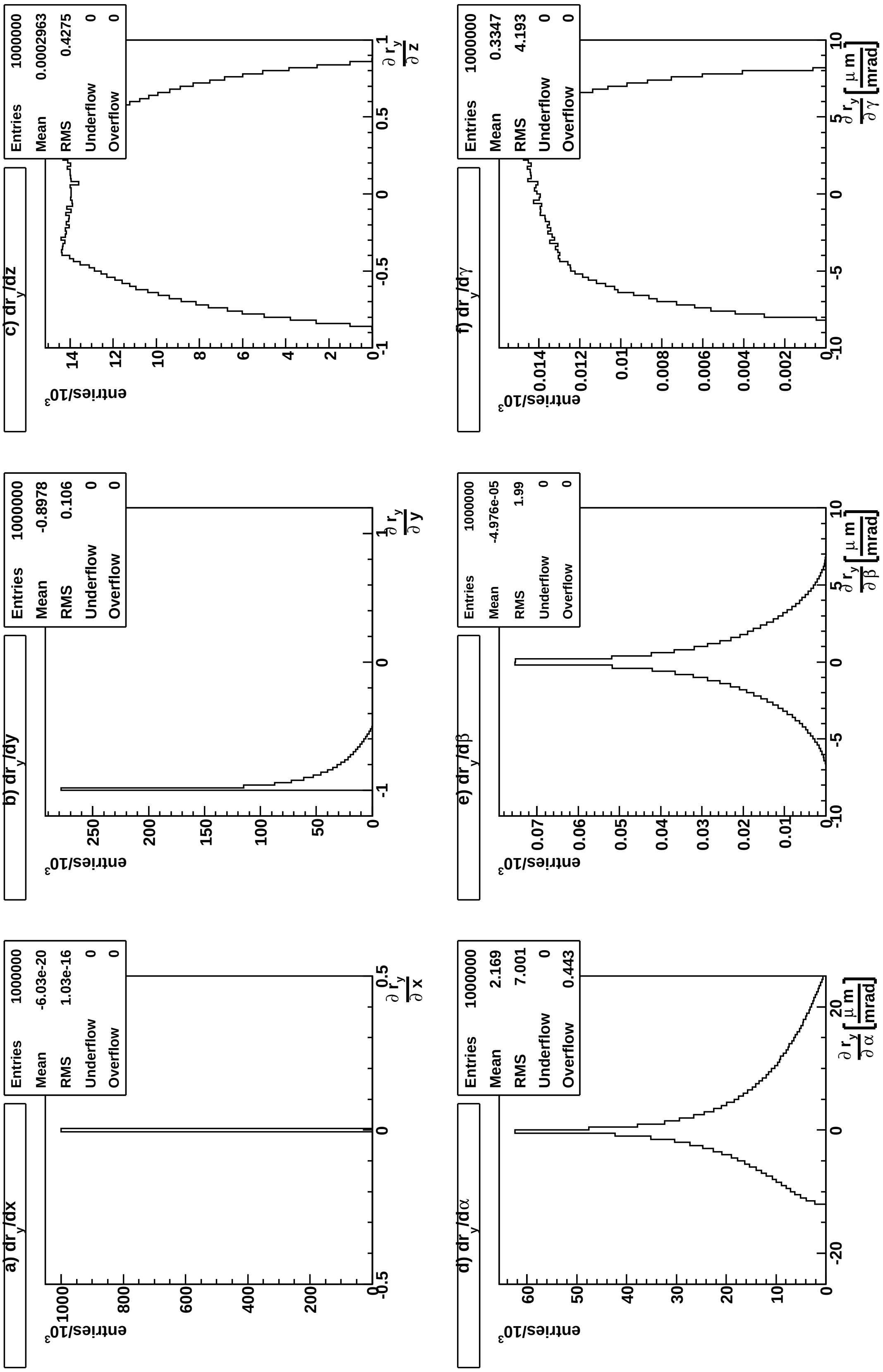

Figure 3.5: *Distribution of the derivatives of DOCA y-residuals with respect to the six alignment parameters*

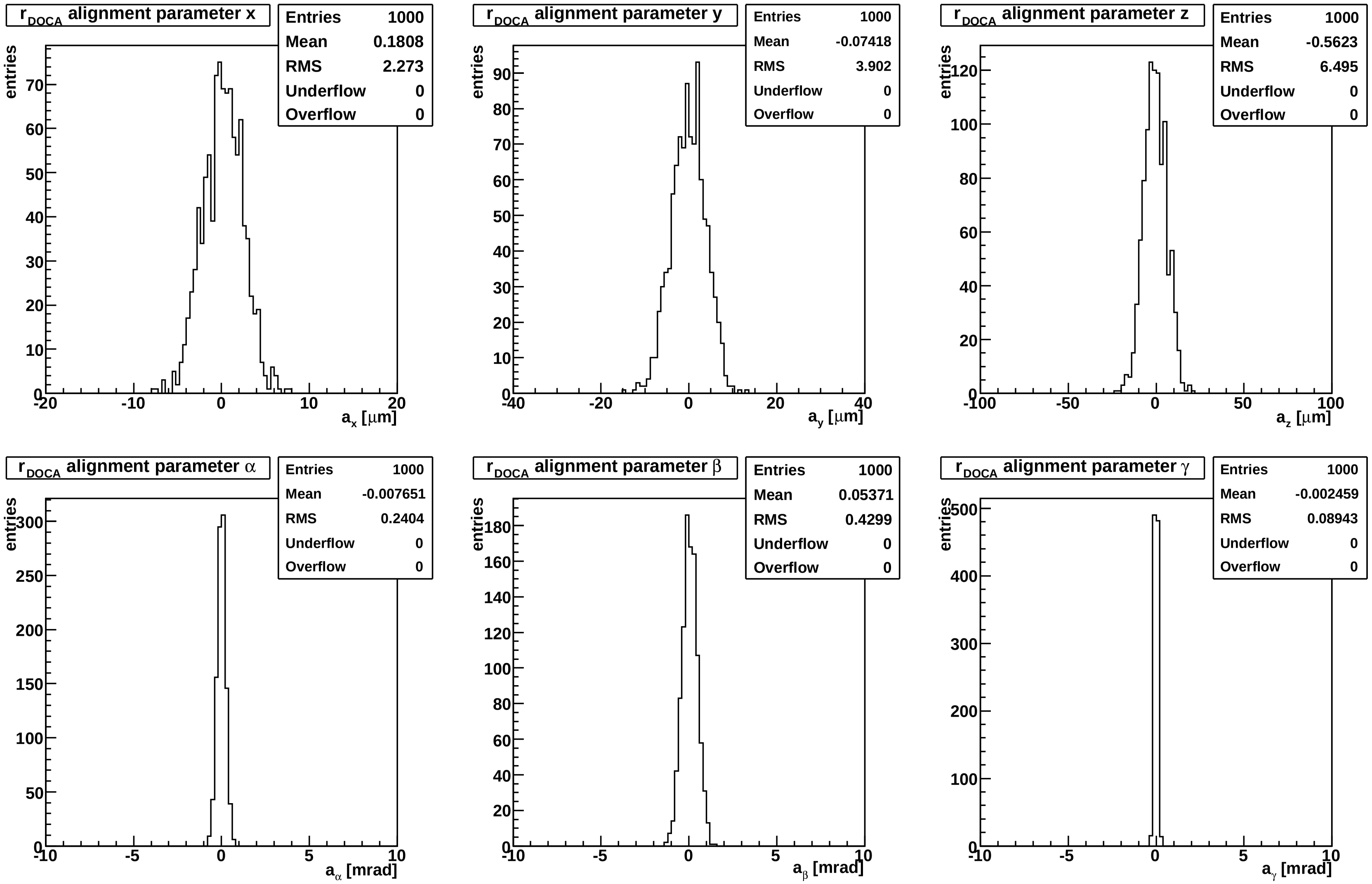

Figure 3.6: *Distribution of the final alignment parameters calculated using DOCA residuals. 1000 runs were produced using nominal alignment and 1000 tracks each.*

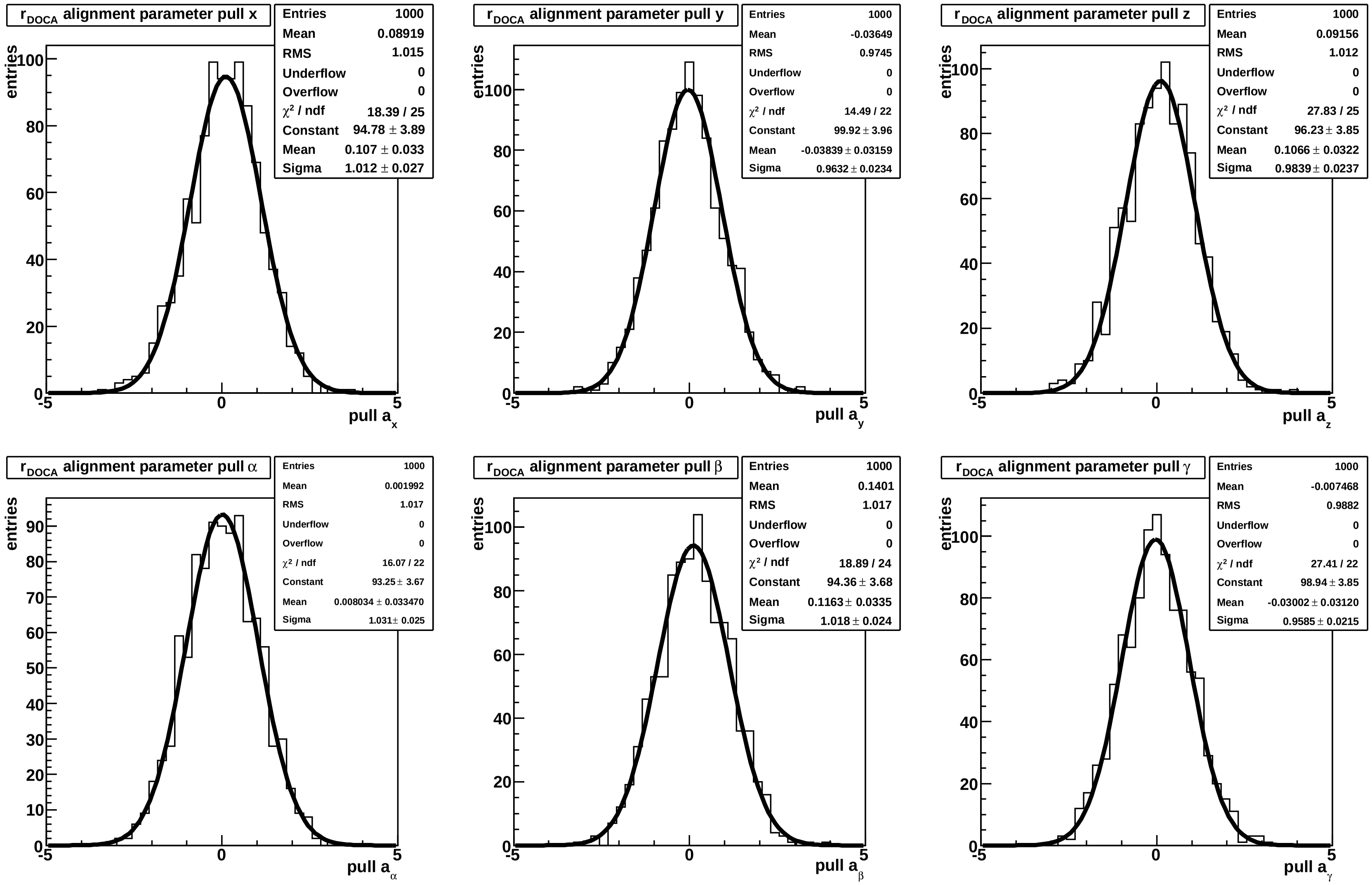

Figure 3.7: *Pull distribution of the alignment parameters calculated using DOCA residuals. 1000 runs were produced using nominal alignment and 1000 tracks each.*

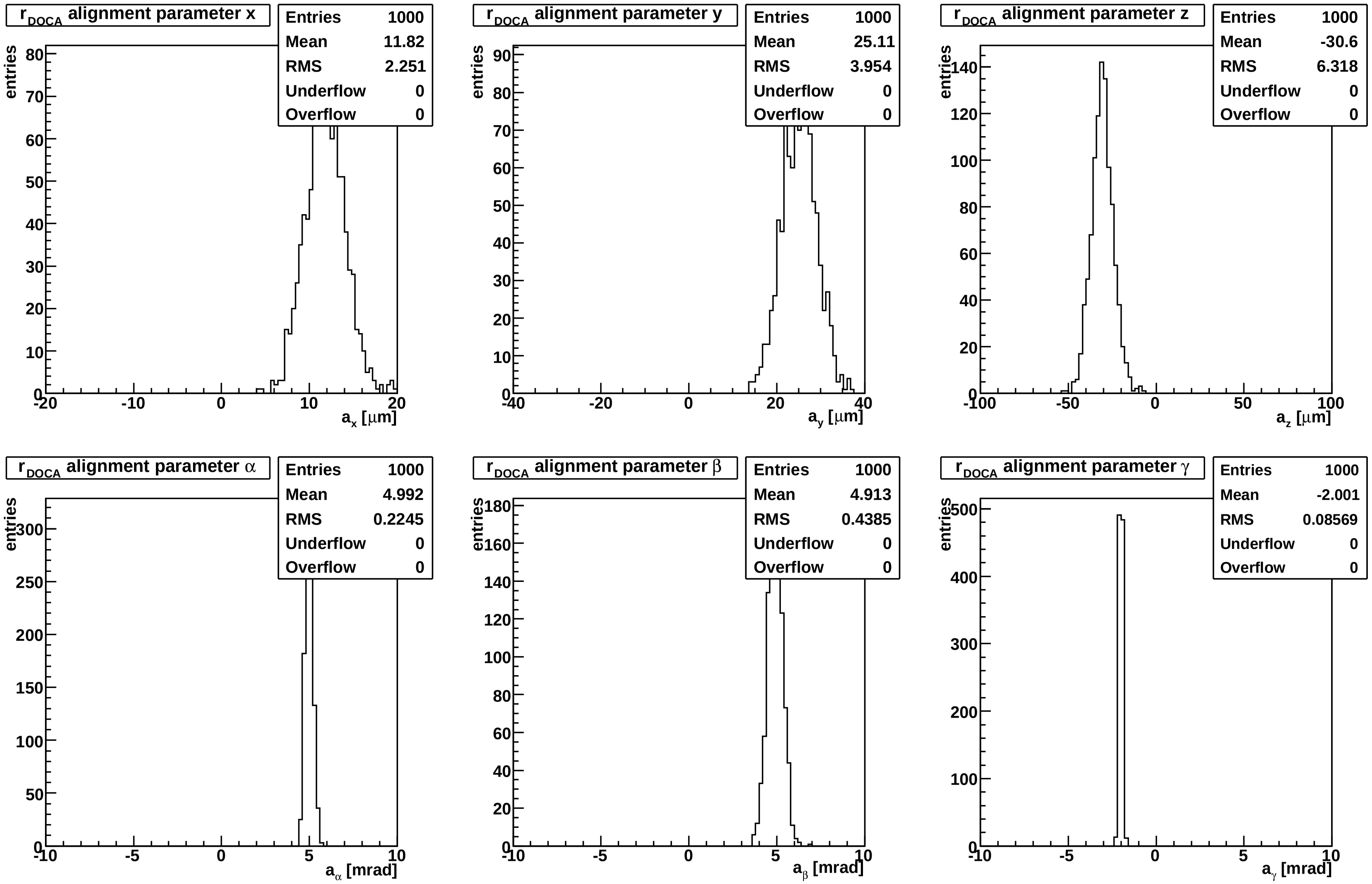

Figure 3.8: *Distribution of the final alignment parameters calculated using DOCA residuals. 1000 runs were produced with 1000 tracks each. The applied misalignment was* $\Delta x$*=12* $\mu m$*,* $\Delta y$*=25* $\mu m$*,* $\Delta z$*= -30* $\mu m$*,* $\Delta\alpha$*=5 mrad,* $\Delta\beta$*=5 mrad,* $\Delta\gamma$*=-2 mrad.*

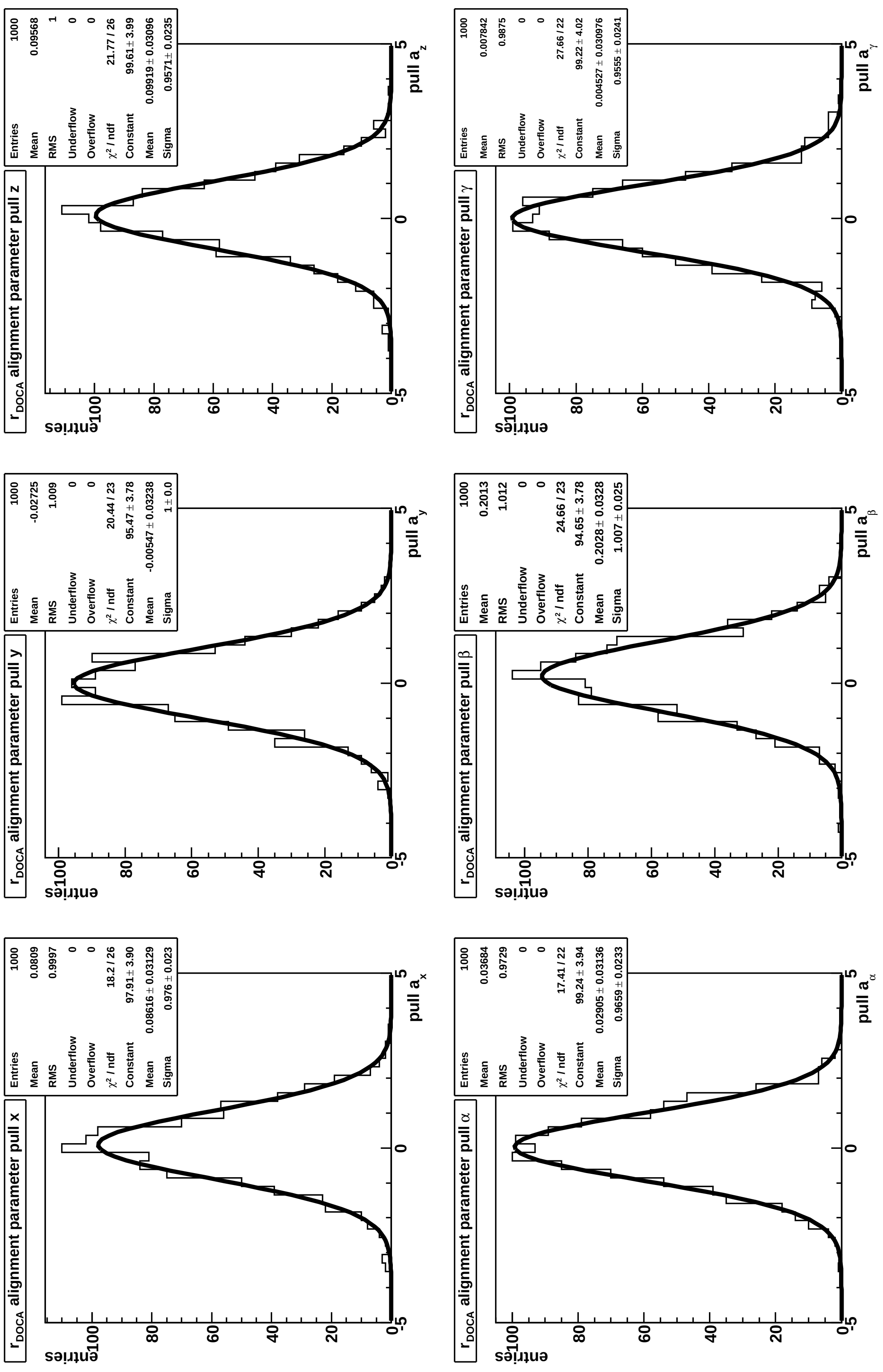

Figure 3.9: *Pull distribution of the alignment parameters calculated using DOCA residuals. 1000 runs were produced with 1000 tracks each. The applied misalignment was the same as in fig. 3.8.*

| Alignment accuracy with the ROOT simulation | | | | |
|---|---|---|---|---|
| | $\|a_{i,true} - a_{i,calculated}\|$ | | $\sigma_{a_i}$ | |
| coordinate | nominal | misaligned | nominal | misaligned |
| x | $0.18\,\mu$m | $0.18\,\mu$m | $2.3\,\mu$m | $2.3\,\mu$m |
| y | $0.07\,\mu$m | $0.11\,\mu$m | $3.9\,\mu$m | $4.0\,\mu$m |
| z | $0.56\,\mu$m | $0.6\,\mu$m | $6.5\,\mu$m | $6.3\,\mu$m |
| $\alpha$ | $7.7 \cdot 10^{-3}$ mrad | $8 \cdot 10^{-3}$ mrad | 0.24 mrad | 0.22 mrad |
| $\beta$ | 0.054 mrad | 0.087 mrad | 0.43 mrad | 0.44 mrad |
| $\gamma$ | $2.5 \cdot 10^{-3}$ mrad | $1 \cdot 10^{-3}$ mrad | 0.09 mrad | 0.09 mrad |

Table 3.1: *Alignment accuracies determined with the ROOT simulation for nominal and misaligned geometry, using 1000 runs with 1000 tracks each.*

However, the corresponding error of the alignment parameters, given by the width of the distributions, suggests that the alignment parameters should be even closer to the expected values. This is also reflected by the pull distributions, which have a mean value not compatible with zero for some degrees of freedom. All pull distributions are gaussian, but for three of them, the mean is about three times its error away from zero. This is certainly unlikely to be compatible with zero. The error calculations have obviously been performed correctly, since the pull distributions have a width compatible with one. It was also observed that tiny changes of the track sample, e. g. adding or removing single tracks or changing the seed of the random number generator result in relatively large shifts of the pull distribution.

If one places the Pixel module such that the tilt angle is removed, the calculated alignment parameters are almost completely compatible with zero within their error range. The corresponding plots are shown in figs. A.7 and A.8 for comparison. This behavior is still under investigation. The worse performance under illumination at a tilt angle is regarded as coming from the implementation of the simulation and not considered as an intrinsic flaw of the algorithm.

Table 3.2 shows the correlation terms resulting from equation (2.12), weighted by the errors of the alignment parameters. The quantities given in the table are calculated as follows:

$$c_{ij} = \frac{\sigma^2_{aij}}{\sqrt{\sigma^2_{aii} \cdot \sigma^2_{ajj}}}. \tag{3.13}$$

Thus, the $c_{ij}$ range from -1 to 1 and give the strength of the correlation between two coordinates. As one can see, x and z are strongly correlated due to the tilt angle. Naturally, the translations in the plane are correlated with the corresponding rotations, that is x with $\beta$ and y with $\alpha$. Due to the tilt angle, also z gets correlated with $\beta$. The translations x and y are essentially uncorrelated. The two rotations $\alpha$ and $\gamma$ are strongly anti-correlated. The correlations are within the expectations already acquired during previous sections. Later on, they can be compared to the values obtained from Athena.

| Correlations $c_{ij}$ in the ROOT simulation | | | | | | |
|---|---|---|---|---|---|---|
| coordinates | x | y | z | $\alpha$ | $\beta$ | $\gamma$ |
| x | 1 | $3.72 \cdot 10^{-4}$ | 0.978 | $-2.35 \cdot 10^{-3}$ | 0.750 | $2.36 \cdot 10^{-3}$ |
| y | – | 1 | $4.29 \cdot 10^{-4}$ | 0.176 | $3.39 \cdot 10^{-4}$ | $-0.166$ |
| z | – | – | 1 | $-2.08 \cdot 10^{-3}$ | 0.781 | $2.10 \cdot 10^{-3}$ |
| $\alpha$ | – | – | – | 1 | $-1.61 \cdot 10^{-3}$ | $-0.9488$ |
| $\beta$ | – | – | – | – | 1 | $1.33 \cdot 10^{-3}$ |
| $\gamma$ | – | – | – | – | – | 1 |

Table 3.2: *Correlations of the local coordinates from the ROOT simulation.*

## 3.5 Discussion

An implementation of our alignment approach has been programmed in the ROOT framework. The study of the alignment of one Pixel module showed the feasibility of the approach.

The choice of residuals was made based on the results of the approach. The use of “single” residuals was discarded as it became clear that this results in a loss of performance. It was decided to use DOCA residuals, since they ease the implementation and portability for the existing Athena-based alignment considerably compared to in-plane residuals.

The error calculations were verified with the simulation and found to be correct given the assumption of no track error and the missing clustering.

The derivatives of residuals were studied and their distributions could be explained based on the features of the simulation and the geometry of the simulated setup.

With 1000 simulated ideal tracks, the simulation is able to constrain the alignment parameters within the values given by table 3.1. These numbers are well within the tolerances required by the TDR (c. f. table 1.1). However, the errors of the alignment parameter show that the parameters should even be better for this high amount of statistics.

Correlation terms have been obtained and show that the expectations of the correlation between the different degrees of freedom are essentially correct.

# Chapter 4

# Implementation of the algorithm in Athena

The main work of this thesis was to implement the alignment of Pixel modules into the existing Athena algorithm, which had previously handled only the alignment of SCT modules.

## 4.1 Athena

The common software framework for both online triggering and offline reconstruction is called Athena [39]. It is written in $C^{++}$ and is based on the Gaudi architecture developed for the LHCb experiment [40]. Athena consists of the Gaudi core and concepts, together with ATLAS-specific enhancements.

Athena is event-driven, meaning that the unit of data which is processed by all components is one event. An event represents the detector response to one triggered bunch-crossing and contains raw data objects such as hits and clusters from the tracking detectors and the calorimeter system, collections of reconstructed tracks, etc. This data is processed event by event by the different algorithms, e. g. calibration algorithms, cluster formation algorithms or track fitting algorithms. The communication and distribution of data between different algorithms is done via a transient data store called StoreGate. The transient store supplies the algorithms with data from memory and receives their results. This data is read into the transient store from data files at the beginning of an Athena run and can be stored in files at the end of the run. The algorithms also use features provided by the various services of the framework, for example the histogram service or the message service. All Athena-specific objects derive from the three base classes `DataObject`, `Algorithm` and `Converter`. The steering of an individual Athena-job and the passing of parameters to the different algorithms is performed by scripts written in Python [41]. The whole framework is modular and the same code is used both online for the high level trigger and offline for processing of previously stored data [42].

## 4.2 The `Chi2AlignAlg` algorithm

The implementation of the $\chi^2$-alignment-approach can be found in the ATLAS software repository kept in CVS at CERN [43]. The algorithm is called `Chi2AlignAlg` and is found in the package `offline/InnerDetector/InDetAlignAlgs/SiRobustAlignAlgs`. Fig. 4.1 shows a diagram illustrating the components of the package. Like all algorithms, `Chi2AlignAlg` has to implement certain Athena-specific methods, namely `initialize()`, `execute()` and `finalize()`.

At the beginning of the job, Athena calls the `initialize()` method of each algorithm which will run within this job. Basic features of the framework are set up and the algorithms can initialize their properties and sub-algorithms, if needed. At this stage, also a copy of the so-called ConditionsDatabase is generated in memory, which is retrieved either from a database server or loaded from a file. The ConditionsDatabase stores all alignment parameters and calibration constants for the ATLAS detector. The `execute()` methods of the algorithms are then called in turn for each event in the event collection. The algorithms perform their actions, e. g. track finding and fitting or alignment. Finally, `finalize()` is called to end all algorithms properly, write out the results into files and shut down all services of the framework [42].

The algorithm `Chi2AlignAlg` makes use - amongst others - of two additional packages belonging to the local $\chi^2$ approach, which were factored out of recent versions of Athena:

During `initialize()`, `Chi2AlignAlg` creates an instance of the `Chi2AlignModule` class for each silicon module (Pixel and SCT) which it is going to align. This data-object-like class is located in the `InDetAlignEvent/SiRobustAlignEvent` package and stores the relevant alignment quantities of one silicon module - namely a six-dimensional vector and a $6 \times 6$ matrix. These are accumulated during running and are eventually used for the alignment solution. `Chi2AlignAlg` also makes use of several Tools which perform helper functions specific to our alignment approach. These tools reside in the `InDetAlignTools/SiRobustAlignTools` package.

For each event, `Chi2AlignAlg` loops through all tracks in the track collection and refits them using only Pixel and SCT hits by using the `refit()` method of the `RefitSiOnlyTool`. This refitting is necessary to incorporate changes into the tracks which enter by different alignment parameters of the modules during subsequent iterations. Additionally, using only silicon hits removes any dependence on TRT hits and therefore removes possible bias on the silicon alignment.

The algorithm loops over all hits for each of the refitted tracks and calculates the relevant alignment quantities as described below. The corresponding `Chi2AlignModule` instance accumulates these values and the algorithm proceeds with the next hit.

At the end of an alignment run, the alignment correction for each module is calculated using equation (2.11) and is written out to a file. Additionally, an NTuple file is created containing information about all processed hits and tracks to allow for monitoring and creating histograms.

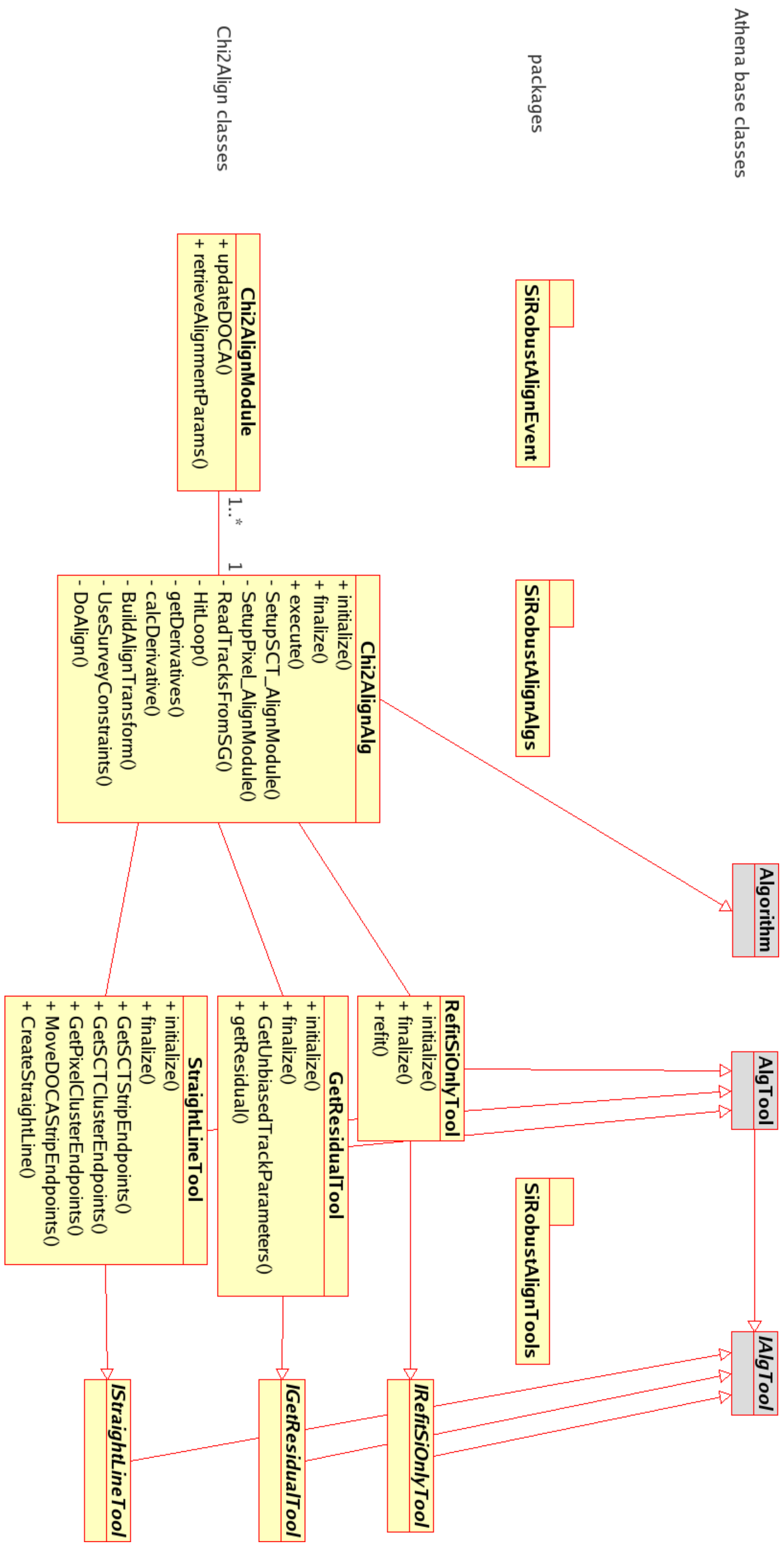

*Figure 4.1: UML diagram of the class structure of* `Chi2AlignAlg`*. The three packages* `SiRobustAlignEvent`*,* `-Algs` *and* `-Tools` *are shown with their classes. The classes inherit from Athena base classes shown in grey. Only the important methods are shown.*

## 4.3 Calculation of residuals

For each pixel hit on the refitted track the algorithm looks up the the stored hit coordinates. The necessary unbiasing of track parameters, as described in section 2.2, is done by using the `removeFromState()` feature of the `KalmanUpdator`. This tool allows for easy recalculation of track parameters after a hit is removed, without requiring that the actual track fit be redone [44, 45].

`Chi2AlignAlg` then uses the `StraightLineTool` to create a pair of strips which are parallel to the x- and y-edges of the module and cross at the hitpoint, as done in the ROOT toy program. These "virtual strips" can have arbitrary crossing points due to the existence of pixel clusters, which occur when multiple adjacent pixels give a signal when crossed by one particle. The `GetResidualTool` is then used to calculate the signed distance (the residual) between these strips and the track. This is done via the `TrkExtrapolator`. This tool can extrapolate track parameters to an arbitrary surface, which is in our case the strips (called a `StraightLineSurface` in Athena). Extrapolation means that this operation yields track parameters at the point of closest approach of the track to the given surface, which are exactly the parameters needed for our DOCA residuals. The quantity `localR` is the signed shortest distance transverse to the `StraightLineSurface` and is therefore appropriate as definition of the residual in `Chi2AlignAlg`.

All distributions shown in the following sections are produced with the input sample described in section 5.1. Figs. 4.2 and 4.3 show the residuals calculated by the algorithm separated by the various barrel layers and endcap discs. Unlike the distributions in the ROOT toy program, these distributions are not uniform, but are instead gaussian shaped. This is due to the track fitting procedure, the non-uniform detection efficiency across the area of a pixel and the possibility of having clustered hits. In this sense, x-residuals denote distances from strips parallel to the long edge of the pixel to the track, since the residuals are perpendicular to the local y-direction. For y-residuals, it is the other way round. The RMS of the barrel x-residuals is about 18 $\mu$m, slightly larger than the expected resolution of a pixel of $\frac{50\,\mu m}{\sqrt{12}} = 14.4\,\mu m$. On the other hand, the y-residuals, having an RMS of only 85 $\mu$m, are considerably narrower than expected from a single pixel ($\frac{400\,\mu m}{\sqrt{12}} = 115.5\,\mu m$). Thus, clustering seems to be especially important along the y-direction in the barrel. In the endcap, clustering obviously plays a less important role, as can be seen from the RMS values, which are considerably closer to the expected values. The pull distributions for the residuals therefore show that the error estimate of endcap residuals is nearly correct, whereas barrel residuals have approximately 25% overestimated errors. The different errors for residuals coming from cluster hits therefore have to be taken into account, which has not yet been done properly.

### 4.3.1 Pixel clustering

Pixel clustering is dealt with at reconstruction level at the stage of calibrating raw data into objects which can be handled by the tracking procedure. This process is termed prepared raw data formation. The pixel clusters are constructed by a `SiClusterizationTool`, which is called by the `InDetPrepRawDataFormation` algorithm. This tool collects adjacent pixels showing a hit into a pixel cluster and calculates the cluster position. Two strategies are possible at present: digital clusterization, where the cluster position is the arithmetic

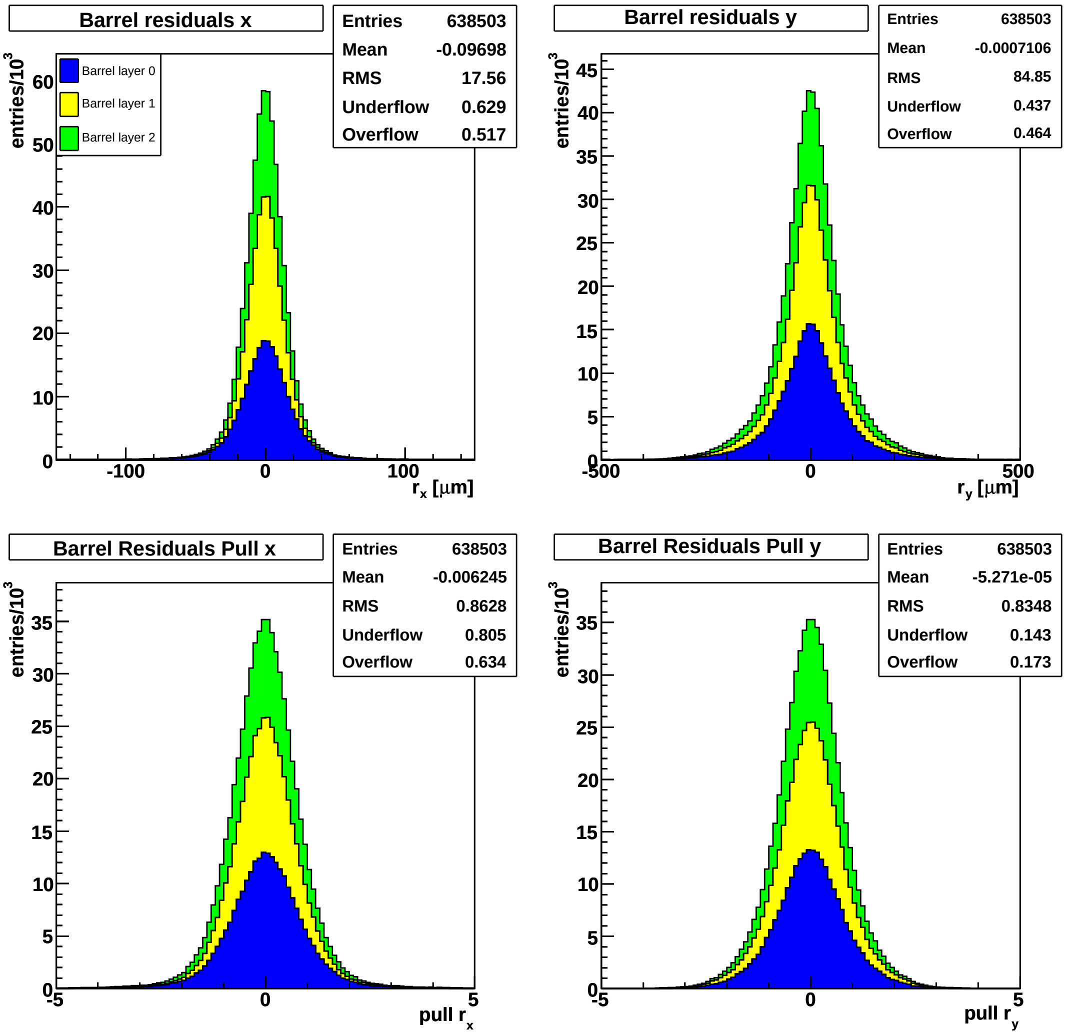

Figure 4.2: *Residual distributions with the corresponding pull distributions for the Pixel barrel modules for both coordinates.*

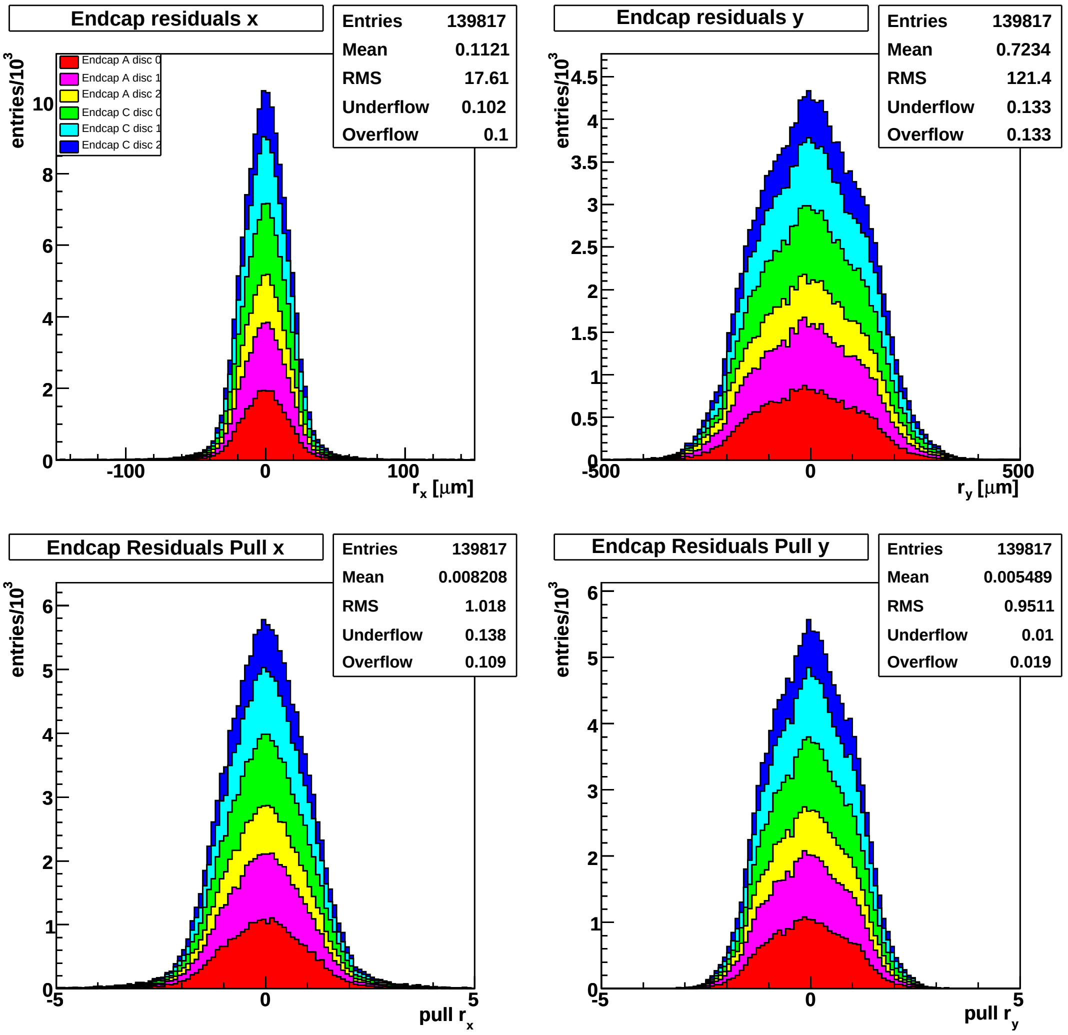

Figure 4.3: *Residual distributions with the corresponding pull distributions for the Pixel endcap modules for both coordinates.*

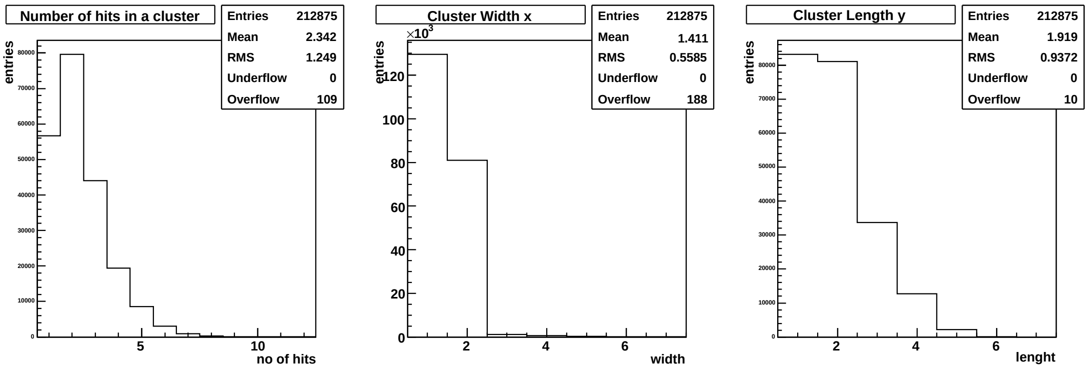

Figure 4.4: *Cluster size distributions in the barrel.*

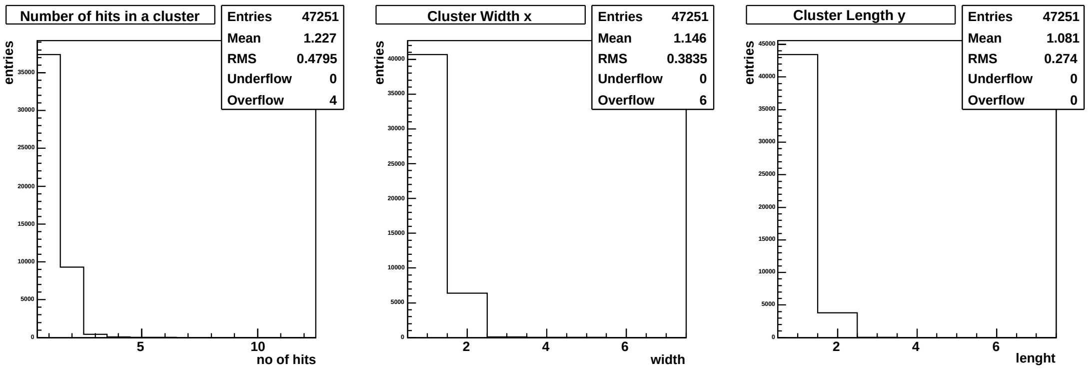

Figure 4.5: *Cluster size distributions in the endcap.*

mean of the pixel positions, and ToT interpolation (analog clustering), where the pixel positions are weighted with the measured time-over-threshold. An absolute charge calibration is not yet in place, but the relative calibration of ToT times allows for a good analog clustering without absolute charge knowledge in the approximation of linear dependence of ToT on collected charge.

For our studies, the ToT-clustering was used. The frequency of various cluster sizes in both pixel coordinates is depicted in figs. 4.4 and 4.5. Overall, cluster hits are quite probable (73% in the barrel, 20% in the endcap). In the barrel, a cluster made of two adjacent pixels has the highest probability. The cluster width in x, i.e. the number of pixel rows contributing to the cluster, is constrained to be one or two rows, due to the fixed tilt angle in this direction. In the y-coordinate, up to 5 pixels can light up with a significant probability due to the small incident angle, which is possible for barrel modules at large $\eta$-values. In the endcap, mostly single pixel hits are observed due to almost perpendicular incident angles.

## 4.4 Residual errors

The errors of the residuals are calculated as

$$\sigma^2_{r\ DOCA} = \sigma^2_{Hit} + \sigma^2_{Track}. \tag{4.1}$$

$\sigma_{Track}$ takes into account the error of the track fit. It is directly obtained from the `TrkExtrapolationTool`. This tool gives the correctly propagated error matrix of the track parameters, of which the error of `localR` is used.

The finite resolution of the detector elements is responsible for $\sigma_{Hit}$. Different pixel cluster sizes have an impact on the hit resolution. At present, three error strategies are implemented in the `InDetPrepRawDataFormation`: The position error of the cluster in one dimension can be computed as $\frac{pixelwidth}{\sqrt{12}}$ (standard method) or as $\frac{clusterwidth}{\sqrt{12}}$. In addition there exists an error table which parameterizes cluster errors depending on size and position in the Pixel detector (new method). All these errors denote the error of the position measurement in the plane of the sensor. For DOCA residuals, which are not in this plane, equation (3.9) is used:

$$\sigma_{Hit,DOCA} = \sigma_{Hit,in-plane} \cdot \sin\delta, \tag{4.2}$$

where $\delta$ is the corresponding angle in a flat projection, as seen in fig. 3.2.

The distribution of the two error contributions $\sigma_{Hit}$ and $\sigma_{Track}$ can be seen in fig. 4.6. The hit errors are computed using the standard error estimate. Additionally, the squared ratio of errors from equation (2.10) is shown, which is a measure of the correlation between different hits on the same track, as seen in section 2.2.

The x-residuals have an almost constant hit error, which is due to an almost constant incident angle of around 90° in the endcaps and around 70° in the barrel. y-residual errors are almost constant in the endcaps; in the barrel they have a large spread due to the varying incident angle.

The track fit errors of endcap hits have rather small values compared to the barrel distributions. In the track errors for x-residuals, two classes of barrel hits are clearly present. These have been identified as hits on the innermost Pixel layer (error roughly 17 $\mu$m) and hits on the other two layers (error roughly 8 $\mu$m). The reason for the larger track error on the inner layer is that it corresponds to the first point of a track and thus no interpolation between enclosing hits is possible. Therefore these hits get values greater than 1 for the squared error ratios and the total error is dominated by the correlation to other hits on subsequent barrel layers. Thus, these modules could need more iterations to converge and maybe even get oscillating alignment parameters around the true position. A vertex constraint on the fitted tracks should significantly improve the track error for the inner barrel layer and thus provide a good alignment for this layer.

As discussed in section 4.3, the error estimates are not correct because of the large fraction of cluster hits, for which the classical error estimate is obviously non-optimal. However, this should not influence the general validity of the alignment approach: For correct error estimates, $\sigma_{Hit}$ is expected to develop several spikes in the x-residuals. For y-residuals, the large smearing coming from the incident angle is expected to cover the effects from different cluster sizes. $\sigma_{Track}$ should not change very much for either coordinate, since the tracking has already tried to consider imperfect hit resolutions internally. This issue

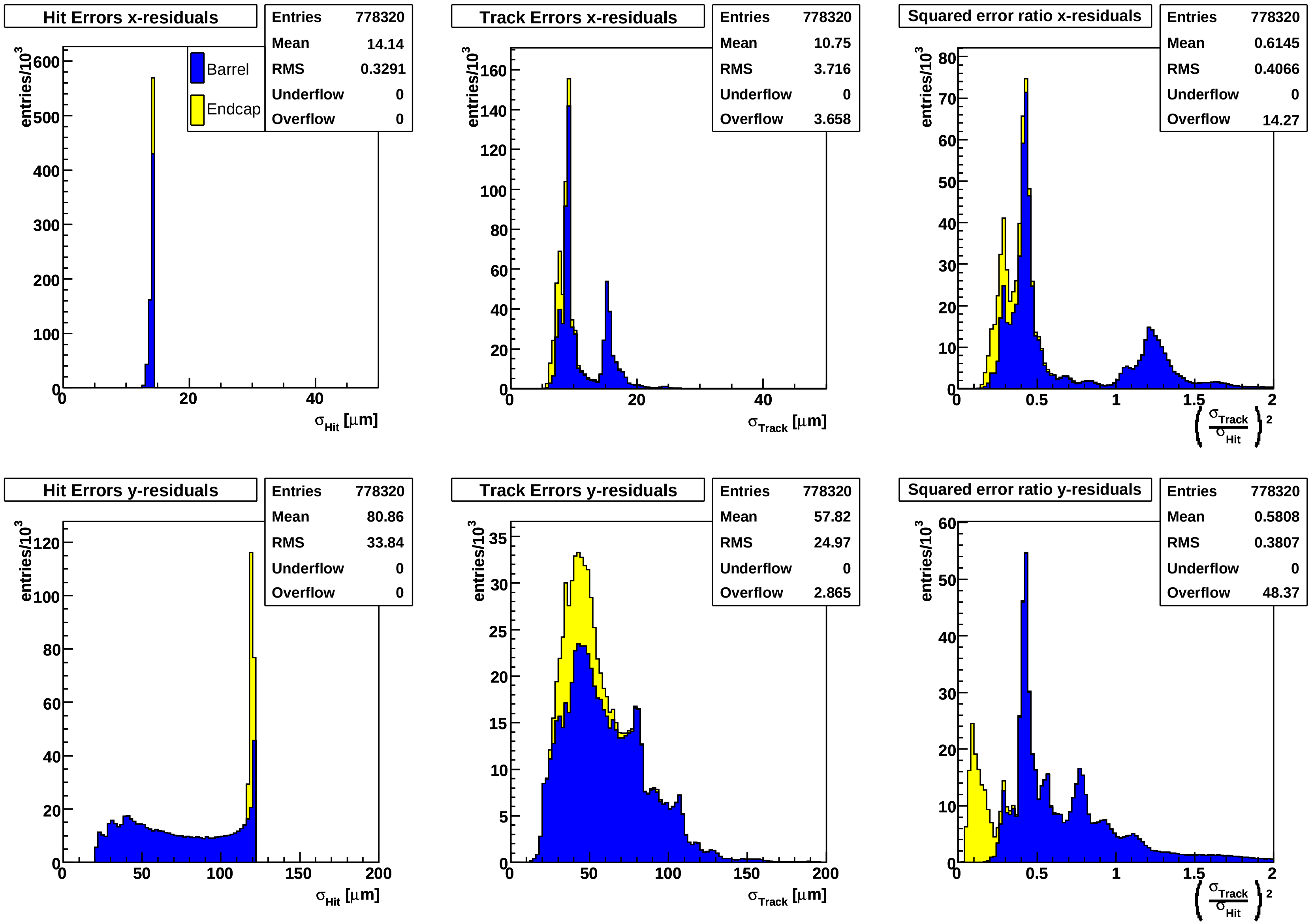

Figure 4.6: *Error distribution for x-residuals (upper plots) and y-residuals (lower plots) split up into hit resolution error and track fitting error. The squared ratio of them denotes the influence of module-to-module correlations. Barrel hits are shown in blue, endcap hits in yellow.*

will be studied using the error table from the `SiClusterizationTool` in newer releases of Athena.

## 4.5 Calculation of derivatives

For calculating the derivatives numerically, the `StraightLineTool` is used to shift and rotate the x- and y-strips by a small amount for all six degrees of freedom. The track extrapolation is then repeated using a different strip position to obtain residuals and the derivatives are calculated using equation (3.11).

In figures 4.7 and 4.8, the derivatives are shown for barrel modules on the innermost layer at $\eta = 0$. They are directly comparable to figures 3.4 and 3.5. The ranges for which derivatives occur are the same as in the ROOT simulation, and only very small differences in the shape of the distributions are visible for some degrees of freedom. The definition of the sign of the residuals is obviously different from the ROOT simulation.

These small differences in distribution shape are due to the different input distributions of the residuals. The DOCA-residual distributions in the ROOT simulation are nearly uniform distributions. Athena residuals acquire a gaussian shape, as discussed in previous sections.

The derivative distributions of other barrel layers essentially look the same as shown in figures 4.7 and 4.8. But there are differences for different $\eta$ positions on the staves: For larger $|\eta|$, $\frac{\partial r_x}{\partial \alpha}$ and $\frac{\partial r_x}{\partial \gamma}$ get a slope due to different crossing angles of the tracks. $\frac{\partial r_y}{\partial y}$ shifts to the left and eventually becomes a spike near zero. $\frac{\partial r_y}{\partial z}$ wanders towards $\pm 1$ and becomes slimmer and taller, thus giving greatly increased sensitivity in the z-direction for modules further out in $\eta$.

Figures 4.9 and 4.10 show the derivative plots for all endcap modules. Here the small variations of the crossing angle about the right angle show up in many slim and spiky distributions. Sensitivity for z is nearly lost, since $\frac{\partial r_x}{\partial z} \approx 0$ and $\frac{\partial r_y}{\partial z}$ is quite small. Another feature is the change of sign between the two endcaps due to the definition of `localR`. This sign change then shows up in the derivatives $\frac{\partial r_x}{\partial x}$ and $\frac{\partial r_y}{\partial y}$.

Overall, it can be seen from the distributions that the coordinates most sensitive to alignment should be x, $\alpha$ and $\gamma$ in the barrel and x and $\gamma$ in the endcap part of the Pixel detector.

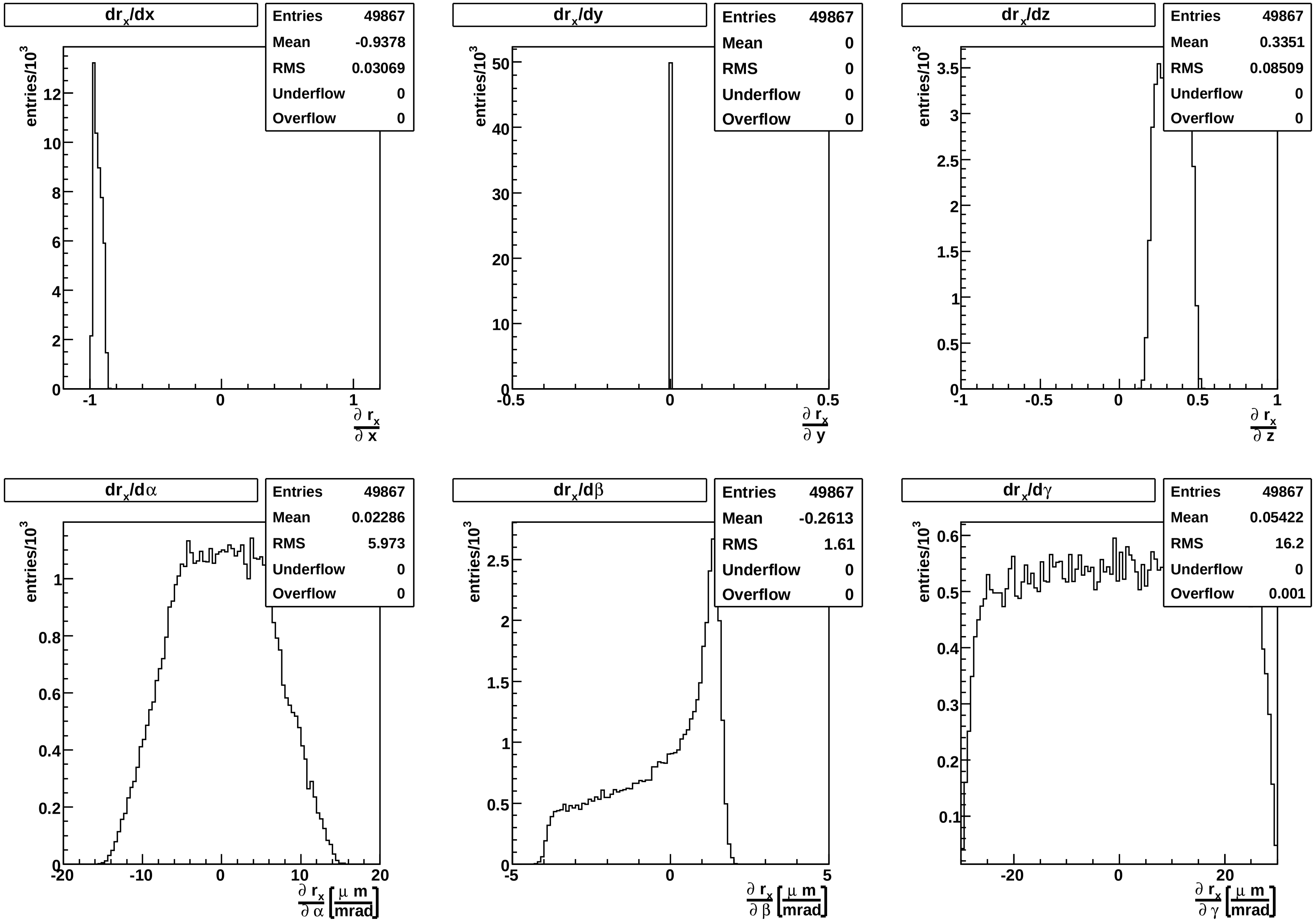

Figure 4.7: *Residual derivatives of $x$ residuals for the 22 modules on the inner barrel layer which are central on a stave. This can be directly compared to fig. 3.4*

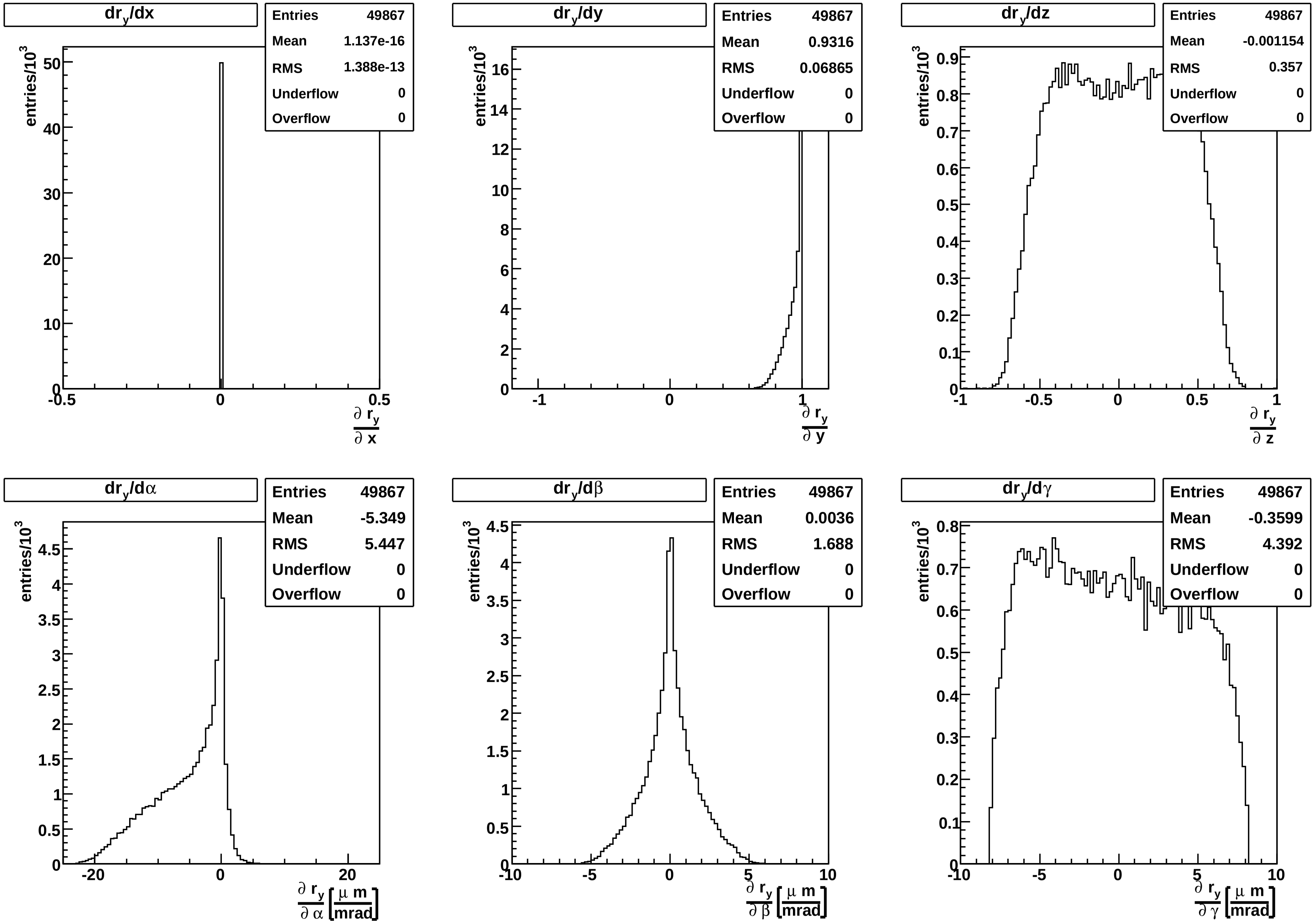

Figure 4.8: *Residual derivatives of y residuals for the 22 modules on the inner barrel layer which are central on a stave. This can be directly compared to fig. 3.5.*

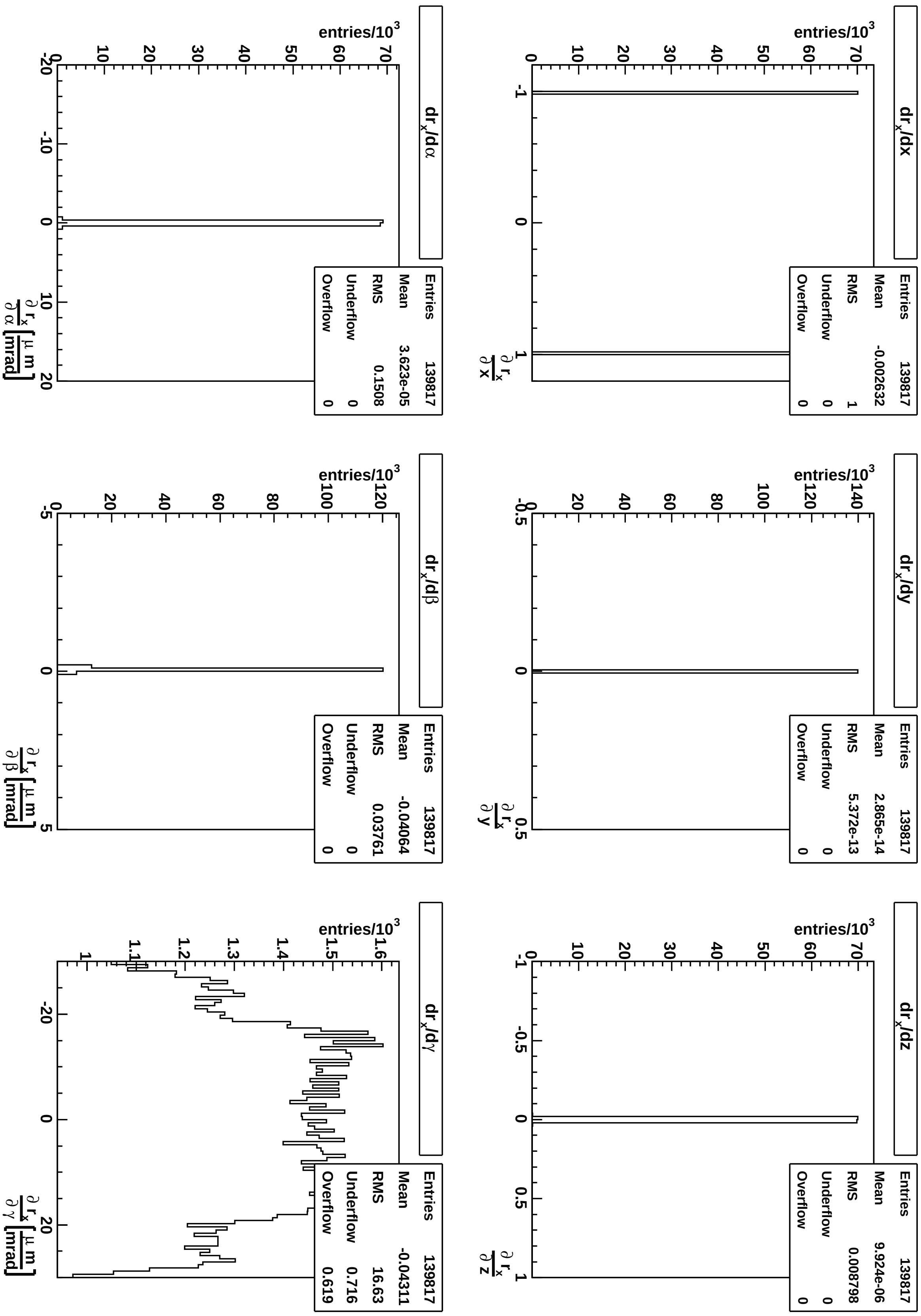

Figure 4.9: *Residual derivatives of x residuals in the endcap part of the Pixel detector.*

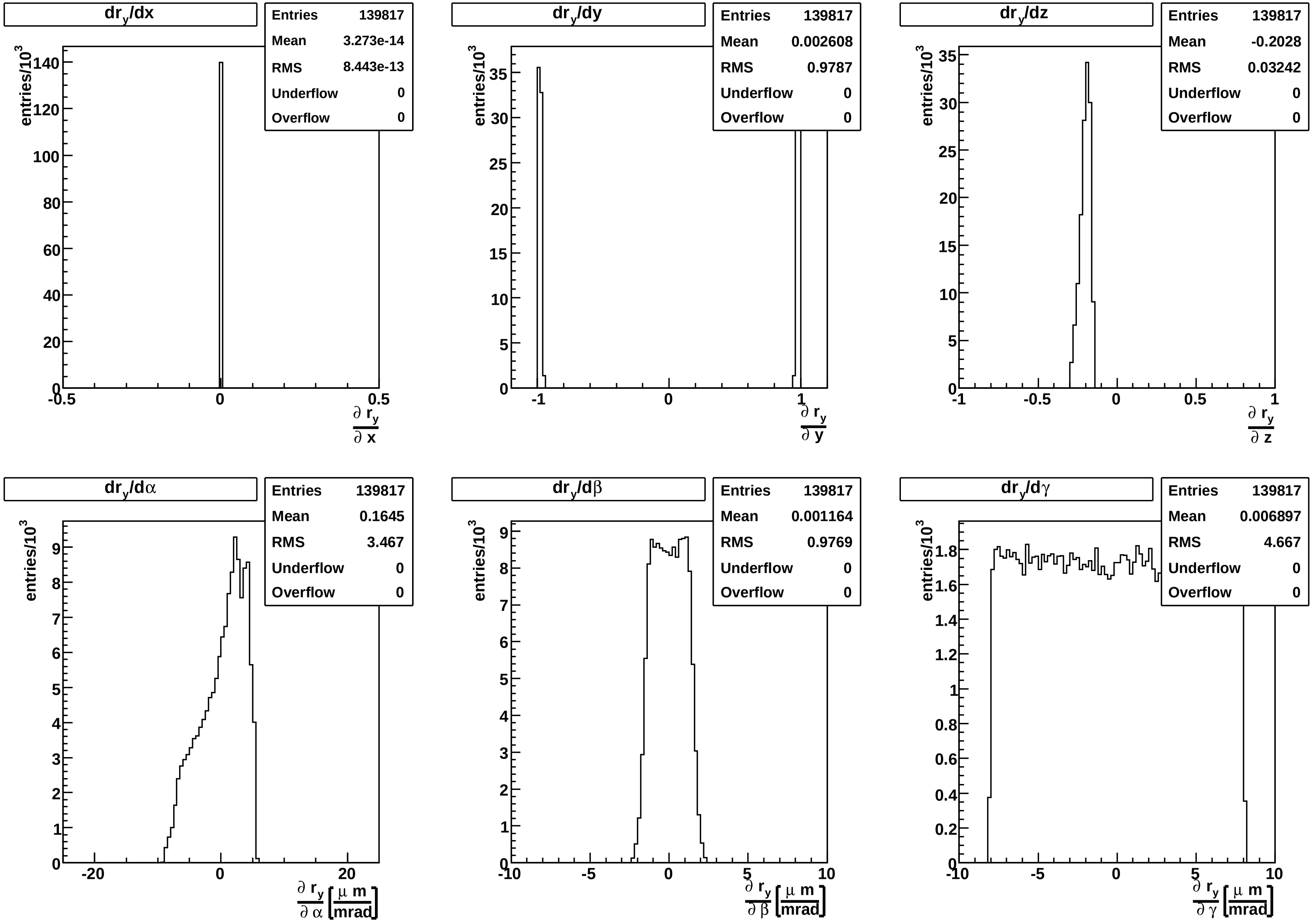

Figure 4.10: *Residual derivatives of $y$ residuals in the endcap part of the Pixel detector.*

# Chapter 5

# Validation and results

## 5.1 Multiple muon sample

As input sample for alignment test with full ATLAS geometry, a sample of 1 million simulated muon tracks, which is intended to also validate the global $\chi^2$ alignment algorithm, was taken [46, 47]. The sample consists of 100 000 events with 10 muons per event coming from a single vertex. The vertex spreads around the nominal interaction point with a three-dimensional gaussian distribution, having $\sigma_{\hat{z}} = 10\,mm$ and $\sigma_{\hat{x}\hat{y}} = 1\,mm$. It was simulated with Athena release 11.0.4 and covers $|\eta| < 3$ with a flat distribution. The transverse momenta of the muons are simulated with a uniform distribution and range from 2 to 50 GeV/c.

The events were reconstructed with Athena release 11.0.42 and saved to Event Summary Data (ESD) files. ESD is a file format “which contains the detailed output of the detector reconstruction and will be produced from the raw data. It will contain sufficient information to allow particle identification, track re-fitting, jet calibration etc. thus allowing for the rapid tuning of reconstruction algorithms and calibrations” [48]. This allows for fast alignment runs with track refitting, but without the need to repeat the time-consuming pattern recognition.

Due to a memory leak in other parts of Athena, only 30 000 events in the ESD files could be processed. The alignment procedure was then executed with `Chi2AlignAlg` from the version `SiRobustAlignAlgs-00-00-37` of the alignment package. Since the inner detector has an acceptance region of $|\eta| < 2.5$, not all muon tracks are reconstructed. Additionally, a cut on the momentum is imposed and only tracks with a momentum greater than 10 GeV/c are considered to minimize the impact of multiple coulomb scattering on the alignment. After applying these cuts, 235845 tracks are used for alignment runs.

In figure 5.1, the reconstructed track quantities are shown. One can see that the generated distributions are reconstructed well: The vertex from which the muons emerge has gaussian shape with the correct widths. The distributions in the track angles are uniform with $-\pi < \phi < \pi$ and $|\eta| < 2.5$. The reconstructed transverse momentum ranges from 2 to 50 GeV/c with a flat distribution, then falls off until about 60 GeV/c. Thus, the momentum extends to about 300 GeV/c at $|\eta| = 2.5$. The number of reconstructed tracks is expected to be on average $10 \times \frac{2.5}{3} \approx 8.3$ due to the acceptance of the Inner Detector, which is fulfilled. The fit quality of the tracks, given by $\chi^2$ divided by the number of degrees of freedom, peaks around one and falls off soon.

With this track sample, each module gets a certain constant number of hits. Their distribution can be seen in fig. 5.2. It was seen during the studies that a minimal number of hits per module is crucial for the determination of reasonable alignment parameters. As the acceptance of the whole Pixel detector decreases towards $|\eta| = 2.5$, the outermost Pixel barrel modules receive only few hits: The interval of $\eta$ which they are covering is smaller than in the central parts and additionally, only few tracks leave enough hitpoints on the subsequent layers of the Inner Detector to yield a usable track fit. It was decided to only align modules which received at least 50 hits. Thus, the 44 outer modules on barrel layer 0 were not aligned. With more statistics due to a larger track sample and by the help of the survey data, which constrains these modules relative to the neighboring ones on the same stave, this issue will be addressed in subsequent studies. In fig. 5.2 one can also see the 7-peak structure in barrel layer 0: 2 positions at a time out of the 13 possible ones on a barrel stave share the same $|\eta|$ range, thus getting about the same number of tracks. In the other two layers, this peak structure is more compressed, since the modules do not reach out as far in $|\eta|$. The endcap modules on disk 0 and disk 1 nearly get the same number of hits. Disk 2 already falls into an $\eta$-region where the acceptance of the Pixel detector is small and thus less tracks are reconstructed in this region.

All tracks were then refitted using the `KalmanFitter` and without TRT hits, as described in section 4.2.

## 5.2 Results with nominal alignment

Figures 5.3 – 5.6 show the results of one alignment iteration over the selected sample for both the barrel and the endcap part of the Pixel detector.

For the barrel, the most sensitive coordinates are x, z and the angles, especially $\gamma$, just as discussed in section 4.5. Sensitivity in z again comes into play by the incident angle and even constrains z better than y. The distributions for layer 0 are narrower than those for layer 1, which are in turn narrower than the distributions for layer 2. This is expected due to the difference in the number of hits these layers get due to their larger distance from the interaction point. The centers of the distributions lie within $2\,\mu$m or less from the origin, with a perceived shift of the inner layers to positive values in x and z. It is investigated, if a problem in the track fitting could be the cause of this shift. The distribution in y is not gaussian but seems to include two different classes of alignment parameters. These are from modules which are in central regions of the barrel, have less propability for long clusters in y, and thus get more correct alignment input, and modules on the outer parts of the barrel having essentially wrong residual errors due to the large effect of the clustering. The widths of the angular distributions are all smaller than 1 mrad. Due to the strict cut on the number of hits per module, only 1412 modules get aligned, but also no significant tails appear in the distributions.

For the endcap, sensitivity in z is not as good, since tracks cross almost perpendicularly. The x- and y-coordinates are sensitive, together with $\gamma$. y is far more constrained than in the barrel case, where a large number of cluster hits spoils the calculation by their incorrect errors. However, the overall distribution in y is shifted by almost 6 $\mu$m to negative values.

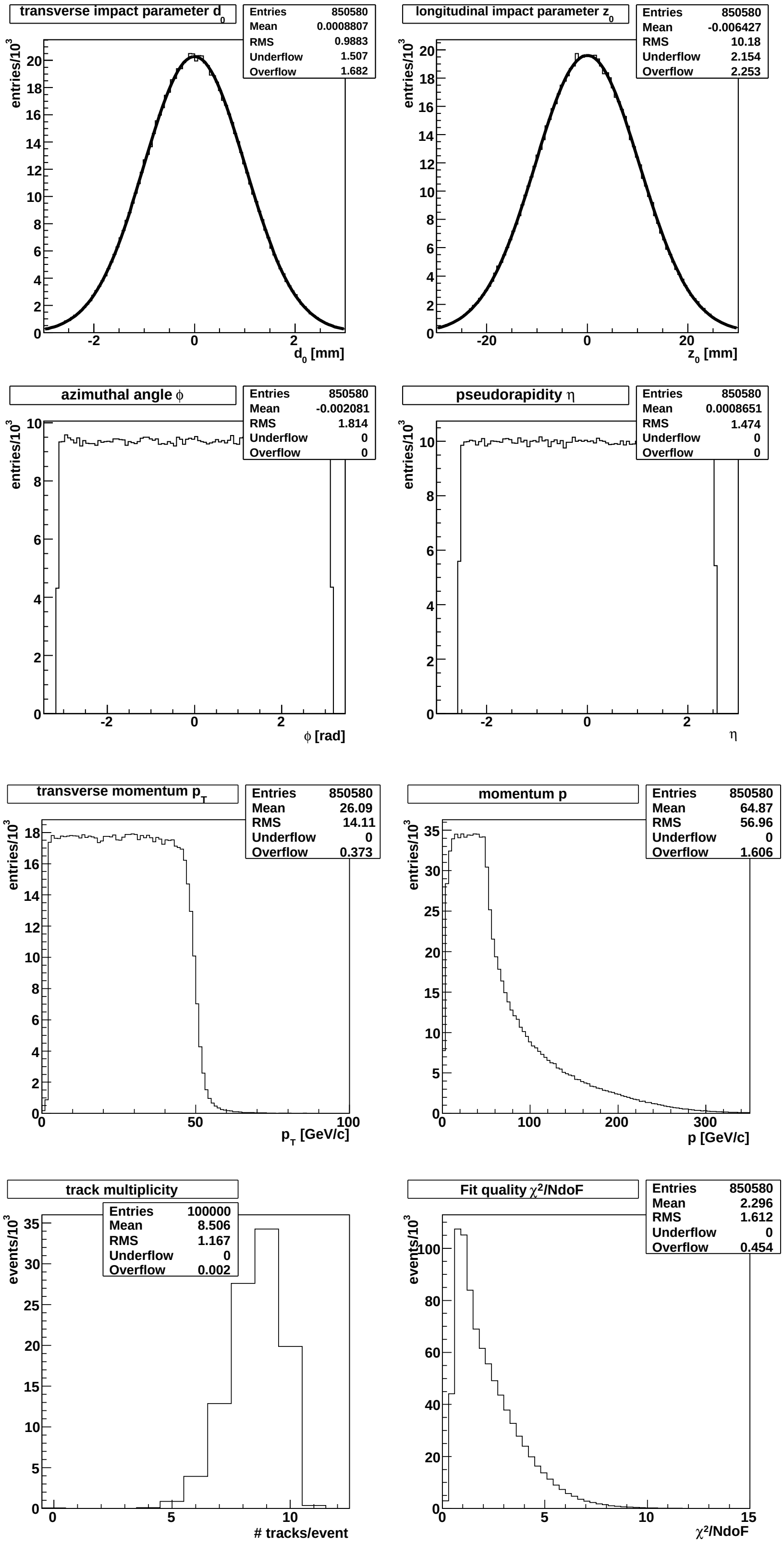

Figure 5.1: *Track quantities reconstructed from the multimuon sample.*

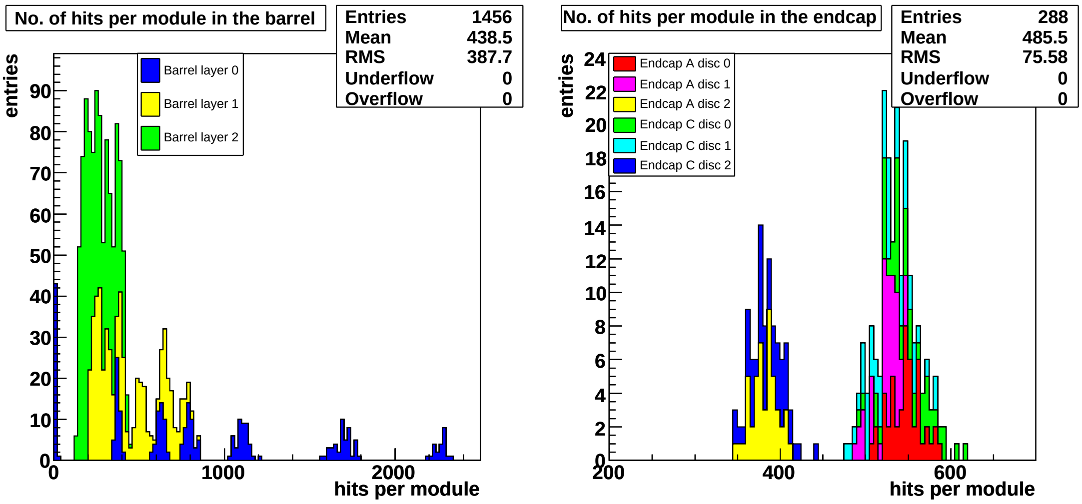

Figure 5.2: *Hits per module distributions on the barrel and endcap.*

This is assumed to be related to the barrel shift described above. The alignment parameter distributions have essentially the same width for all endcap disks since there is no clear separation in terms of the number of hits they get.

The widths of the pull distributions are close to 1, but some discrepancies remain. This is due to the wrong error assignments for the residuals, which are propagated through the alignment algorithm. The shifts of $a_x$, $a_z$ and $a_\beta$ are emphasized in the pull distributions of the barrel and clearly visible since many hits in the module yield low errors on the final alignment parameters.

Tables 5.1 and 5.2 give the correlation between the 6 alignment parameters, analogue to table 3.2. All values are averaged over the whole barrel or endcap. The degrees of freedom in the endcap are all weakly correlated due to the almost perpendicular crossing angles. In the barrel, the values are comparable to those from the ROOT simulation, some discrepancies are due to the varying incident angle for modules at larger $\eta$. From this table, it is understood that the three degrees of freedom x, z and $\beta$ are off at the same time in the alignment parameter distribution because they are correlated.

Correlations $c_{ij}$ for Pixel barrel modules

| coordinates | x | y | z | $\alpha$ | $\beta$ | $\gamma$ |
|---|---|---|---|---|---|---|
| x | 1 | $2.72 \cdot 10^{-4}$ | 0.992 | $1.38 \cdot 10^{-3}$ | 0.745 | $9.05 \cdot 10^{-4}$ |
| y | – | 1 | $-2.66 \cdot 10^{-4}$ | $-0.384$ | $2.54 \cdot 10^{-4}$ | 0.307 |
| z | – | – | 1 | $1.42 \cdot 10^{-3}$ | 0.759 | $9.77 \cdot 10^{-4}$ |
| $\alpha$ | – | – | – | 1 | $8.91 \cdot 10^{-4}$ | $-0.839$ |
| $\beta$ | – | – | – | – | 1 | $-9.00 \cdot 10^{-4}$ |
| $\gamma$ | – | – | – | – | – | 1 |

Table 5.1: *Correlations of the local coordinates for all Pixel barrel modules. The values are averaged over all modules in the barrel.*

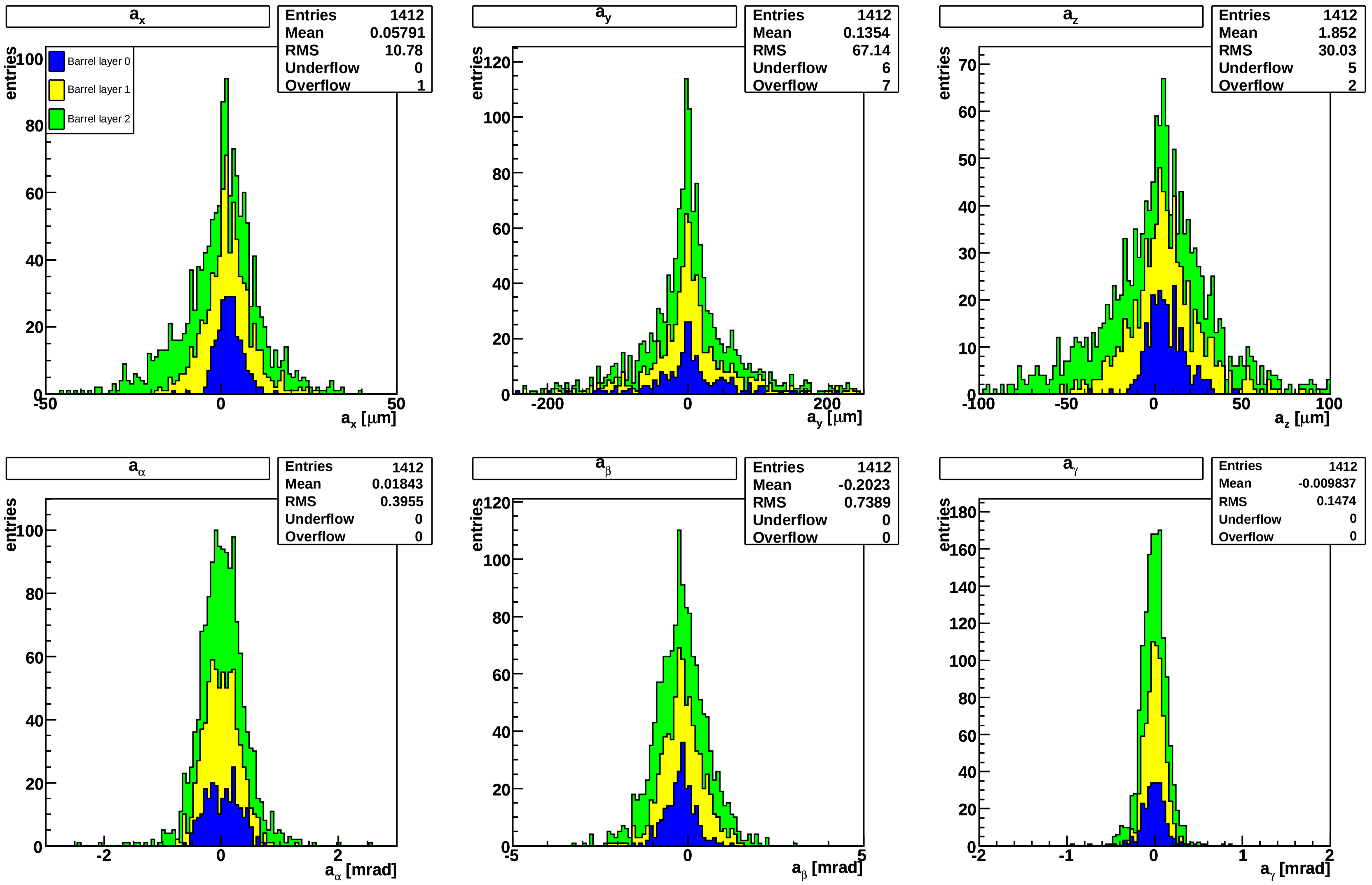

Figure 5.3: *Calculated alignment parameters after 1 iteration for the Pixel barrel modules based on the multimuon sample.*

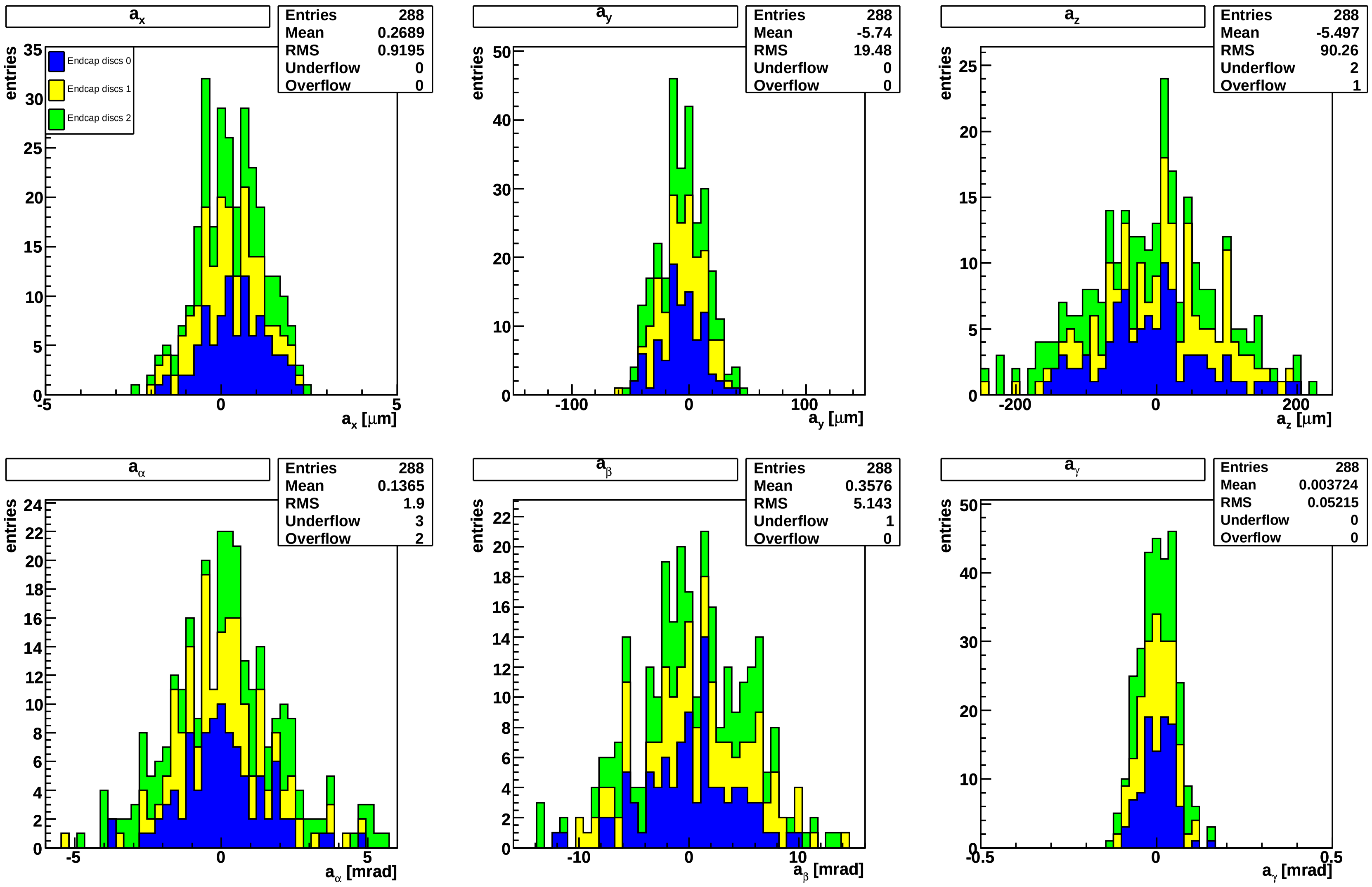

Figure 5.4: *Calculated alignment parameters after 1 iteration for the Pixel endcap modules based on the multimuon sample.*

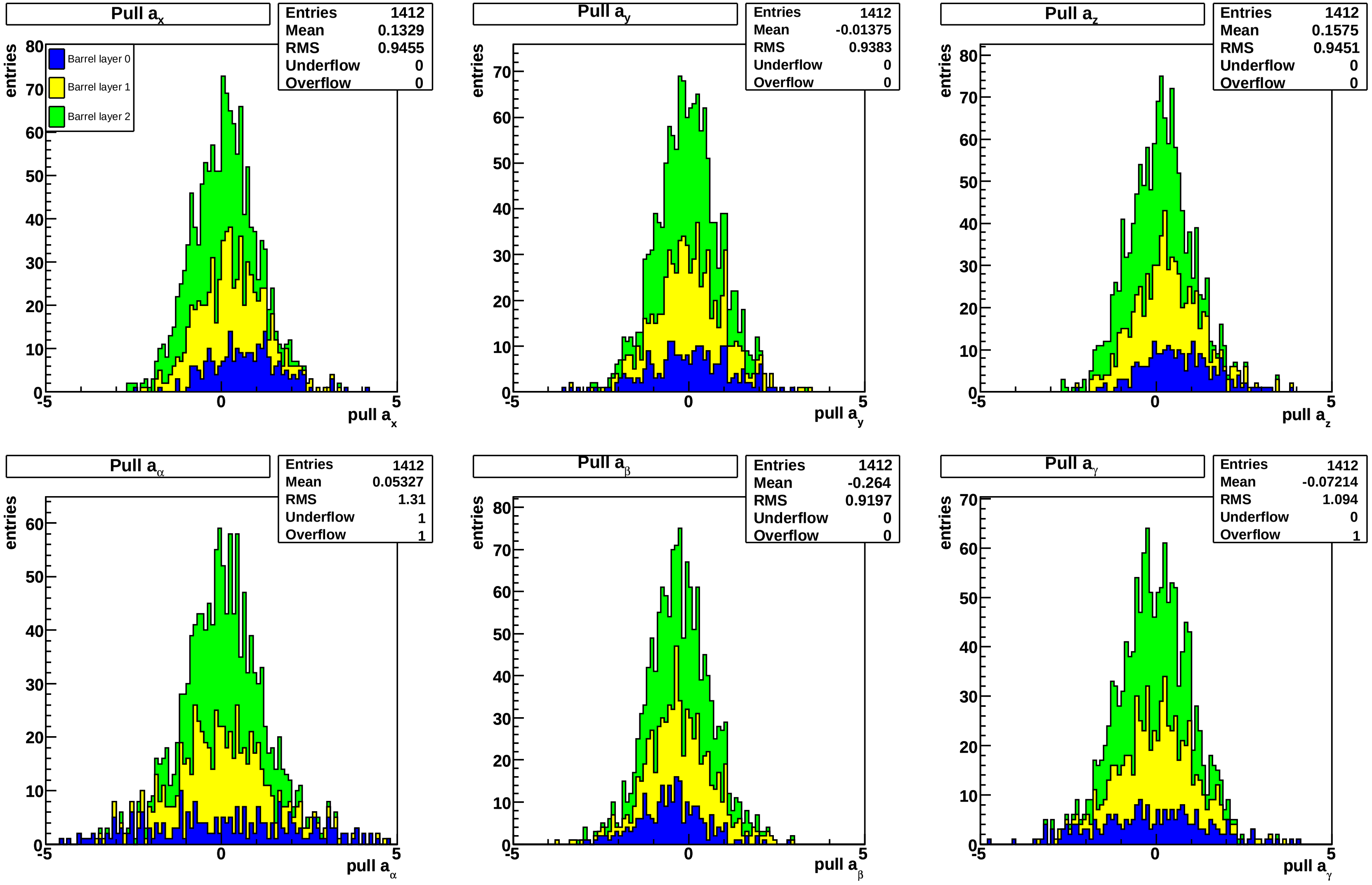

Figure 5.5: *Pull distribution of the alignment parameters after 1 iteration for the barrel modules based on the multimuon sample.*

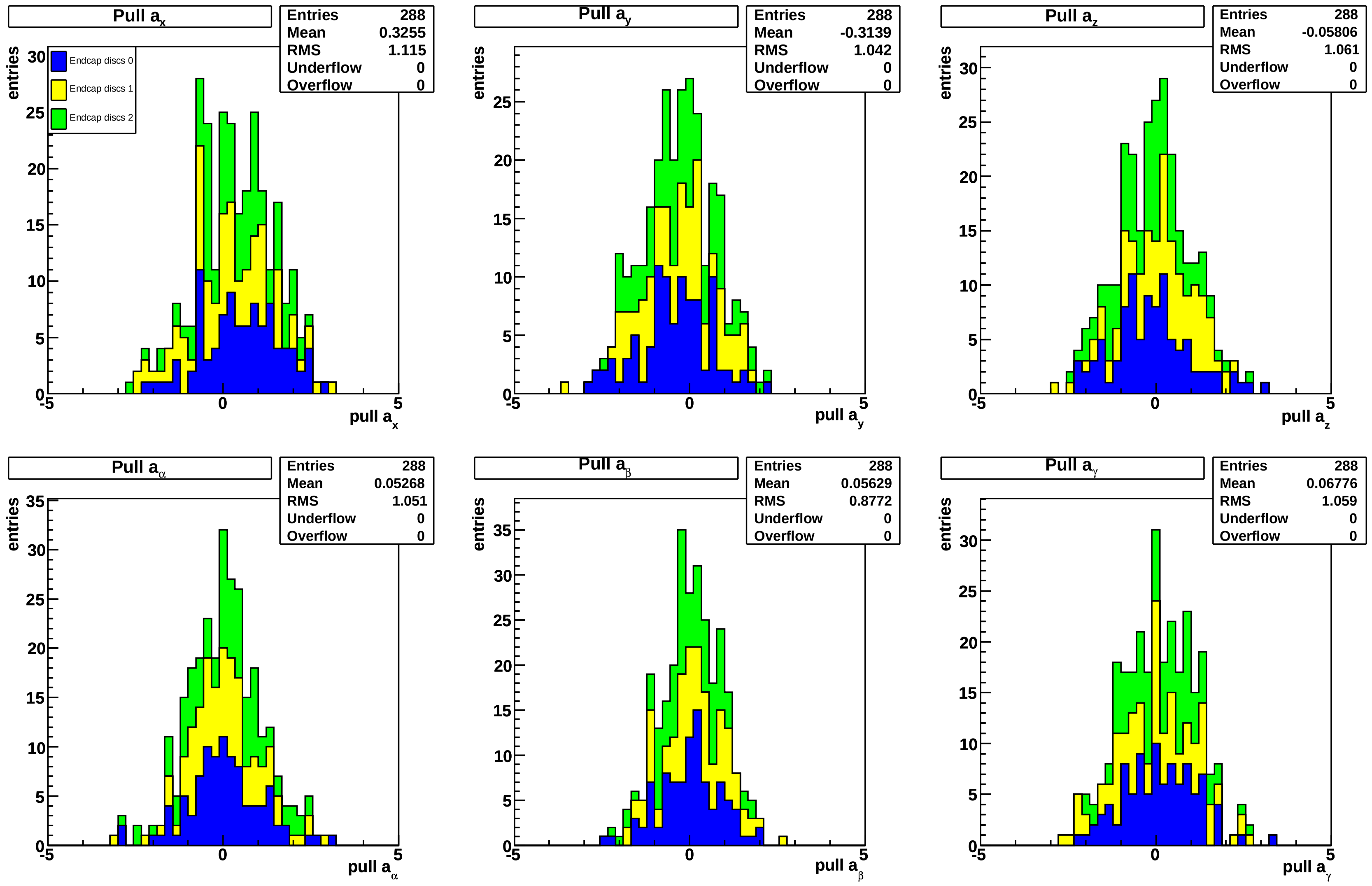

Figure 5.6: *Pull distribution of the alignment parameters after 1 iteration for the endcap modules based on the multimuon sample.*

| Correlations $c_{ij}$ for Pixel endcap modules | | | | | | |
|---|---|---|---|---|---|---|
| coordinates | x | y | z | $\alpha$ | $\beta$ | $\gamma$ |
| x | 1 | $-8.20 \cdot 10^{-4}$ | $-1.70 \cdot 10^{-3}$ | $1.96 \cdot 10^{-3}$ | $6.16 \cdot 10^{-4}$ | $-0.0728$ |
| y | – | 1 | $-1.17 \cdot 10^{-4}$ | $4.14 \cdot 10^{-4}$ | $-9.58 \cdot 10^{-4}$ | $7.33 \cdot 10^{-4}$ |
| z | – | – | 1 | $-0.318$ | $-2.74 \cdot 10^{-6}$ | $-9.67 \cdot 10^{-3}$ |
| $\alpha$ | – | – | – | 1 | $1.56 \cdot 10^{-3}$ | $3.67 \cdot 10^{-3}$ |
| $\beta$ | – | – | – | – | 1 | $-7.34 \cdot 10^{-4}$ |
| $\gamma$ | – | – | – | – | – | 1 |

Table 5.2: *Correlations of the local coordinates for all Pixel endcap modules. The values are averaged over all modules in the endcap.*

### 5.2.1 Iterations

The algorithm intrinsically depends on iterations. The results of the alignment algorithm over 10 iterations on nominal geometry were studied to check for the convergence on the final solution and as a reference for misalignment studies.

At the moment, iterating over a track sample is possible via files containing the alignment parameters computed during an earlier iteration of the algorithm. They are written out at the end of an iteration. At the beginning of the next iteration, the geometry model of the detector can read in the information usually received from the ConditionsDatabase from this file instead. During all iterations, the whole SCT detector was fixed to not introduce any additional effects on the alignment of the Pixel modules.

After 10 iterations with nominal geometry, alignment accuracy limits can be inferred from the alignment parameter distributions. They are shown in table 5.3. Four limits have been obtained: 95% confidence level (CL) is the range where 95% of all entries of the histogram fall in, starting from the center of the histogram. This corresponds to the $2\sigma$ error of a gaussian distribution. Similarly, 68% CL corresponds to $1\sigma$ of a gaussian. The $\sigma_{gauss}$ value is obtained from a gaussian fit onto the distribution. Finally, $\sigma_{statistical}$ gives the mean of the alignment parameter errors of all modules, calculated using eq. (2.13). All numbers in the table were calculated from figures 5.7 and 5.8, which show the final alignment parameters after ten iterations. Additionally, the flow of all modules, i. e. the evolution of alignment parameters over subsequent iterations, is depicted in figures 5.9 and 5.10.

As one can see in these flowplots, the alignment parameters distribute in an interval around zero alignment after the first iteration. Then they essentially stay within this interval. Modules whose alignment parameters get away too far from zero are mostly brought back within the next iteration. In the endcaps, some modules parameters behave more erratic and do not reconverge after one iteration due to the lower quality of input (less tracks, which are almost crossing at a right angle). There is one module in the barrel and two in the endcaps which experience a large movement in their alignment parameter evolution, from which they need some iterations to recover. They are left out in these plots to be able to distinguish the other module's movements with increased resolution. This behavior is under investigation, but it is assumed that badly reconstructed tracks are the reason for it.

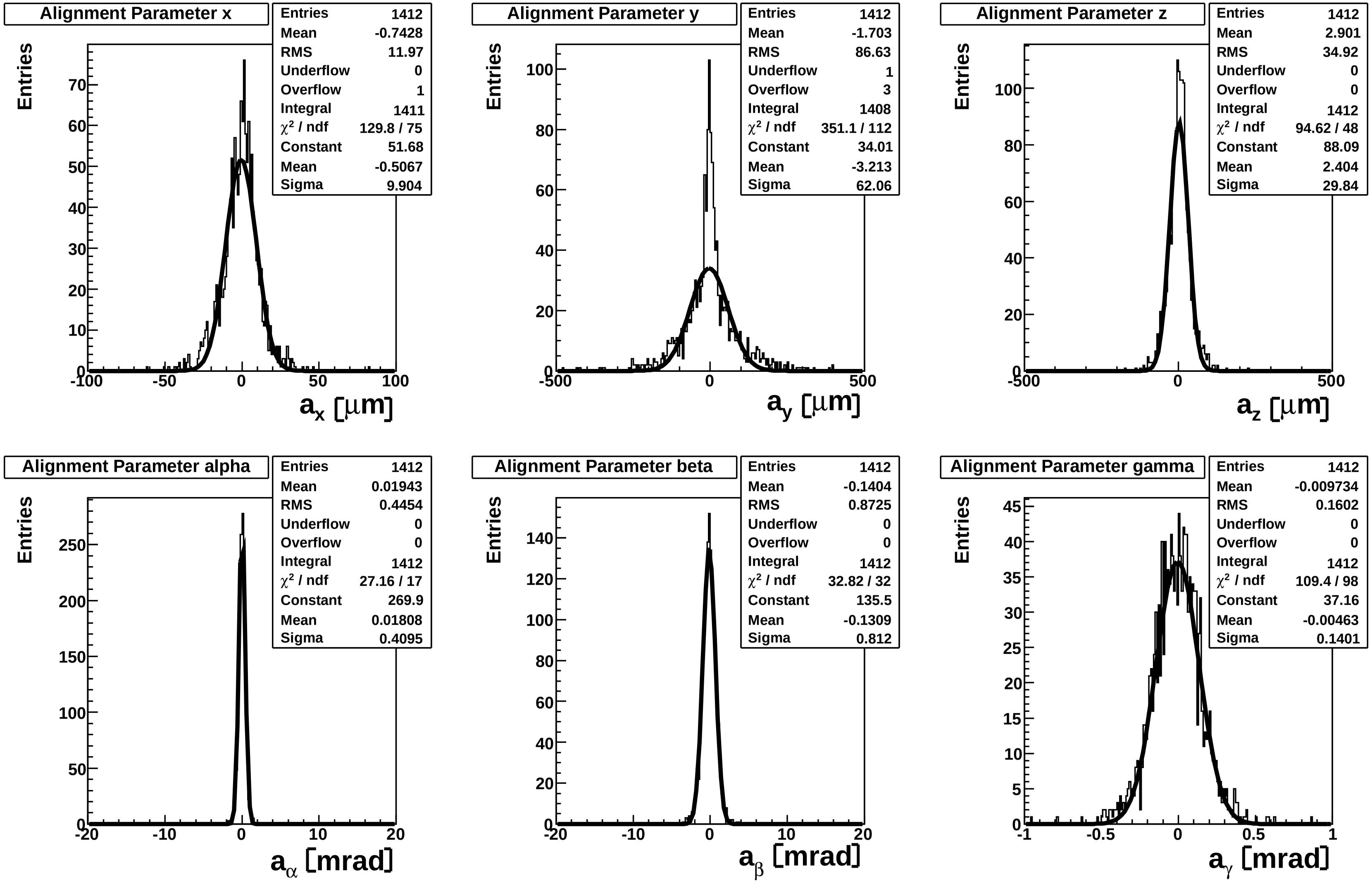

Figure 5.7: *Calculated alignment parameters after 10 iterations for the Pixel barrel modules based on the multimuon sample.*

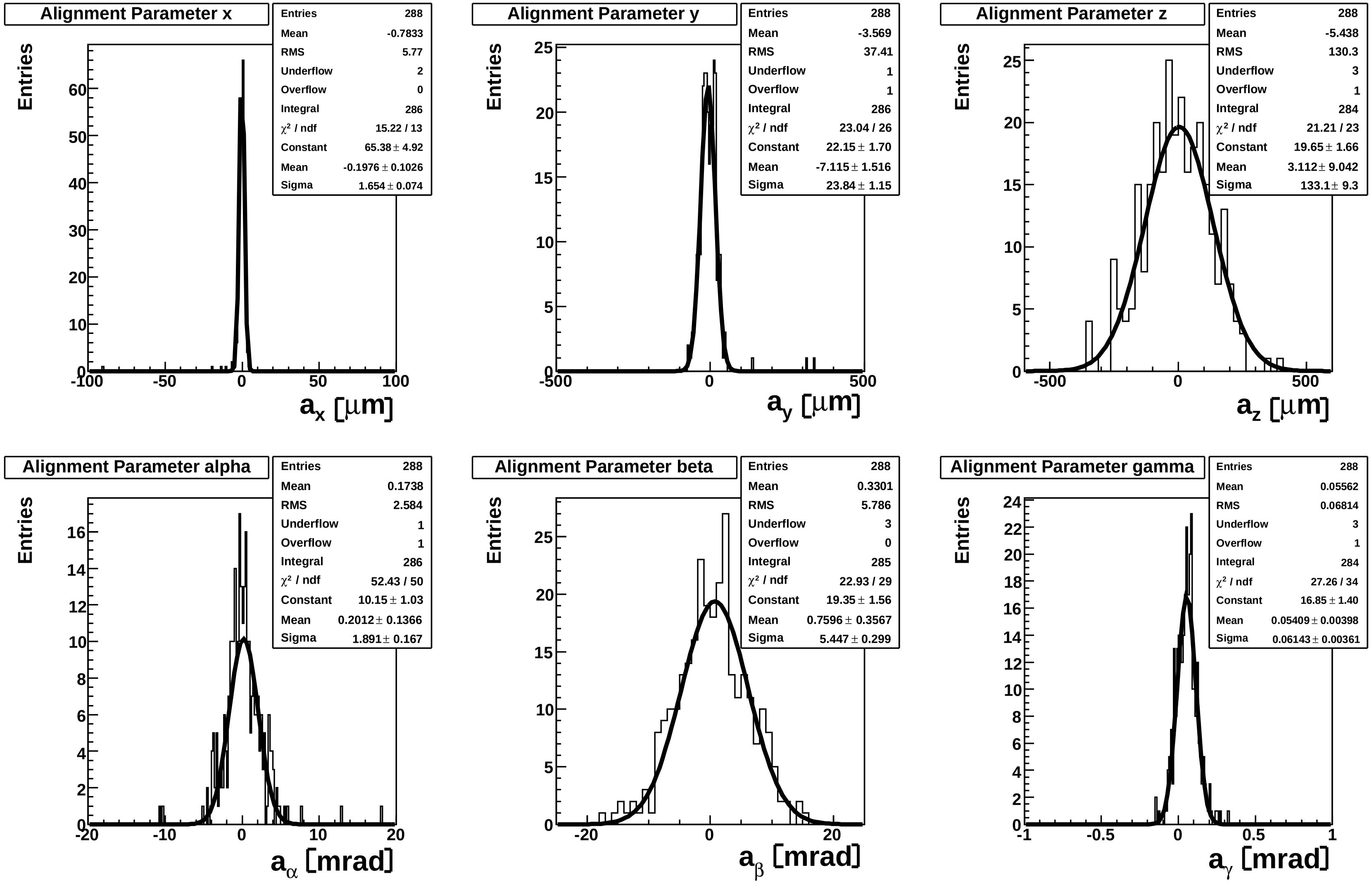

Figure 5.8: *Calculated alignment parameters after 10 iterations for the Pixel endcap modules based on the multimuon sample.*

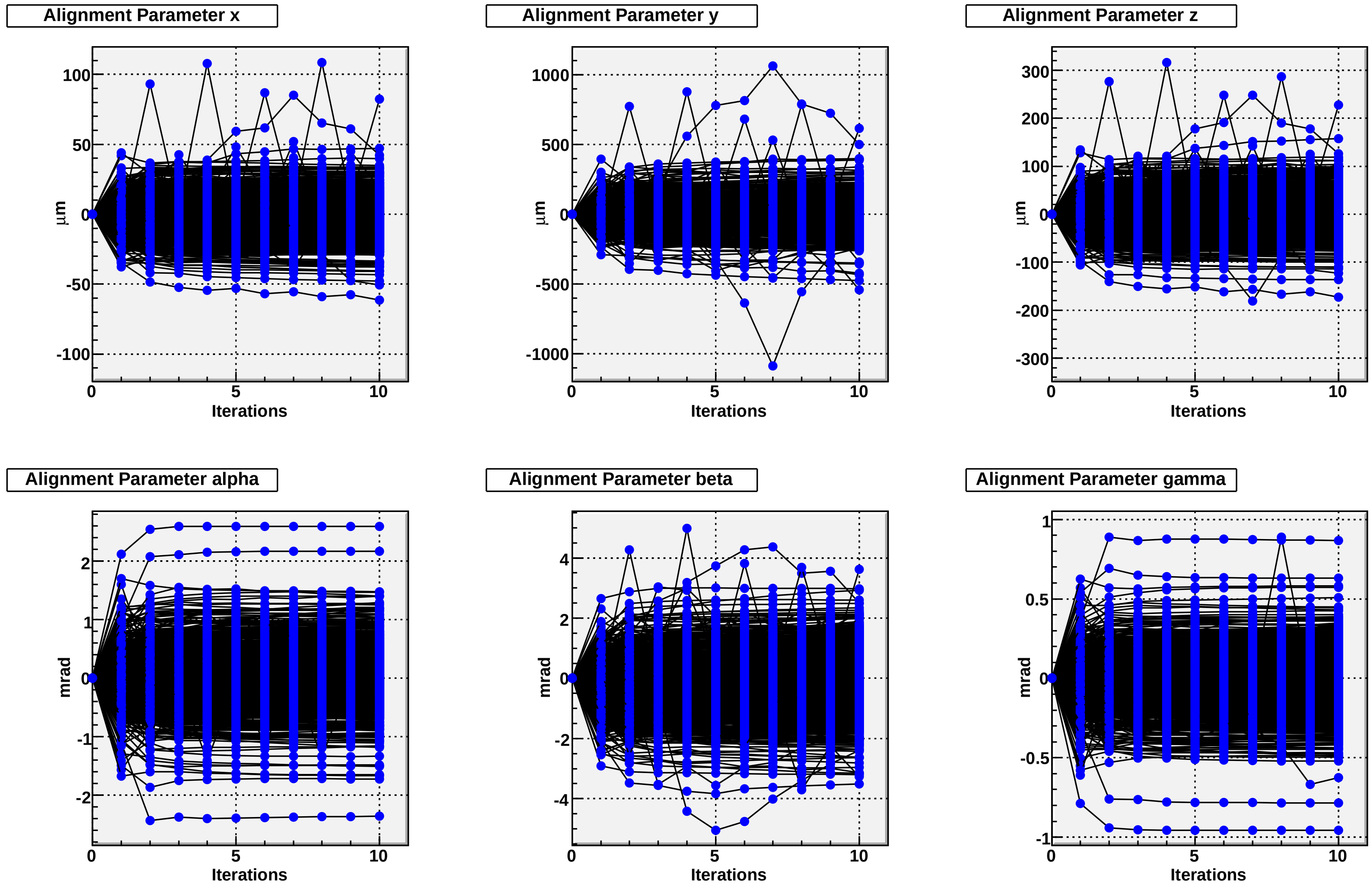

Figure 5.9: *Alignment parameter flow of all aligned barrel modules for 10 iterations based on the multimuon sample. One module is not shown.*

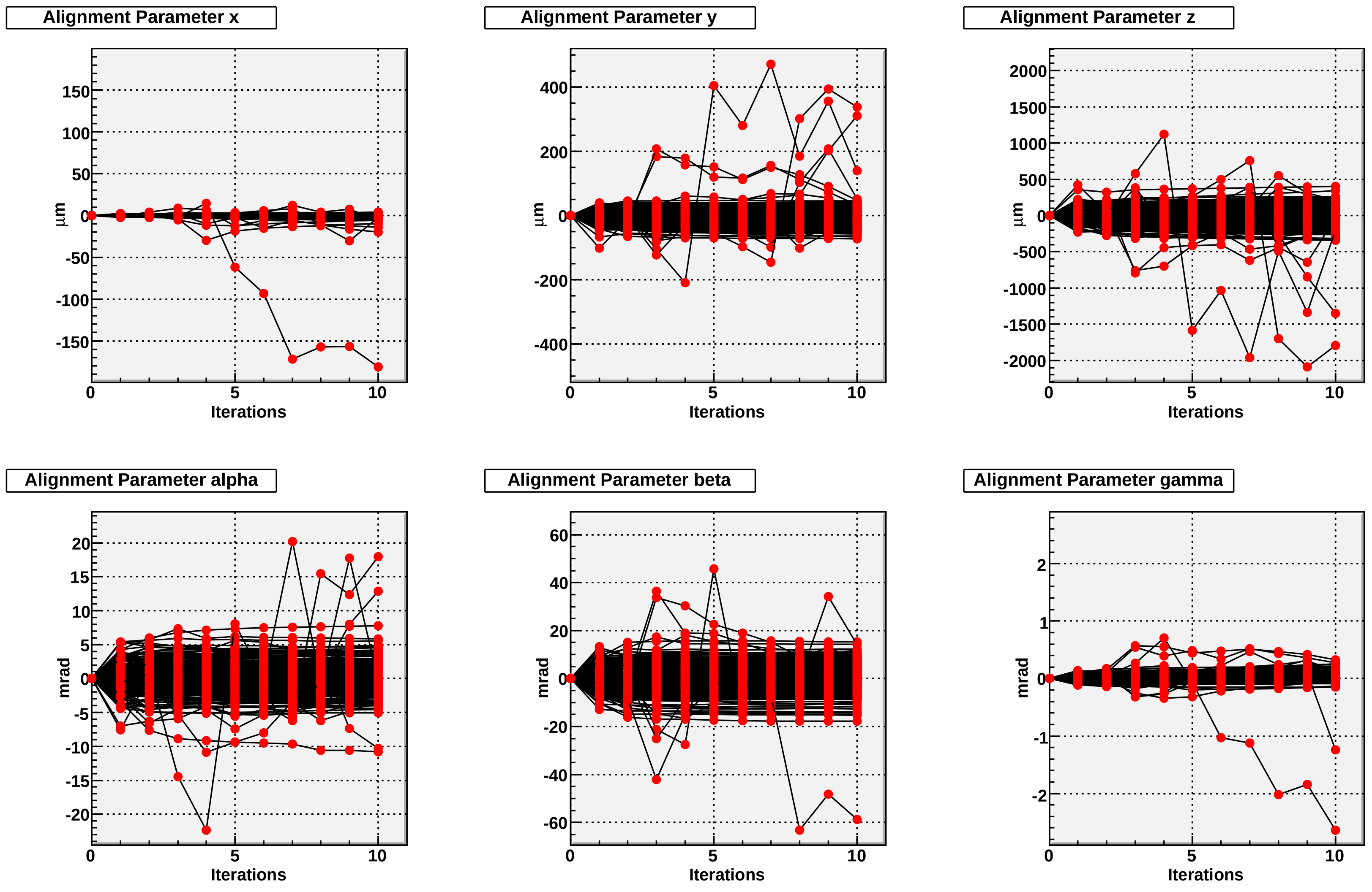

Figure 5.10: *Alignment parameter flow of all endcap modules for 10 iterations based on the multimuon sample. Two modules are not shown.*

| Alignment accuracy limits | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| alignment | Pixel barrel module | | | | Pixel end-cap module | | | |
| parameter | 95% CL | 68% CL | $\sigma_{gauss}$ | $\sigma_{stat}$ | 95% CL | 68% CL | $\sigma_{gauss}$ | $\sigma_{stat}$ |
| $a_x$ [$\mu m$] | 25 | 10 | 9.91 | 9.72 | 4 | 1.8 | 1.65 | 0.715 |
| $a_y$ [$\mu m$] | 195 | 63 | 62.1 | 55.3 | 55 | 25 | 32.1 | 16.2 |
| $a_z$ [$\mu m$] | 75 | 29 | 29.8 | 28.0 | 260 | 130 | 133 | 76.4 |
| $a_\alpha$ [$mrad$] | 0.9 | 0.4 | 0.41 | 0.31 | 4.4 | 2.0 | 1.9 | 1.6 |
| $a_\beta$ [$mrad$] | 1.7 | 0.9 | 0.81 | 0.70 | 12.2 | 5.8 | 5.4 | 5.1 |
| $a_\gamma$ [$mrad$] | 0.32 | 0.15 | 0.14 | 0.12 | 0.18 | 0.09 | 0.061 | 0.043 |

Table 5.3: *Alignment accuracy limits of `Chi2AlignAlg` based on the multimuon track sample.*

## 5.3 Studies with misalignment

Additionally, studies with several misaligned ATLAS detector setups were performed and alignment accuracy limits were deduced for these cases.

### 5.3.1 Misalignment setups

The misalignment setups used to test the alignment are the following: First, the whole barrel layer 1 of the Pixel detector was misaligned with the numbers given by table 5.4. Second, a set of 18 Pixel modules was misaligned by the amounts given in table 5.5. The modules are listed in table 5.6 and were chosen such that two modules on each disc of the endcap and each layer of the barrel are displaced. These modules are distributed randomly over $\eta$ and $\phi$. The size of these misalignments is based on the accuracy our approach is capable of giving with the present sample, derived from the previous chapter, and the recommendations for misalignment studies circulated by the Inner Detector alignment group coordination [49].

Since at the moment there are no Monte-Carlo samples with a simulated misalignment, the misalignment is introduced at reconstruction level, i. e. geometry information that contains shifted positions of the chosen modules is provided to the job. Together with the simulated hits stored for nominal geometry, the alignment algorithm should see the misalignment and correct for it.

| Misalignment sets for barrel layer 1 | |
|---|---|
| Layer1_Set1 | $\Delta x = -30\,\mu$m |
| Layer1_Set2 | $\Delta y = 75\,\mu$m |
| Layer1_Set3 | $\Delta z = 100\,\mu$m |

Table 5.4: *The misalignment sets for displacing barrel layer 1.*

### 5.3.2 Results from misalignment runs

Figure 5.11 shows the flow of alignment parameters for the 18 selected modules. The applied misalignment was 18Modules_Set7. As can be seen, the applied misalignment is

| Misalignment sets for individual misalignment of the 18 chosen modules | |
|---|---|
| 18Modules_Set1 | $\Delta x = -30\,\mu$m |
| 18Modules_Set2 | $\Delta y = 75\,\mu$m |
| 18Modules_Set3 | $\Delta z = 100\,\mu$m |
| 18Modules_Set4 | $\Delta\alpha = 2\,$mrad |
| 18Modules_Set5 | $\Delta\beta = -5\,$mrad |
| 18Modules_Set6 | $\Delta\gamma = 1\,$mrad |
| 18Modules_Set7 | $\Delta x = -30\,\mu$m<br>$\Delta y = 75\,\mu$m<br>$\Delta z = 100\,\mu$m<br>$\Delta\alpha = 2\,$mrad<br>$\Delta\beta = -5\,$mrad<br>$\Delta\gamma = 1\,$mrad |

Table 5.5: *The misalignment sets for displacing individual modules.*

recovered in all six degrees of freedom by the algorithm and the modules return to nominal alignment. The alignment accuracy is eventually determined by the achievable accuracy for nominal alignment. This accuracy is different for barrel and endcap modules, which can be seen in the parameter evolutions. In many degrees of freedom, alignment accuracy is even better than for the full detector. It is assumed, that especially the selected barrel modules are "well behaved" modules by chance. One module is left out in the plot because of its large movement. Looking at the results of alignment using different misalignment sets, it is notable that the small fraction of modules not aligned correctly are the same modules for most of the misalignment sets. This is supporting the hypothesis that bad tracks are the cause regardless of the position movement of the module.

Additionally, fig. 5.12 shows the evolution of the alignment parameters for the whole barrel layer 1 when misaligned with the set Layer1_Set1. Here, the applied misalignment in x is also recovered and the modules converge to stable alignment parameters in the range given by the accuracy for nominal alignment.

## 5.4 Discussion

The `Chi2AlignAlg` algorithm from the alignment package version `SiRobustAlignAlgs-00-00-37` was used together with Athena release 11.0.4. Most modules have a stable alignment parameter evolution over 10 iterations. With the present setup and the studied track sample, one is able to constrain the alignment parameters within the numbers given by table 5.3.

Misalignments introduced into the detector were recovered well within the alignment precision possible with the selected track sample. Erraticly moving modules do so with many of the initial alignment sets. Thus, it is assumed that special bad tracks could be the cause for this instead of a wrong calculation of the input quantities for alignment. More studies are underway to check for the convergence and stability of the alignment including the SCT detector and using arbitrary misalignments.

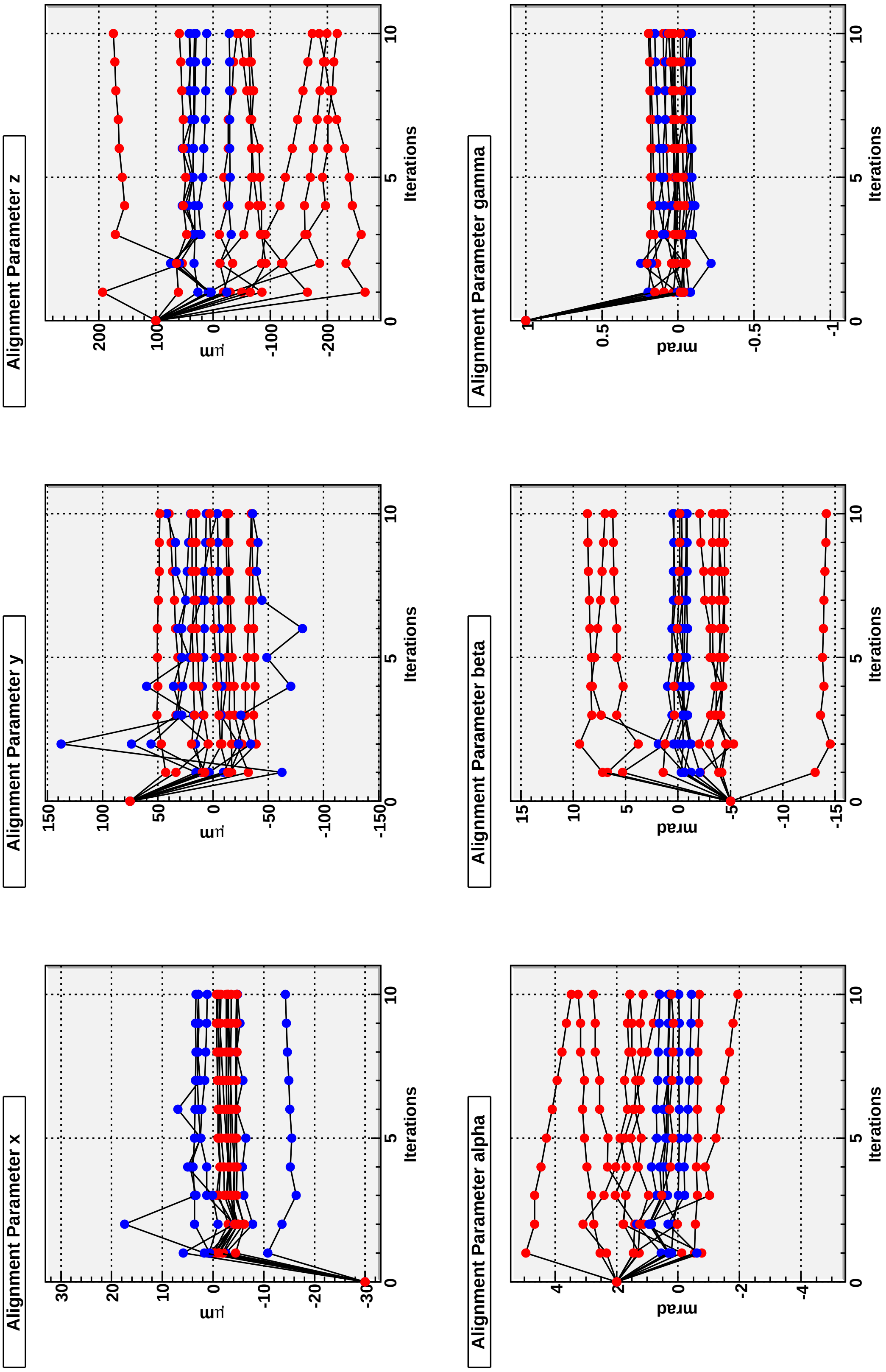

Figure 5.11: *Alignment parameter flow for the 18 selected modules over 10 iterations. The applied misalignment was 18Modules_Set7. Barrel modules are depicted in Blue, endcap modules in red. One module is not shown.*

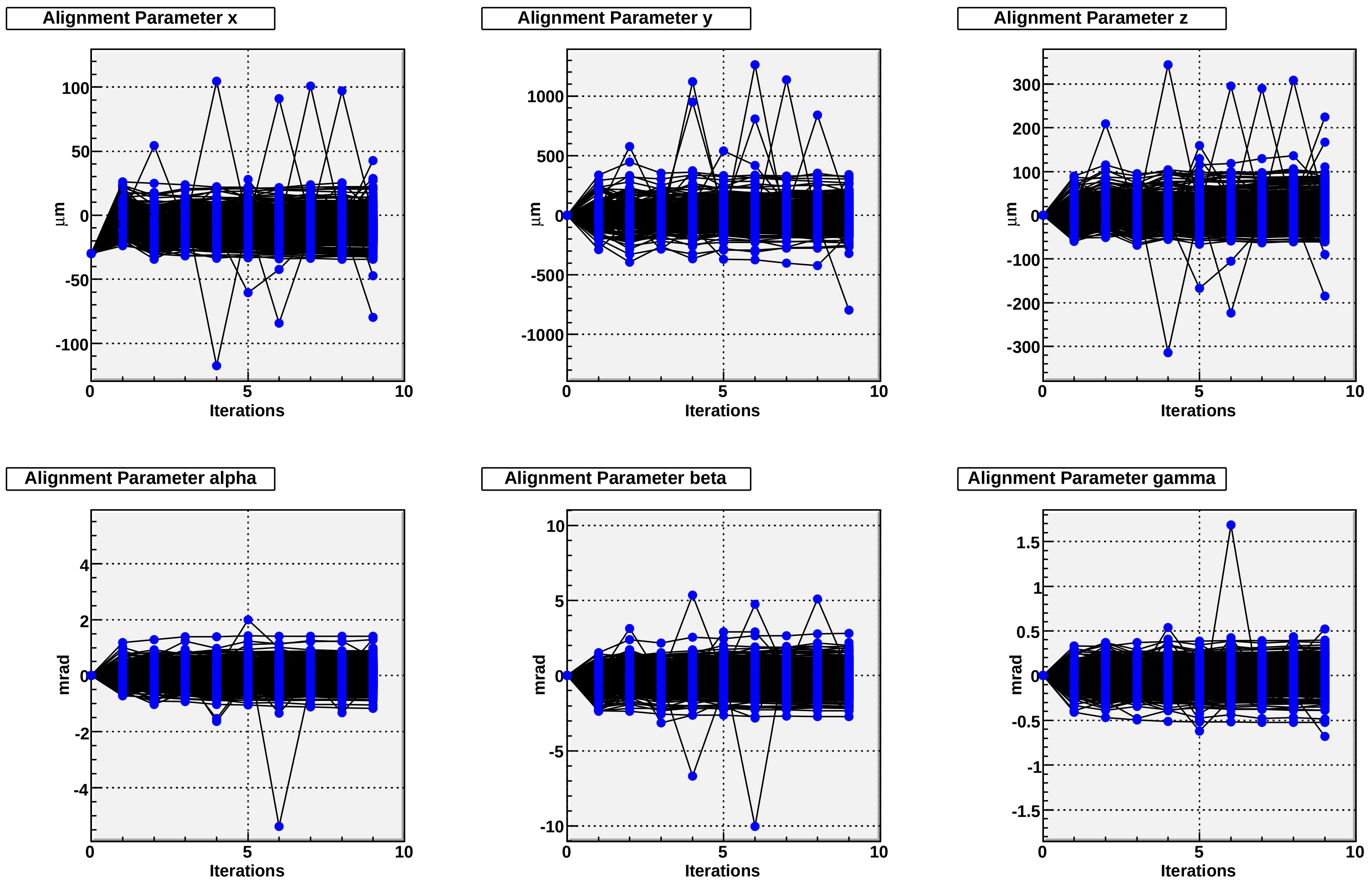

Figure 5.12: *Alignment parameter flow for the barrel layer 1 modules over 9 iterations. The applied misalignment was Layer1_Set1.*

| Pixel modules chosen for single module misalignment | | | |
|---|---|---|---|
| superstructure | Layer / Disk | Phi | Eta |
| Barrel | 0 | 15 | 3 |
| | 0 | 7 | -4 |
| | 1 | 5 | -5 |
| | 1 | 20 | 2 |
| | 2 | 40 | 0 |
| | 2 | 18 | -1 |
| Endcap A | 0 | 11 | 0 |
| | 0 | 50 | 0 |
| | 1 | 23 | 0 |
| | 1 | 47 | 0 |
| | 2 | 2 | 0 |
| | 2 | 30 | 0 |
| Endcap C | 0 | 0 | 0 |
| | 0 | 17 | 0 |
| | 1 | 12 | 0 |
| | 1 | 40 | 0 |
| | 2 | 8 | 0 |
| | 2 | 33 | 0 |

Table 5.6: *The 18 chosen modules for misalignment sets of individual modules. The identifier numbers are given according to the ATLAS readout identifier scheme [50].*

For two of the three translational degrees of freedom, the alignment accuracy given by the initial as-built-precision in the Pixel barrel is reached. In the endcaps, we can improve the most sensitive coordinate with respect to the initial survey. At present, only this degree of freedom meets the alignment accuracy required by the TDR. As discussed in previous sections, this is not an intrinsic problem of `Chi2AlignAlg`, but is due to degraded input. As seen in table 5.3, the statistical error almost completely determines the width of the distributions, systematic errors are too small to be seen. Thus, the track sample needs to be increased to improve the statistical error of the method. A reasonable sample size to constrain both SCT and Pixel modules well is estimated to be at least 10 times larger than the present one. Additionally, a firm track selection and the use of different track types, like cosmic ray tracks, should give additional information and accuracy for the alignment. The error of pixel hits is not yet finally mastered due to the problems with different cluster shapes and sizes. After improvements in this region are made, it is expected that especially the y-coordinate in the barrel part of the Pixel detector is constrained better than at present.

There are also features integrated into `Chi2AlignAlg` which are not part of this thesis work:

A `BookKeepTool` allows the enrichment of overlap hits to efficiently use the surplus of information they give. Overlap hits happen when a particle track crosses two adjacent overlapping modules on the same barrel layer or on the same endcap disk. Overlap hits also yield a constraint on the relative position of these modules, whereas non-overlap hits only connect modules lying behind each other on different layers. Additionally, due to the closeness of the two hits, the error of extrapolation is small which improves the information. This feature is already used for the testbeam data and will be tested on a larger scale for full ATLAS.

Additionally, the Pixel module survey measures relative positions for modules on a stave or a disk, whereas track-based alignment mostly aligns modules on subsequent layers and disks. Thus, track-based alignment and the survey give essentially complementary information. `Chi2AlignAlg` has an interface to the `SurveyConstraintTool`, which allows to merge information from the module survey and from track-based alignment. First tests have already been performed to see how this tool can help track-based alignment approaches [30].

Further studies are being performed to investigate the alignment of other detector geometries. In 2004, the so-called CombinedTestBeam (CTB) run was the first occasion where real data was taken from all ATLAS subdetectors at once [51]. The setup was chosen to mimic one slice of the ATLAS barrel, with six Pixel modules and eight SCT modules present as the silicon part of the Inner Detector. All detectors were built up to scale at the H8 beamline of the CERN SPS synchrotron and irradiated with electrons, positrons, muons and pions in the energy range of 2 to 350 GeV.

The combined TRT and SCT cosmics run is performed at the moment at the CERN SR1 surface test area. There, the barrel parts of the TRT and SCT detectors have been mounted to each other and are now tested during the measurement of cosmic ray particles.

In both cases, Athena is used to reconstruct the events and the alignment performance of the `Chi2AlignAlg` algorithm is studied for these geometries.

# Chapter 6

# Conclusions

Detector alignment plays a crucial role in understanding the detector and measuring physical quantities with ultimate precision. All LHC experiments rely on precisely calibrated and aligned detectors. To fully exploit the physics potential of the ATLAS Inner Detector, the precision of alignment must be significantly better than the intrinsic resolution – that is $\sigma_x \times \sigma_y = 15\,\mu m \times 115\,\mu m$ for the Pixel detector –, and ideally down to $1\,\mu$m for the local x coordinate of the Pixel modules. After exploring all other means, only track based alignment can give the last required precision.

In this thesis, the existing local $\chi^2$-approach for the ATLAS SCT detector was modified to also handle Pixel modules. The algorithm is linear and local with respect to each module. Correlations come in via iterating several times. A derivation of this approach was presented and the modifications it takes to also incorporate pixel residuals have been explained.

A ROOT prototype program was developed and the feasibility of the alignment approach has been proven for this setup. Valuable lessons concerning the correct choice of residuals and the implementation of the DOCA-style residuals for a pixel geometry have been learnt in this test environment. There are ongoing studies to further investigate the performance of the approach. The calculation of residual derivatives by analytical means and the possibility of updating the alignment constants during an alignment run with a Kalman filter technique will first be studied within the ROOT test program, dealing with the performance issue of the tilt angle being part of this effort.

The Athena implementation of our alignment approach, namely the `Chi2AlignAlg` algorithm, was upgraded with code to handle Pixel modules and it could be shown that the algorithm is capable of aligning all 5832 silicon modules of the Inner Detector. Thus, it is at present the only algorithm capable of aligning the complete Pixel and SCT detectors. The alignment precision of the Pixel detector could be derived for a track sample of about 236 000 muon tracks. The obtained values approach the requirements given in the TDR, but do not reach them fully yet. The feasibility of recovering misalignments has been shown.

The work on this approach is very active. There are studies underway which will investigate how a selection of appropriate tracks can influence the performance. A Kalman-filter technique is explored to investigate whether comparable alignment results can be obtained within fewer iterations. Vertex constraints should help aligning the innermost Pixel layer as well as correlating modules in different $\phi$-regions of the Inner Detector. The `Chi2AlignAlg` algorithm will be extended by the possibility to align global structures of the detector, like disks and layers, which represent the physical objects of which the detectors are composed. Thus, global movements of disks and layers are separated from individual alignment corrections of the modules. Furtheron, the algorithm will directly profit from any improvements made in the ATLAS reconstruction chain, for example via better track fitters or improved pixel cluster handling.

Because of its full integration into Athena, the `Chi2AlignAlg` algorithm is flexible enough to not only align the ATLAS setup, but for example also the reduced CTB setup. At this moment, alignment studies for the testbeam are being finished and the performance of our approach is tested in this environment. The next big step is the alignment based on the SCT/TRT barrel combined cosmics run before those two detectors get integrated into ATLAS down in the cavern. This is the first opportunity to align parts of the SCT barrel detector with real data. All this will help being ready to provide a working alignment of the whole silicon inner detector for the day ATLAS starts.

# Appendix A

# Additional plots

## A.1 Additional plots from the small ROOT simulation

### A.1.1 Plots with in-plane residuals

### A.1.2 Plots with perpendicular illumination

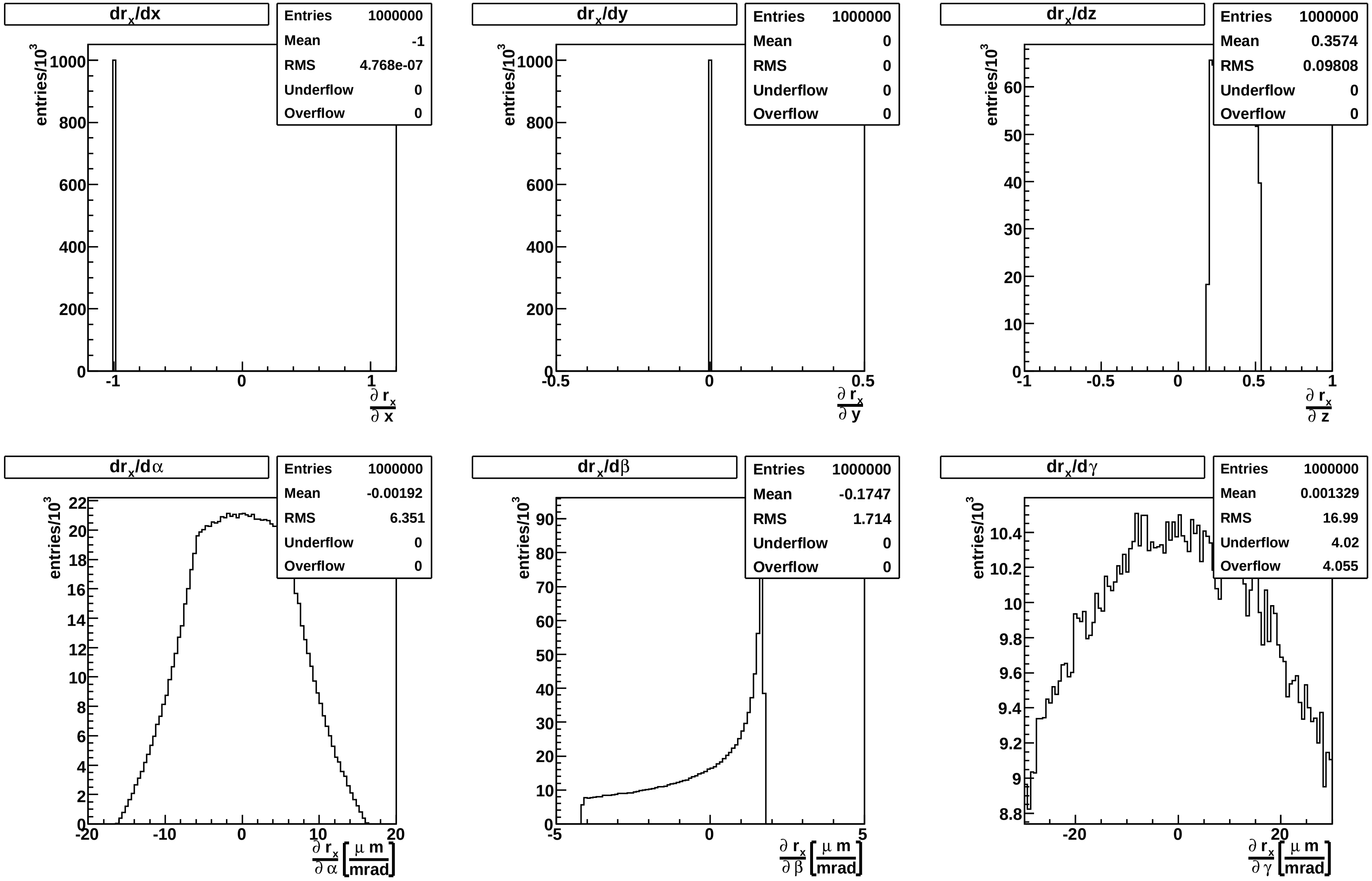

Figure A.1: *Distribution of the derivatives of in-plane x-residuals with respect to the six alignment parameters*

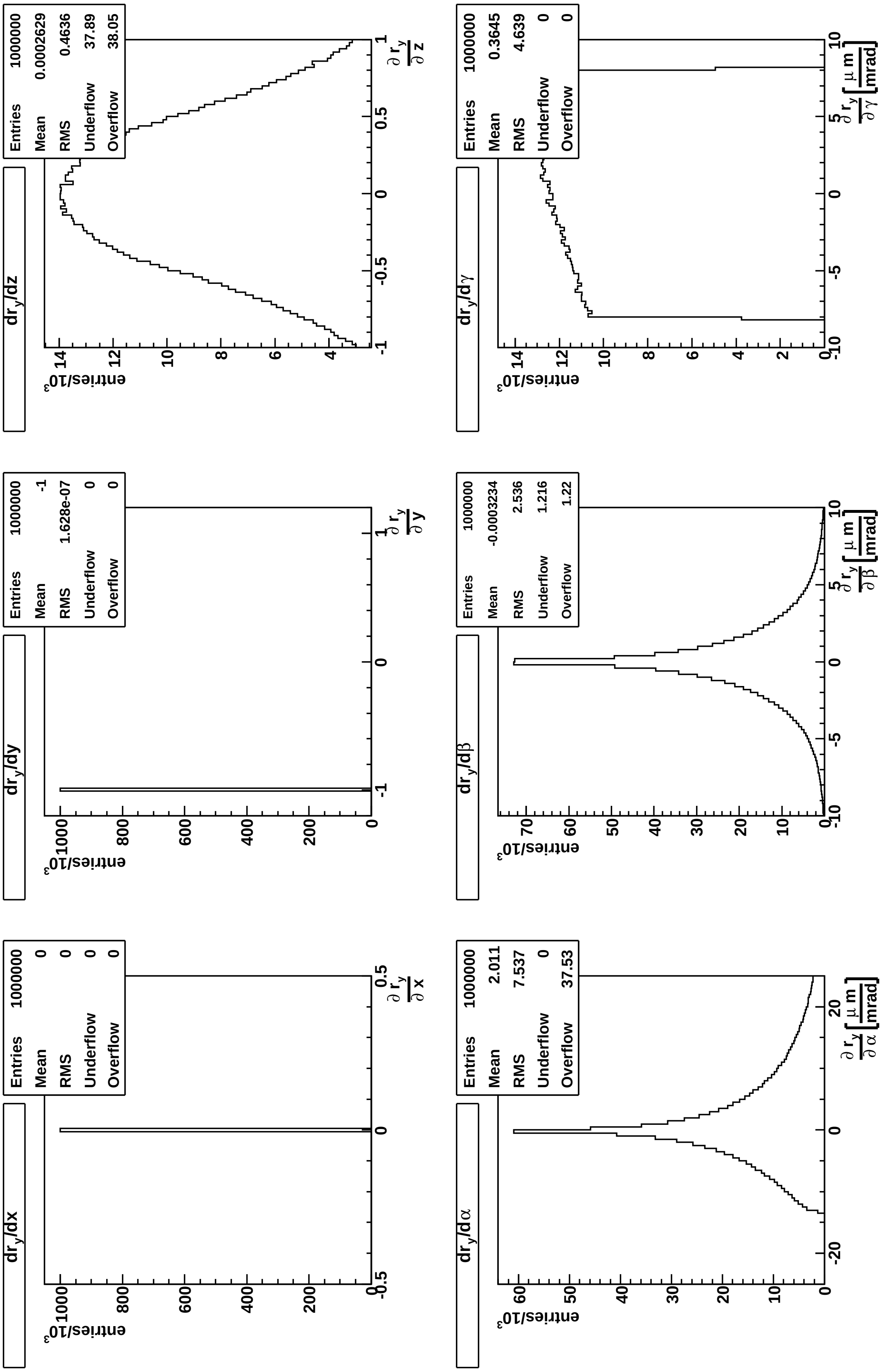

Figure A.2: *Distribution of the derivatives of in-plane y-residuals with respect to the six alignment parameters*

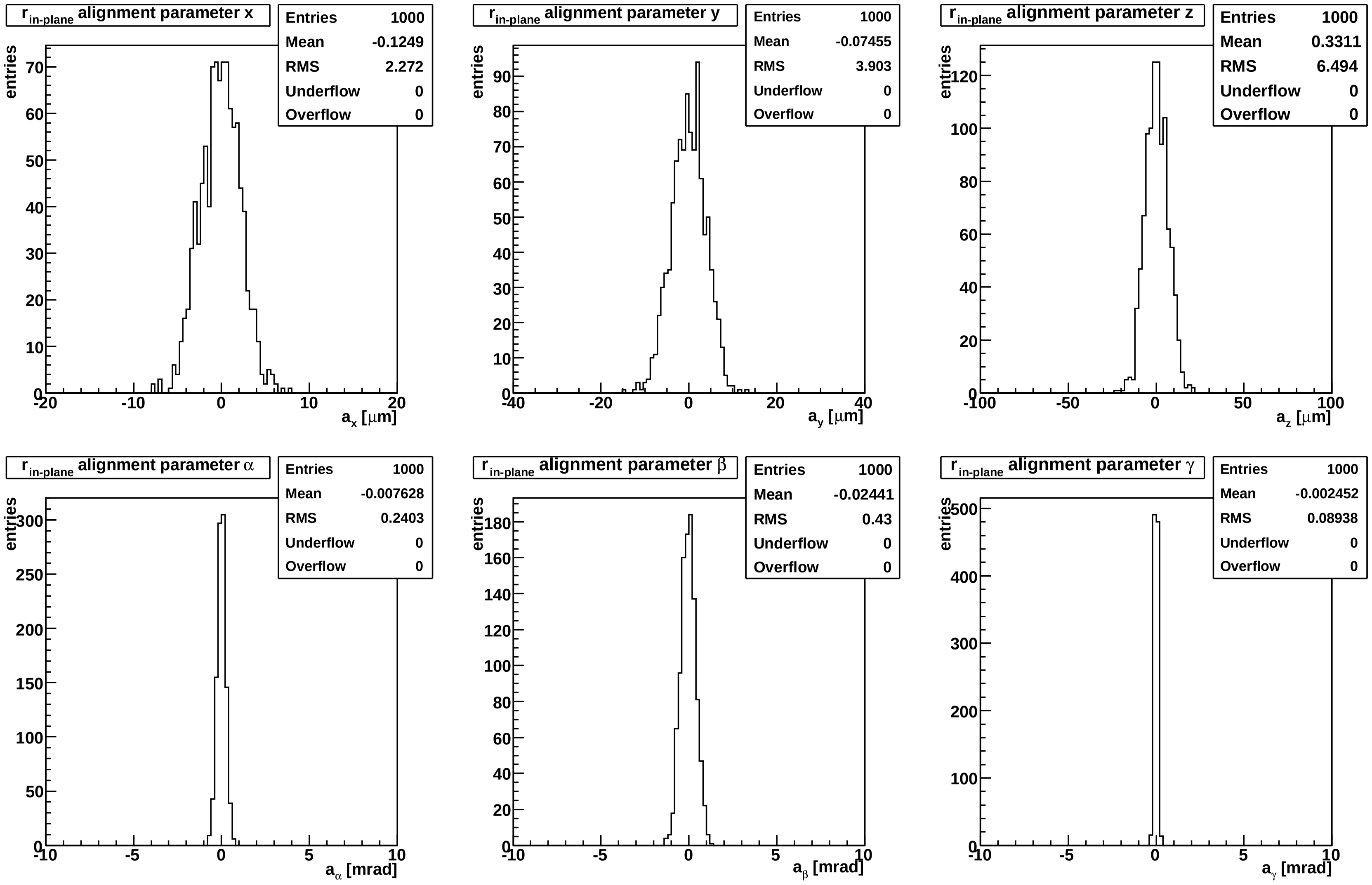

Figure A.3: *Distribution of the final alignment parameters calculated using in-plane residuals. 1000 runs were produced using nominal alignment and 1000 tracks each.*

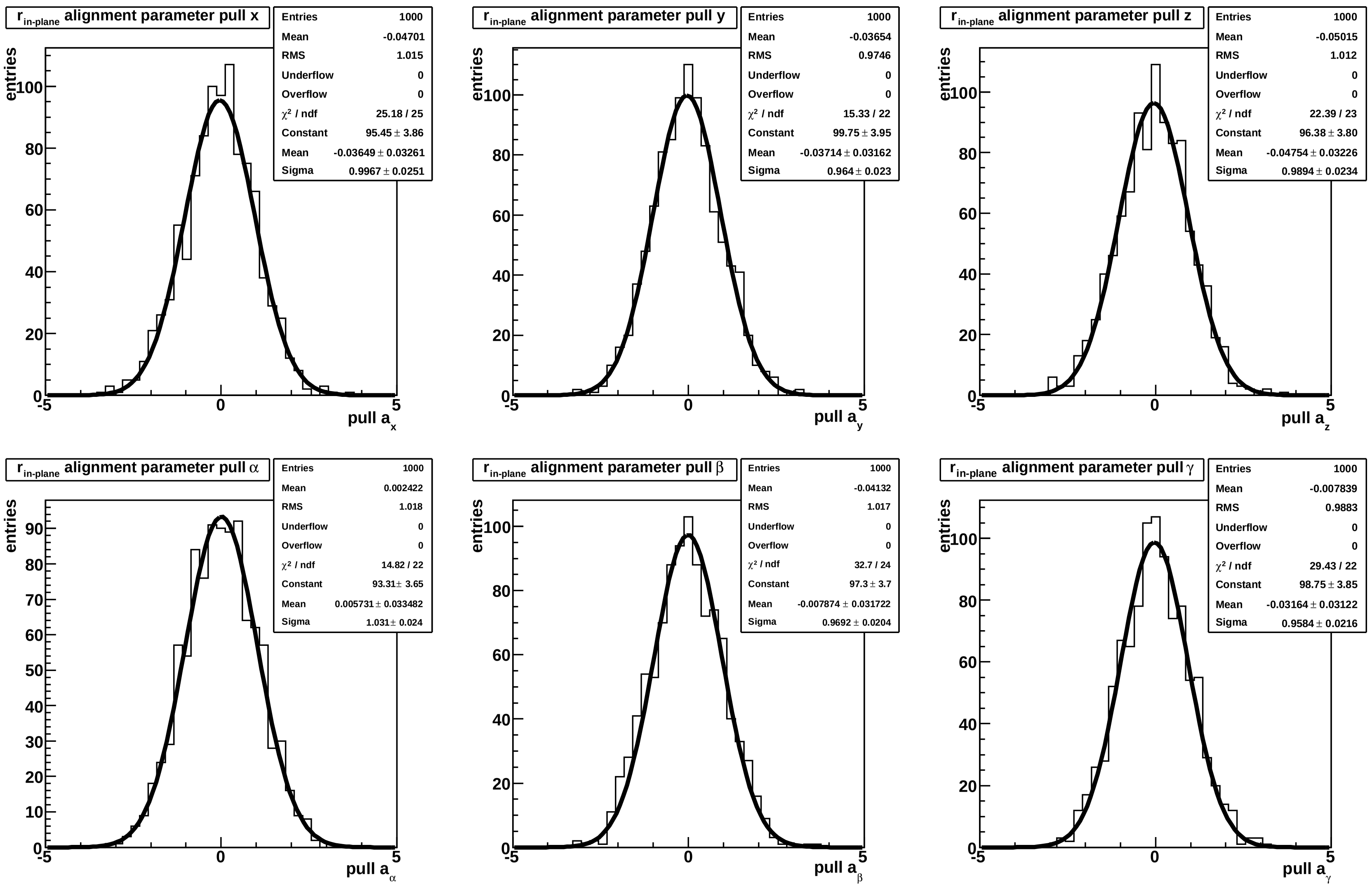

Figure A.4: *Pull distribution of the alignment parameters calculated using in-plane residuals. 1000 runs were produced using nominal alignment and 1000 tracks each.*

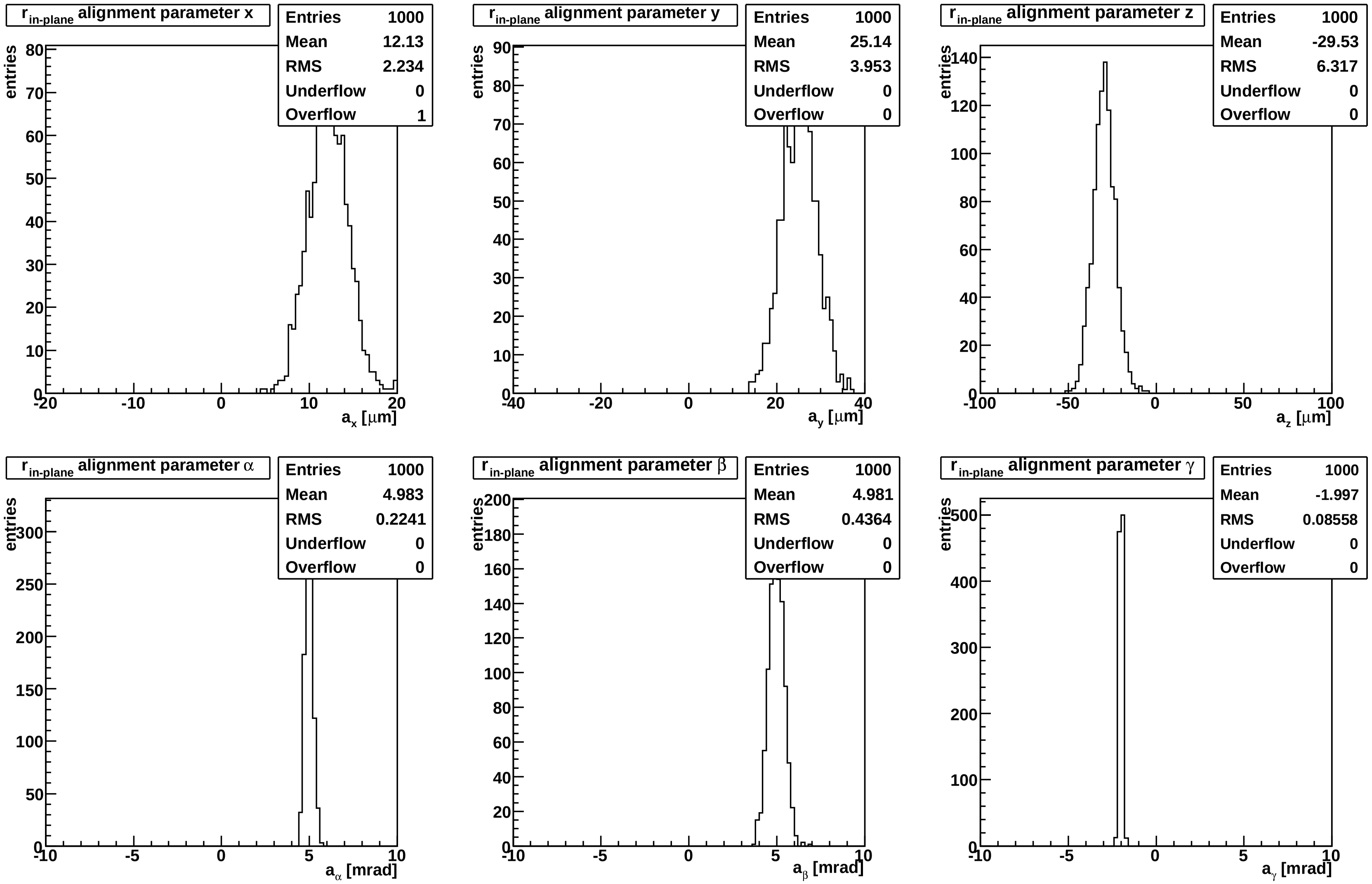

Figure A.5: *Distribution of the final alignment parameters calculated using in-plane residuals. 1000 runs were produced with 1000 tracks each. The applied misalignment was* $\Delta x$*=12* $\mu m$*,* $\Delta y$*=25* $\mu m$*,* $\Delta z$*= -30* $\mu m$*,* $\Delta\alpha$*=5 mrad,* $\Delta\beta$*=5 mrad,* $\Delta\gamma$*=-2 mrad.*

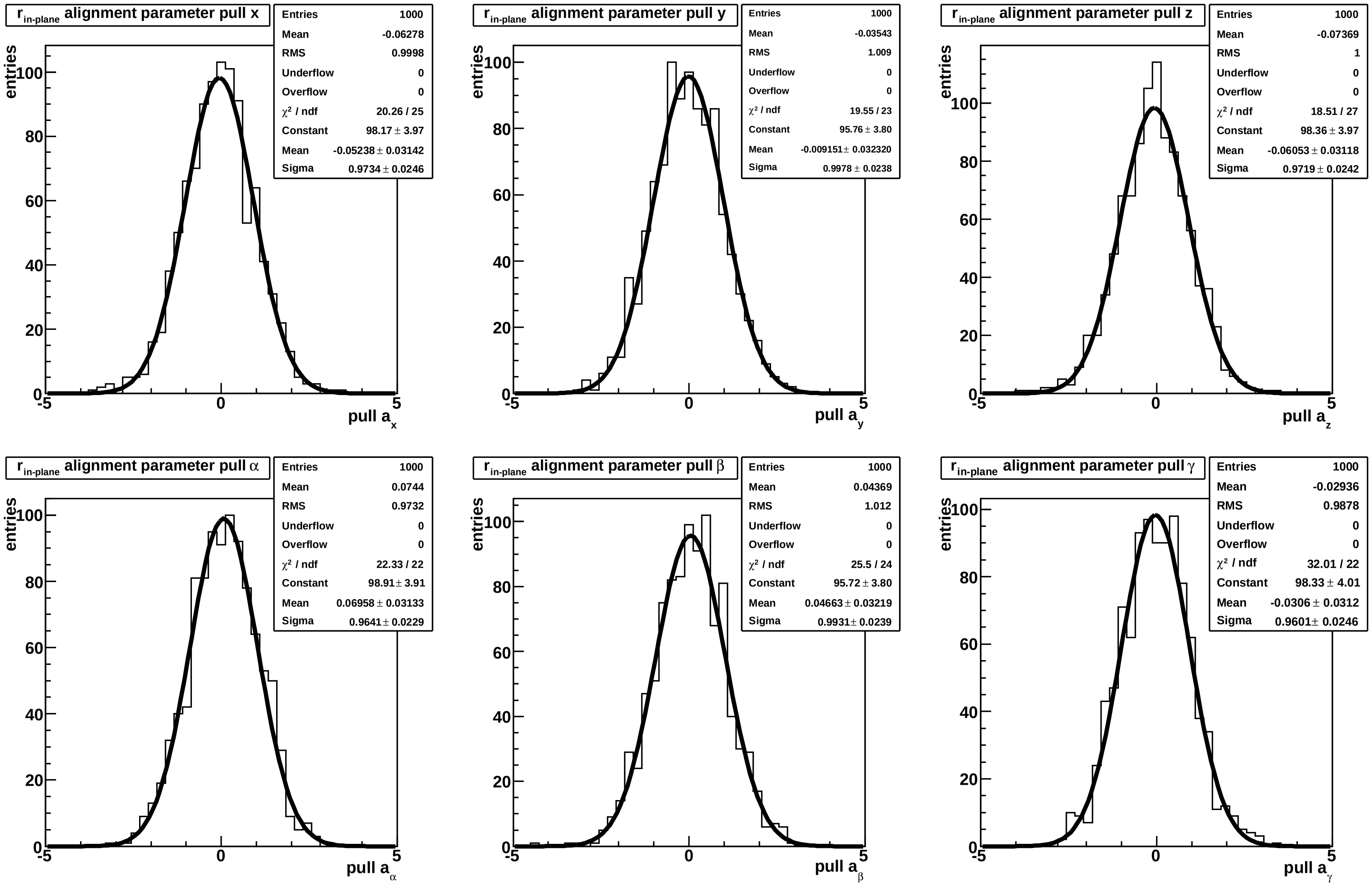

Figure A.6: *Pull distribution of the alignment parameters calculated using in-plane residuals. 1000 runs were produced with 1000 tracks each. The applied misalignment was the same as in fig. A.5.*

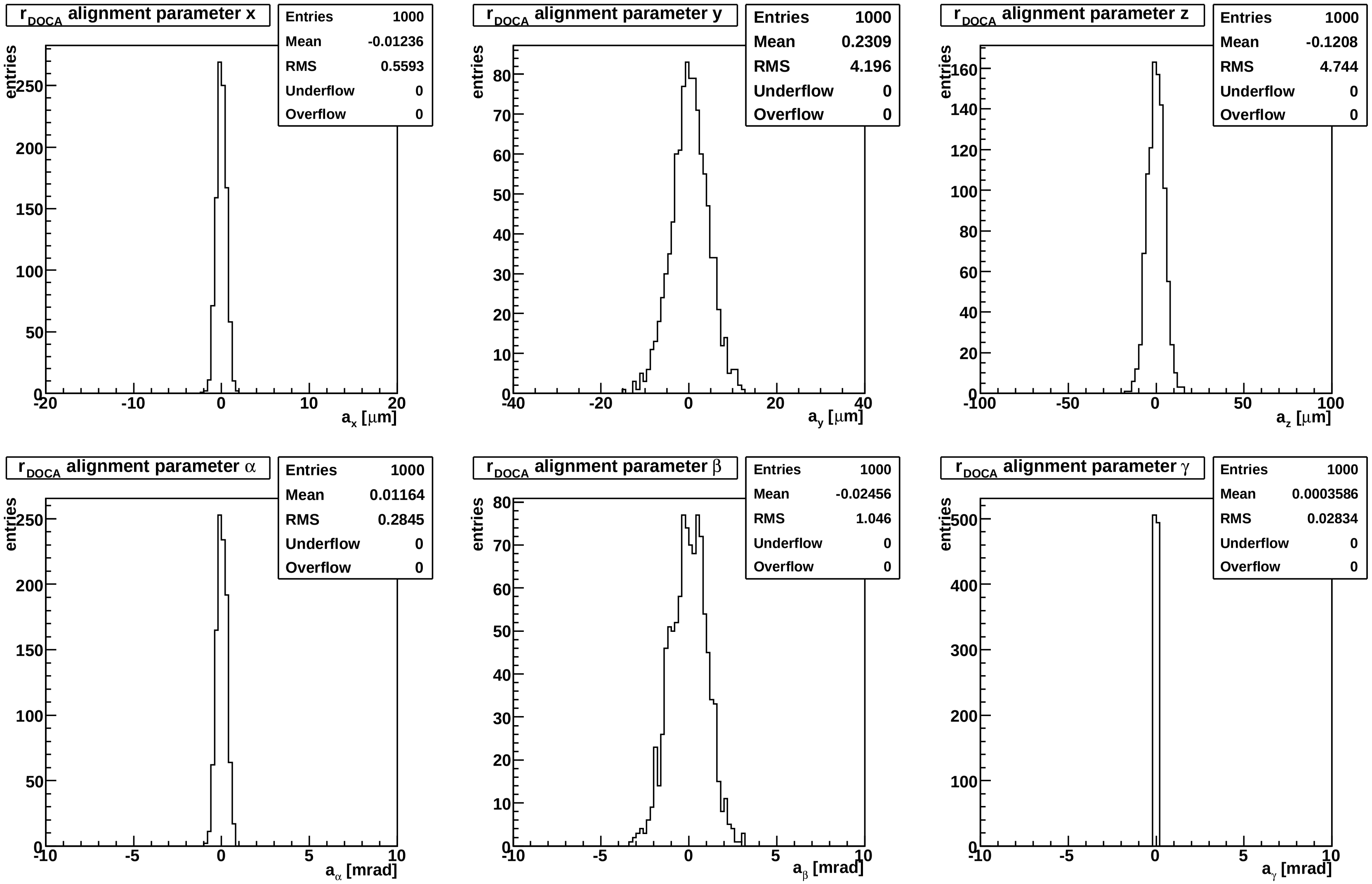

Figure A.7: *Distribution of the final alignment parameters calculated using DOCA residuals. The module was placed at (0, 0, 50.5 mm). 1000 runs were produced with nominal alignment and 1000 tracks each.*

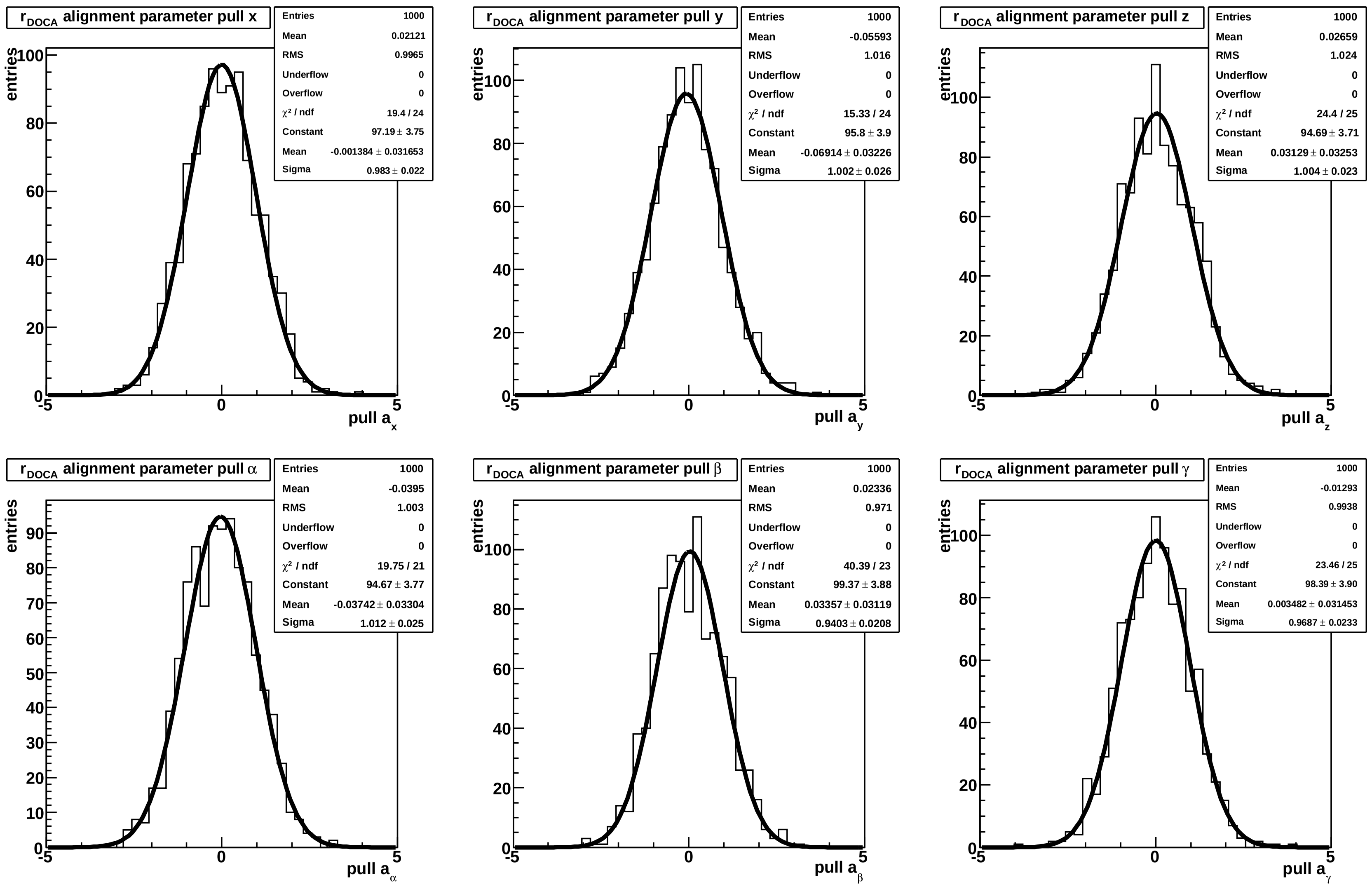

Figure A.8: *Pull distribution of the alignment parameters calculated using DOCA residuals. The module was placed at (0, 0, 50.5 mm). 1000 runs were produced using nominal alingment and 1000 tracks each.*

# List of Figures

# List of Tables

# Bibliography

[1] D. Perkins, *Introduction to High Energy Physics*, Cambridge University Press, 4th edition, 2000.

[2] G. Altarelli, *The Standard Model of Particle Physics*, Encyclopedia of Mathematical Physics, Elsevier (2005), CERN-PH-TH-2005-206, hep-ph/0510281.

[3] P. Higgs, *Broken symmetries, massless particles and gauge fields*, Phys. Lett. **12** (1964) 132.

[4] The LHC Collaboration, *LHC : The Large Hadron Collider Accelerator Project*, CERN-AC-93-03-LHC.

[5] The LHC Collaboration, *LHC Conceptual Design Report*, CERN-AC-95-05-LHC.

[6] The ATLAS Collaboration, *ATLAS Technical Proposal*, CERN-LHCC-94-43.

[7] The CMS Collaboration, *CMS Technical Proposal*, CERN-LHCC-94-38.

[8] The LHCb Collaboration, *LHCb Technical Proposal*, CERN-LHCC-98-004.

[9] The ALICE Collaboration, *ALICE Technical Proposal*, CERN-LHCC-95-71.

[10] *ATLAS detector public page*, `http://atlas.ch/`.

[11] M. Hauschild, *Particle Detectors*, (2006), Lecture given at the 2nd IMPRS block course at MPI.

[12] The ATLAS Collaboration, *ATLAS Detector and physics performance: Technical design report*, CERN-LHCC-99-14.

[13] The ATLAS Collaboration, *ATLAS Muon Spectrometer: Technical design report*, CERN-LHCC-97-22.

[14] The ATLAS Collaboration, *ATLAS Magnet System: Technical design report*, CERN-LHCC-97-18.

[15] The ATLAS Collaboration, *ATLAS Barrel Toroid: Technical design report*, CERN-LHCC-97-19.

[16] The ATLAS Collaboration, *ATLAS Endcap Toroid: Technical design report*, CERN-LHCC-97-20.

[17] The ATLAS Collaboration, *ATLAS Calorimeter Performance: Technical design report*, CERN-LHCC 96-40.

[18] The ATLAS Collaboration, *ATLAS liquid argon calorimeter: Technical design report*, CERN-LHCC-96-41.

[19] The ATLAS Collaboration, *ATLAS Tile Calorimeter: Technical design report*, CERN-LHCC 96-42.

[20] The ATLAS Collaboration, *ATLAS Inner Detector: Technical design report Vol. 1*, CERN-LHCC-97-16.

[21] The ATLAS Collaboration, *ATLAS Inner Detector: Technical design report Vol. 2*, CERN-LHCC-97-17.

[22] The ATLAS Collaboration, *ATLAS Central Solenoid: Technical design report*, CERN-LHCC-97-21.

[23] The ATLAS Collaboration, *ATLAS Pixel Detector: Technical design report*, CERN-LHCC-98-13.

[24] S. Haywood, *Module Overlaps in the SCT and Pixels*, ATLAS Note, ATL-INDET-2002-016 (2002).

[25] G. Lutz, *Semiconductor Radiation Detectors*, Springer, 1999.

[26] L. Andricek, *Silicon Detectors for High Energy and Astrophysics Experiments*, `http://cds.mppmu.mpg.de/cdsagenda/askArchive.php?base=agenda&categ=a0439&id=a0439s2t3/document`, Presentation given at MPI Young Scientist Workshop 2004, Ringberg.

[27] I. Reisinger, *Spatial and vertex resolution studies on the ATLAS Pixel Detector based on Combined Testbeam 2004 data*, diploma thesis (2006), University of Dortmund.

[28] D. Dobos, *Production accompanying testing of the ATLAS Pixel module*, `http://e4.physik.uni-dortmund.de/pub/EIV/Vortr%E4ge/Dobos2004-4.pdf`, talk at the University of Dortmund Lehrstuhlseminar.

[29] M. García-Sciveres, *The ATLAS Pixel detector*, ATLAS Note, ATL-COM-INDET-2004-004 (2004).

[30] T. Golling, *Alignment of the Silicon Tracking Detector using Survey Constraints*, ATLAS Note, ATL-INDET-PUB-2006-001 (2006).

[31] A. Salzburger, *Atlas coordinate frames*, `http://twiki.cern.ch/twiki/bin/view/Atlas/CoordinateFrames`, ATLAS Computing Wiki.

[32] V. Re et al., *Performance of the BABAR silicon vertex tracker*, Nuclear Instruments and Methods in Physics Research **A501** (2003) 14.

[33] R. Härtel, *Iterative local $\chi^2$ alignment approach for the ATLAS SCT detector*, diploma thesis (2005), MPI Munich.

[34] P. Brückman, A. Hicheur, and S. Haywood, *Global $\chi^2$ approach to the Alignment of the ATLAS Silicon Tracking Detectors*, ATLAS internal Note, ATL-COM-INDET-2005-004 (2005).

[35] G. Cowan, *Statistical Data Analysis*, Oxford Science Publications, 1998.

[36] *ROOT system homepage*, `http://root.cern.ch/`.

[37] R. Brun et al., *ROOT Users Guide 5.08*, 2005.

[38] I. Bronstein et al., *Taschenbuch der Mathematik*, Verlag Harri Deutsch, 5th edition, 2001.

[39] S. Lloyd, *The ATLAS Workbook*, `https://twiki.cern.ch/twiki/bin/view/Atlas/WorkBook`.

[40] *Gaudi Project*, `http://proj-gaudi.web.cern.ch/proj-gaudi/welcome.html`, Gaudi framework project website.

[41] *The Python Programming Language*, `http://www.python.org`, Official website.

[42] The ATLAS Collaboration, *Athena - The ATLAS Common Framework - Developer Guide*, 2004, draft v8.0.0.

[43] *ATLAS CVS repository*, `http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/`, ATLAS offline packages in the CVS repository.

[44] M. Regler, R. Frühwirth, and W. Mitaroff, *Filter Methods in Track and Vertex Reconstruction*, International Journal of Modern Physics C **7** (1996) 521.

[45] W. Liebig, *TrkKalmanFitter*, `http://twiki.cern.ch/twiki/bin/view/Atlas/TrkKalmanFitter`, ATLAS Computing Wiki.

[46] S. González-Sevilla, *Multimuons in the GRID*, `http://agenda.cern.ch/askArchive.php?base=agenda&categ=a0624&id=a0624s7t5/transparencies`, Presentation held at ID alignment meeting, CERN.

[47] S. González-Sevilla, *Multimuon samples*, `http://ific.uv.es/~segonzal/multimuons/multimu.html`.

[48] T. LeCompte, *ATLAS Event Data Model*, `http://twiki.cern.ch/twiki/bin/view/Atlas/EventDataModel`, ATLAS Computing Wiki.

[49] J. Schieck and D. Froidevaux, *Suggestion for CSC misalignment numbers*, `https://hypernews.cern.ch/HyperNews/Atlas/get/IDAlignment/15.html`, Hypernews article from the Inner Detector Alignment list.

[50] S. Goldfarb and A. Schaffer, *Definition of Off-line Readout Identifiers for the ATLAS Detector*, ATL-SOFT-2001-004 (2001).

[51] B. Di Girolamo et al., *Beamline instrumentation in the 2004 combined ATLAS testbeam*, ATLAS Note, ATL-TECH-PUB-2005-001 (2005).

# Acknowledgments

The last lines of this thesis are dedicated to the people who helped and supported me during this time.

First I want to thank Prof. Umbach and Prof. Bethke for their collaboration and for giving me the opportunity to perform the research for this thesis at the Werner-Heisenberg-Institut although I was a student at Würzburg university. They put a lot of effort into making this collaboration possible and squeezed me into their timetables, for which I'm grateful. I also owe a warm thank you to Prof. Fraas for making bureaucratic things less complicated.

A huge thanks goes to my three group supervisors, Richard Nisius, Stefan Kluth and Jochen Schieck. From the beginning on I never got lost. I was integrated into the group from day one and never needed to hesitate to ask any question. Thank you for spending so much effort in giving hints and tips, stimulating my work and finally proofreading this thesis attentively.

A large contribution to making my time at the MPI worthwhile was due to my officemates and my officemates from next door. Thank you for fruitful discussions and a personal atmosphere that never made me regret my choice. Here I also have to mention the whole family of high-energy physics at the MPI, at CERN and at other institutes, with lots of open-hearted people willing to help and to give support.

Finally my gratitude is given to my whole family and all the people who support me in my private life. I hope they all know what they mean to me.

# Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen Hilfsmittel als die angegebenen benutzt habe.

München, den 26.5.2006

Tobias Göttfert