# Iterative local $\chi^2$ alignment approach for the ATLAS SCT detector

Diploma Thesis
supervised by Prof. Dr. S. Bethke

Roland Härtel
Physik-Department
Technische Universität München
and
Max-Planck-Institut für Physik
Werner Heisenberg Institut

Munich, November 28, 2005

## Abstract

An approach for the alignment of SCT modules in the ATLAS Inner Detector with particle tracks was developed and implemented in the ATLAS software framework Athena. The approach uses distance of closest approach residuals and a linear least squares minimization to derive the most probable set of alignment parameters for each module. The procedure is iterative, i.e. with the first set of alignment constants a track refit is done and the alignment algorithm is repeated. Correlations between modules are only implicitly taken into account due to the improvement of track parameters through the iterations.

A derivation of the underlying concepts is presented. The achievable accuracy and limits of the alignment approach were studied with Monte Carlo simulated tracks in the Athena framework. The results and limitations obtained with the present versions of Athena and the proposed alignment software are presented.

# Contents

# Acknowledgements

My most heartfelt gratitude goes out to my parents. Their guidance and encouragement, as well as their unceasing support, emotionally and financially, made it possible for me to study physics and to write this thesis.

I want to thank Prof. S. Bethke for the opportunity he gave me to work at the Werner-Heisenberg-Institut and to become a member of the ATLAS collaboration.

The warmest thank you goes to my three thesis advisors Stefan Kluth, Richard Nisius and Jochen Schieck. I wouldn't know where to begin with the praise, with all they did for me. From introducing me to the field of high energy physics and always lending me a supportive hand during my first tentative steps there to finally proofreading my thesis and having the patience to help me knock it into shape.

Special mention is due of course for Sven Menke, the "Athena-pompier" that was always there to put out any fire and help me wrestle down the unruly framework software. With Sven at my back it is save to say: Athena makes me happy.

This acknowledgements would be incomplete without mentioning the people at CERN that made this year in the ATLAS collaboration so special for me: Markus Elsing and his "Tracking-Crew" Wolfgang Liebig, Ed Moyse, Andi Salzburger and Andi Wildauer. The warm welcome I always received from them and their dedication to give this few percent more that make all the difference, were a constant source of motivation and inspiration.

I want to thank my two flatmates Max and Schuppi for being such good friends and a lifeline to reality. They helped me to keep both feet firmly planted on the ground.

Finally, all my love goes out Tine, the lady of my heart. I hope I can make it up to her for all the long hours I spent in the arms of high energy physics instead of hers.

# Overview

The main emphasis of this thesis was the study of detector alignment with particle tracks. The work is described in five chapters that are structured as follows:

- **Chapter 1 – Introduction**
  The LHC and the multipurpose detector ATLAS are described briefly and the physics motivation and goals that are the rationale to conduct such an enormous experiment are outlined. The SCT subdetector of ATLAS receives special focus and the concept of alignment is introduced.
- **Chapter 2 – Local $\chi^2$ Alignment Algorithm**
  The general ideas behind track based detector alignment are presented and the local $\chi^2$ alignment approach we propose for the ATLAS SCT and the mathematical concepts behind it are outlined in detail.
- **Chapter 3 – Prototype Program with ROOT**
  A small scale tracking and geometry-description testbed environment was developed in ROOT. Our local $\chi^2$ alignment approach was implemented as a prototype program within this testbed environment. The prototype program was both, a proof-of-principle application and a starting point for the implementation of our alignment approach into the ATLAS software framework Athena. Details about the implementation of the prototype program as well as tests and their results are presented.
- **Chapter 4 – Iterative Local $\chi^2$ Alignment in Athena**
  By far the biggest part of the thesis is presented in this chapter: the implementation of our local $\chi^2$ alignment algorithm into the ATLAS software framework Athena. Details of the implementation and the input parameters for our local $\chi^2$ alignment algorithm – residuals, residual errors and residual derivatives – are described. Tests with nominal ATLAS SCT setup and various misaligned setups are outlined. The results of these tests are presented and discussed.
- **Chapter 5 – Conclusions**
  The main points of the preceding chapters are summarized. Ongoing developments and unresolved issues are pointed out and prospective future developments of our alignment approach discussed briefly.

# Chapter 1

# Introduction

## 1.1 Standard Model

The Standard Model of particle physics is a successful theory to describe in a coherent way the properties and interactions of fundamental particles up to energies of $\mathcal{O}(200\ \text{GeV})$ [1].

The Standard Model is a quantum field theory that describes the interactions of spin-$\frac{1}{2}$ fermions. The interactions are mediated by fields or more exactly by field quanta, the spin-1 gauge bosons. The bosons arise from the requirement of local gauge invariance of the fermion fields and are a manifestation of the symmetry group of the Standard Model, that is $SU(3)\times SU(2)\times U(1)$ [2–5].

The fundamental fermions that are described by the Standard Model are leptons and quarks [6] and they are grouped in three generations. The interactions between the fermions belong to two sectors. Firstly the strong interaction described by Quantum Chromo Dynamics (QCD) with the symmetry group SU(3) and secondly the electroweak interaction with the symmetry group $SU(2)\times U(1)$. The electroweak interaction is a unification of Quantum Electro Dynamics (QED) and the weak interaction.

The gauge bosons – the mediators of the interaction fields – are per se massless but the electroweak symmetry group $SU(2)\times U(1)$ is spontaneously broken. This leads to the emergence of massive gauge bosons, namely the $W^{\pm}$ and the $Z^0$. A theory to describe the electroweak symmetry breaking is the Higgs mechanism [7] which postulates the existence of a massive neutral scalar boson, the Higgs boson.

From theoretical arguments (unitarity violation of the cross-section of WW-boson scattering) we expect that in the energy range up to 1 TeV signs of the mechanism of electroweak symmetry-breaking or new particles predicted by theories beyond the Standard Model must exist. This sets the energy scale for the Large Hardon Collider (LHC) that is currently built at CERN, the European Laboratory for Particle Physics near Geneva. It also sets the challenge for the two multipurpose particle detectors at LHC - the ATLAS and the CMS experiment - to be able to detect a wide range of experimental signatures of the production and decay of the Higgs bosons or new particles.

## 1.2 LHC

In the 27 km long former LEP tunnel a new accelerator and collider is currently installed, the Large Hadron Collider (LHC) [8, 9]. In fall 2005 LHC is beyond the stage of a computer sketch. The first parts – dipole bending magnets and short straight sections – are already installed in the tunnel (see figure 1.1). The LHC is designed to accelerate and collide two proton beams with a proton-proton center of mass energy of 14 TeV. The LHC can be used for heavy ion collisions as well.

Figure 1.1: *Photograph of a section of the LHC already installed in the tunnel.*

The beam particles, the protons, are not elementary particles but composite objects which consist of partons, namely quarks and gluons. Therefore not the total center of mass energy is available for a parton-parton reaction but only the fraction carried by the two interacting partons. In the light of this it should be possible for the LHC to probe energy regions up to 5 TeV [10].

The LHC is built and integrated into the existing infrastructure at CERN as shown in figure 1.2. The injector and pre-accelerator chain consists of Linac, Booster, PS and SPS and will supply the LHC with two high intensity beams of protons or lead ions. In case of proton-proton collisions each beam will be injected into the LHC ring with a proton energy of 450 GeV. After the injection the energy is ramped up to the final value of 7 TeV per proton. Super-conducting dipole bending magnets produce a 8.3 T magnetic field to guide the proton beam through the accelerator ring.

There are two high luminosity interaction points where the two multipurpose detectors ATLAS (A Toroidal Lhc ApparatuS) [11] and CMS (Compact Muon Solenoid) [12] will make full use of the discovery potential of LHC. In addition there are two low luminosity

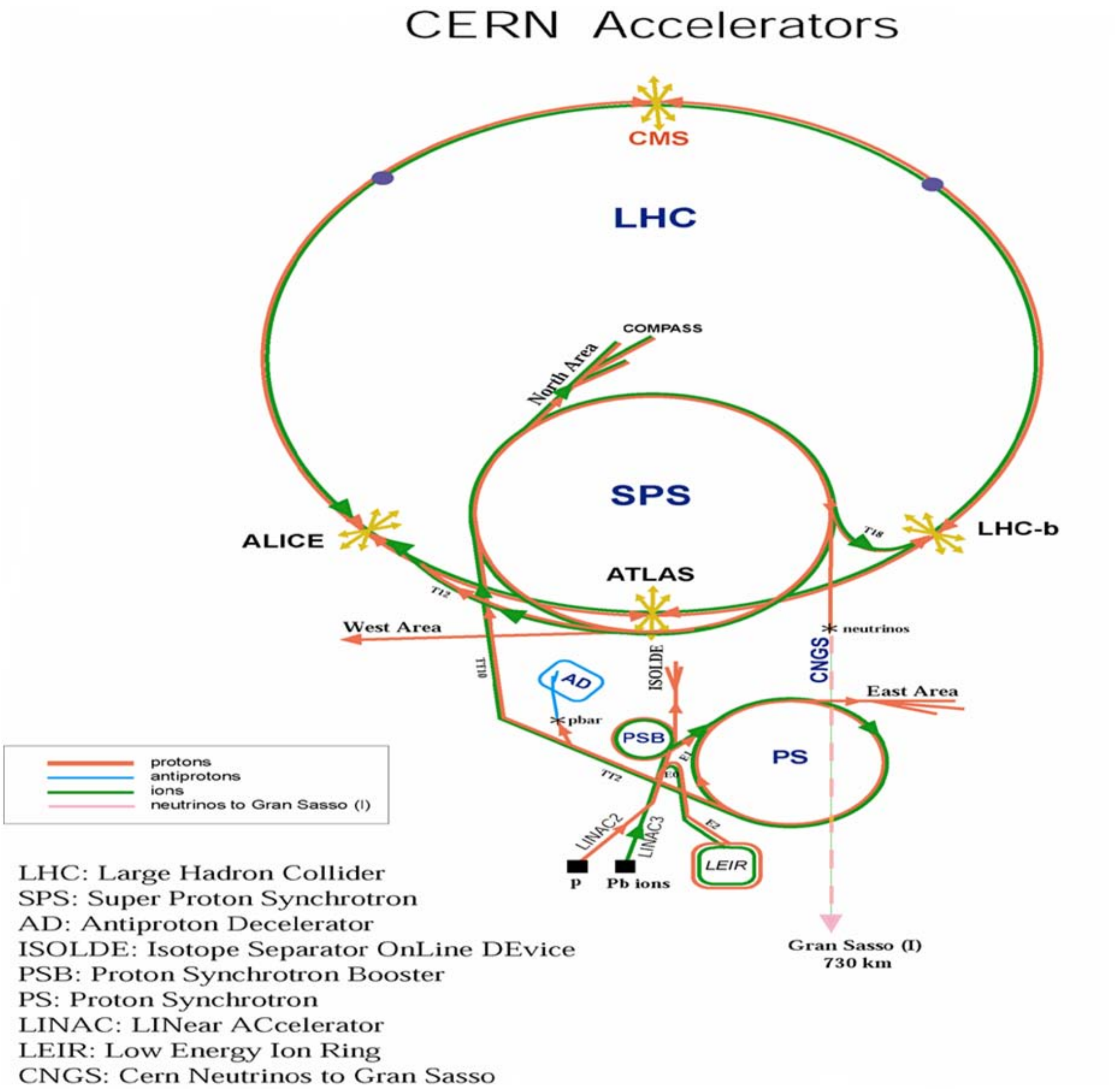

Figure 1.2: *Accelerator infrastructure at CERN. The positions of the LHC experiments Alice, ATLAS, LHCb and CMS are shown as well. The TOTEM experiment is at the same site as CMS and is not listed in the drawing.*

experiments, LHCb dedicated to b-physics [13] and TOTEM specialized to forward scattered proton analysis [14]. Finally, there is an experiment dedicated to heavy ion collisions and the search for the quark-gluon-plasma, the ALICE (A Large Ion Collider Experiment) detector [15].

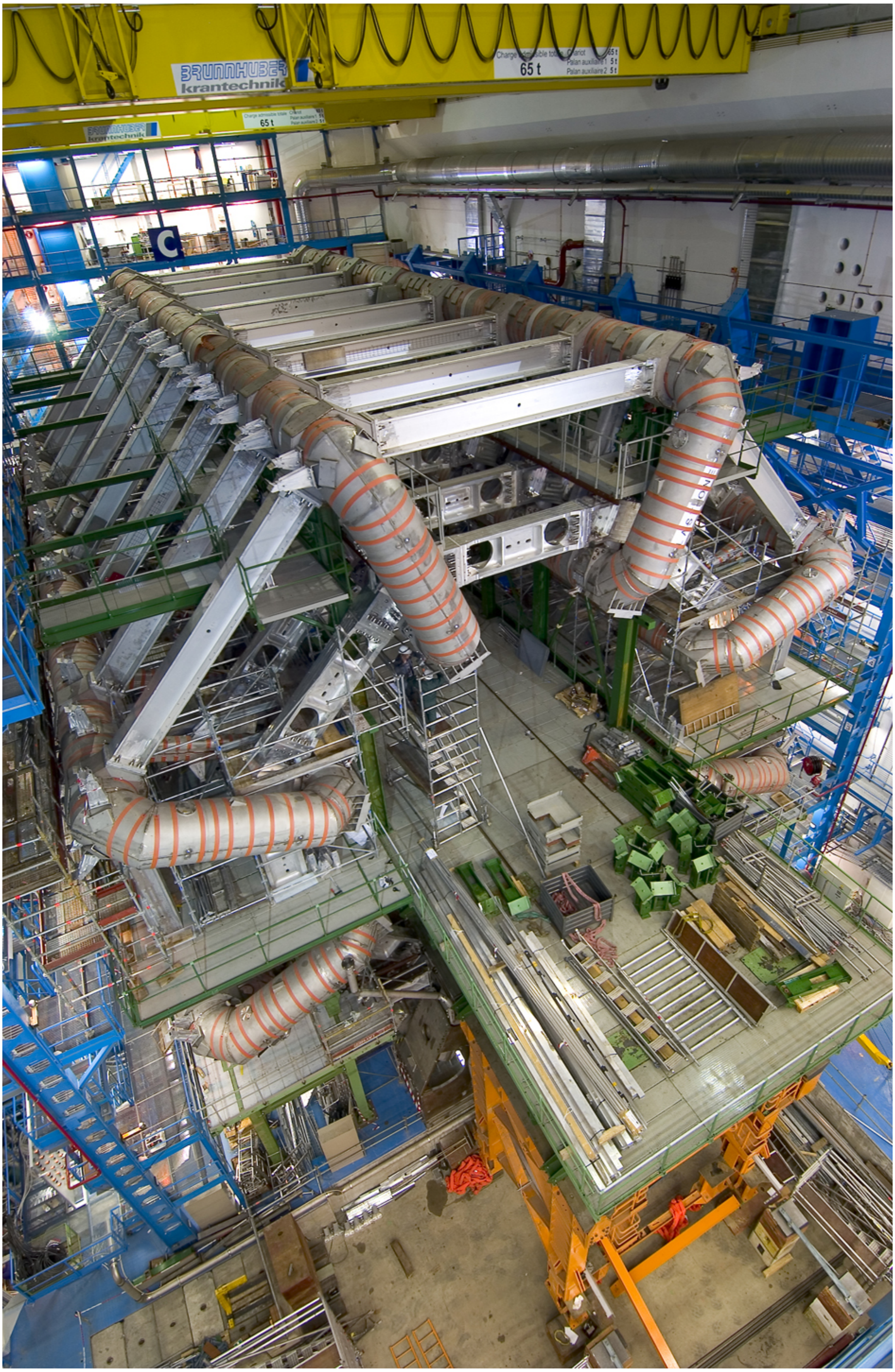

Figure 1.3: *The photograph shows the growing ATLAS detector in its underground construction site. The eight coils of the barrel air-toroid magnet system are fully installed already.*

## 1.3 ATLAS

The ATLAS detector is a multipurpose particle detector designed to make optimal use of the discovery potential of LHC. A selection of some of the most important items on the physics program of ATLAS are the following [16]:

- Discovery of the mechanism of electroweak symmetry breaking, e.g. the discovery of the Higgs boson.
- Searches for experimental signatures predicted by theories beyond the Standard Model, like supersymmetric particles, extra space dimensions or dark matter candidates.
- Precise measurements of properties of Standard Model particles, like CP-violation in B-Meson decay and the masses of the W-boson and the top-quark.

The items on the physics program influenced the guiding principles for the design of ATLAS. For example one expected signature of supersymmetric particles is missing energy, if the lightest supersymmetric particle (LSP) is stable, i.e. if R-parity is conserved [17]. A stable LSP would be a massive neutral weakly interacting particle that cannot directly be detected by ATLAS and thus creates the missing energy signature. To accurately detect missing energy the ATLAS detector must be as hermetic as possible and able to detect particles in the forward regions close to the beamline. Another example for a physics topic that influenced design and construction and will influence the operation of ATLAS is the precision measurement of the W-boson mass. The W mass can be calculated from theory very precisely. In the radiative corrections of the W mass there are terms $\sim \mathrm{m}_{top}^2$ and $\sim \ln(\mathrm{m}_{Higgs})$ [16]. Consequently, during early LHC running it is possible to constrain the Higgs mass (or to rule out the existence of a Standard Model Higgs boson) by accurately measuring the W mass and the top-quark mass. It is estimated to be possible to measure the top-quark mass with a precision of 2 GeV at LHC [16]. For the W mass to not become the dominant uncertainty of the Higgs mass estimation, it must be measured with a precision of 15 MeV or better [16] where the current world-average of the W mass has a precision of 38 MeV [18]. To achieve such a precision the systematic uncertainties must be under control, as statistical errors are not an issue for W mass measurement at the LHC. The most dominant uncertainty is the absolute energy and momentum scale of the decay leptons of the W bosons. Consequently, for precision measurement of the W mass the ATLAS detector must achieve high energy and momentum resolution. This requires that we have a very precise understanding of the detector, i.e. very good calibration and alignment.

The ATLAS detector was designed to fulfill the requirements from the items on the physics program. In figure 1.3 the growing ATLAS detector is shown in its underground construction site. When construction is finished ATLAS will look like the schematic drawing in figure 1.4. The ATLAS detector is a sophisticated device and is described in detail in the Technical Design Reports. Only a brief outline of the different subdetector systems is given here together with the corresponding references. The SemiConductor-Tracker (SCT) is covered more thoroughly in section 1.4.

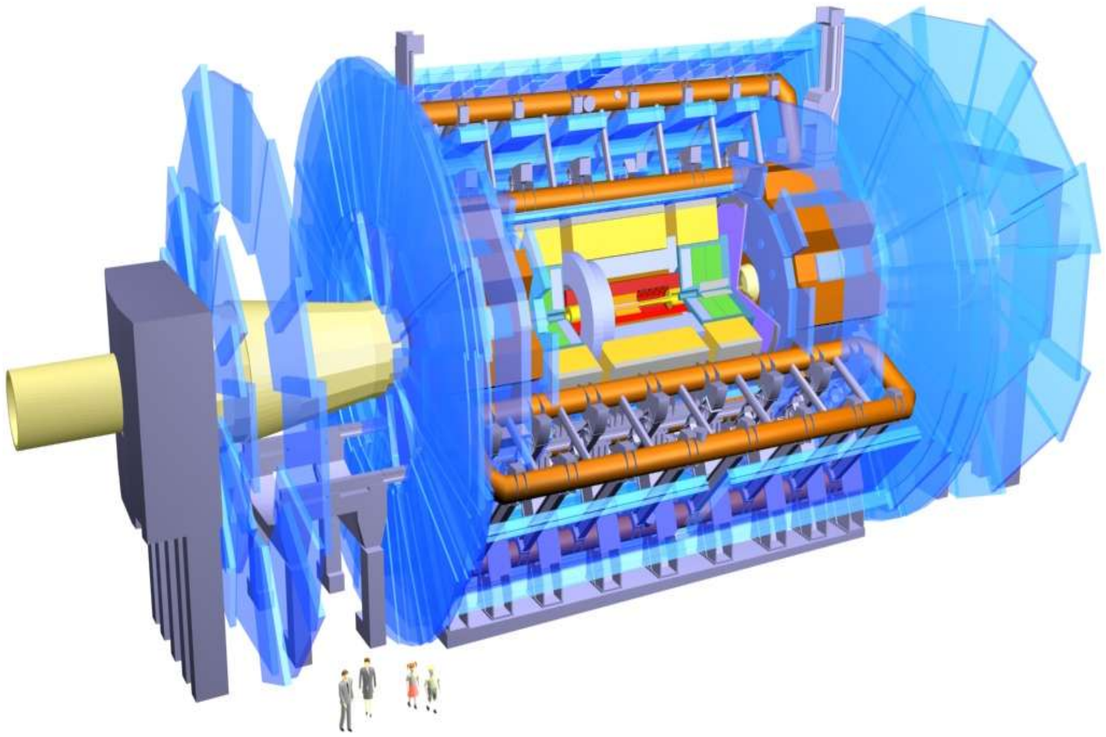

Figure 1.4: *A schematic view of the ATLAS detector.*

The magnet system of ATLAS can be seen in figure 1.5. It consists of a super-conducting solenoid that provides a homogeneous 2 Tesla magnetic field within the volume of the Inner Detector. The muon spectrometer has its own unique magnetic field, namely an toroidal field – with a peak field strength of 4 Tesla – generated by eight super-conducting air core toroids in each end-cap and around the barrel [19–22].

We already stated that hermeticity was a guiding principle for the design of ATLAS. In a proton-proton collision the two interacting partons mostly have very different momenta and so their center of mass frame will likely be boosted along the beamline. So for ATLAS it is vital to cover a large range in $\theta$, the deflection angle from the beamline. Another variable commonly used in high energy physics experiments to describe the deflection with respect to the beamline is the pseudorapidity $\eta$, defined as

$$\eta = -\ln\left(\tan\left(\frac{\theta}{2}\right)\right), \tag{1.1}$$

which is a good approximation of the true relativistic rapidity of a relativistic particle when its mass and momentum are not known [18].

The ATLAS muon system is by far the biggest subdetector system and the main reason for the enormous size of the whole detector. It provides both a precision muon spectrometer and a stand-alone trigger subsystem [23]. The precision measurements are provided by

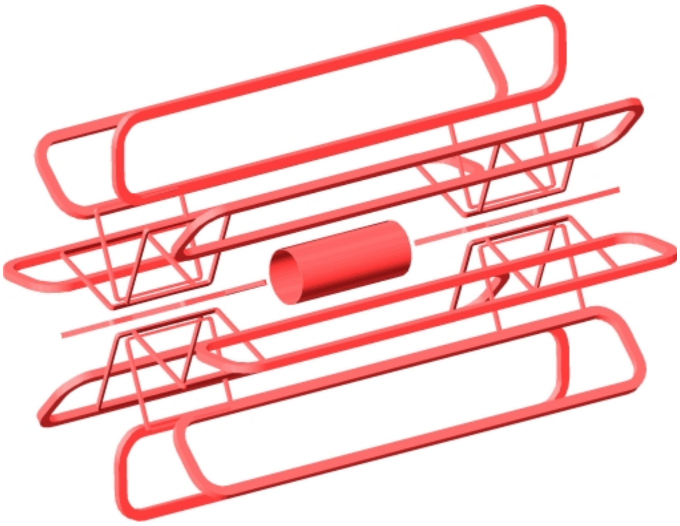

Figure 1.5: *A schematic view of the ATLAS magnet system.*

Monitored Drift Tubes (MDTs) and, in the range $2 < |\eta| < 2.7$, by radiation hard Cathode Strip Chambers (CSCs). The trigger system covers the range $|\eta| < 2.4$ and consists of both Resistive Plate Chambers (RPCs) and Thin Gap Chambers (TGCs).

The Calorimeter system consists of an electromagnetic calorimeter (EMC) covering the region $|\eta| < 3.2$, a hadronic barrel calorimeter (TileCal) covering the region $|\eta| < 1.7$, hadronic end-cap calorimeters (HEC) covering the region $1.5 < |\eta| < 3.2$ and forward calorimeters (FCAL) covering the region $3.2 < |\eta| < 4.9$. All ATLAS calorimeters are sampling calorimeters, but they use different absorptive and active materials. The TileCal uses iron as absorber and plastic scintillators as active material. The iron also serves as flux return yoke of the central solenoidal magnetic field. In all other calorimeters (EMC, HEC and FCAL) liquid argon (LAr) is used as active material. The EMC uses lead as absorber, the HEC uses copper and FCAL uses copper in its electromagnetic part and tungsten in its hadronic part [24–26].

The Inner Detector of ATLAS is designed to make high precision measurements of kinematic parameters of charged particle tracks in the range $|\eta| < 2.5$. It consists of three subsystems, the Transition Radiation Tracker (TRT), the SemiConductorTracker (SCT) and the Pixel detector. An illustration of the Inner Detector is shown in figure 1.6.

The TRT is a straw detector designed to operate at very high rates. It has $50 \times 10^3$ straws in the barrel region and $320 \times 10^3$ straws in the two end-caps together. The TRT provides particle identification by measuring transition radiation photons and it provides about 30 measurements per charged particle track which is vital for track-finding and momentum resolution. The SCT is a silicon strip detector that provides eight precision measurements per charged particle track. Two strip layers are glued back-to back with a stereo angle of 40 mrad and constitute a SCT module. In the barrel region there are 2112 SCT modules and in the forward region 1976 SCT modules in total, subdivided in two end-caps. The Pixel detector consists of 1744 pixel modules very close to the interaction point. The pixel detector provides high precision measurements for primary and secondary vertex positions with pixel points $50\ \mu\text{m} \times 400\ \mu\text{m}$. [27–29]

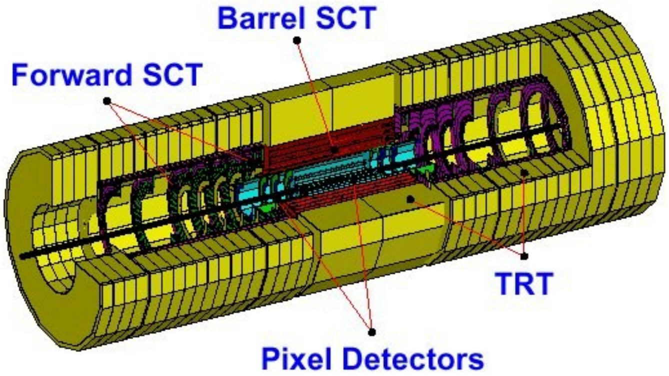

Figure 1.6: *A schematic view of the ATLAS Inner Detector.*

## 1.4 SCT

The Inner Detector is used for track finding, vertex reconstruction and momentum reconstruction. No single subdetector of the Inner Detector can do these tasks alone. TRT, SCT and Pixel detector provide complementary information about particle tracks, e.g. for the momentum measurement the relative precision of the different measurements is balanced so that no single measurement dominates the momentum resolution. Because SCT alignment is the main topic of this thesis we will explain the SCT detector in more detail.

### 1.4.1 Principle of operation

Silicon detectors are asymmetric pn-doted semiconductor junctions and to work as a detector for charged particles the pn diode is reverse-biased by applying a positive voltage on the n side [18]. Thus the depletion zone is artificially increased and a large charge carrier free volume inside the diode is created. If an ionizing particle passes through the depletion zone it produces electron-hole pairs along its path, the number being proportional to the energy loss. The externally applied electric field separates the pairs before they recombine – electrons drift towards the anode, holes to the cathode. The process is sketched in figure 1.7.

For ATLAS SCT the high voltage (HV) is applied on the back side of the diode ($n^+$ ohmic bulk contact in figure 1.7). A typical HV value for a non-iradiated ATLAS SCT diode is 150 V.

The SCT signal readout is as follows. Attached to the $p^+$ strips are aluminium readout strips that are insulated from the diode by a thin silicon-oxide layer. The collected charge in the $p^+$ strip produces a mirror charge in the aluminum strip. This signal is read out, amplified, shaped and discriminated. A binary signal is then transmitted to the data acquisition system, 1 if the readout signal was above threshold or 0 if it was below.

The intrinsic position resolution of silicon detectors is around 5 $\mu$m and is limited by transverse diffusion of the electrons and holes during charge collection [18]. The spatial

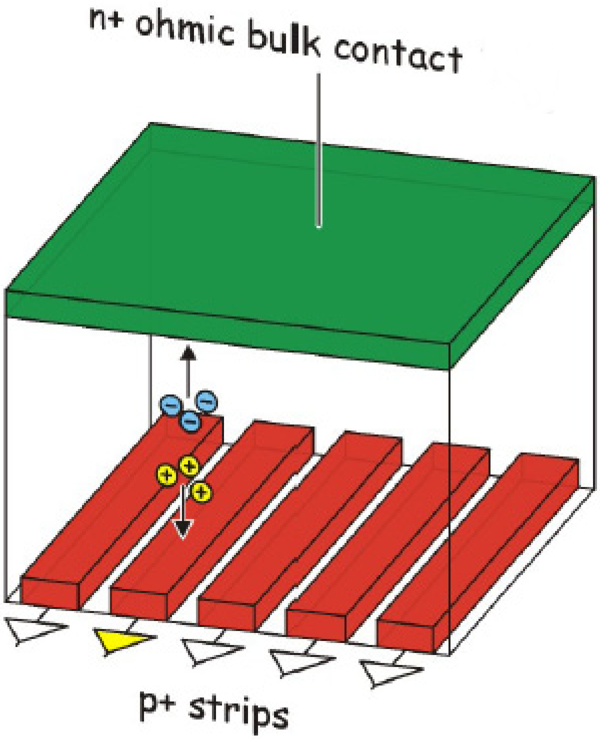

Figure 1.7: *An illustration of the principle of operation of a silicon detector. (drawing from [30]).*

resolution of the SCT detector is mainly limited by the distance of two neighboring readout strips, the pitch, as we will see in section 1.4.2.

Another important effect that needs to be taken into account is the so called Lorentz shift. As the SCT is operated within a magnetic field of 2 T the free charge carriers produced by ionizing radiation in the depletion zone are subject to the Lorentz force while they drift towards the readout strips. This results in an offset of a few microns between readout-position and the path of the ionizing particle. For ATLAS this effect is corrected for during offline event reconstruction.

### 1.4.2 Detector layout

The smallest mechanical unit of the ATLAS SCT detector that we consider in the following is a SCT module. Each SCT module has two readout sides – with 768 aluminum readout strips on each side – that are glued back-to-back with a stereo angle of 40 mrad.

SCT modules that are mounted in the barrel part of the detector have four rectangular silicon diodes, two on each side, as can be seen in figure 1.8. One readout side – the so called r-$\phi$-side – has its aluminum strips aligned along the beamline, the readout strips on the other side – the so called stereo-side – are rotated by 40 mrad with respect to the beamline. The sensitive part of a SCT barrel module is about 12 cm × 6 cm in size and the strip pitch is 80 $\mu$m.

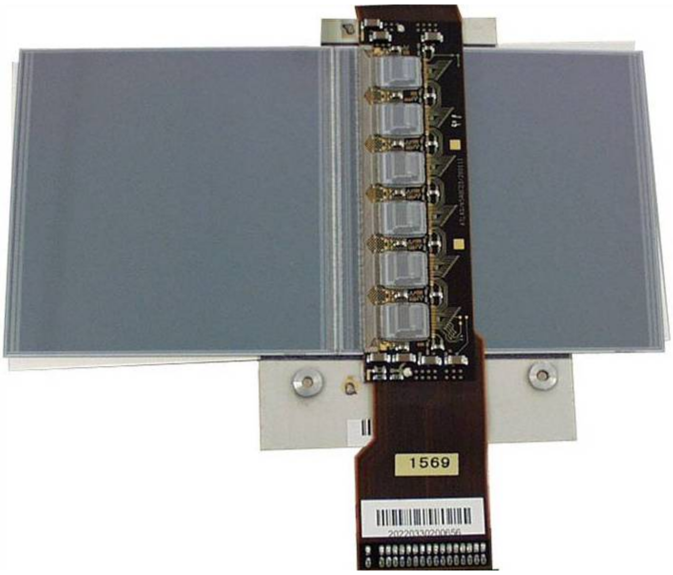

Figure 1.8: *ATLAS SCT barrel module. The readout strips run from left to right and are not visible.*

SCT end-cap modules have wedge-shaped silicon diodes where the strips of the r-$\phi$-side are aligned along lines that emerge radially from the beamline and the readout strips of the stereo-side are rotated by 40 mrad with respect to this. The strip pitch increases with radial distance from the beamline and is between 55 $\mu$m and 95 $\mu$m [31]. This results in a fanning angle of the readout strips. The radial coverage of the end-caps requires four different types of modules with either one silicon diode (short modules) or two silicon diodes (long module) on each readout side. There are three radial positions in the end-caps: inner, middle and outer position [32]. On the outer position only long modules are mounted. On the middle position mostly long modules are mounted. The middle modules on disk 7 have only one diode on each readout side (due to the high radiation

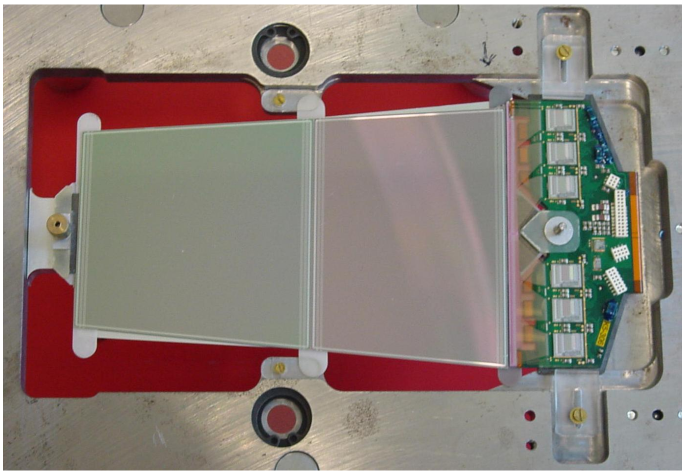

Figure 1.9: *ATLAS SCT long middle module in a support frame. The readout strips run from left to right and are not visible.*

environment) and are called short middle modules (see table 1.2). On the inner position only short modules are mounted. A long middle module is shown in figure 1.9.

The spatial resolution perpendicular to the readout strips is not limited by the intrinsic resolution of the silicon diode (about 5 $\mu$m) but by the strip pitch. The single hit resolution is 16 $\mu$m perpendicular to the readout strips and 580 $\mu$m along the readout strip [33].

The barrel part of the SCT detector consists of four cylindrical superstructures – called barrel layers – on which a total of 2112 SCT barrel modules are mounted. To achieve an overlap of neighboring modules the SCT barrel modules are mounted with a 10° tilt at the surface of the cylindrical barrel structures [34]). Details about the configuration of the barrel SCT are shown in table 1.1. A fully assembled SCT barrel layer is shown in figure 1.10.

| SCT barrel configuration | | |
|---|---|---|
| SCT barrel layer | mean radius | number of modules |
| 0 | 30 cm | 384 |
| 1 | 37 cm | 480 |
| 2 | 44 cm | 576 |
| 3 | 51 cm | 672 |

Table 1.1: *List of the mean radius and the number of modules of the four SCT barrel layers [34].*

The SCT end-cap modules are mounted on nine disks in both forward regions. The two SCT end-caps are called end-cap A and end-cap C where end-cap A is along the beamline in positive global ẑ-direction (the ATLAS coordinate system is presented in section 4.1). Details about the configuration of the SCT end-caps are listed in table 1.2. A fully assembled end-cap disk is shown in figure 1.11. The inner and outer module types can be seen. The insensitive region of the disk, where the readout electronic can be seen, is covered by the diodes of middle modules on the other side of the disk.

| SCT end-cap configuration | | | | |
|---|---|---|---|---|
| Disk | # inner modules | # middle modules | # outer modules | total |
| 0 | 0 | 40 | 52 | 92 |
| 1 | 40 | 40 | 52 | 132 |
| 2 | 40 | 40 | 52 | 132 |
| 3 | 40 | 40 | 52 | 132 |
| 4 | 40 | 40 | 52 | 132 |
| 5 | 40 | 40 | 52 | 132 |
| 6 | 0 | 40 | 52 | 92 |
| 7 | 0 | 40 * | 52 | 92 |
| 8 | 0 | 0 | 52 | 52 |

Table 1.2: *List of the number of the various module types mounted on the nine SCT end-cap disks [35]. * On disk 7 short middle modules are mounted.*

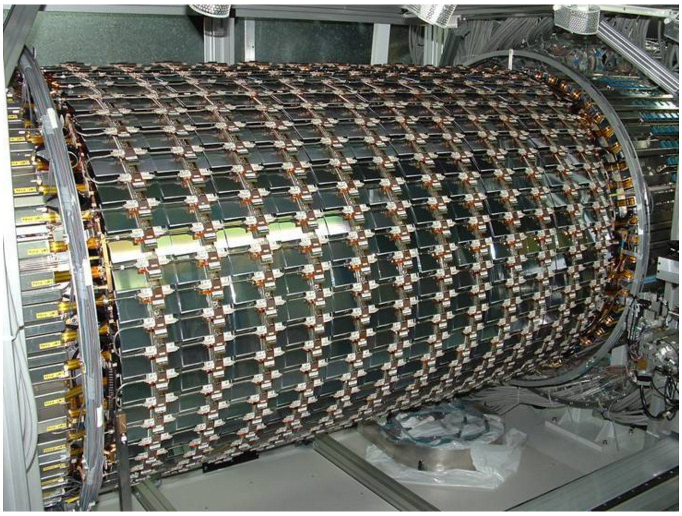

Figure 1.10: *The fully assembled SCT barrel layer 1.*

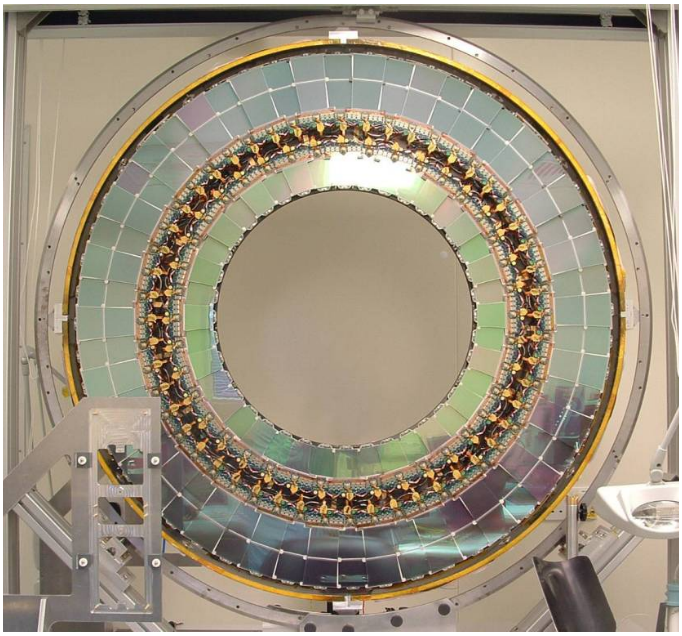

Figure 1.11: *The fully assembled SCT end-cap C disk 6. Visible is one side covered with inner and outer modules.*

### 1.4.3 Coordinate system

The local coordinate system of a SCT module is a righthanded three-dimensional orthogonal frame. It is aligned with the r-$\phi$-side of the SCT module. We will denote the local frame with (x,y,z). The local coordinate frame is sketched in figure 1.12.

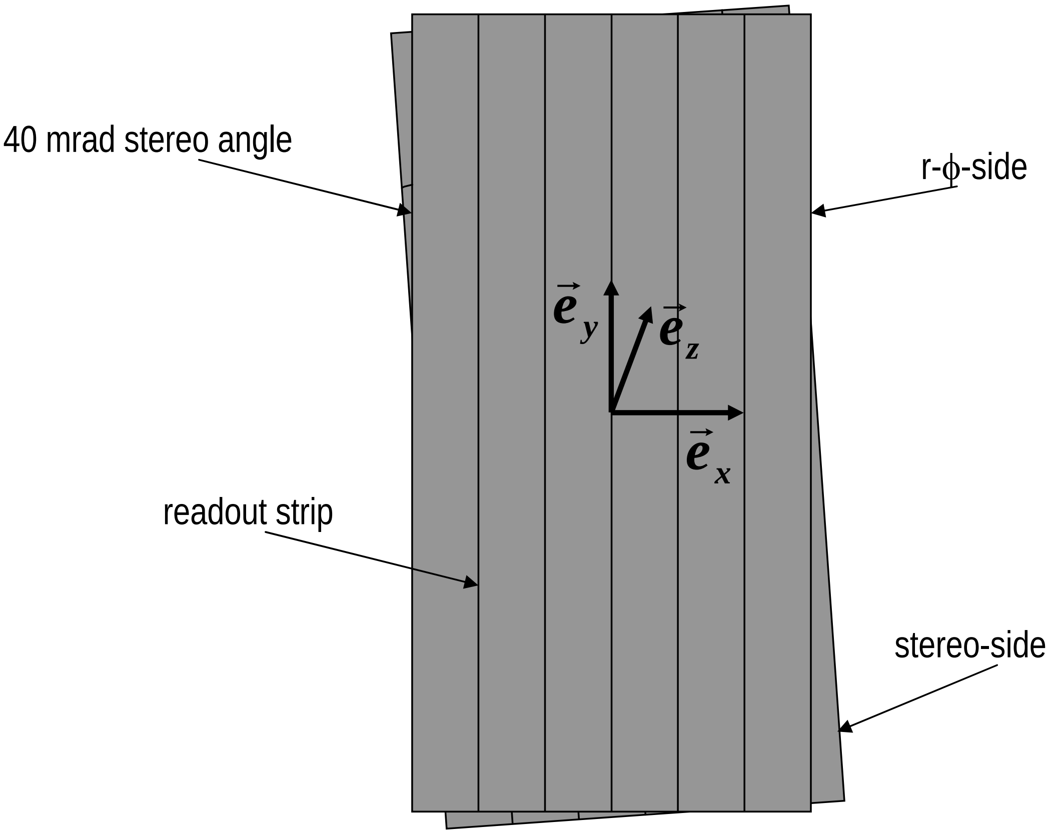

Figure 1.12: *The sketch shows the r-$\phi$-side of a barrel SCT module in the front and the stereo-side in the back. The two rectangular silicon diodes are drawn as one long silicon diode. The coordinate system of the r-$\phi$-side is sketched.*

The origin of the local frame (0,0,0) is at the center-of-gravity of the r-$\phi$-side of the SCT module. The x-axis lies in the plane of the r-$\phi$-side and is perpendicular to the readout strips. The y-axis lies in the plane of the r-$\phi$-side as well and runs along the readout strip direction. The z-axis is normal to the plane of the r-$\phi$-side. We will denote rotations around the axes (x,y,z) with the angles ($\alpha$,$\beta$,$\gamma$) respectively.

## 1.5 Alignment

To make best use of the high spatial resolution of the SCT detectors (16 $\mu$m perpendicular to the readout strip axis and 580 $\mu$m along the strip axis) it must be matched by equally precise geometrical calibration. This geometrical calibration – the process of determining the exact position of the active detector components within a global reference frame – is called alignment.

For ATLAS SCT there are various complementary alignment approaches. The first step for alignment is to build the SCT modules very precisely and to measure the as-built position of the silicon diodes on a module even more precisely, down to the level of 2 $\mu$m for the local x-coordinate [36]. The modules are then mounted precisely on the superstructures, like end-cap disks and barrel layers. The position of a mounted module on an end-cap disk is surveyed with a precision of about 10 $\mu$m [36]. The mounting precision for SCT barrel modules on the barrel layer cylinders is known to be 25 $\mu$m [37] and the mechanical precision of the carbon fiber cylinders themselves is about 100 $\mu$m. The relative position of end-cap disks with respect to each other is measured to about 100 $\mu$m [36]. The relative position of the barrel layers with respect to each other is not known at the moment but estimated to be worse than 100 $\mu$m [38]. Finally the relative position of the whole end-caps with respect to the barrel will be surveyed to about 100 $\mu$m.

Adding up all these uncertainties results in an initial as-built alignment precision for each individual module. At the moment the initial alignment is estimated to be of the order of what is listed in table 1.3. [38]

| Initial SCT alignment precision | | |
|---|---|---|
| alignment precision | SCT barrel module | SCT end-cap module |
| x | 100 $\mu$m | 50 $\mu$m |
| y | 100 $\mu$m | 50 $\mu$m |
| z | 500 $\mu$m | 500 $\mu$m |

Table 1.3: *Initial as-built alignment precision for SCT barrel and end-cap modules [38]. Coordinate axes (x,y,z) denote the local frame of a SCT module (see section 1.4.3).*

The numbers in table 1.3 must be compared with the required alignment precision. In order to not degrade track parameter (see sections 2.1 and 4.1) resolution by more than 20% alignment accuracies must be better than those listed in table 1.4. [27]

| Required SCT alignment precision | | |
|---|---|---|
| alignment precision | SCT barrel module | SCT end-cap module |
| x | 12 $\mu$m | 12 $\mu$m |
| y | 50 $\mu$m | 50 $\mu$m |
| z | 100 $\mu$m | 200 $\mu$m |

Table 1.4: *Required alignment precision for SCT barrel and end-cap modules [27]. Coordinate axes (x,y,z) denote the local frame of a SCT module (see section 1.4.3).*

With the numbers from table 1.4 we can give a rough estimation of alignment precision required for the rotational degrees of freedom. Alignment precision for $\alpha$ should be about

3 mrad, alignment precision for $\beta$ should be about 6 mrad and alignment precision for $\gamma$ should be about 0.2 mrad.

Even more ambitious is the alignment accuracy required to measure the W-boson mass with a precision of 15 MeV. It is estimated that an alignment accuracy of about 1 $\mu$m perpendicular to the readout strips is needed to achieve a momentum resolution that is precise enough [39].

Consequently, additional alignment information about the SCT detector after installation and during operation is necessary. A frequency scanning interferometer (FSI) is installed within the SCT end-cap and barrel support structures. The FSI measures distances between superstructures and is designed to monitor deformations and movements of end-cap disks and barrel cylinders with a precision of about 2 $\mu$m. The position of individual SCT modules have to be extrapolated from the FSI measurements. The alignment accuracy of the FSI for individual modules is not known.

The only other source of alignment information after installation and during operation is track based alignment which we will discuss in some detail in the following chapters.

# Chapter 2

# Local $\chi^2$ Alignment Algorithm

In this chapter we present the fundamental algebraic formalism, the core of our alignment algorithm proposed for the ATLAS SCT detector. Detector alignment is an essential link in the chain from detector construction to final physics analysis. To fully exploit the resolution of the ATLAS SCT detector – and thus its physics potential – it must be aligned with a precision significantly better than its intrinsic resolution. The condition from [27] that alignment uncertainty should not degrade track parameters by more than 20% requires an alignment accuracy – for example – perpendicular to the readout strips of an SCT module of about 12 $\mu$m. This precision can only be achieved by a track based alignment approach since constraints from survey and monitoring with FSI are estimated to be less precise than this.

## 2.1 Track based alignment

Generally speaking track reconstruction is a two step process: pattern recognition and track fitting. Pattern recognition takes a prepared sample of hits and flags certain hits to belong to one "particle trajectory". Then track fitting takes over, where a best fit under certain conditions (particle hypothesis, magnetic field map, tracking model, etc.) is performed to describe the passage of a charged particle through the detector volume. The track fit must yield certain important parameters like particle momentum and direction which are vital to any subsequent physics analysis of the event.

The ATLAS Inner Detector is designed such that under normal circumstances a charged particle generates more hits than necessary to constrain the set of equations of the underlying tracking model. A charged particle track originating from the interaction point will on average produce 10 hits in the Pixel and SCT detectors and about 30 hits in the TRT. For a helical track model five track parameters need to be fitted to the hits whereas for a straight line model it is only four parameters [40]. Thus one has to deal with an overconstrained set of equations to which only approximate solutions are possible[1]. We present more details about the ATLAS track parametrization and about track reconstruction in chapter 4.

---

[1] This is of course intentional. Only with an overconstrained fit statistical error analysis is possible.

The resulting track is an estimate of the true particle trajectory, where the hits scatter around the fitted trajectory. The distances between a track and its associated hits are called residuals and it is the task of a track fitting algorithm to produce track fits with the smallest possible set of track residuals. This is normally done with least squares minimization.

While track fitting assumes that the positions of the hits in space are perfectly known and tries to adjust the resulting tracks accordingly, track based alignment does things exactly the other way round. Tracks are considered to be perfect and residuals are minimized by modifying the detector positions. For a large sample of tracks, the residual distribution of a module with no misalignment should be centered around zero and for a module with misalignment it should be shifted. The situation is sketched in figure 2.1. The upper most module and the two lower modules are at their nominal position. Consequently, for these modules the distribution of residuals from many tracks is centered around zero. The second from top module is shifted with respect to its nominal positions and thus the residual distribution associated with this module is shifted and not centered around zero.

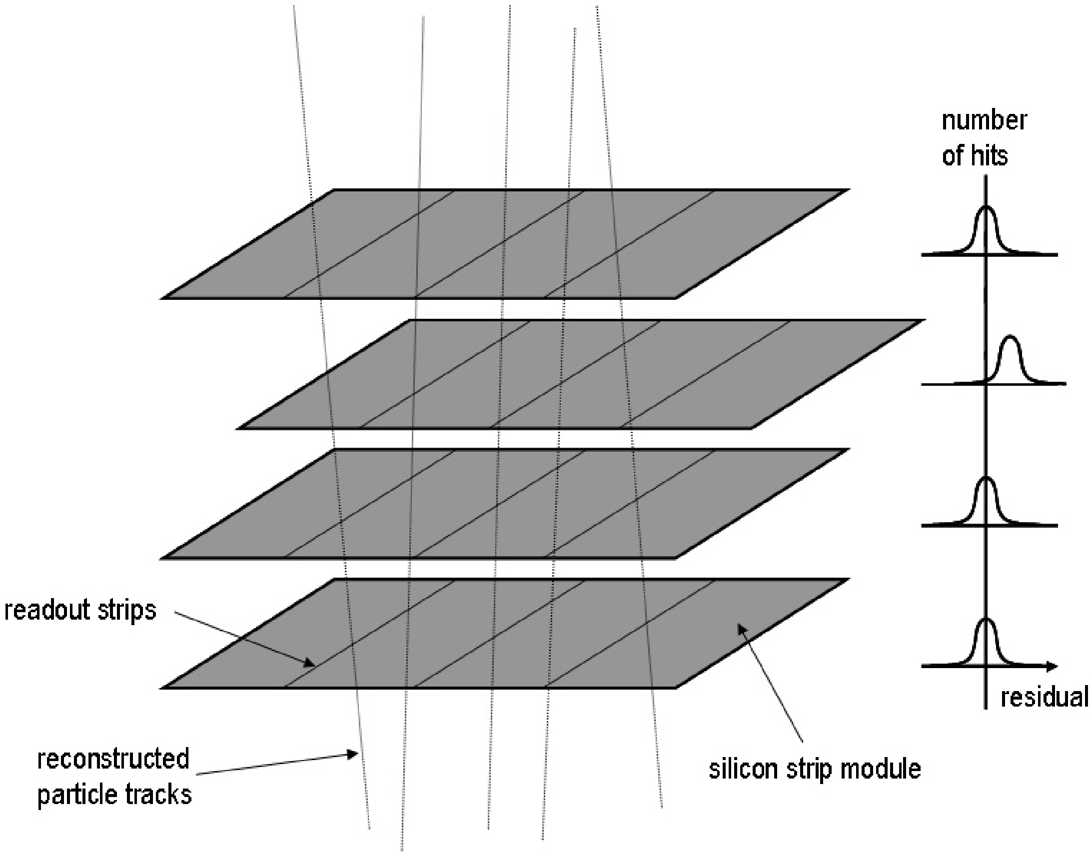

Figure 2.1: *The sketch shows four layers of modules and particle tracks passing through them. The second from top module is shifted with respect to its nominal position.*

Thus, hits, tracks and the resulting residuals are the common ground of all track based alignment approaches. However, they differ in the derivation of alignment corrections from input residual distributions. The approach we pursue for ATLAS SCT alignment treats each of the 4088 modules independently and uses a linearized $\chi^2$-minimization of the residuals with respect to all six degrees of freedom of a rigid body to infer the most likely set of alignment corrections for each given module.

## 2.2 Algebraic Derivation

The equations describing this in a mathematically rigorous fashion are presented in this section.

We derive the alignment algorithm from the following $\chi^2$-function:

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

Here $\vec{r}_i = \vec{r}_i(\vec{a}, \vec{\pi}_i)$ is the vector of residuals measured for track $i$. $\vec{r}_i(\vec{a}, \vec{\pi}_i)$ is a function of alignment parameters $\vec{a}$ and of the track parameters $\vec{\pi}_i$. $\vec{a}$ is the vector of all alignment parameters of all modules that have hits associated to one of the tracks. $\vec{\pi}_i$ is the vector of track parameters of track $i$, e.g. for straight line tracks 4 parameters and for helix tracks 5 parameters [41]. $m$ is the number of tracks that are used for alignment and $V_i$ is the covariance matrix of the residual measurements of track $i$.

We assume that $\chi^2$ for a perfectly aligned detector resides in a minimum, therefore we require:

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

For ease of reading the dependence of $\chi^2$ on $(\vec{\pi}_1, \ldots, \vec{\pi}_m)$ is suppressed in the notation.

The term $\frac{d}{d\vec{a}}$ in equation (2.2) we define as follows:

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

To solve equation (2.2) we rewrite $\chi^2$ as a Taylor expansion around the expansion point $\vec{a}_0$ where $\vec{a}_0$ is the vector of initial alignment parameters.

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

As we attempt a linear least square minimization we only expand $\chi^2$ up to first order in $\Delta\vec{a}$ with $\Delta\vec{a} = (\vec{a} - \vec{a}_0)$.

Requirement (2.2) in equation (2.4) yields:

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

$\sum_{tracks}$ is a shorthand notation for $\sum_{i\,\in\,tracks}^{m}$ and will be used in all the following equations as well. $\frac{d}{d\vec{a}_0}$ is a shorthand notation for the derivative with respect to $\vec{a}$ evaluated at $\vec{a} = \vec{a}_0$. A few intermediate steps are shown in appendix A to demonstrate the manipulations we employed to get the final result of equation (2.5). They are omitted here to keep focused on the derivation of alignment parameter corrections.

The term $\frac{d\vec{x}}{d\vec{a}}$ in equation (2.5) we define as follows:

$$
\frac{d\vec{x}}{d\vec{a}} \equiv \begin{pmatrix} \frac{dx_1}{da_1} & \frac{dx_2}{da_1} & \cdots \\ \frac{dx_1}{da_2} & \frac{dx_2}{da_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{2.6}
$$

From equation (2.5) follows immediately a formal solution for the alignment parameter correction $\Delta\vec{a}$:

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

Equation (2.7) is a set of $n$ coupled linear equations where

$$
n = number\ of\ degrees\ of\ freedom\ of\ one\ module \times number\ of\ modules. \tag{2.8}
$$

For the 4088 SCT modules of the full ATLAS setup with the six degrees of freedom of a rigid body $n = 24528$. One of the alignment approaches for the ATLAS Inner Detector, namely the global $\chi^2$ approach [42], tries to solve equation (2.7) by inverting the large matrix on a dedicated parallel computer system.

It is the core concept of our approach to solve equation (2.7) approximately by breaking it up into many small independent sets of equations, namely one set of six coupled linear equations for each module. This approach was used for alignment of the BABAR SVT detector [43].

To split up equation (2.7) we need to make certain assumptions about it. Some of these assumptions are very well motivated, but others are quite ad hoc. To show the exact nature of the simplifications we make we have to dissect equation (2.7) first.

To evaluate the $\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}$ terms in equation (2.7) we write out the total derivative:

$$\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{a}} = \frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_i)}{\partial\vec{a}} + \frac{d\vec{\pi}_i}{d\vec{a}}\frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_i)}{\partial\vec{\pi}_i} \tag{2.9}$$

Now we have expressed one total derivative as a function of another, namely $\frac{d\vec{\pi}_i}{d\vec{a}}$. The track parameters $\vec{\pi}_i$ have a dependence on the alignment parameters of the $n$ modules that have hits associated to the track $i$.

$$\vec{\pi}_i = \vec{\pi}_i(\vec{a}_1, \vec{a}_2, ..., \vec{a}_n) \tag{2.10}$$

To get a handle on $\frac{d\vec{\pi}_i}{d\vec{a}}$ we need to go back to equation (2.1) and realize that we can also minimize $\chi^2(\vec{a}, \vec{\pi}_1, \ldots, \vec{\pi}_m)$ with respect to $\vec{\pi}_i$ and thus get an expression for $\Delta\vec{\pi}_i = \vec{\pi}_i - \vec{\pi}_{i\,0}$. This goes in complete analogy to equations (2.2) - (2.7) and yields:

$$\Delta\vec{\pi}_i = \vec{\pi}_i - \vec{\pi}_{i\,0} = -\left(\left(\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{\pi}_{i\,0}}\right) V_i^{-1} \left(\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{\pi}_{i\,0}}\right)^T\right)^{-1} \left(\left(\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{\pi}_{i\,0}}\right) V_i^{-1} \vec{r}_i(\vec{a},\vec{\pi}_{i\,0})\right) \tag{2.11}$$

A detailed derivation of equation (2.11) is shown e.g. in [42]. Again we used the $\frac{d}{d\vec{\pi}_{i\,0}} = \frac{d}{d\vec{\pi}_i}\Big|_{\vec{\pi}_i=\vec{\pi}_{i\,0}}$ shorthand notation. It is worth noting that in equation (2.11) the residual $\vec{r}_i(\vec{a},\vec{\pi}_{i\,0})$ is no longer a function of $\vec{\pi}_i$ but has been evaluated with the initial track parameters $\vec{\pi}_{i\,0}$ already.

In equation (2.11) appears a term that can be identified as the covariance matrix $C_\pi$ of the track parameters $\pi$.

$$C_\pi = \left(\left(\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{\pi}_{i\,0}}\right) V_i^{-1} \left(\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{\pi}_{i\,0}}\right)^T\right)^{-1} \tag{2.12}$$

In the following equations we will use $C_\pi$ as a notational shorthand form for the above expression.

By using equation (2.11) as an equation for $\vec{\pi}_i$ we can now get from total to partial derivatives:

$$\frac{d\vec{\pi}_i(\vec{a})}{d\vec{a}} = -\left(\frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_{i\,0})}{\partial\vec{a}}\right) \cdot V_i^{-1} \cdot \left(\frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_i)}{\partial\vec{\pi}_{i\,0}}\right)^T \cdot C_\pi \tag{2.13}$$

Reinserting (2.13) into (2.9) we get:

$$\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{a}} = \frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_i)}{\partial\vec{a}} - \left(\frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_{i\,0})}{\partial\vec{a}}\right) \cdot V_i^{-1} \cdot \left(\frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_i)}{\partial\vec{\pi}_{i\,0}}\right)^T \cdot C_\pi \cdot \frac{\partial\vec{r}_i(\vec{a},\vec{\pi}_i)}{\partial\vec{\pi}_i} \tag{2.14}$$

To see the underlying structure more clearly we need to switch to index notation and look at the entries $\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_l}$ of the matrix $\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{a}}$. Here $r_{i\,k}$ means the residual of track $i$ on module $k$ and $\vec{a}_l$ is the vector of alignment parameters of module $l$.

In index notation equations (2.9) and (2.14) become:

$$\begin{aligned}\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_l} &= \frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{a}_l} + \frac{d\vec{\pi}_i}{d\vec{a}_l}\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{\pi}_i} = \\ &= \left(\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{a}_l}\right) - \left(\frac{\partial r_{i\,l}(\vec{a}_l,\vec{\pi}_{i\,0})}{\partial\vec{a}_l}\right)\cdot V_i^{-1}\cdot\left(\frac{\partial r_{i\,l}(\vec{a}_l,\vec{\pi}_i)}{\partial\vec{\pi}_{i\,0}}\right)^T\cdot C_\pi\cdot\left(\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{\pi}_i}\right)\end{aligned} \tag{2.15}$$

With equation (2.15) we are now in the position to discuss the two cases $k = l$ and $k \neq l$. For the $k = l$ case we can remove from the track parameters $\vec{\pi}_i$ the dependence on $\vec{a}_k$ and thus force $\frac{d\vec{\pi}_i}{d\vec{a}_k} = 0$. To do so we must use unbiased track parameters, i.e. track parameters calculated with the hit on module $k$ removed from the track. The track parameters then do not depend on the alignment parameters of module $k$ anymore and – most importantly – are not biased by any misalignment of module $k$ (hence the name). So for $k = l$ and "k-unbiased" track parameters $\vec{\pi}_i$ equation (2.15) reduces to

$$\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_k} = \frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{a}_k}. \tag{2.16}$$

It is obvious that we cannot remove all hits from track $i$ to unbias it entirely. Without hits there is no track. Even the removal of one hit already degrades the quality of the track fit. As we will see in chapter 4 removing one measurement from track $i$ will leave its fit quality good enough for the purpose of our alignment approach. Any misalignments from modules $l \neq k$ will still bias the track parameters and consequently the residual and derivative calculation.

In case of $k \neq l$ the term $\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{a}_l} = 0$ as $r_{i\,k}$ does not explicitly depend on $\vec{a}_l$ and so equation (2.15) becomes

$$\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_l} = -\left(\frac{\partial r_{i\,l}(\vec{a}_l,\vec{\pi}_{i\,0})}{\partial\vec{a}_l}\right)\cdot V_i^{-1}\cdot\left(\frac{\partial r_{i\,l}(\vec{a}_l,\vec{\pi}_i)}{\partial\vec{\pi}_{i\,0}}\right)^T\cdot C_\pi\cdot\left(\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{\pi}_i}\right). \tag{2.17}$$

This is not necessarily zero. It is a gauge of the correlation between measurements from modules $k$ and $l$. More exactly, from equation (2.17) we see that there is a anti-correlating effect between modules. This can be easily understood by going back to figure 2.1.

If initial alignment parameters have the second from top module at its nominal position directly above and below the other three modules, then tracks fitted through hits on the four modules are pulled to the left. The track parameters are biased by the misalignment. When trying to correct for this misalignment the second from top module is for sure shifted from the initial position towards its actual position, i.e. to the right. But the other three modules are shifted from their initial positions to the left, because of the biased track parameters. The "left-right" anti-correlation from this example is a direct consequence of equation (2.17).

In equation (2.17) we identify

$$\left(\frac{\partial r_{i\,l}(\vec{a}_l,\vec{\pi}_i)}{\partial\vec{\pi}_{i\,0}}\right)^T\cdot C_\pi\cdot\left(\frac{\partial r_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{\partial\vec{\pi}_i}\right)=U_i \tag{2.18}$$

as the error contribution of the track parameters to the residual error. Consequently, $V_i^{-1}\cdot U_i$ is a weight factor of the correlation term in equation (2.17). For a diagonal measurement covariance matrix $V_i$ and a diagonal track covariance matrix $U_i$ the term $V_i^{-1}\cdot U_i$ is the squared ratio of track error and measurement error of the residual.

The origin of correlations between modules in track based alignment is the common error of the track parameters of one track. Modules that have hits associated to the same track share a common error source and are thus correlated. In the light of this we can understand the intrinsic weight factor in equation (2.17). Residual measurements where the contribution of the track error is small compared to the hit error, only have a weak correlation to other residual measurements on the track.

Nevertheless the anti-correlation effect cannot be discussed away. Still we need to suppress the correlations given by equation (2.17) if we want to solve the alignment fit equation (2.7) for each module individually. We chose to ignore equation (2.17) and set

$$\frac{dr_{i\,k}(\vec{a}_k,\vec{\pi}_i)}{d\vec{a}_l}\equiv 0. \tag{2.19}$$

In section 2.3 we consider an iterative algorithm that indirectly restores the correlating effects which we suppress here.

Having said all this, we can now write equation (2.9) in a simplified form:

$$\begin{aligned}\frac{d\vec{r}_i(\vec{a},\vec{\pi}_i)}{d\vec{a}} &= \begin{pmatrix}\frac{dr_{i\,1}(\vec{a}_1,\vec{\pi}_i)}{d\vec{a}_1} & \frac{dr_{i\,2}(\vec{a}_2,\vec{\pi}_i)}{d\vec{a}_1} & \cdots\\ \frac{dr_{i\,1}(\vec{a}_1,\vec{\pi}_i)}{d\vec{a}_2} & \frac{dr_{i\,2}(\vec{a}_2,\vec{\pi}_i)}{d\vec{a}_2} & \cdots\\ \vdots & \vdots & \ddots\end{pmatrix} \approx \begin{pmatrix}\frac{\partial r_{i\,1}(\vec{a}_1,\vec{\pi}_i)}{\partial\vec{a}_1} & & 0\\ & \frac{\partial r_{i\,2}(\vec{a}_2,\vec{\pi}_i)}{\partial\vec{a}_2} & \\ 0 & & \ddots\end{pmatrix} = \\ &= \mathbf{1}\begin{pmatrix}\frac{\partial r_{i\,1}(\vec{a}_1,\vec{\pi}_i)}{\partial\vec{a}_1}\\ \frac{\partial r_{i\,2}(\vec{a}_2,\vec{\pi}_i)}{\partial\vec{a}_2}\\ \vdots\\ \frac{\partial r_{i\,n}(\vec{a}_n,\vec{\pi}_i)}{\partial\vec{a}_n}\end{pmatrix}\end{aligned} \tag{2.20}$$

The only non diagonal part of equation (2.7) then is the covariance matrix of the residual measurements $V_i$. The entries $\sigma^2_{i\,kk}$ in the diagonal of $V_i$ correspond to the squared measurement error of residual $r_{i\,k}$. The non-diagonal elements $\sigma_{i\,kl}$ of $V_i$ correspond to correlated measurement errors due to multiple coulomb scattering (MCS). The non-diagonal nature of $V_i$ can be suppressed by only using high momentum tracks (i.e. $p\geq 10$ GeV) [2] to stem the effect of MCS.

---

[2] In the following we set $\hbar=c=1$

Assuming that MCS effects are negligible we can write:

$$V_i \approx \begin{pmatrix} \sigma_{i\,1}^2 & & 0 \\ & \sigma_{i\,2}^2 & \\ 0 & & \ddots \end{pmatrix} = \mathbf{1} \begin{pmatrix} \sigma_{i\,1}^2 \\ \sigma_{i\,2}^2 \\ \vdots \\ \sigma_{i\,n}^2 \end{pmatrix} \tag{2.21}$$

and equation (2.7) disentangles to

$$\Delta\vec{a}_k = -\left(\sum_{tracks} \frac{1}{\sigma_{i\,k}^2} \left(\frac{\partial r_{i\,k}(\vec{a}_k)}{d\vec{a}_{k\,0}}\right) \left(\frac{\partial r_{i\,k}(\vec{a}_k)}{d\vec{a}_{k\,0}}\right)^T\right)^{-1} \cdot \left(\sum_{tracks} \frac{1}{\sigma_{i\,k}^2} \left(\frac{\partial r_{i\,k}(\vec{a}_k)}{d\vec{a}_{k\,0}}\right) r_{i\,k}(\vec{a}_{k\,0})\right) \tag{2.22}$$

where $\Delta\vec{a}_k$ denotes the vector of alignment corrections for module $k$.

From this stage onward we will only look at the alignment corrections of individual modules, so we drop the index $k$ in the notation.

The linear approximation of the covariance matrix for the vector of alignment parameters $\vec{a}$ is given by [44]

$$\left[\sigma_a^2\right] = 2\left(\frac{\partial^2\chi^2}{\partial a_0^2}\right)^{-1} \approx \left(\sum_{tracks} \frac{1}{\sigma_i^2} \left(\frac{\partial r_i(\vec{a})}{d\vec{a}_0}\right) \left(\frac{\partial r_i(\vec{a})}{d\vec{a}_0}\right)^T\right)^{-1} . \tag{2.23}$$

And thus the error $\sigma_{a\,j}$ of the alignment parameter $a_j$ is given by

$$\sigma_{a\,j} = \sqrt{\sigma_{a\,jj}^2} \tag{2.24}$$

## 2.3 Iterative approach

To overcome the manifest limitations regarding correlations, this approach is designed to run iteratively. In figure 2.2 the basic concept behind the iterative approach is shown. Initial detector layout and hit positions are fed into the track reconstruction machinery. Once hits are associated to tracks and optimal track fits have been done, residuals and derivatives for each module and all tracks are calculated. Then, after the whole track sample has been analyzed the $\chi^2$-minimization as outlined in section 2.2 is done for each module. Thus the best fit estimates for alignment corrections for each module are derived and the detector layout is updated. With this updated detector layout and the original hits a new run of track reconstruction and subsequent alignment is started with the same track sample. This procedure is expected to converge to final alignment corrections for each module. If convergence criteria are met the alignment task is finished.

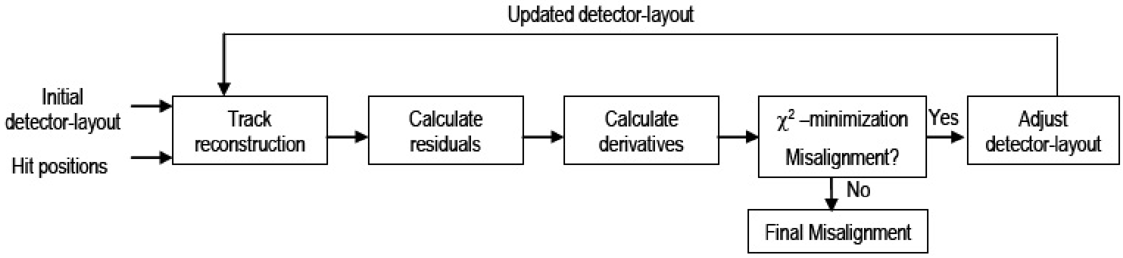

Figure 2.2: *Flowchart of iterative local* $\chi^2$ *alignment.*

As each module is treated independently to derive the alignment corrections, no correlations with respect to other modules are taken into account during one iteration. However, track parameters with a better fit quality (due to a better aligned detector layout used for track reconstruction between iterations) gradually bring correlations into play.

# Chapter 3

# Prototype Program with ROOT

## 3.1 ROOT

ROOT is an object-oriented analysis framework aimed at solving the data analysis challenges of high energy physics [45].

While not specialized for the ATLAS experiment like the dedicated ATLAS software framework Athena (see section 4.1), ROOT is a good test bed for code design and prototype development. Powerful geometry and detector description packages, tracking packages and math libraries, especially linear algebra routines and of course histogramming and fitting tools are provided within the ROOT framework.

## 3.2 Geometry setup and tracking model

Part of the price for not working in the ATLAS software framework Athena was the necessity to implement ATLAS SCT specific geometry and detector description into ROOT. For sake of simplicity we decided to use only one SCT barrel module for alignment studies with the prototype program. Detector description of the SCT module within the ROOT geometry manager resembles a real ATLAS SCT barrel module as closely as possible. The simulated module has two readout sides that are rotated with respect to each other by a stereo angle of 40 mrad. One side we will call the r-$\phi$-side and the other side we will call stereo-side. Each readout side is 12 cm × 6 cm in size, 285 $\mu$m thick and has 768 readout strips with a strip pitch of 80 $\mu$m. The centers-of-gravity of the two readout sides are spaced 885 $\mu$m apart in local z-direction (see section 1.4.3). We simulate each readout side as one "long" silicon diode and not as two "short" silicon diodes that are wirebonded to each other (see 1.4.2). See figure 3.1 for details.

In the context of the ROOT prototype program we will use three coordinate systems. Firstla, a global coordinate frame in which the position of the SCT module is defined and secondly two local coordinate frames, one for each readout side of the SCT module. The global frame is a righthanded three-dimensional orthogonal frame ($\hat{x}$,$\hat{y}$,$\hat{z}$). The origin (0,0,0) of this frame is our simulated interaction point from which all tracks emerge. The local frames of the two readout sides are righthanded three-dimensional orthogonal frames

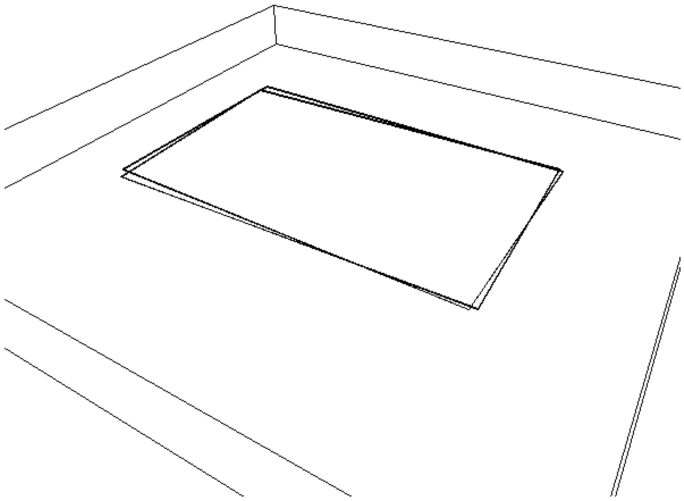

Figure 3.1: *Setup of the SCT module geometry with ROOT GeoModel. The two readout sides of a SCT module are visible. They are rotated by an angle of 40 mrad with respect to each other. The simulated readout strips on each side of the SCT module are not visible.*

as well. We will denote the local frame aligned with the r-$\phi$-side of the SCT module with (x,y,z), as outlined in section 1.4.3. We will describe all alignment parameters and all rotations of the SCT module with respect to the r-$\phi$ frame. The three angles ($\alpha$,$\beta$,$\gamma$) denote rotations around the r-$\phi$ coordinate axes (x,y,z) respectively. The other local frame (x',y',z') is aligned with the stereo-side where z' is normal to the plane of the stereo-side, y' is along the readout strips of the stereo-side and x' is perpendicular to the readout strips. We only need the stereo frame for our definition of the in-plane residual (see section 3.3.1).

In the geometry setup of our ROOT prototype program we placed the SCT module at the global coordinates (0, 20 cm, -44 cm) and tilted the SCT module with an angle $\beta = 10°$. This corresponds to a SCT module on barrel layer 2.

We implemented a simple tracking model with straight line tracks. All tracks come from the origin of the global coordinate frame and are aimed at the SCT module with a random spread of the track direction to homogeneously hit the entire module cross section, as shown in figure 3.2.

Our tracking model in the ROOT prototype program has certain limitations. The tracks are perfect straight lines without any associated tracking error and we only simulate the SCT readout as a single strip readout without cluster hit formation (see section 4.3.4). For a real SCT module the $p^+$ strips collect the free charge carriers in the diode volume below the area of ± half the strip pitch around the center of the strip. This results in a finite hit resolution that is mainly limited by the strip pitch. In the ROOT prototype program we approximate the aluminium readout strip that is attached to the $p^+$ with a straight line and associate a hit to the readout strip if the track passes within a distance of ± half the strip pitch. The result is a finite hit resolution that is in good agreement

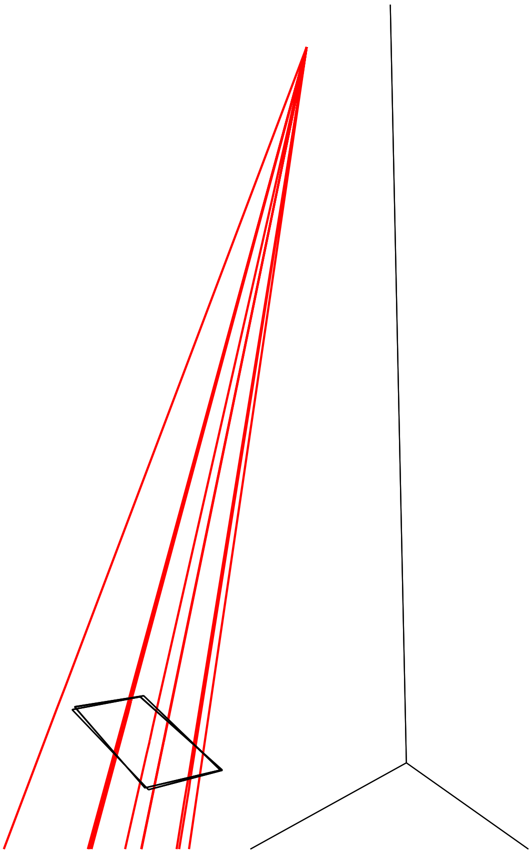

Figure 3.2: *Straight line tracks originate from a common vertex, propagate through the detector volume and hit the SCT module.*

with the finite hit resolution of a real SCT module.

## 3.3 Implementation

From equation (2.22) it is evident that we need three types of input to perform the linearized $\chi^2$-minimization and infer alignment parameters. We need residuals, residual errors and residual derivatives with respect to the alignment parameters.

### 3.3.1 Residual and residual error estimates

In section 2.1 we already gave a general definition of a residual as the distance between a track and an associated hit. Commonly this is interpreted as the distance between track and hit in the plane of the detector. The distance vector lies in the plane of the detector and we will call this type of residual an in-plane residual $r_{in-plane}$.

In an alternative definition is the residual is defined as the Distance Of Closest Approach (DOCA) between track and hit. Normally this means that the distance vector does not lie in the measurement plane. We will call this type of residual a DOCA residual $r_{DOCA}$. See figure 3.3 for details.

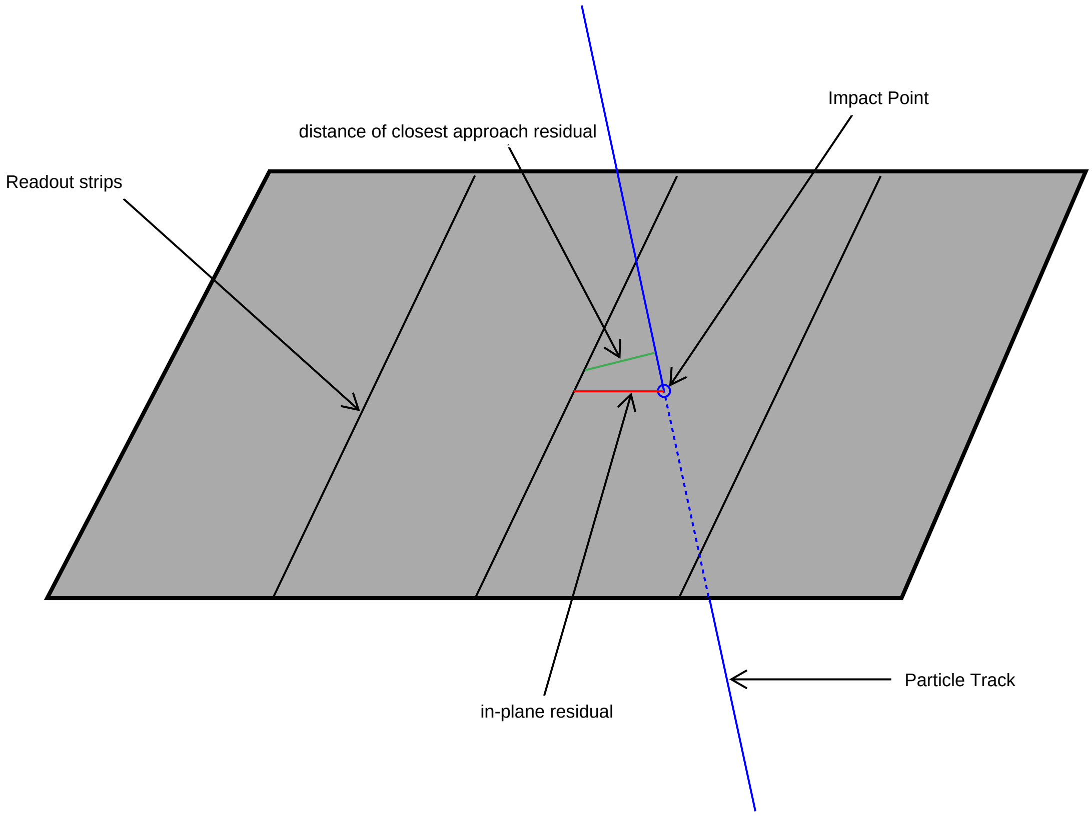

Figure 3.3: *Two definitions for "residual" are possible. The red line between track and readout strip is the in-plane residual. The green line is the distance of closest approach (DOCA) residual.*

The calculation of $r_{in-plane}$ in the tracking and geometry schema adopted for the ROOT prototype program is as follows. In the local frame of each readout side the only relevant coordinate to describe a readout strip is the x-coordinate (or x'-coordinate). The in-plane residual is then defined as the signed distance between the x-coordinate of the readout strip and the x-coordinate of the impact point of the track on the plane of the readout side.

$$r_{in-plane} = x_{impact\ point} - x_{readout\ strip} \tag{3.1}$$

We calculate $r_{DOCA}$ by calculating the distance between crossing straight lines. One line represents the straight line track and the other line represents the SCT readout strip that was hit. Both lines are described by a point and a direction vector.

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

$$\text{readout strip: } \vec{x} = \vec{c} + \kappa\vec{d} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} + \kappa \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} \tag{3.3}$$

The necessary input for this representation of a line is readily accessible in the tracking and geometry schema adopted for the ROOT prototype program.

The signed shortest distance between the two lines $r_{DOCA}$ is calculated as follows [46]:

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

The sign of $r_{DOCA}$ is determined by the sign of the determinant in the numerator of equation (3.4) and is incidentally in our limited tracking model with all tracks coming from the same direction always the same as the sign of $r_{in-plane}$.

All investigations with the prototype program in ROOT were done with both $r_{in-plane}$ and $r_{DOCA}$. In figure 3.4 distributions of $r_{in-plane}$ and $r_{DOCA}$ for both readout sides are shown. We generated straight line tracks until 10000 hits on each readout side were accumulated. The distinctive top hat shape of the residual distribution has two reasons. Firstly the straight line tracks have no fitting uncertainty, consequently the edges of the residual distributions are not smeared out. And secondly we consider only single strip hits in our prototype program. Cluster hits, where two neighboring readout strips are associated to a track, would produce a triangular shaped residual distribution (see section 4.3.4). Without track smearing and using only single strip hits every distance from the center of a readout strip up to ± half the strip pitch is equally probable as the impact point of a track. A flat residual distribution is the result. The observed RMS of the residual distributions is $\sigma = 23\ \mu$m as expected for a top hat distribution with a width of 80 $\mu$m (see appendix A).

Because we simulate perfect straight line tracks without track fit error, the RMS of the residual distributions is the only error contribution of our residual measurement and so we use it as the residual error estimate.

To simulate the effects of track smearing, to test the robustness of our alignment algorithm against such effects and to validate proper error estimation and propagation, in addition we simulated a random gaussian readout with a standard deviation equal to the RMS of the top hat distribution, i.e. $\sigma = 23\ \mu$m. As shown in figure 3.5, gaussian shaped $r_{in-plane}$ and $r_{DOCA}$ distributions are the result.

All investigations with the prototype program in ROOT were done with both top-hat shaped and gaussian shaped residual distributions.

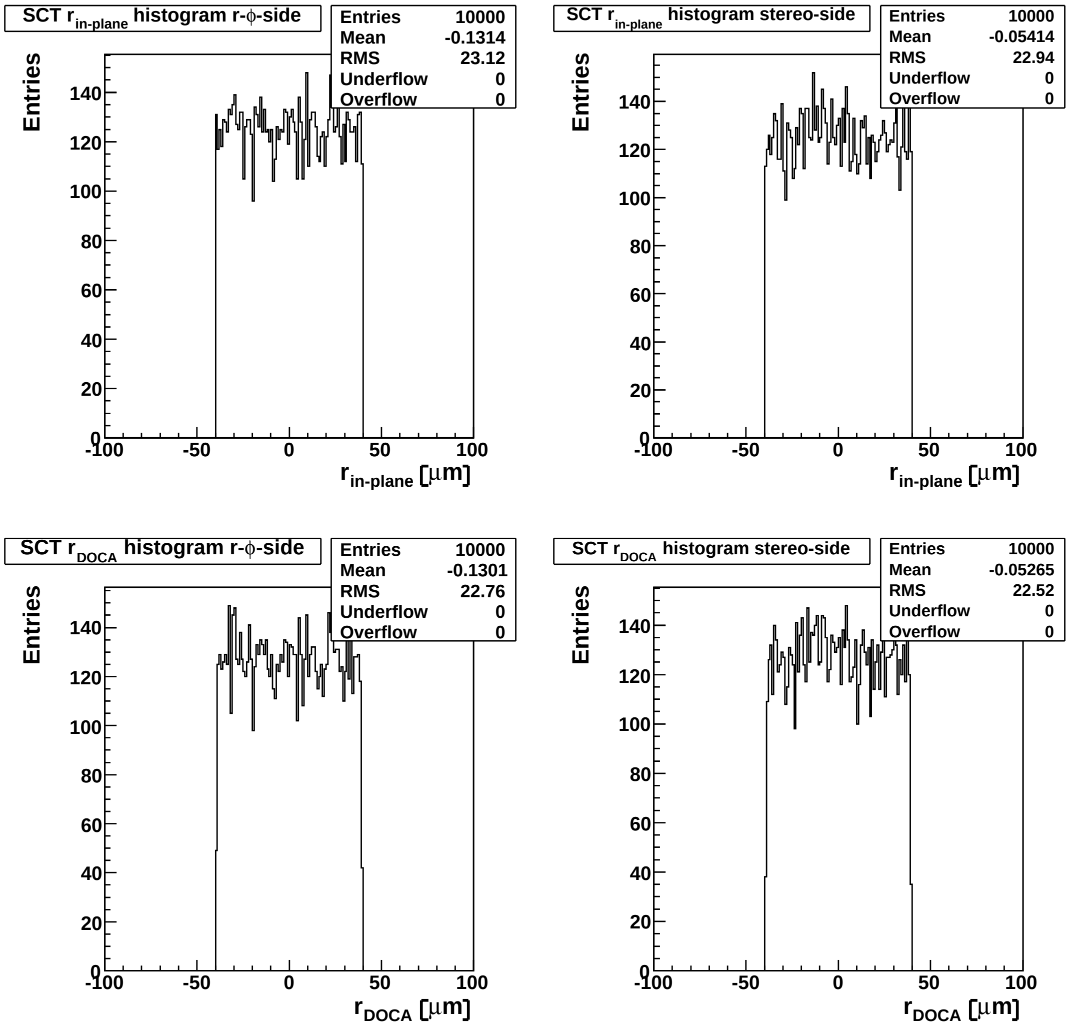

Figure 3.4: *Residual distributions for* $r_{in-plane}$ *(upper left and upper right plots) and* $r_{DOCA}$ *(lower left and lower right plots). The top hat shape is a result of the simplifications within the ROOT prototype program. The RMS of the distribution is in agreement with the expected standard deviation of 23* $\mu m$.

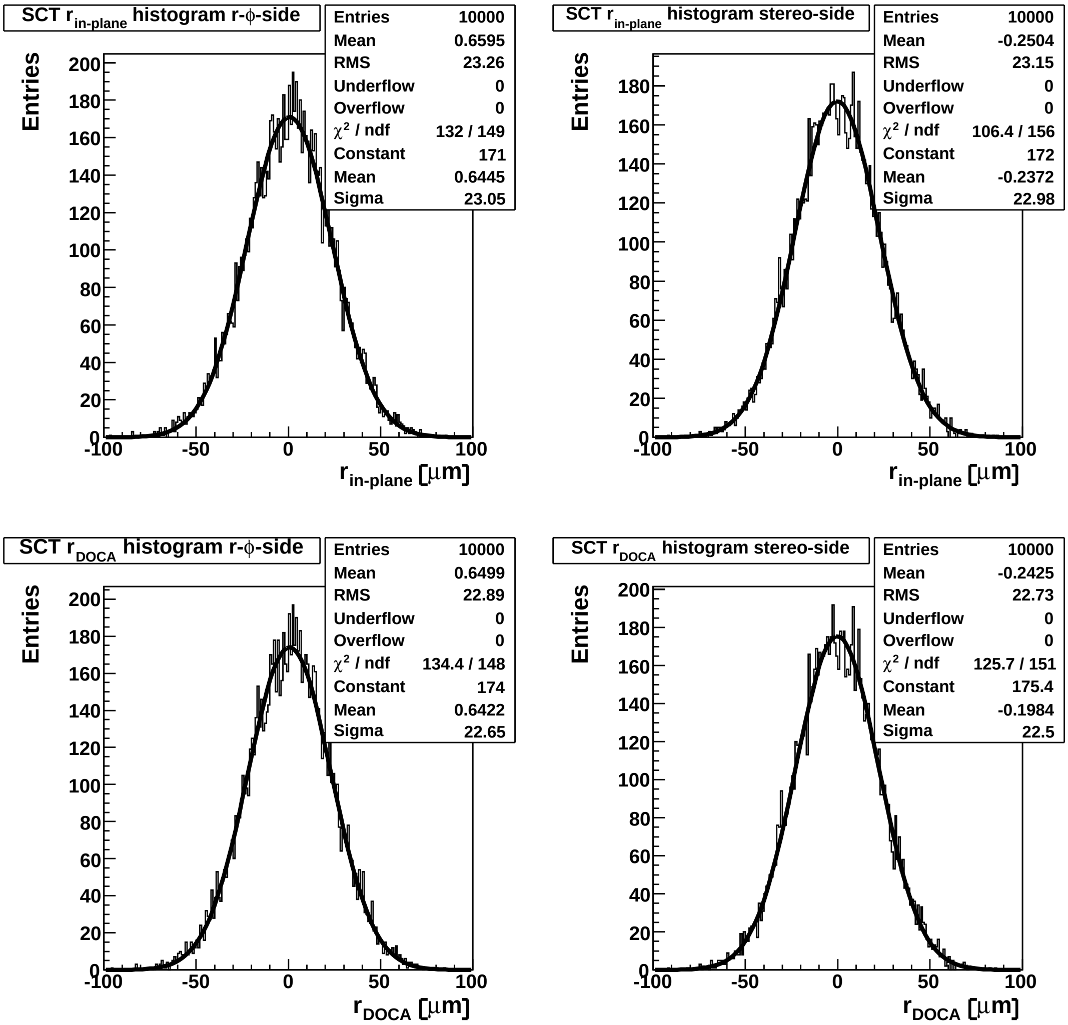

Figure 3.5: *Residual distributions for* $r_{in-plane}$ *(upper left and upper right plots) and* $r_{DOCA}$ *(lower left and lower right plots). The gaussian shape is a result of the random gaussian smearing of the hit position on the readout side plane.*

### 3.3.2 Residual derivatives

We calculate the derivatives of $r_{in-plane}$ and $r_{DOCA}$ with respect to the six alignment parameters $a_i$ numerically with the difference quotient.

$$\frac{\partial r_{in-plane}}{\partial a_i} = \frac{r_{in-plane}(a_i + h_i) - r_{in-plane}(a_i - h_i)}{2h_i} \tag{3.5}$$

$$\frac{\partial r_{DOCA}}{\partial a_i} = \frac{r_{DOCA}(a_i + h_i) - r_{DOCA}(a_i - h_i)}{2h_i} \tag{3.6}$$

$r_{in-plane}(a_i + h_i)$ and $r_{DOCA}(a_i + h_i)$ are computed by shifting the SCT module from its original position in the r-$\phi$ frame about the parameter $h_i$ and redoing the $r_{in-plane}$ and $r_{DOCA}$ calculations for both readout sides.

Distributions of $\frac{\partial r_{in-plane}}{\partial a_i}$ and $\frac{\partial r_{DOCA}}{\partial a_i}$ based on top hat shaped residual distributions are shown in figures 3.6 and 3.7. The coordinate axes (x,y,z) denote the coordinates of the r-$\phi$ frame and ($\alpha$,$\beta$,$\gamma$) denote the angles of rotation around (x,y,z) respectively. The absolute values of $\frac{\partial r_{in-plane}}{\partial a_i}$ and $\frac{\partial r_{DOCA}}{\partial a_i}$ are a direct measure of the sensitivity of track based alignment to constrain the alignment parameter $a_i$. As expected, a misalignment in x-direction is the translational degree of freedom that is constrained most. A misalignment along the x-direction would result in a direct shift of the residual distribution. Misalignments in y-direction on the other hand are only weakly constrained as only the small stereo angle gives rise to any sensitivity along this directions at all. Only hits on the stereo side yield a non-zero value of $\frac{\partial r_{in-plane}}{\partial y}$ and $\frac{\partial r_{DOCA}}{\partial y}$. Our alignment algorithm is only sensitive to the z-direction because the incident angle of the tracks with respect to the module plane is not 90°. Due to the tilt angle of the SCT module of 10° sensitivity from the x-coordinate is transferred to the z-coordinate and consequently the values of $\frac{\partial r_{in-plane}}{\partial z}$ and $\frac{\partial r_{DOCA}}{\partial z}$ are non-zero.

The rotational degrees of freedom are fairly well constrained, as even tiny rotations result in obvious distortions of the residual distributions. This is because of the large lever arm a rotation has due to the size of the SCT module (12 cm × 6 cm). The different widths of the derivative distributions for the rotational degrees of freedom are due to the different lever arms (rotations with $\beta$ only have a lever arm of up to 3 cm whereas rotations with $\alpha$ and $\gamma$ have lever arm of up to 6 cm. The coordinate axes that are affected by a rotation have different sensitivities. A rotation with $\gamma$ results in a displacement along the x-direction and is thus strongly constrained. Rotations with $\alpha$ and $\beta$ however, result mainly in shifts along the z-direction and are consequently not as strongly constrained.

In section 4.3.6 we discuss the residual derivative distributions in more detail. Distributions of $\frac{\partial r_{in-plane}}{\partial a_i}$ and $\frac{\partial r_{DOCA}}{\partial a_i}$ based on gaussian shaped residual distributions are shown in appendix B and look similar to the distributions in figures 3.6 and 3.7.

### 3.3.3 Alignment procedure

Residual measurements, residual error estimates and residual derivatives from both r-$\phi$- and stereo-side of the module are fed into equation (2.22). We can combine the measurements from both readout sides because two assumptions hold true. Firstly the two

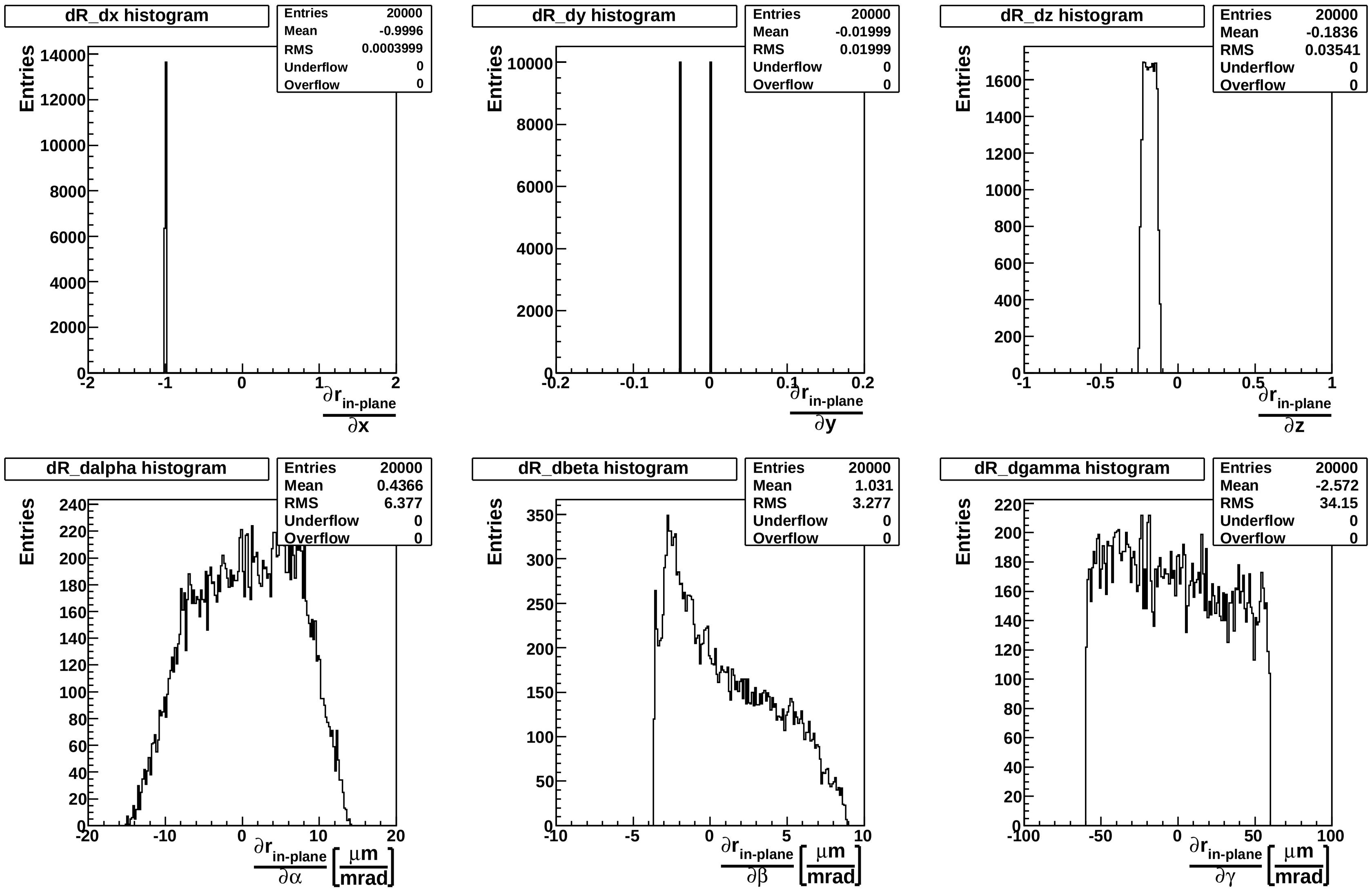

Figure 3.6: *Distribution of* $\frac{\partial r_{in-plane}}{\partial a}$ *for 10000 hits on both sides of the SCT module with top hat shaped* $r_{in-plane}$ *distribution.*

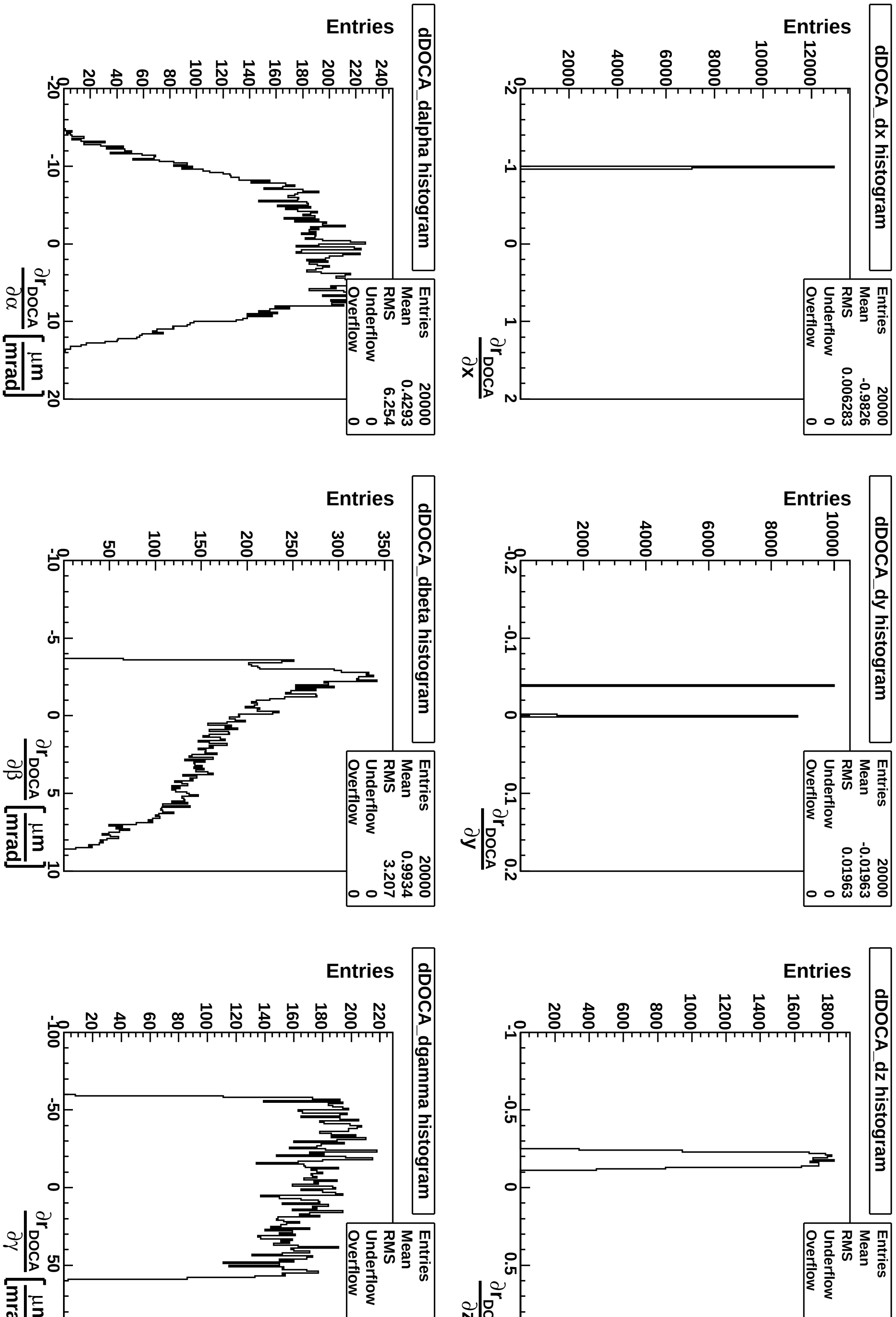

Figure 3.7: *Distribution of* $\frac{\partial r_{DOCA}}{\partial \alpha}$ *for 10000 hits on both sides of the SCT module with top hat shaped* $r_{DOCA}$ *distribution.*

readout sides of a SCT module are fixed with respect to each other. Together both sides constitute a module with only the six degrees of freedom of a rigid body and no internal degrees of freedom, like twisting or bending. And secondly, the r-$\phi$- and stereo-side residual measurements are uncorrelated. A scatter-plot showing the uncorrelated behavior of measurements from r-$\phi$- and stereo-side is shown in appendix B. Therefore the two measurements can be treated as independent input for the linearized $\chi^2$-minimization.

## 3.4 Tests and results

To test the statistical behavior of all components of the ROOT prototype program and to validate the error estimation and error propagation we repeated the complete simulation-reconstruction-alignment chain 500 times with nominal alignment. Each time straight line tracks were generated until 1000 hits were accumulated on both sides of the SCT module for a total of 2000 hits on the module[1]. The computed alignment parameters were filled into pull distribution histograms. This is shown for alignment with $r_{in-plane}$ in figure 3.8 and for alignment with $r_{DOCA}$ in figure 3.9 respectively.

The pull is defined as

$$Pull_{a_i} = \frac{Misalignment_{i\ Truth} - a_{i\ calculated}}{\sigma_{i\ calculated}}. \tag{3.7}$$

The error $\sigma_i$ of the calculated alignment parameter $a_i$ is computed according to equations 2.23 and 2.24 respectively.

If the error estimates for the alignment parameters are correct we expect a pull distribution with a RMS of 1 and if the alignment parameters are not biased the pull distribution should be centered at 0. In figures 3.8 and 3.9 we can see that both expectations are fulfilled for top hat shaped $r_{in-plane}$ and $r_{DOCA}$ distributions as input. For gaussian shaped $r_{in-plane}$ and $r_{DOCA}$ distributions as input the pull distributions are shown in appendix B and both expectations are fulfilled as well.

---

[1] That means a sample of one million hits was processed. For technical reasons this sample size could not be increased so more repetitions or more hits were not possible.

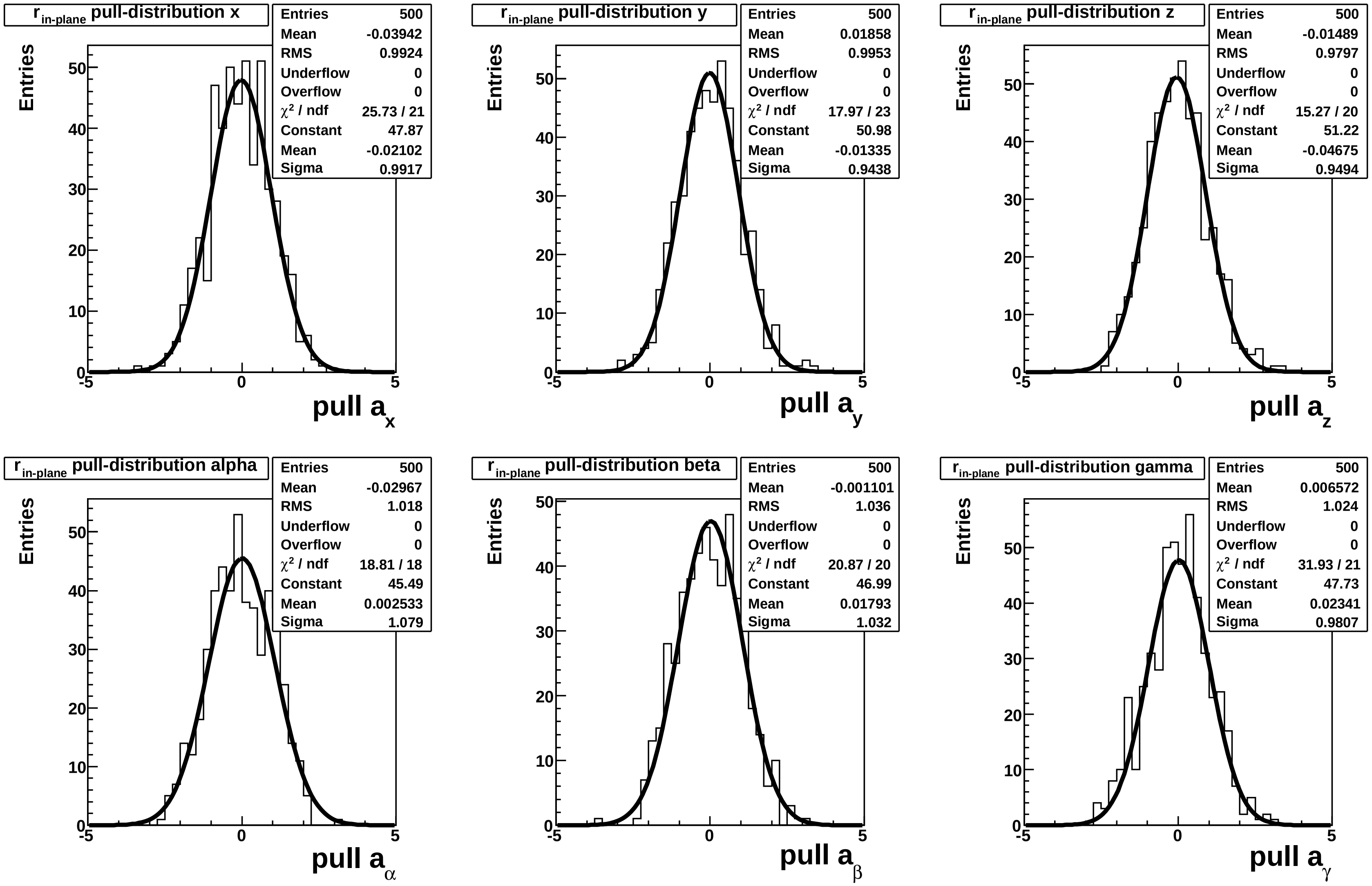

Figure 3.8: *Pull distributions for the six alignment parameters with top hat shaped $r_{in-plane}$ distribution.*

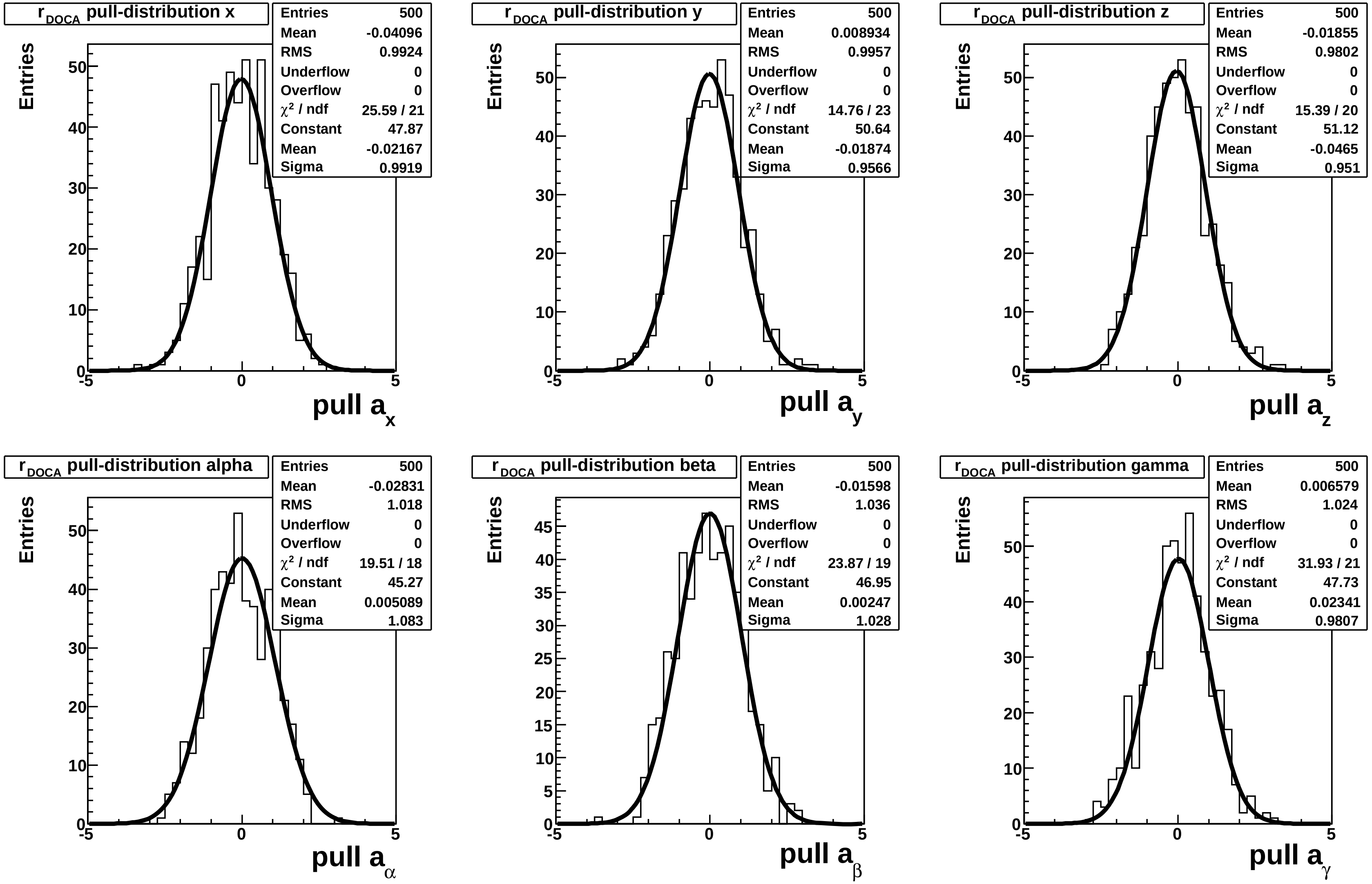

Figure 3.9: *Pull distributions for the six alignment parameters with top hat shaped* $r_{DOCA}$ *distribution.*

As the core of our alignment algorithm is a linearized $\chi^2$-minimization it is instructive to look at the $\chi^2$ distributions and the $\chi^2$ probability distributions, P($\chi^2$). In figure 3.10 these distributions are plotted with top hat shaped $r_{in-plane}$ and $r_{DOCA}$ distributions as input and in figure 3.11 with gaussian shaped residual distributions as input.

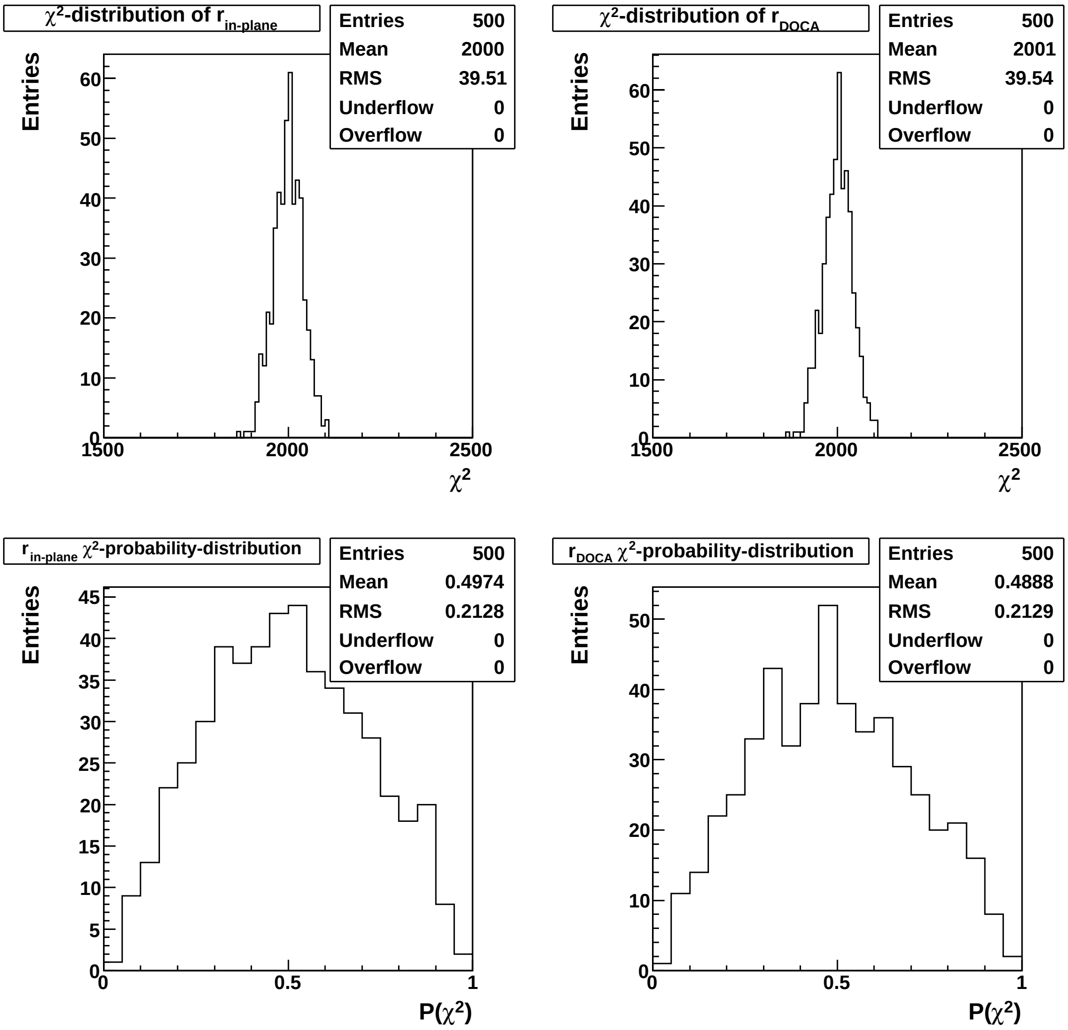

Figure 3.10: *$\chi^2$ and P($\chi^2$) distributions based on top hat shaped $r_{in-plane}$ and $r_{DOCA}$ distributions for 500 runs.*

The Number of Degrees Of Freedom ($n_{dof}$) of each $\chi^2$ is 2000 because of the 2000 residual measurements for each run. With gaussian shaped input distributions we expect for the mean $\mu$ of the $\chi^2$ distribution $\mu = n_{dof} = 2000$ and a standard deviation of $\sigma = \sqrt{2\ n_{dof}} \approx 63$ [47]. We also expect flat P($\chi^2$) distributions. We observe in figure 3.11 that our expectations for $\chi^2$ distributions based on gaussian shaped input are fulfilled. For top hat shaped input we observe for the $\chi^2$ distributions in figure 3.10 that $\mu = n_{dof}$ but that the RMS is smaller than $\sqrt{2\ n_{dof}}$. The corresponding P($\chi^2$) distributions are not flat but are shaped as expected for top hat shaped input [48].

By comparing the top hat shaped and the gaussian shaped input distributions (figures 3.4 and 3.5) we can explain these observations. For the top hat shaped residual distributions

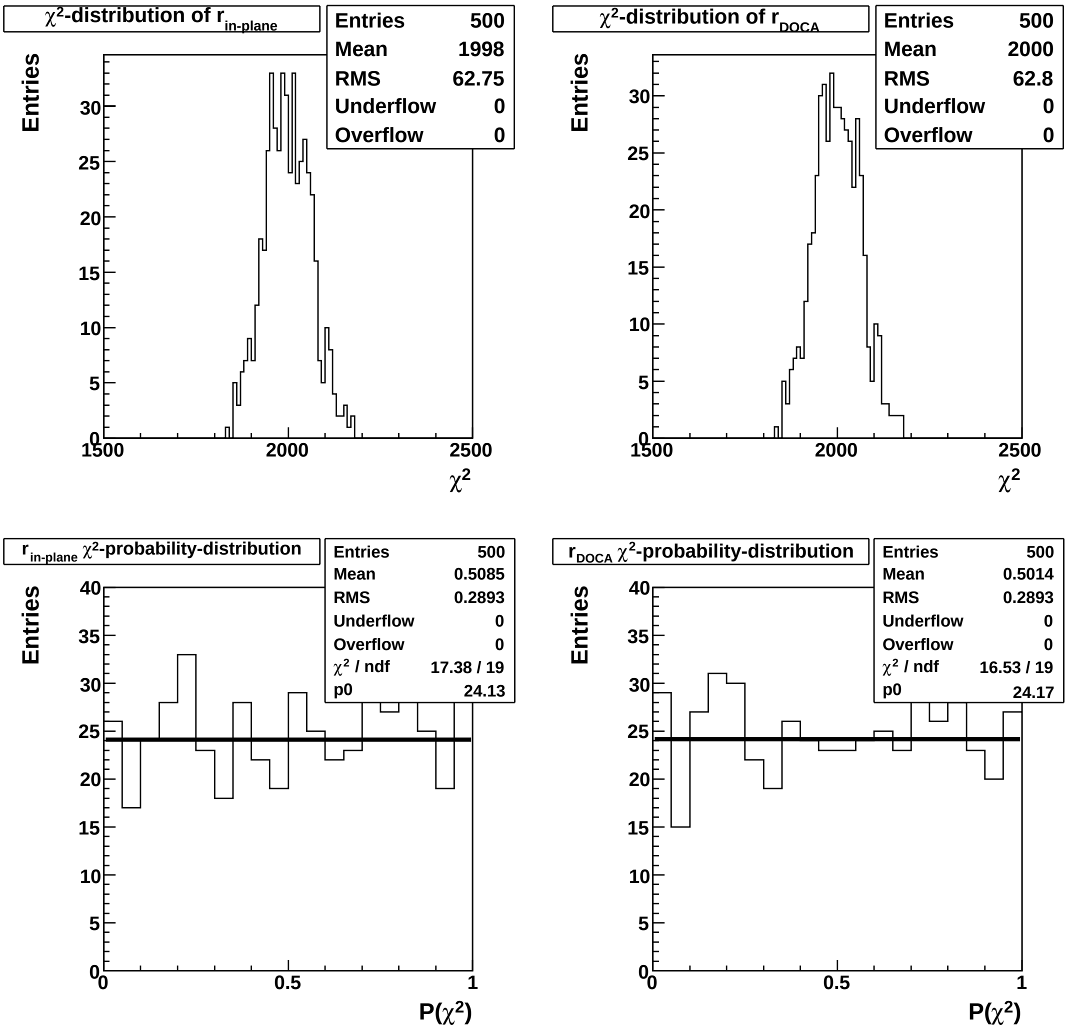

Figure 3.11: *$\chi^2$ and P($\chi^2$) distributions based on gaussian shaped $r_{in-plane}$ and $r_{DOCA}$ distributions for 500 runs.*

the probability for small residuals (and consequently for small $\chi^2$) is smaller than for gaussian shaped residual distributions. For the top hat shaped residual distributions the probability of residuals larger than half the strip pitch is zero and consequently large values of $\chi^2$ are suppressed. This is the reason why the $\chi^2$ distribution based on top hat shaped input has a smaller RMS than the one based on gaussian shaped input and why small and large values in the P($\chi^2$) distributions are suppressed and intermediate values more likely.

To validate that our alignment algorithm can recover misalignments we repeated the complete simulation-reconstruction-alignment chain 500 times with misalignments. As before, each time straight line tracks were generated until 1000 hits were accumulated on both sides of the SCT module for a total of 2000 hits on the module. We simultaneously misaligned all six degrees of freedom with the following misalignments: $\Delta$x = 50 $\mu$m, $\Delta$y = 200 $\mu$m, $\Delta$z = -200 $\mu$m, $\Delta\alpha$ = 5.0 mrad, $\Delta\beta$ = -5.0 mrad, $\Delta\gamma$ = 5.0 mrad. The resulting alignment parameter distributions for alignment with top hat shaped $r_{in-plane}$ and $r_{DOCA}$ are shown in figures 3.12 and 3.13.

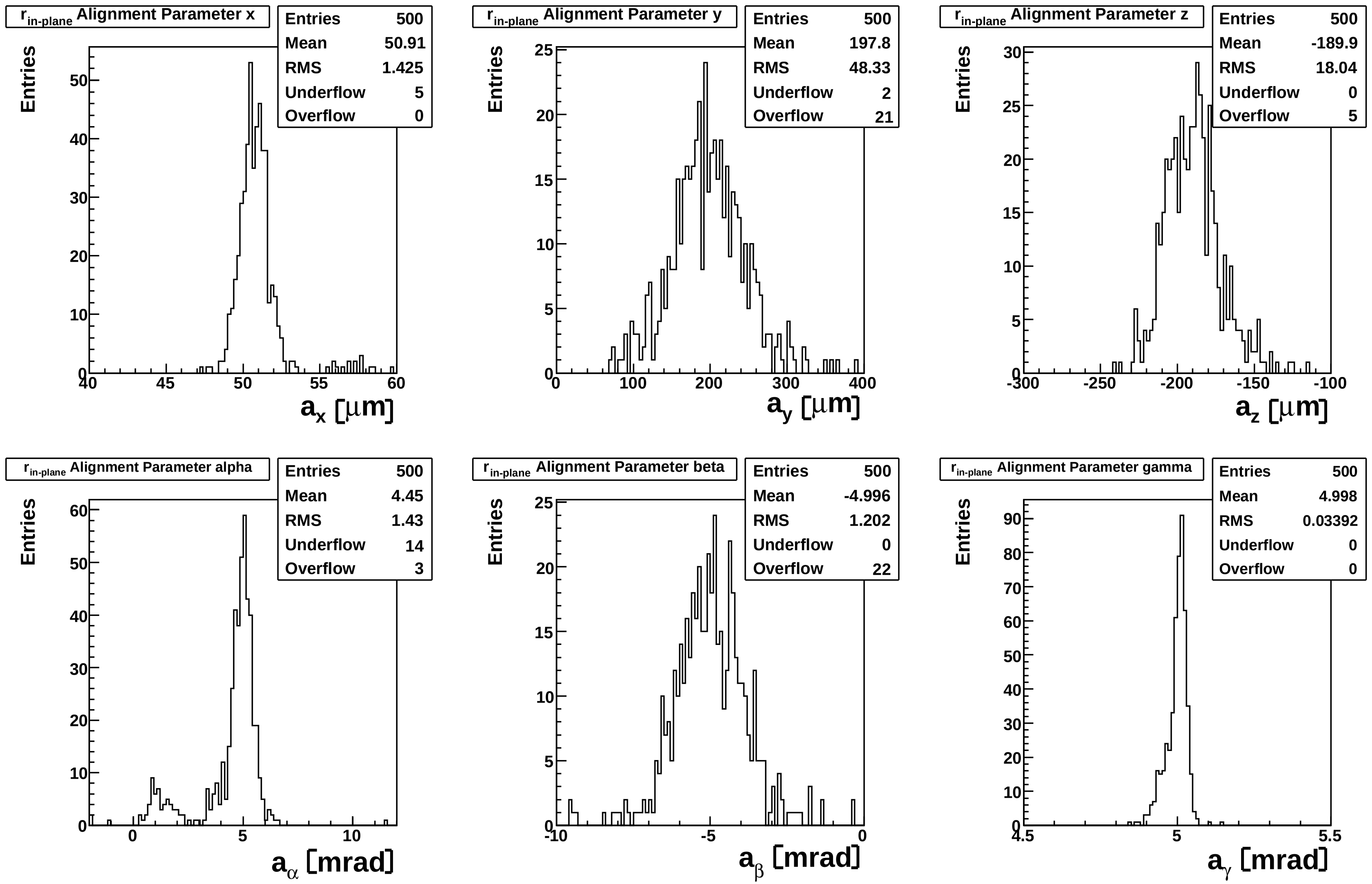

Figure 3.12: *Alignment parameter distributions for the six alignment parameters with top hat shaped* $r_{in-plane}$ *distribution and applied misalignment.*

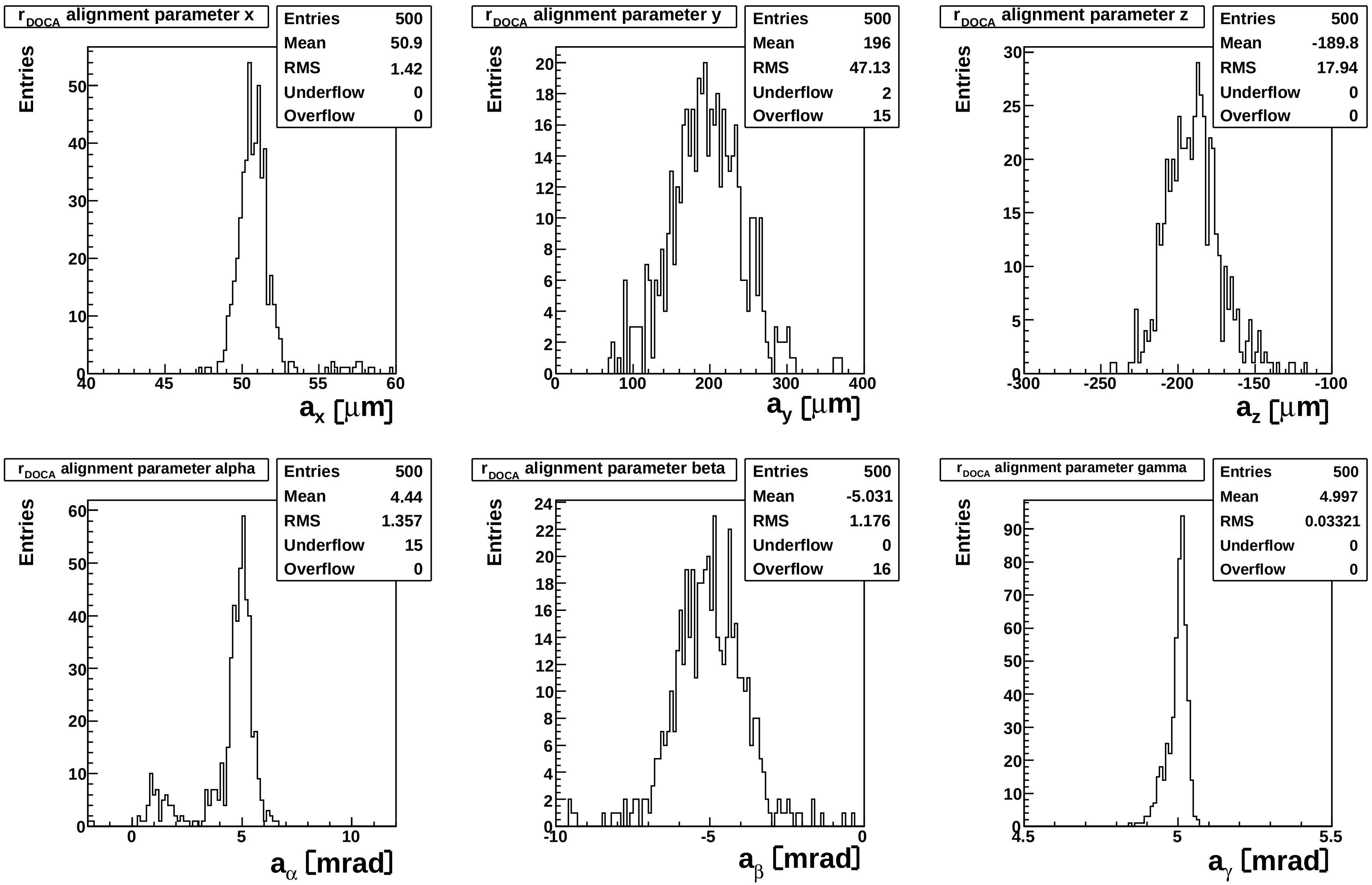

Figure 3.13: *Alignment parameter distributions for the six alignment parameters with top hat shaped* $r_{DOCA}$ *distribution and applied misalignment.*

From the RMS of the alignment parameter distributions in figures 3.12 and 3.13 we conclude that within the ROOT prototype program our alignment algorithm recovers the applied misalignments with an accuracy that meets the required alignment accuracy (see section 1.5 and table 1.4). However, there are tails in the alignment parameter distributions where the calculated alignment parameters are outside of the required alignment accuracy. The origin of the second peak in both $a_\alpha$ distributions is unknown.

## 3.5 Discussion

We successfully developed and implemented a prototype version of our proposed local $\chi^2$ alignment approach in ROOT. Within the prototype program we could test the alignment machinery in a controlled environment.

We validated that both definitions of residual – in-plane residual and distance of closest approach residual – are usable for our alignment approach and yield comparable alignment results. We chose to use the distance of closest approach residual for the implementation of our proposed local $\chi^2$ alignment approach within the ATLAS software framework Athena. There are two main arguments for this choice. Firstly, the need to handle two local coordinate frames to calculate $r_{in-plane}$ and especially the derivatives $\frac{\partial r_{in-plane}}{\partial a_i}$ is cumbersome. Secondly, because of the fanning angle of the readout strips of SCT end-cap modules (see section 1.4.2) the definition of $r_{in-plane}$ for SCT end-cap modules is not as simple as equation (3.1). Conversely, the prospect of having a homogeneous definition of residual for different types of detectors and the comparable performance of both definitions in the prototype program encouraged us to use $r_{DOCA}$ for the implementation of our proposed local $\chi^2$ alignment approach within Athena.

We validated that both top hat shaped and gaussian shaped residual distributions are suitable as input for our alignment approach and that errors are estimated and propagated correctly for both types of residual distributions. This result is of special importance for the implementation of our proposed local $\chi^2$ alignment approach within Athena. There the residual distributions are a convolution of a top hat shaped and gaussian shaped distributions (see 4.3.4).

We showed that our alignment algorithm is able to recover misalignments and that within the ROOT prototype program the achievable alignment accuracy fulfills the requirements stated in section 1.5. In general we can say that the proof of principle was successful and that we could establish and validate the full chain of track based detector alignment, from particle tracks to alignment parameters within the ROOT prototype program.

# Chapter 4

# Iterative local $\chi^2$ alignment in Athena

The bulk of this thesis was the implementation of our local $\chi^2$ alignment approach into the existing structure of the ATLAS software framework Athena.

## 4.1 Athena

All software activity for the ATLAS experiment is supported and connected by a common framework called Athena.

"All levels of processing of ATLAS data, from high-level trigger to event simulation, reconstruction and analysis, take place within the Athena framework; in this way it is easier for code developers and users to test and run algorithmic code, with the assurance that all geometry and conditions data will be the same for all types of applications (simulation, reconstruction, analysis, visualization)." [49]
An example of the interplay between reconstruction, detector description and visualization is shown in figure 4.1.

## 4.2 Coordinate systems

The global coordinate system we use in the following sections is the tracking frame. It is a righthanded three-dimensional orthogonal frame ($\hat{x}$,$\hat{y}$,$\hat{z}$) where the $\hat{z}$-axis is aligned with the magnetic field orientation of the central solenoidal magnetic field and the origin (0,0,0) is at the interaction point [41]. To denote directions in this frame we use the angular coordinates $\phi$ and $\theta$. $\phi$ is the azimuthal angle and $\theta$ the deflection angle with respect to the $\hat{z}$-axis. In addition to the angle $\theta$ we also use the pseudorapidity $\eta$ as defined in equation (1.1). The other coordinate system we use is the local frame of each SCT module. The local frame is a right-handed three-dimensional orthogonal frame (x,y,z) as well. The local frame is aligned with the r-$\phi$-side of the SCT module and is in accordance with the definition of the local frame in section 1.4.3.

The track parameters in Athena also have a global and a local representation.

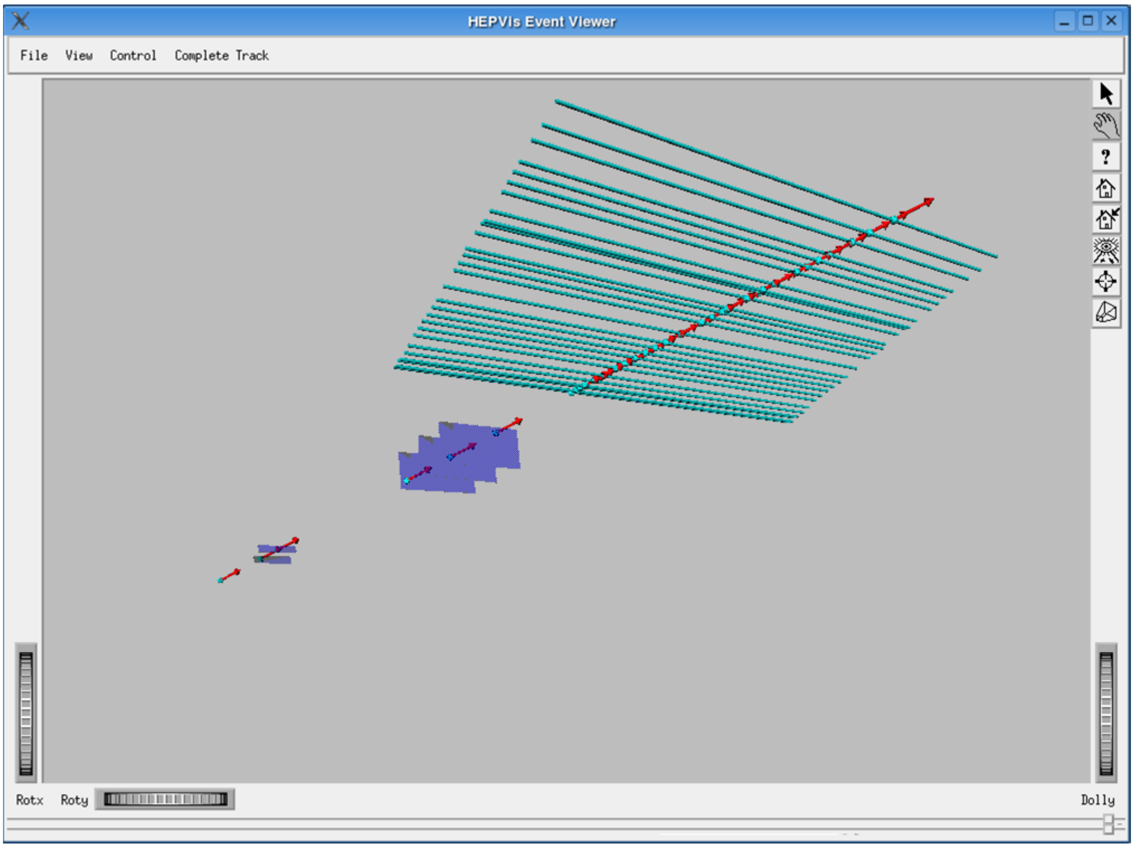

Figure 4.1: *HepVis is a three dimensional event display and an example for the interplay between different software components in Athena. Detector description software and reconstruction software provide the input for the visualization software HepVis. In the event display a single particle track coming from the interaction point can be seen. From left to right the track has hits on two pixel barrel modules, on three SCT barrel modules and on several TRT straws. The position track parameters on each surface are denoted by turquoise balls and the momentum track parameters by red arrows.*

$$\pi_{global} = (\vec{x}, \vec{p}, q) \tag{4.1}$$

Here $\vec{x}$ is the global position, $\vec{p}$ the global momentum and $q$ the charge of the track. This global representation is independent from the underlying track model. The local representation of the track parameters for a helix track is given by five parameters:

$$\pi_{local} = (l_1, l_2, \phi, \theta, \frac{q}{p}) \tag{4.2}$$

Here local means that the track parameters are defined on a surface and $(l_1, l_2)$ denote the two coordinates in the local frame of the surface. However, the three parameters $(\phi, \theta, \frac{q}{p})$ are a representation of the momentum in the global frame.

## 4.3 Implementation

The complexity and scale of the ATLAS experiment requires that Athena is modular, robust and flexible enough to meet the needs of the experiment throughout its operational lifetime. Therefore Athena is split into many packages where each package is dedicated to a narrow task.

Athena is an event-driven framework designed to execute a set of user-specified algorithms event by event. An event in Athena consists of all the data recorded or simulated for a triggered event, e.g. a bunch-crossing or an event with cosmic ray particles. A set of events is called a run. The algorithms are software programs within Athena that process and analyze the data of all events in a run. To run within Athena an algorithm has to implement three methods: `initialize()`, `execute()` and `finalize()`.

At the start of an Athena job, before the first event of the run is processed, Athena calls the `initialize()` methods of all user-specified algorithms. In its `initialize()` method each algorithm creates the instances of helper classes it needs, books histograms and ntuples and retrieves the helper tools (called AlgTools in Athena) it needs. During `initialize()` the framework itself launches common services that are needed by the algorithms, like histogramming- and ntuple-service, the detector description service or StoreGate. StoreGate is a transient datastore where all software components of an Athena job (Algorithms, AlgTools and Services) can deposit data that can then be accessed by the other components. StoreGate enables the data flow between the components of an Athena job. All alignment parameters and calibration constants for the ATLAS detector are stored in the conditions database. During `initialize()` a copy of the conditions database is either retrieved from an official database server or created from a local file that contains the relevant alignment parameters and calibration constants.

Once all `initialize()` methods are processed the run is started. Athena then loops through all events of a run and for each event calls the `execute()` methods of all user-specified algorithms. In their `execute()` methods the algorithms process the event data, for example do pattern recognition, track fitting and subsequent track-based detector alignment. The algorithms share information with each other via StoreGate.

At the end of the Athena job, after the last event of the run, Athena calls the `finalize()` method of each algorithm. In the `finalize()` method an algorithm produces the final results of the processed run and writes out histograms, ntuples or objects from StoreGate to file.

### 4.3.1 Chi2AlignAlg

Within Athena our local $\chi^2$ algorithm is called Chi2AlignAlg. It is an official part of the ATLAS software framework and located in the package InnerDetector/InDetAlignment/SiRobustAlign [50, 51].

The schematic diagram in figure 4.2 sketches the data flow during one Athena job with Chi2AlignAlg. Only a fraction of the complex network of software components of an Athena job is shown.

During `initialize()` Athena creates a transient copy of the conditions database. By transient copy we mean that the lifetime of this copy of the conditions database is only

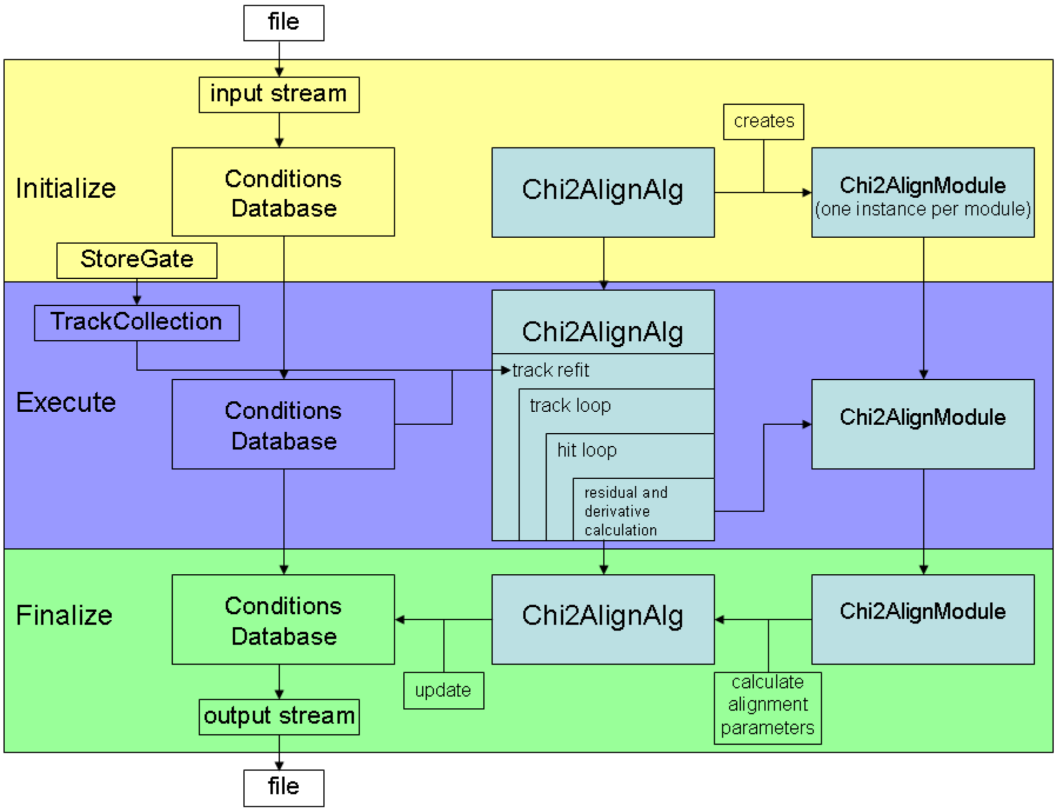

Figure 4.2: *A schematic overview of the data flow during one Athena job with Chi2AlignAlg. Details about the different software components are given in the text. Iterations are possible by inserting the alignment parameters from the output file of one job as input parameters for the conditions database of the next job.*

the duration of one Athena job. The conditions database supplies the initial alignment parameters that are needed by Chi2AlignAlg. In section 2.2 we discussed the role of initial alignment parameters. In its `initialize()` method Chi2AlignAlg creates multiple instances of a helper class called Chi2AlignModule. One instance for each SCT module is created. During the run each instance of Chi2AlignModule will accumulate the input information for alignment (residuals, residual errors and residual derivatives) for its SCT module.

In its `execute()` method Chi2AlignAlg retrieves a TrackCollection from StoreGate. A TrackColletion contains instances of the common track class in Athena (TrkTrack) [41]. The TrackCollection has been written to StoreGate by another algorithm possibly implementing a track selection. All tracks in the TrackCollection are then refitted. The technical details and reasons for the track refit are outlined in section 4.3.3. Chi2AlignAlg then loops through all refitted tracks and for each track loops through all hits and calculates residuals, residual errors and residual derivatives. For each SCT module this information is stored in its Chi2AlignModule instance.

After the run – during `finalize()` – each Chi2AlignModule instance calculates the alignment parameters for its SCT module with equation (2.22). Chi2AlignAlg then updates the conditions database with this new alignment parameters.

As outlined in section 2.3 our alignment approach is designed to run iteratively. The Athena EventLoopManager – the software component that loops through all events in a run – is only capable to loop through a run once. To iterate through the events of a run more than once a MultipleEventLoopManager was made available upon our request. However, the mechanism to iterate within one Athena job is not yet fully functional and under development. A workaround-solution for us was the following: It is possible during `initialize()` to create the transient conditions database from an input file and it is also possible during `finalize()` to write out the updated conditions database to an output file. By using the output file of a previous Athena alignment job as the input file of the next alignment job we were able to start an iteration cycle as sketched in figure 2.2.

### 4.3.2 Track sample

As input tracks for all subsequent studies and tests we used a sample of 400k simulated single pion events. The sample was simulated with Athena release 10.3.0 with a flat distribution both in $\phi$ and $\eta$ in the range $|\eta| < 5$. More details about the track sample can be found in [52]. The sample consists of 20k events each for $\pi^+$ and $\pi^-$ at ten different particle energies. The energies are 1 GeV, 3 GeV, 5 GeV, 10 GeV, 20 GeV, 50 GeV, 100 GeV, 200 GeV, 500 GeV, 1000 GeV. As discussed in section 2.2 we apply a momentum cut of $p \geq 10$ GeV to reduce the impact of multiple coulomb scattering on the track fit quality and thus to reduce correlated errors between SCT modules. After the momentum cut we are left with a sample of 280k single pion events. As the sample was simulated with a flat $\eta$-distribution only 140k events are within the acceptance region $|\eta| < 2.5$ of the SCT detector. An additional quality cut was imposed to get isolated tracks without wrongly associated hits: all events with secondary tracks from an hadronic interaction of the primary pion were rejected and not used for alignment. With these cuts we are left with a final sample of 132131 tracks that have hits in the SCT. We will call this the selected track sample.

With the selected track sample each SCT module receives a certain fixed number of hits. The distributions of the number of hits per module ($\mathrm{N}_{Hits}$) for all SCT barrel and end-cap modules are shown in figure 4.3. We observe that the mean of both distributions is about 260 hits per module. This is what we expect with 130k Tracks, eight SCT hits per track, i.e. two hits on four SCT modules and 4088 modules in total. The tails of both distributions are towards higher $\mathrm{N}_{Hits}$ which is a bias to the good, as more hits on a module reduce the statistical error of the alignment parameter estimation. The cross sections of SCT modules that are closer to the interaction point correspond to larger solid angles than the cross sections of modules that are further away. For example the SCT modules on barrel layer 0 receive almost twice as many hits than those on barrel layer 3.

It is noteworthy that the selected track sample and thus the number of hits per module is small. We expect that a track sample one order of magnitude larger is needed to achieve the required alignment accuracy (see section 1.5). Such a sample was not available for the Athena release we used for our studies, i.e. 10.5.0.

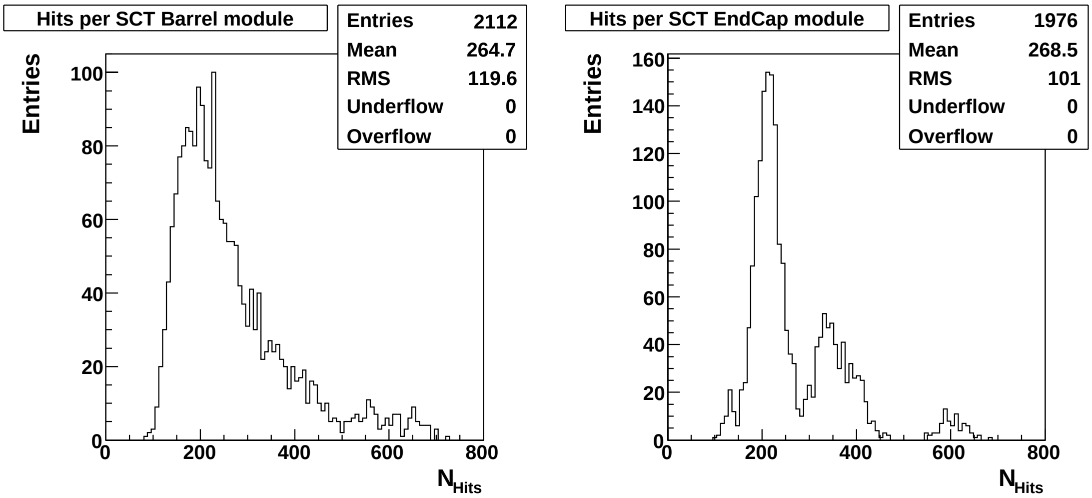

Figure 4.3: *Distributions of the number of hits ($N_{Hits}$) on SCT barrel and end-cap modules with the selected track sample.*

### 4.3.3 Track reconstruction

We did the initial reconstruction of the simulated single pion sample with Athena release 10.5.0. The reconstruction algorithm we used was xKalman.

"xKalman is a reconstruction package for the ATLAS Inner Detector. It by itself encapsulates a complete suite of all necessary Event Data Model classes, services, tools and algorithms to fully reconstruct tracks, do vertexing and more. It is therefore its own copy of the Inner Detector software. The package was born in ATRECON in fortran and the C++ code is still very much written in this style, making it in parts hard to read. Never the less, it is a very powerful package." [53]

The TrackCollection for each event that xKalman registered to StoreGate was persistified and written out to an ESD file.

"The Event Summary Data (ESD) should contain the detailed output of reconstruction, so that refinements of combined reconstruction (e.g. particle identification, track refitting, jet calibration) can be done from the ESD, following improvements of algorithm or of calibration/alignment." [54]

As stated above, ESD files are very well suited for our purpose. With the reconstruction information contained in the ESD file we can do track refitting without redoing the CPU-intensive pattern recognition.

We split the initial reconstruction of the single pion sample into 40 parallel Athena jobs that were processed at the lxbatch farm at CERN. Each of the 40 reconstruction jobs took eight hours to complete. This is to be compared with the time it takes to run one alignment job (i.e. one iteration) with the selected track sample. On a computer comparable to the computers in the lxbatch farm it takes 45 minutes for one Athena alignment job to finish.

As outlined above, pattern recognition for the single pion sample was done with xKalman. Before Chi2AlignAlg uses the tracks for alignment they are refitted with a KalmanFilter-TrkFitterTool [55]. There are two main reasons for the track refit. Firstly it serves to incorporate the updated alignment parameters into the track parameters. Secondly for the refit the track is stripped of its TRT hits and refitted as a silicon-only track, i.e. the track refit is done only with the hits from the Pixel and SCT subdetectors. We remove the TRT hits from the tracks to avoid a bias from the TRT subdetector on our alignment of the SCT detector.

The refitted tracks are then used by Chi2AlignAlg to obtain the input parameters for our alignment approach, that is residuals, residual errors and residual derivatives, discussed in turn in sections 4.3.4 - 4.3.6.

We obtained all the following results with Athena release 10.5.0 and Chi2AlignAlg from tag SiRobustAlign-00-00-24 of the SiRobustAlign package.

### 4.3.4 Residuals

A common definition of the residual for different types of detectors allows for a homogeneous, modular and reusable program code. Consequently, as discussed in section 3.5 we decided to use the distance of closest approach ($r_{DOCA}$) as a definition for the residual, since it has a common definition for SCT barrel modules and for all four types of SCT end-cap modules. An in-plane residual would need a different definition for SCT barrel modules and SCT end-cap modules due to the fanning angle of the readout strips on SCT end-cap modules. In addition the definition of $r_{DOCA}$ is flexible enough to be extended to other types of detectors, like the ATLAS Pixel detector.

Like in the ROOT prototype program $r_{DOCA}$ is calculated as the distance between a track and a straight line. The straight line represents the readout strip that is associated with the hit. Extrapolating the track parameters to the straight line (a StraightLineSurface in Athena terminology) with a TrkExtrapolationTool yields local track parameters at the point of closest approach of the track to the StraightLineSurface [56]. The local track parameters are given in the AtaStraightLine-representation where the two local coordinates $(l_1, l_2)$ are $(localR, localZ)$ and denote the radial distance from the straight line and the z-position along the line. More details on track extrapolation can be found in [57].

The radial distance *localR* is a signed quantity and thus complies exactly with our definition of $r_{DOCA}$.

$$r_{DOCA} = localR \tag{4.3}$$

The sign convention of *localR* is described in [41]. Distributions of $r_{DOCA}$ for SCT barrel modules are shown in figure 4.4 and distributions of $r_{DOCA}$ for SCT end-cap modules are shown in figure 4.5.

Not all hits in an SCT module are single strip hits. A certain percentage of hits are cluster hits. For tracks that originate from the interaction point 23% of all SCT barrel module hits and 1.7% of all SCT end-cap module hits are cluster hits. The difference between the number of SCT barrel module cluster hits and SCT end-cap module cluster hits is explained below. A cluster hit is produced if two or more neighboring SCT readout strips collect a charge above the discrimination threshold and produce a hit signal.

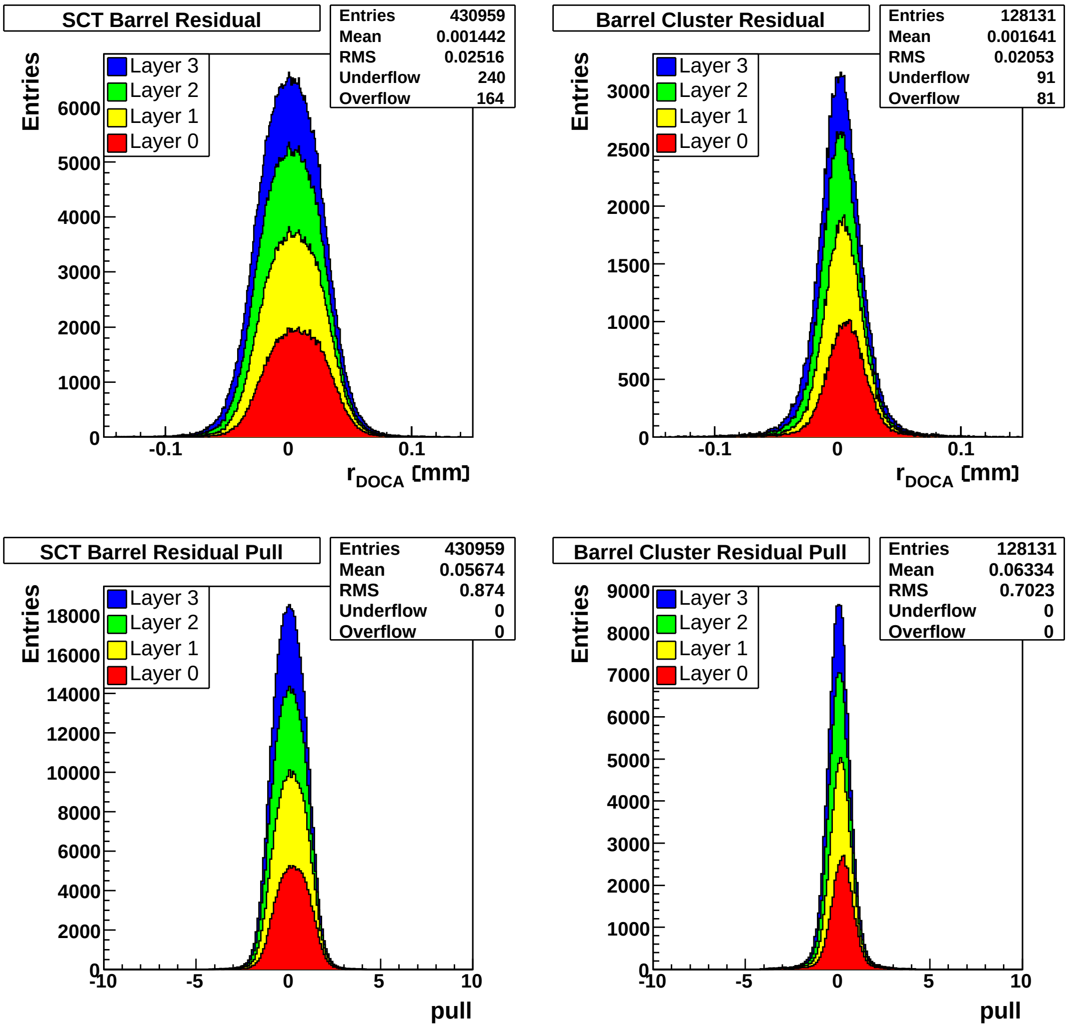

Figure 4.4: *Residual and pull distributions of single strip hits (left) and cluster hits (right) on all SCT barrel modules with refitted unbiased tracks.*

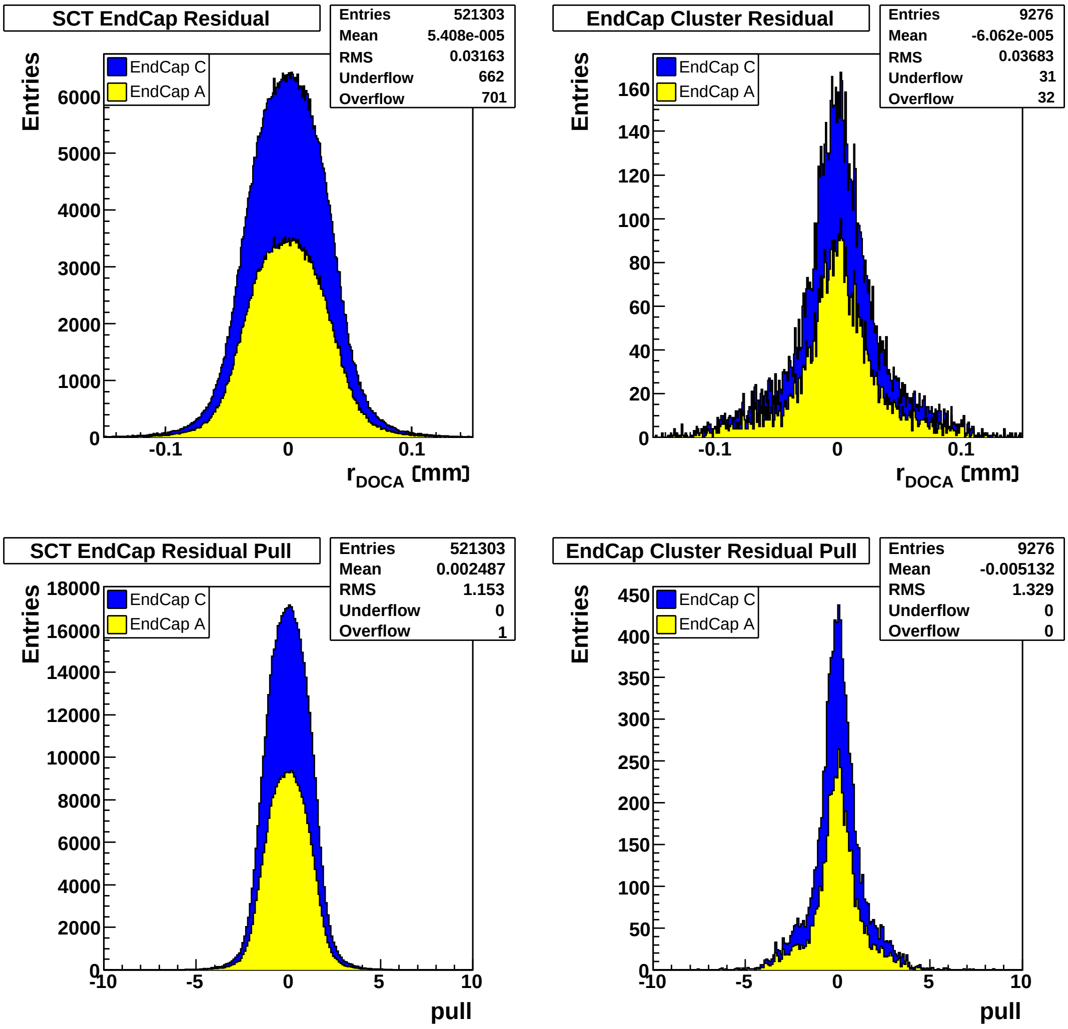

Figure 4.5: *Residual and pull distributions of single strip hits (left) and cluster hits (right) on all SCT end-cap modules with refitted unbiased tracks.*

During reconstruction the neighboring single strip hits are joined together and form a cluster hit. $r_{DOCA}$ distributions for SCT barrel module and SCT end-cap module cluster hits are shown in figures 4.4 and 4.5. The $r_{DOCA}$ distribution of two-strip cluster hits without any smearing from track fitting uncertainty is not top hat shaped like for single strip hits, but triangular shaped. The triangular shape comes from the convolution of two neighboring top hat distributions. To represent a cluster hit and to calculate $r_{DOCA}$ we use a StraightLineSurface as well. The straight line of a cluster hit runs parallel to the readout strips and through the center-of-gravity of the cluster hit.

The difference between the number of cluster hits in SCT barrel modules and SCT end-cap modules is a purely geometrical effect. The incident angle of tracks from the interaction point on the SCT module with respect to the most sensitive coordinate, i.e. the x-coordinate, is different for SCT barrel and SCT end-cap modules. Incident angle distributions for SCT barrel and SCT end-cap modules are shown in figure 4.6. The reason for the different incident angles is that to achieve an overlap of neighboring modules the SCT barrel modules are mounted with a 10° tilt at the surface of the cylindrical barrel structures [34]). Consequently, the normal vector of the SCT module plane – the z-coordinate in the local frame – is tilted by 10° with respect to a vector that emerges radially from the beamline. The path of a charged particle from the interaction point within the silicon diode crosses readout strip boundaries and the produced electrons drift towards neighboring readout strips. For SCT end-cap modules on the other hand the incident angle is 90° and the transverse component of the path of a charged particle, i.e. the movement along the local x-coordinate, is negligible. The probability of a cluster hit is suppressed accordingly. The incident angle with respect to the local y-axis only plays a minor role for cluster hit probability. The dependence of cluster hit probability on incident angles was studied experimentally [33] and our observations in the simulation are in accordance with the experimental results.

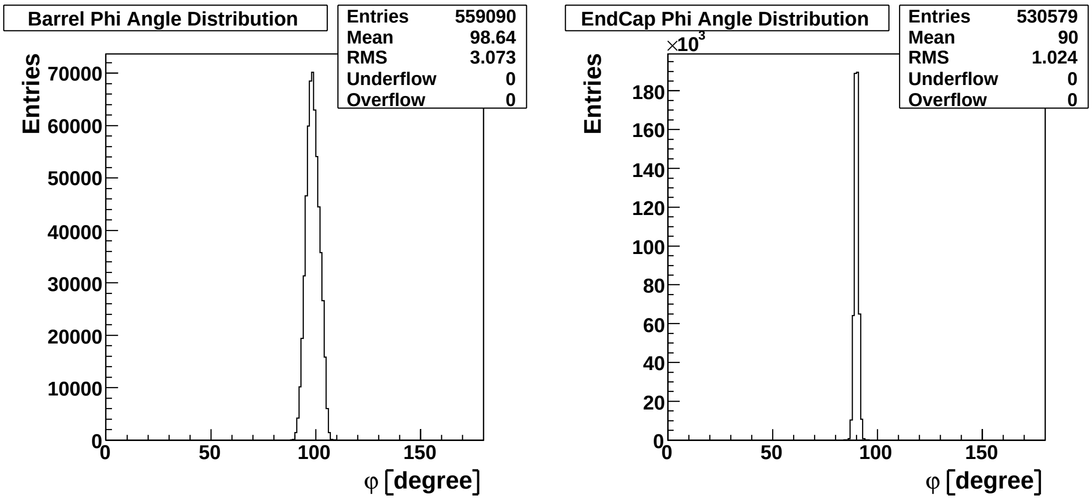

Figure 4.6: *$\varphi$ impact angle of the tracks on SCT modules with respect to the local x-coordinate.*

We obtained the residual distributions in figures 4.4 and 4.5 with unbiased track parameters as explained in section 2.2. To unbias the track parameters as described in section 2.2, i.e. to remove the hit information from the fit of the track parameters, we use the inverse Kalman update capability of the Athena KalmanUpdatorTool [58]. To remove a bias from the track parameters that was introduced by a misaligned SCT module it is in principle not sufficient to remove only the hit information for which we calculate the residual. If a track has a hit on the other side of the SCT module as well, this hit information must also be removed to get true unbiased track parameters, because the two readout sides are assumed to be fixed with respect to each other and thus correlated. The functionality to get true unbiased track parameters is not implemented in Chi2AlignAlg yet. We did all studies and tests with semi-unbiased track parameters where only one hit at a time was removed from the track.

In figures 4.4 and 4.5 the $r_{DOCA}$ distributions for single strip hits are not top hat shaped and the $r_{DOCA}$ distributions for cluster hits are not triangular shaped. This comes from the fact that the distributions are convoluted with a gaussian distribution that represents the errors of the track fit. As we saw in chapter 3 the linear least squares method works for top hat and gaussian distributed residuals as long as the residual distributions are unbiased and we estimate the residual error correctly [48].

To assess our estimate of the residual error the pull $= \frac{r_{DOCA}}{\sigma_{r\ DOCA}}$ is also shown in figures 4.4 and 4.5. We discuss the error estimation in detail in section 4.3.5.

If our estimate of the residual error is correct we expect pull distributions with a RMS of 1. This is not the case. For SCT barrel modules we seem to overestimate the residual error, hence the RMS of the distribution is smaller than 1. Whereas for SCT end-cap modules we underestimate the error and consequently the RMS is bigger than 1. The reason for this is a wrong track fit error from the KalmanFitter in release 10.5.0 [59, 60].

Another characteristic of the $r_{DOCA}$ distribution for SCT barrel modules in figure 4.4 is a visible shift of the distribution for SCT barrel layer 0 modules with respect to layers 1, 2 and 3. This is discussed in more detail in section 4.4.1.

Some of the tests in section 4.4 yield poor results. This is in large parts attributable to a degraded input. We consider this to be not an intrinsic flaw of Chi2AlignAlg. To explain what we mean with degraded input a comparison between residuals from unbiased silicon-only refitted tracks and residuals from original xKalman tracks (still biased and with TRT hits) is drawn. In figure 4.7 xKalman residuals for SCT barrel modules are shown. xKalman residuals for SCT end-cap modules are shown in appendix B.

Several things are noteworthy. Firstly, the residual distribution for barrel layer 0 is not shifted for xKalman tracks and we can conclude that the shift was introduced by the track refit. Secondly, as one can see from the "Entries" number in the statistics box, during the refit some tracks are lost. This is known to occur most frequently for tracks that have hits near the edge of a module which is exactly the overlap region between SCT modules which is of special importance for track based alignment. Thirdly, the number of outlier hits with large residuals is much reduced for original xKalman tracks compared to refitted tracks (as can be seen from the "Underflow" and "Overflow" entries in the statistics boxes of figures 4.4 and 4.7). The incorrect treatment of outlier hits during track refit is known [61] and superior track refit tools are under development.

The three issues highlighted here all propagate through the alignment machinery of Chi2AlignAlg and degrade the calculated alignment corrections.

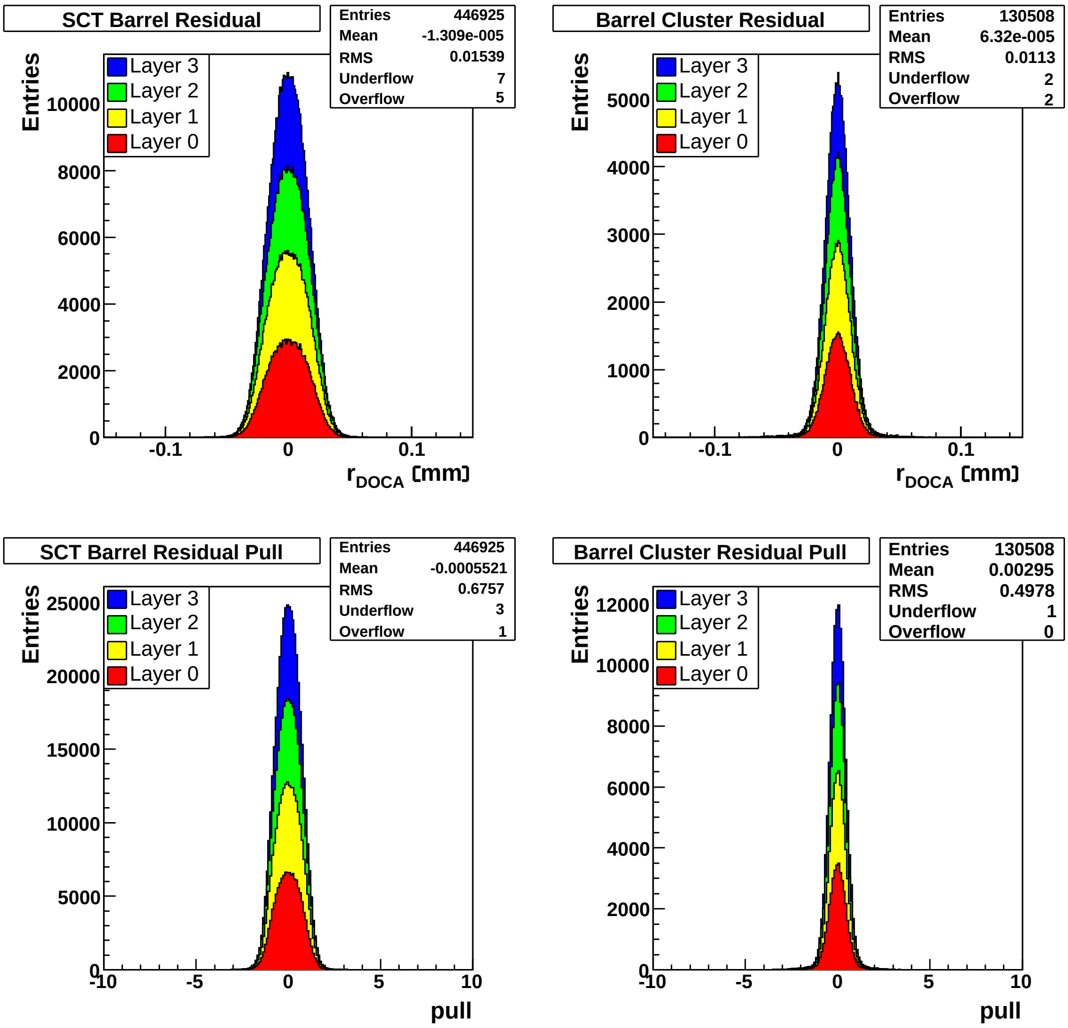

Figure 4.7: *Residual and pull distributions of single strip hits and cluster hits on all SCT barrel modules with xKalman tracks.*

### 4.3.5 Residual error estimates

We calculate the error $\sigma_{r\ DOCA}$ of $r_{DOCA}$ as follows:

$$\sigma^2_{r\ DOCA} = \sigma^2_{Hit} + \sigma^2_{Track} \tag{4.4}$$

Here $\sigma_{Hit}$ denotes the error component of $r_{DOCA}$ that comes from the finite resolution of the SCT and $\sigma_{Track}$ denotes the fit error of the local track parameter *localR*. The distributions for $\sigma_{Track}$ and $\sigma_{Hit}$ are shown in figure 4.8.

As in the ROOT prototype program we calculate $\sigma_{Hit}$ as

$$\sigma_{Hit} = \frac{strip\ pitch}{\sqrt{12}} \sin(\varphi) \tag{4.5}$$

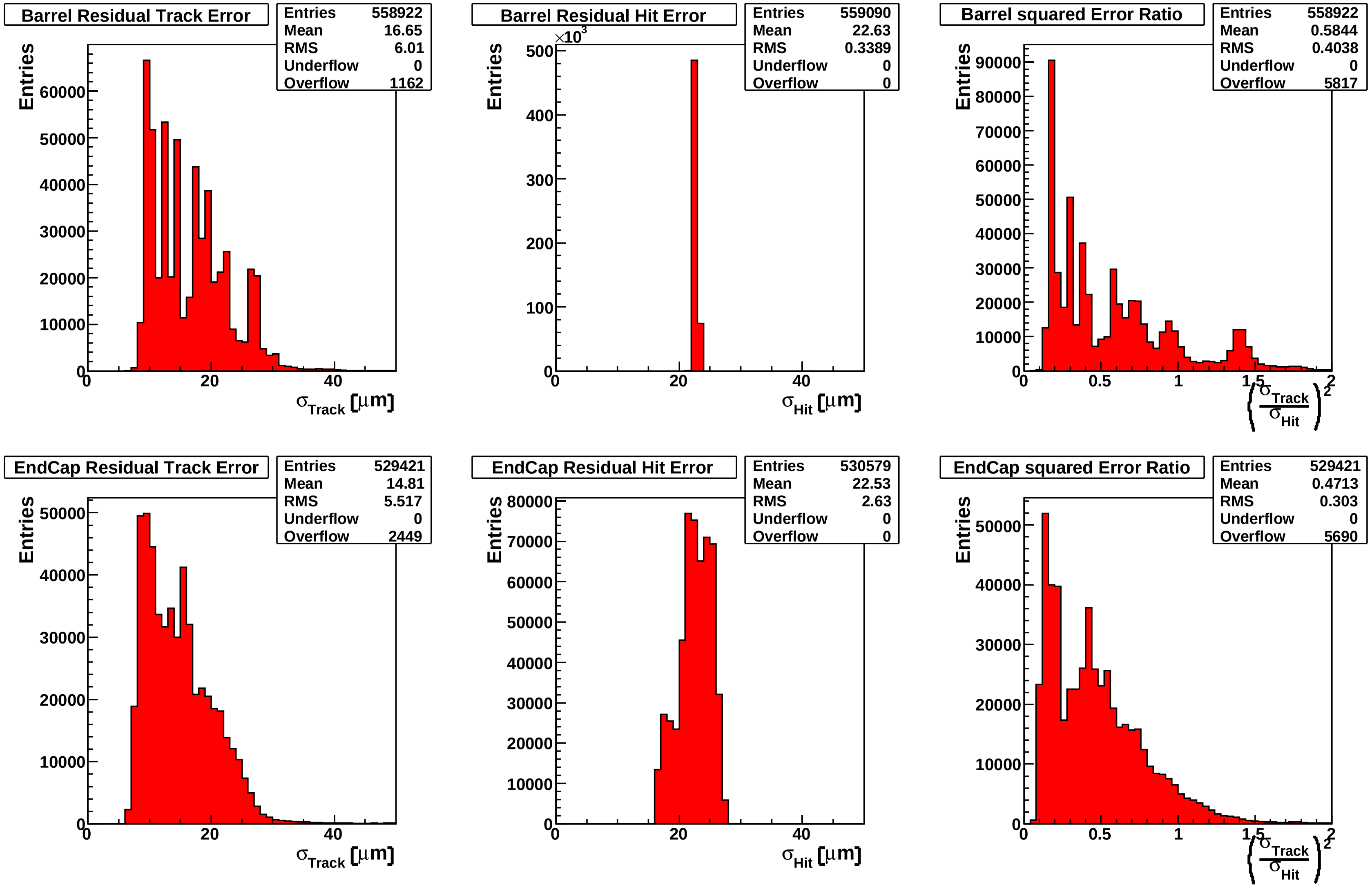

Figure 4.8: *Distributions of the different error components of* $r_{DOCA}$ *for SCT barrel modules (upper plots) and SCT end-cap modules (lower plots).* $\sigma_{Track}$ *is the track fitting error and* $\sigma_{Hit}$ *is the error contribution from the finite resolution of the SCT. The squared error ratio is a measure how correlated with other measurements the* $r_{DOCA}$ *measurement is.*

where $\varphi$ is the incident angle with respect to the local x-coordinate (see figure 4.6).

Due to the fanning angle of the readout strips of SCT end-cap modules the strip pitch is not constant, but a function of the local y-coordinate. The varying strip pitch of SCT end-cap modules can be seen in figure 4.8. To calculate the correct strip pitch we use the impact point of the track to get the local y-coordinate along the strip.

To calculate $\sigma_{Hit}$ for cluster hits we also use equation (4.5). This is not correct as equation (4.5) only holds for top hat shaped distributions. The correct handling of cluster hits and especially cluster hit $r_{DOCA}$ errors in Chi2AlignAlg is still under development.

$\sigma_{Track}$ is provided by the TrkExtrapolationTool. The track extrapolation not only propagates track parameters to a local surface but also the covariance matrix of the track parameters. The track fit error of *localR* is then obtained from the local covariance matrix. The substructure of the $\sigma_{Track}$ distribution in figure 4.8 comes – among other things – from tracks with different numbers of hits, from tracks with overlap hits, i.e. hits on two SCT modules in the same layer, from tracks with outlier hits or from tracks with different momenta. This substructure and its impact on the performance of Chi2AlignAlg is still under investigation.

Also shown in figure 4.8 is the ratio $\frac{\sigma^2_{Track}}{\sigma^2_{Hit}}$. In section 2.2 we identified this ratio as a measure of the correlation between modules. The value of this ratio is an indication wether the $r_{DOCA}$ measurement is dominated by the uncorrelated finite hit resolution of the SCT or by correlated track fit errors.

### 4.3.6 Residual derivatives

Residual derivatives with respect to the six degrees of freedom for SCT barrel modules are shown in figure 4.9 and for SCT end-cap modules in figure 4.10. We calculate the derivatives numerically with the difference quotient from equation (3.6). The coordinates (x,y,z) denote the coordinates in the local frame and ($\alpha$,$\beta$,$\gamma$) are the angles of rotation around (x,y,z) respectively. We already discussed in section 3.3.2 that $\frac{\partial r_{DOCA}}{\partial a_i}$ is a direct measure of the sensitivity of track based detector alignment to constrain the alignment parameter $a_i$.

As we simulated a SCT barrel module in our ROOT prototype program figure 4.9 can be directly compared to figure 3.7. The sensitive coordinate, i.e. local x, is constrained the most by track based detector alignment. Compared to this the local y-coordinate is only weakly constrained as only hits on the stereo-side of a SCT barrel module contribute a non-zero $\frac{\partial r_{DOCA}}{\partial y}$ (see section 4.4.1). Due to the tilt angle of 10° some of the sensitivity of the local x-coordinate is transferred to the z-coordinate. The $\frac{\partial r_{DOCA}}{\partial z}$ distribution directly reflects the incident angle distribution for SCT barrel modules in figure 4.6. The flipped sign of $\frac{\partial r_{DOCA}}{\partial x}$ and $\frac{\partial r_{DOCA}}{\partial z}$ in figure 4.9 compared to figure 3.7 is due to an opposite sign convention for $r_{DOCA}$ in the ROOT prototype program and in Chi2AlignAlg in Athena. The dip at zero for the $\frac{\partial r_{DOCA}}{\partial \gamma}$ distribution in figure 4.9 has a geometrical reason. To yield a small $\frac{\partial r_{DOCA}}{\partial \gamma}$ value the impact point of a track has to be very close to the center of rotation, i.e. the center of gravity of the r-$\phi$-side. The surface area and thus the hit probability decreases for smaller distances between impact point and center of rotation. Again due to the tilt angle this behavior is transferred in part to the $\frac{\partial r_{DOCA}}{\partial \alpha}$ distribution. Of course this argument begs the questions why this was not observed in the ROOT prototype program. This discrepancy between the prototype program and the implementation of Chi2AlignAlg in Athena is not understood. The asymmetry of $\frac{\partial r_{DOCA}}{\partial \beta}$ is also a consequence of the tilt angle of 10°. If the tilt angle was -10° the $\frac{\partial r_{DOCA}}{\partial \beta}$ distribution would be flipped.

In figure 4.10 the $\frac{\partial r_{DOCA}}{\partial a_i}$ distributions of all SCT end-cap modules are shown. The two peaks in the $\frac{\partial r_{DOCA}}{\partial x}$ distribution come from modules on end-cap A (peak at +1) and modules on end-cap C (peak at -1). The $\frac{\partial r_{DOCA}}{\partial y}$ distribution of SCT end-cap modules compared to the $\frac{\partial r_{DOCA}}{\partial y}$ distribution of SCT barrel modules shows the gain in sensitivity along the local y-coordinate due to the fanning angle of the readout strips. The $\frac{\partial r_{DOCA}}{\partial z}$ distribution reflects the incident angle distribution for SCT end-cap modules (see figure 4.6). As the mean incident angle is 90°, the local z-coordinate is only weakly constrained by track based detector alignment with tracks from the interaction point (see section 4.4.1). The incident angle of 90° also explains the small values for $\frac{\partial r_{DOCA}}{\partial \alpha}$ and $\frac{\partial r_{DOCA}}{\partial \beta}$. These rotational degrees of freedom of SCT end-cap modules are only weakly constrained by track based alignment compared to SCT barrel modules as there is no tilt angle and thus no sensitivity is transferred from the x-coordinate (see section 4.4.1).

The substructure in the $\frac{\partial r_{DOCA}}{\partial a_i}$ distributions in figure 4.10 comes from the four different types of SCT modules used in the end-caps and from the module configurations on the end-cap disks. This substructure and its impact on the performance of Chi2AlignAlg is under investigation.

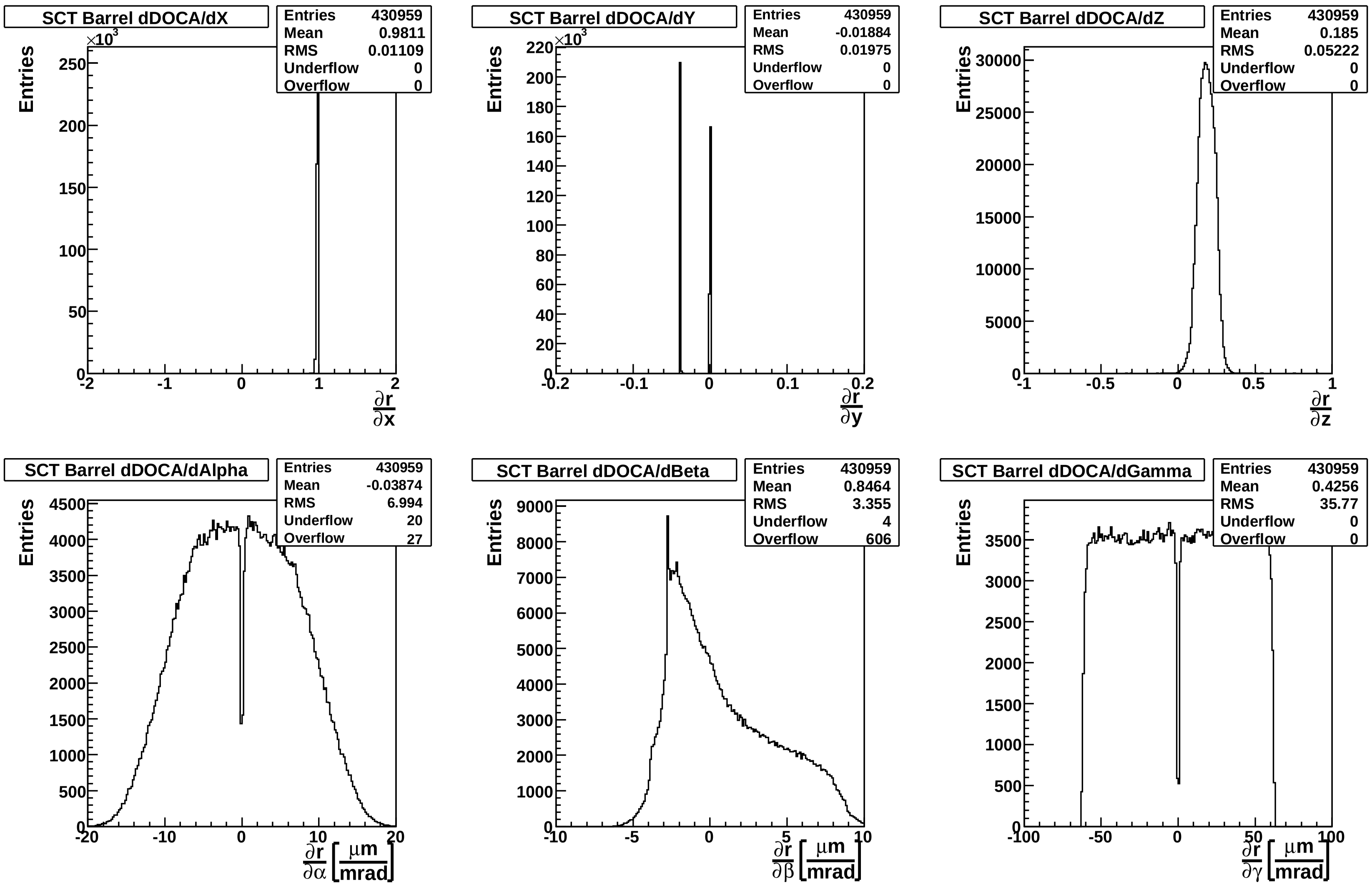

Figure 4.9: *Distributions of the residual derivatives* $\frac{\partial r_{DOCA}}{\partial a_i}$ *for all SCT barrel modules The coordinates (x,y,z) denote the local coordinate axes.* $(\alpha,\beta,\gamma)$ *denote rotation angles around these axes.*

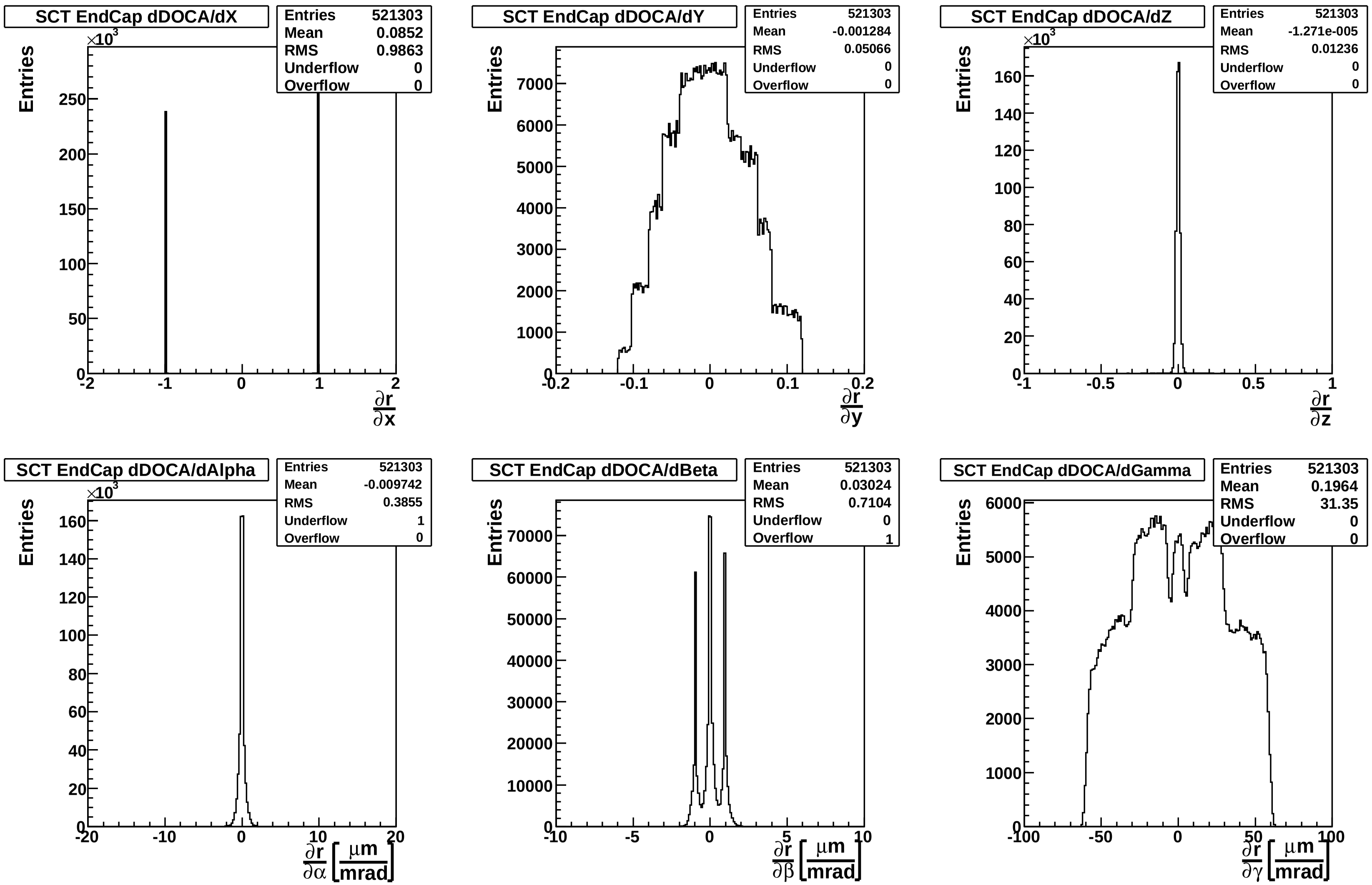

Figure 4.10: *Distributions of the residual derivatives* $\frac{\partial r_{DOCA}}{\partial a_i}$ *for all SCT end-cap modules The coordinates (x,y,z) denote the local coordinate axes.* ($\alpha$,$\beta$,$\gamma$) *denote rotation angles around these axes.*

## 4.4 Alignment tests and results

To test the performance of the Chi2AlignAlg – especially the convergence after a few iterations – several tests with nominal ATLAS SCT setup and various misaligned setups were done.

### 4.4.1 Tests with nominal alignment

All figures in sections 4.3.4 - 4.3.6 were done with the nominal ATLAS detector geometry we described in section 1.4.2. Many features of Chi2AlignAlg can be tested with nominal alignment. Also, the performance of Chi2AlignAlg with nominal alignment serves as a reference for tests with misaligned detector geometries.

In the following sections we will always associate the residual-, alignment parameter-, and pull-distributions to the Athena job, i.e. the iteration, they were produced in. For example the initial residual distribution of all SCT modules on barrel layer 0 in figure 4.11 is called "from iteration 1".

We already stated in section 4.3.4 that there is a manifest shift of 6 $\mu$m of the residual distribution of barrel layer 0. This shift is an artifact of release 10.5.0 and comes from a wrong handling of the Lorentz shift in the barrel pixel modules [62]. This is a nice example of a perceived misalignment due to correlation effects between modules that we ignore completely during one iteration (see section 2.2 and figure 2.1).

We stated in section 4.3.3 that xKalman is "its own copy of the Inner Detector software". xKalman does not rely on an Athena tool to handle the Lorentz shift, but corrects for it internally. This is the reason why the xKalman residual distributions are not affected by this software bug as we saw already in section 4.3.4 (see figure 4.7, for reference xKalman residual distributions for barrel layer 0 are shown in appendix B).

Our approach does not take correlation effects into account, that is they are not corrected for, and Chi2AlignAlg does not do pixel detector alignment yet. Consequently, the 6 $\mu$m shift is perceived as a real misalignment and thus removed as can be seen in figure 4.12 where the residual distribution of barrel layer 0 from iteration 2 is shown. The 6 $\mu$m shift is reduced to 2 $\mu$m. During subsequent iterations the alignment parameters for all SCT modules are adjusted such that the residual distributions are centered at 0 and with the smallest possible rms. This is shown for SCT barrel modules in figure 4.13 - 4.15 and for SCT end-cap modules in figures 4.16 - 4.18. We observe that through the iterations some single strip hits are misidentified as cluster hits. Due to the internal handling of single strip hits and cluster hits we estimate that this does not influence the performance of Chi2AlignAlg. Nonetheless it is an indication of possibly incorrect detection and treatment of cluster hits within Chi2AlignAlg.

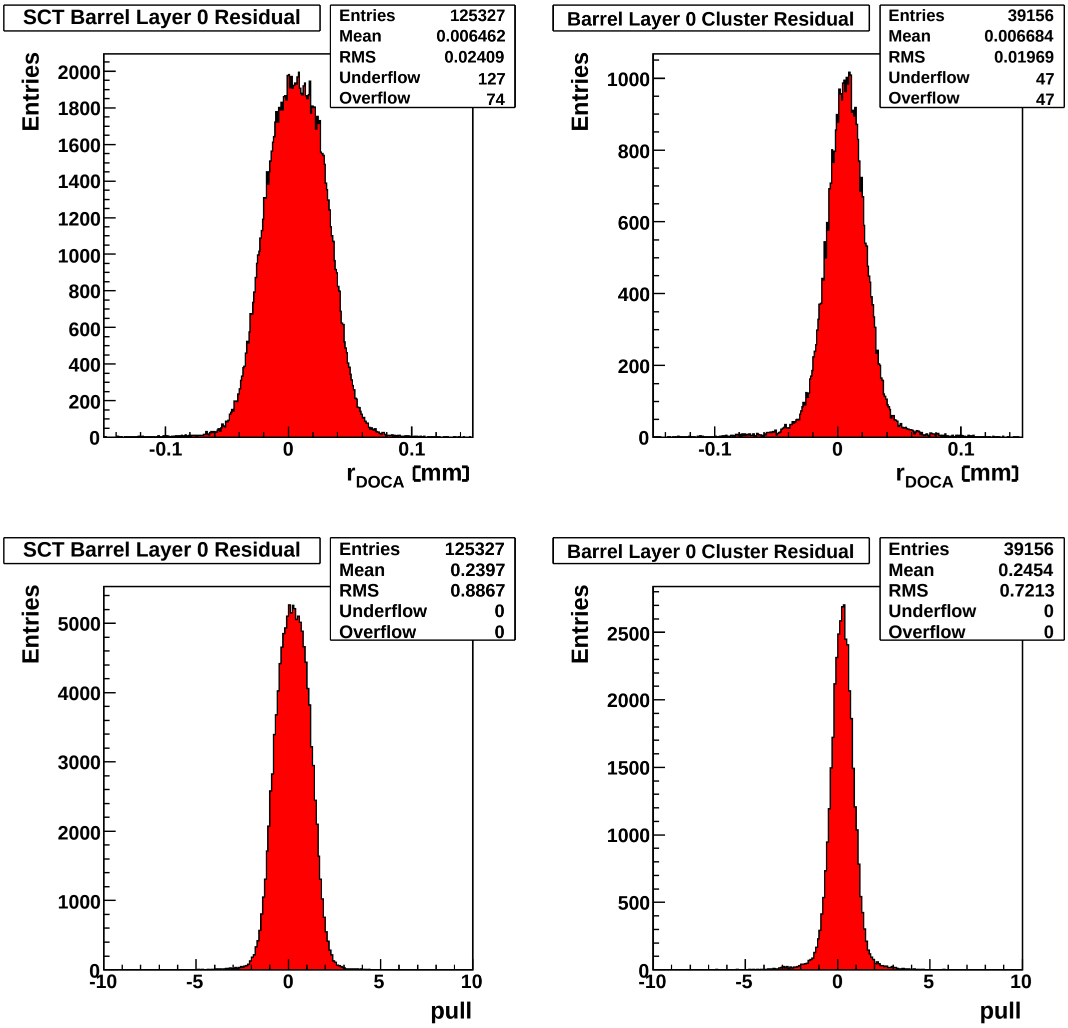

Figure 4.11: *Residual and pull distributions from iteration 1 of single strip hits and cluster hits on all SCT modules of barrel layer 0 with refitted unbiased tracks. The 6 $\mu$m shift is clearly visible.*

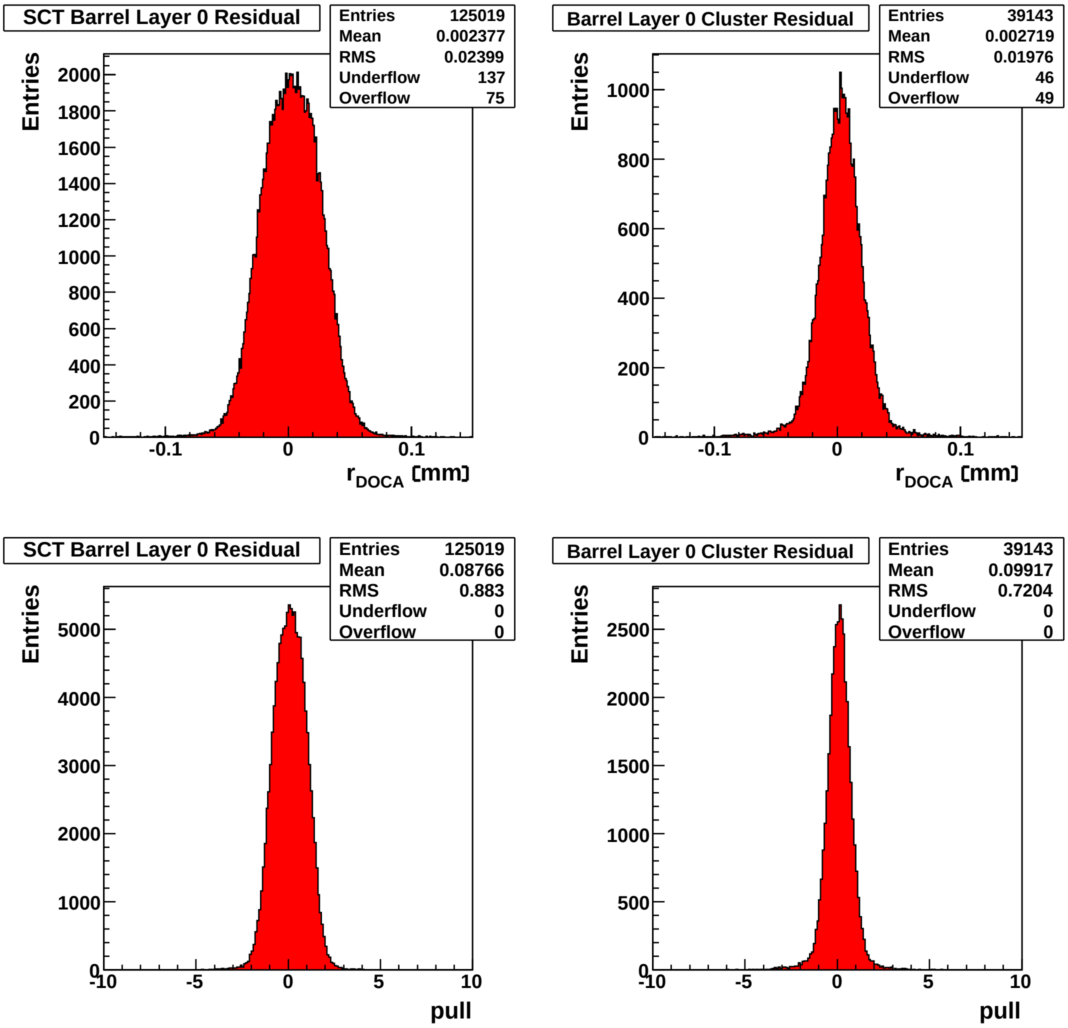

Figure 4.12: *Residual and pull distributions from iteration 2 of single strip hits and cluster hits on all SCT modules of barrel layer 0 with refitted unbiased tracks. The shift is reduced to 2* $\mu m$*.*

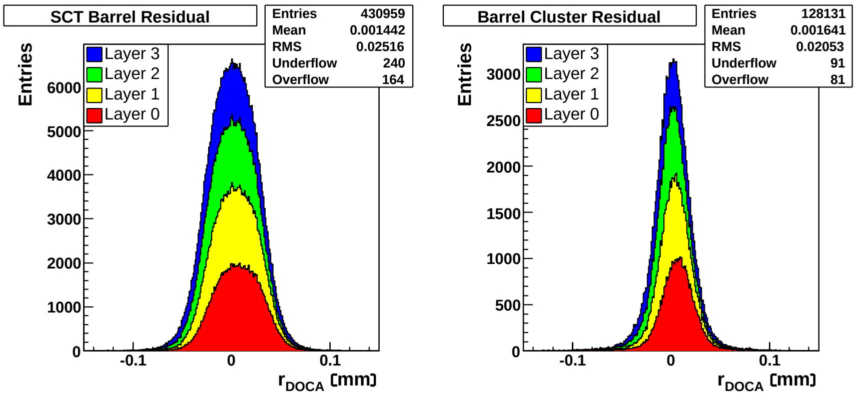

Figure 4.13: *Residual distributions from iteration 1 of single strip hits and cluster hits on all SCT barrel modules with refitted unbiased tracks.*

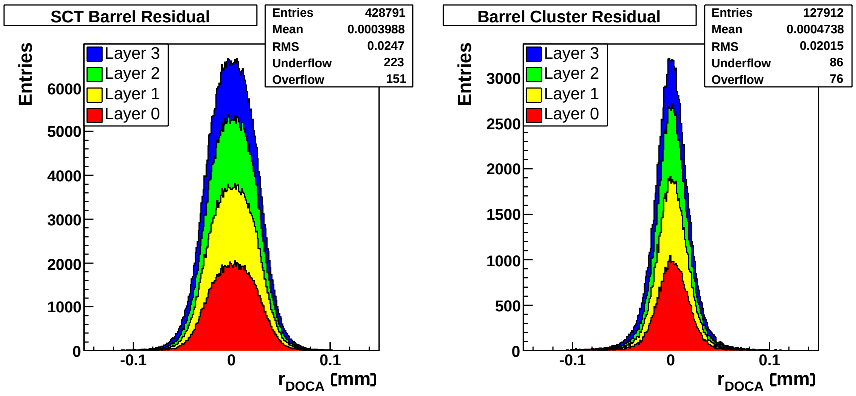

Figure 4.14: *Residual distributions from iteration 5 of single strip hits and cluster hits on all SCT barrel modules with refitted unbiased tracks.*

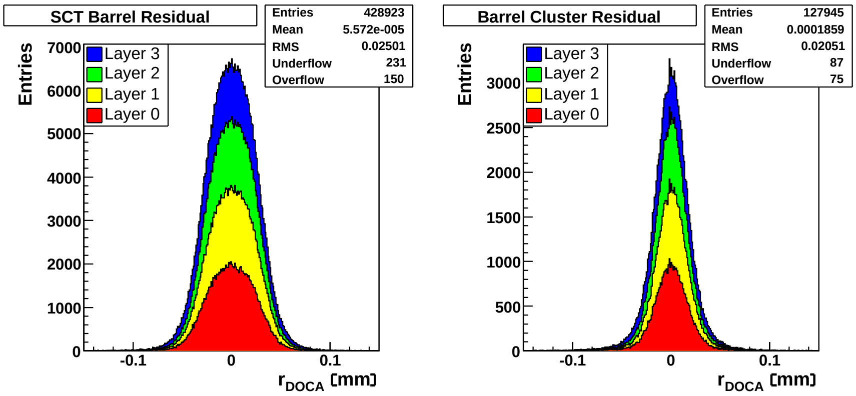

Figure 4.15: *Residual distributions from iteration 10 of single strip hits and cluster hits on all SCT barrel modules with refitted unbiased tracks.*

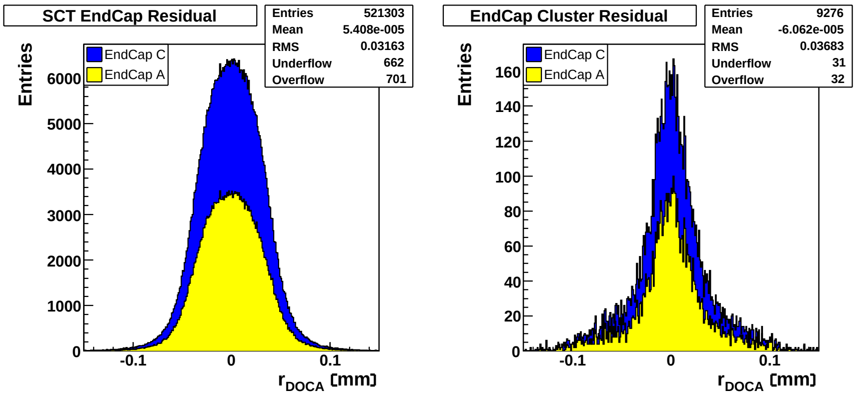

Figure 4.16: *Residual distributions from iteration 1 of single strip hits and cluster hits on all SCT end-cap modules with refitted unbiased tracks.*

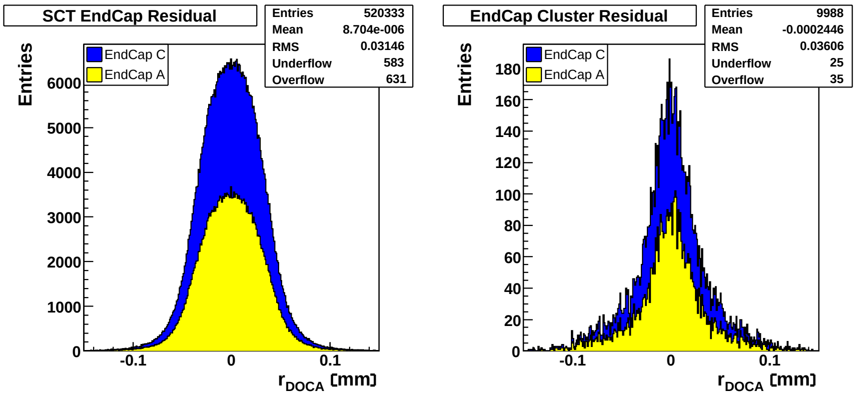

Figure 4.17: *Residual distributions from iteration 5 of single strip hits and cluster hits on all SCT end-cap modules with refitted unbiased tracks.*

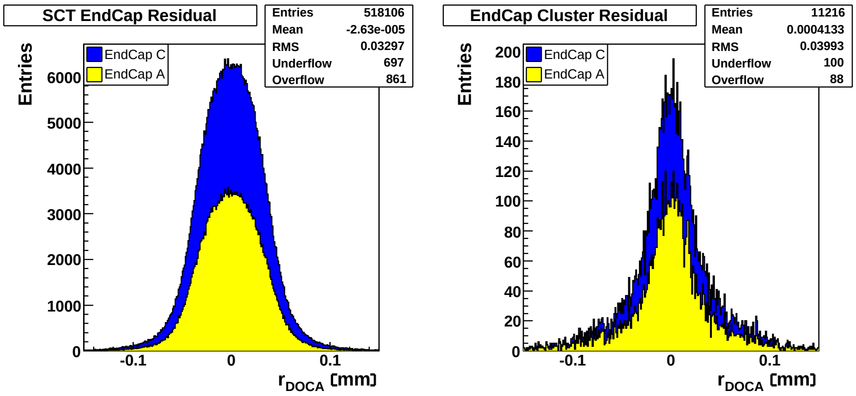

Figure 4.18: *Residual distributions from iteration 10 of single strip hits and cluster hits on all SCT end-cap modules with refitted unbiased tracks.*

As for the ROOT prototype program we determine the pull distributions for the alignment parameters with equation (3.7). For iteration 1 the pull distributions for SCT barrel modules are shown in figure 4.19 and for SCT end-cap modules in figure 4.20. In figures 4.19 and 4.20 the pull of all 2112 SCT barrel modules and all 1976 SCT end-cap modules is shown. These pull distributions are comparable to figure 3.9 in the same sense as the canonical average is comparable to the time average in statistical mechanics. In Athena we run 2112 alignment jobs for SCT barrel modules in parallel whereas in the ROOT prototype program we ran 500 alignment jobs sequentially.

Having said this we can now interpret figures 4.19 and 4.20 in the same way as figure 3.9. For unbiased alignment parameters with a correct error estimate according to equations (2.23) and (2.24) we expect pull distributions centered at 0 and with a rms of 1. For figure 4.19 both conditions are only fulfilled approximately and for figure 4.20 only poorly. We attribute this behavior to the shortcomings of Chi2AlignAlg and of Athena release 10.5.0 as outlined above.

It is instructive to look at the differential pull distributions of a later iteration, e.g. iterations 5 and 10. Differential pull distributions for SCT end-cap modules for iteration 5 and 10 are shown in figures 4.21 and 4.22 (for reference, differential pull distributions for SCT barrel modules for iterations 5 and 10 are shown in appendix B). By differential pull we mean the alignment correction with respect to the previous iteration normalized by its error, i.e.

$$\Delta\ pull_{a_i\ iteration\ n} = \frac{\Delta a_{i\ iteration\ n}}{\sigma_{i\ iteration\ n}}. \tag{4.6}$$

We assumed for equation (2.2) that $\chi^2$ for a perfectly aligned detector resides in a minimum and so we try to minimize it. As $\chi^2$ approaches a minimum the necessary alignment corrections $\Delta a_i$ become smaller from iteration to iteration. The error $\sigma_i$ associated with the alignment correction $\Delta a_i$ does not decrease from iteration to iteration as it is basically the initial residual error propagated through the $\chi^2$-minimization. While $\sigma_{Track}$ of the residual is affected by alignment corrections, $\sigma_{Hit}$ is not. Consequently for the differential pull distributions we expect narrow distributions centered at 0. The distributions in figures 4.21 and 4.22 fulfill this expectations in general, but we observe large tails that bias the RMS of the distributions to larger values. The tails come from modules with significant alignment corrections even after 5 or 10 iterations. In other words, there are modules that do not converge on a stable alignment after a few iterations.

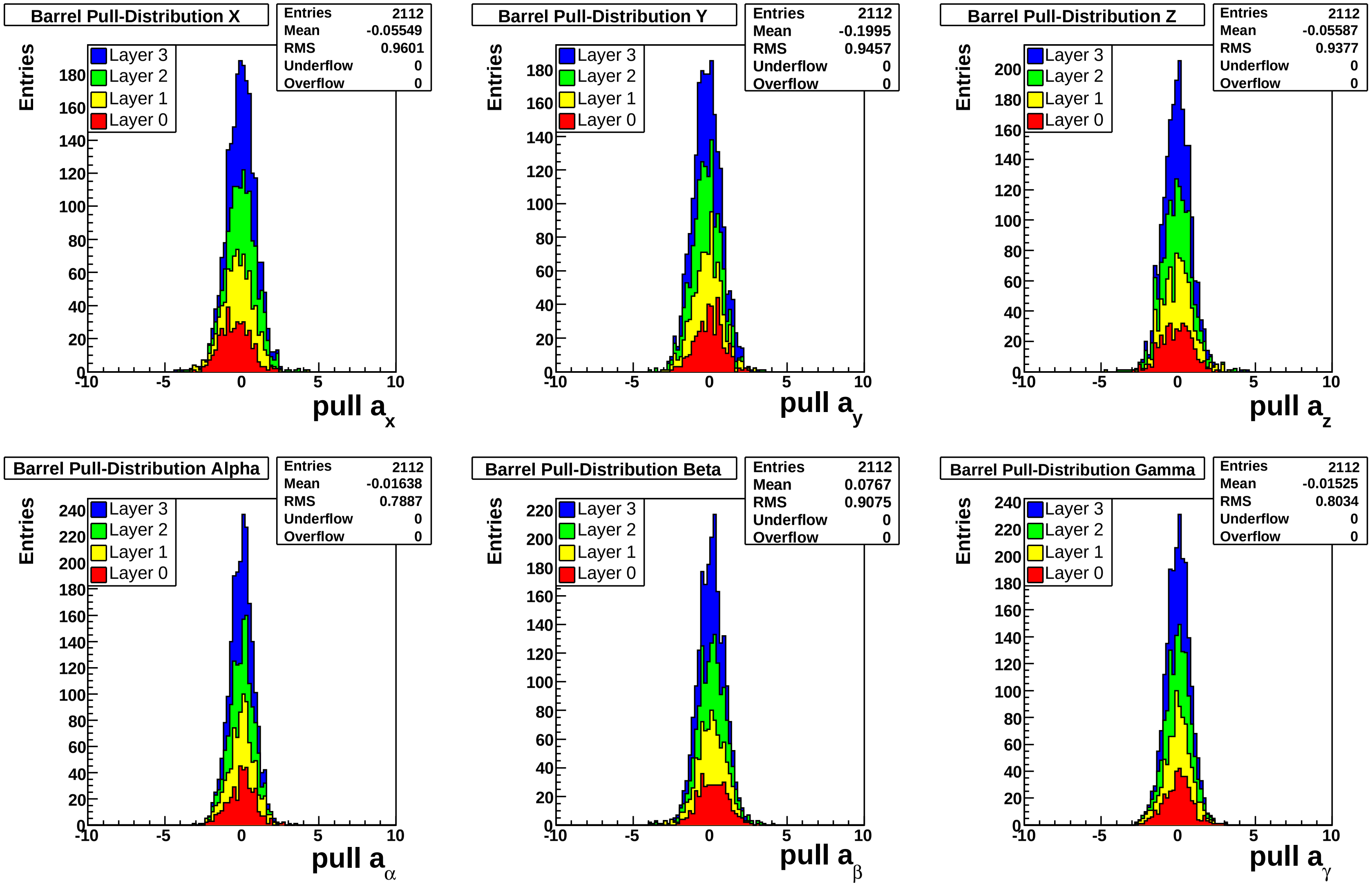

Figure 4.19: *Alignment parameter pull distributions from iteration 1 for all six degrees of freedom of all SCT barrel modules.*

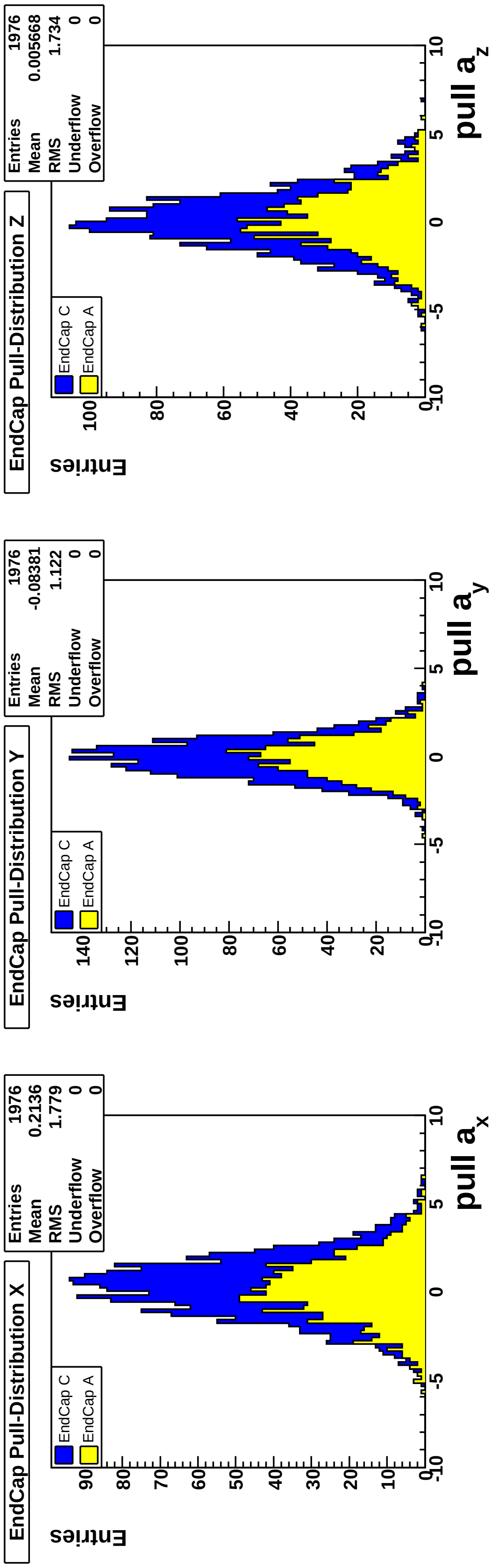
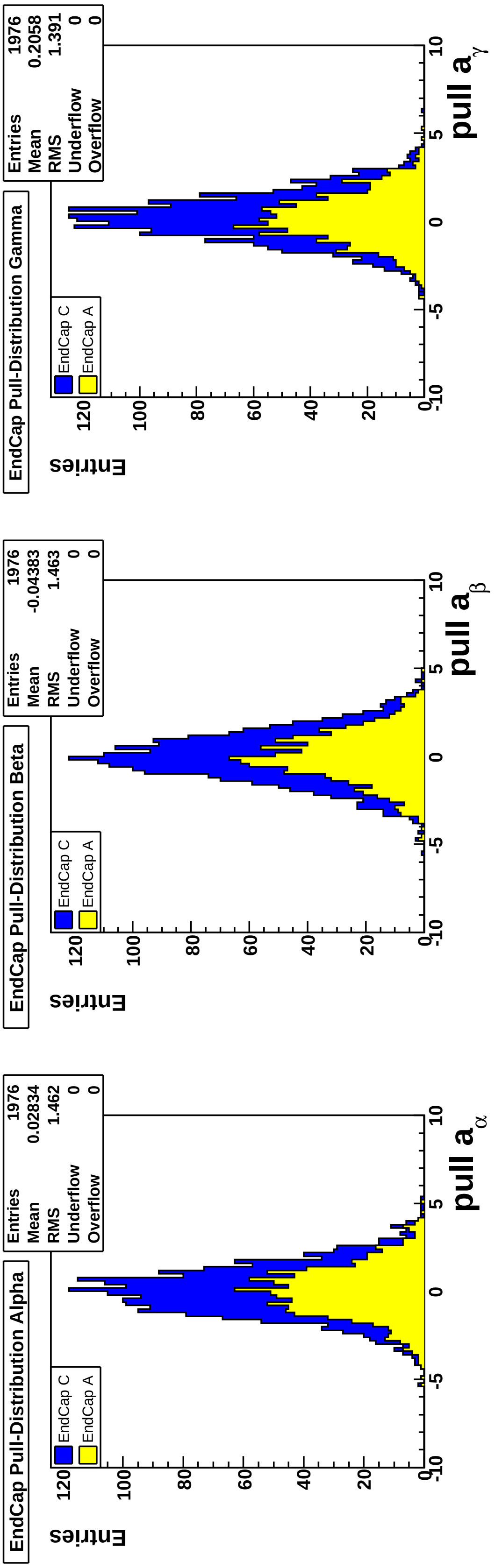

Figure 4.20: *Alignment parameter pull distributions from iteration 1 for all six degrees of freedom of all SCT end-cap modules.*

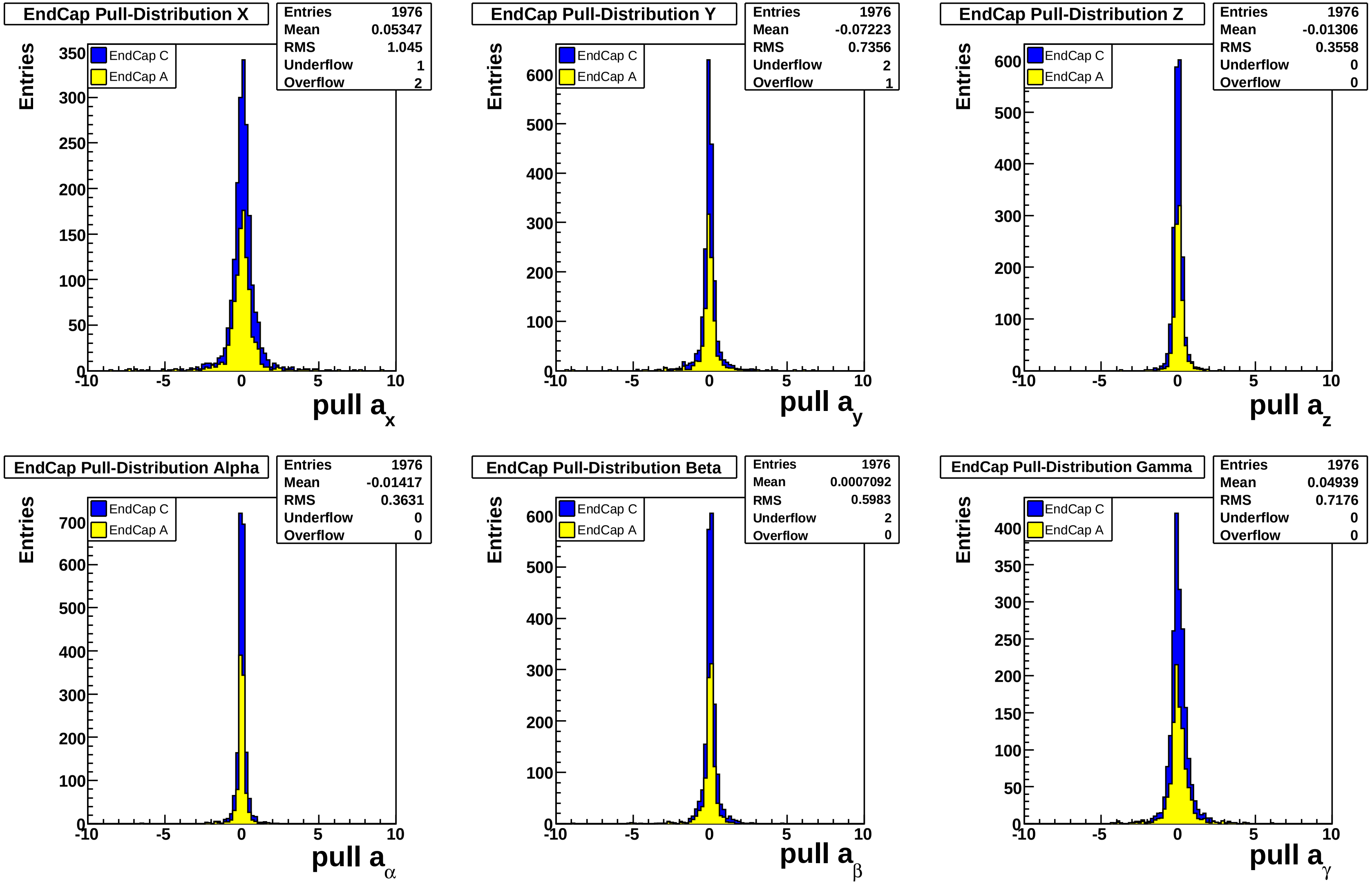

Figure 4.21: *Differential pull distributions from iteration 5 for all six degrees of freedom of all SCT end-cap modules.*

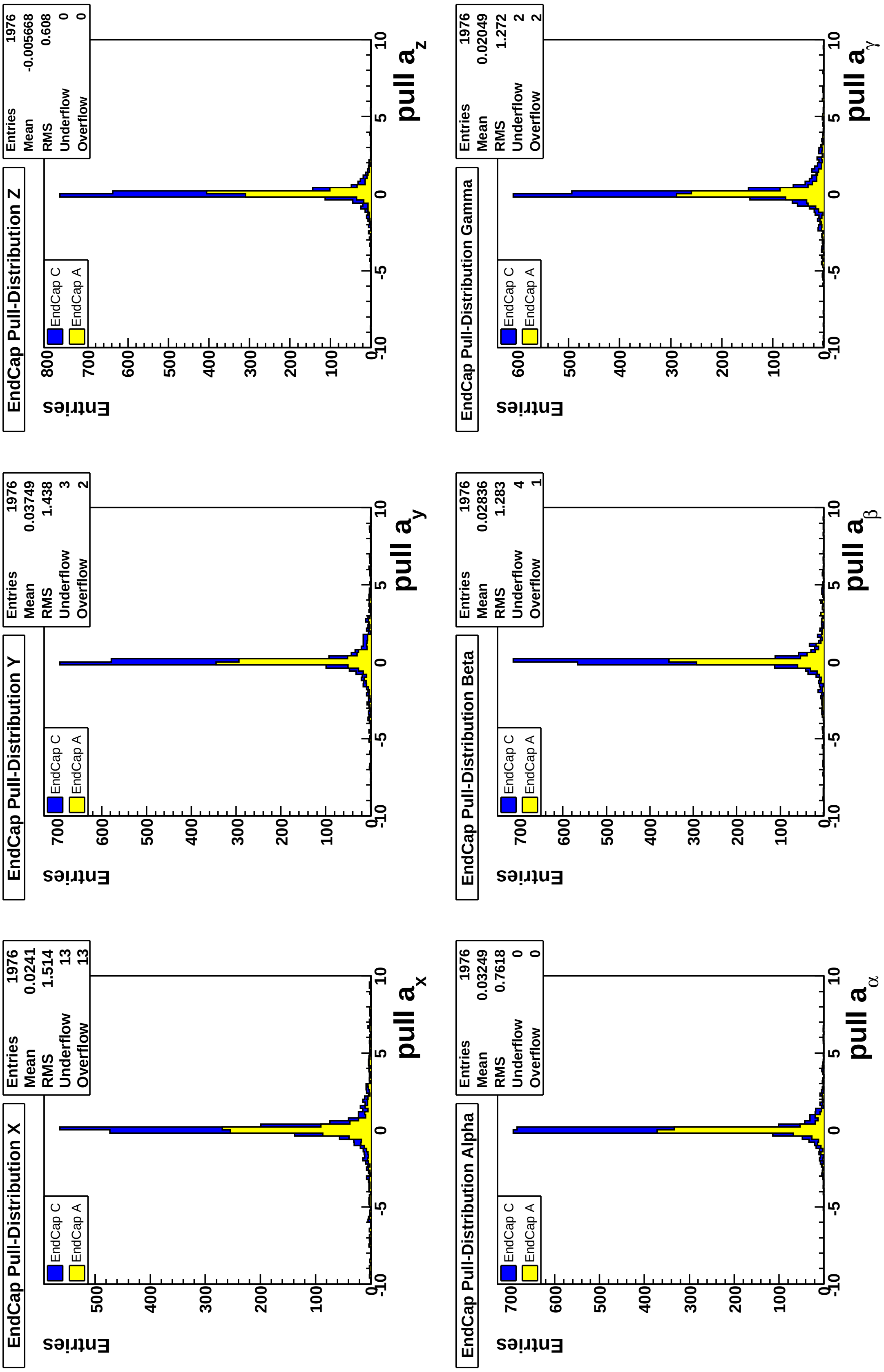

Figure 4.22: *Differential pull distributions from iteration 10 for all six degrees of freedom of all SCT end-cap modules.*

The flow of alignment parameters through the iterations is shown for two SCT modules of end-cap A in figures 4.23 and 4.24. In figure 4.23 the module clearly converges on a stable set of six alignment parameters. The true nominal alignment, i.e. 0, is within the statistical error estimate of four of the six alignment parameters after ten iterations. For alignment parameter $a_x$ nominal alignment is within three standard deviations and for $a_\alpha$ within a little more than one standard deviation.

The alignment parameter flow for the module in figure 4.24 is erratic. We observe no convergence of the six alignment parameters. The true nominal alignment is not within the error estimate for any of the six alignment parameters after ten iterations. Our estimate is that modules with an erratic convergence behavior are hit by a comparatively high number of "pathological" tracks that have bad fit quality and outlier hits. We will come back to this issue in section 4.4.3. For reference the alignment parameter flow of two SCT barrel modules is shown in appendix B.

The flow of alignment parameters through the iterations for all SCT modules of end-cap A and end-cap C – without error bars – is shown in figures 4.25 and 4.26. We can observe that the majority of SCT modules do converge on a stable alignment after a few iterations. Only few modules show an erratic behavior similar to figure 4.24. From the spread of the alignment parameters of modules that do converge on a stable alignment we can draw conclusions how much Chi2AlignAlg can constrain each of the six degrees of freedom. This statement is highly correlated to what we said in section 4.3.6 about the residual derivatives being an indication how much track based alignment can constrain a specific degree of freedom.

By analyzing the alignment parameter distributions for the SCT end-caps and the SCT barrel after ten iterations we can give an estimate of the alignment accuracy limits. Alignment parameter distributions after ten iterations for SCT barrel and end-cap modules are shown in appendix B. For the setup of Athena and Chi2AlignAlg we chose for this study and with the selected track sample as outlined in section 4.3.2 the estimates of the alignment accuracy limits are listed in table 4.1. We obtained the 95% confidence level estimate for each alignment parameter by taking a symmetric interval around zero that contained 95% of the entries [63]. The $2\sigma$ gaussian fit estimate comes from a gaussian fit of the alignment parameter distribution. This can be seen for example in figure B.14 where the standard deviations of the gaussian fits are shown. The $2\sigma$ statistical error is the mean (over all modules) of the error of the alignment parameters calculated with equations (2.23) and (2.24).

The difference between the accuracy of $a_x$ for SCT barrel and end-cap modules is because of the tilt angle of the barrel modules and the sensitivity transferred from the local x-coordinate to the local z-coordinate (see section 4.3.6).

We already stated in section 4.3.4 that the residual error for SCT barrel modules is overestimated and for SCT end-cap modules underestimated. The residual errors are propagated through the alignment machinery and consequently, we observe in figures 4.19 and 4.20 that the statistical error for the alignment parameters is overestimated for SCT barrel modules and underestimated for SCT end-cap modules. This behavior is mirrored in the alignment accuracy limits in table 4.1.

We observe that a gaussian fit does not describe the alignment parameter distribution correctly. Therefore our estimates of the alignment accuracy achievable with the present setup are the 95% CL alignment accuracy limits.

| Alignment accuracy limits | | | | | | |
|---|---|---|---|---|---|---|
| alignment parameter | SCT barrel module | | | SCT end-cap module | | |
| | 95%CL | $2\sigma$ gauss fit | $2\sigma_{stat}$ | 95%CL | $2\sigma$ gauss fit | $2\sigma_{stat}$ |
| $a_x$ | 96 $\mu$m | 73 $\mu$m | 57 $\mu$m | 38 $\mu$m | 28 $\mu$m | 9.8 $\mu$m |
| $a_y$ | 505 $\mu$m | 341 $\mu$m | 337 $\mu$m | 290 $\mu$m | 179 $\mu$m | 113 $\mu$m |
| $a_z$ | 490 $\mu$m | 350 $\mu$m | 298 $\mu$m | 2205 $\mu$m | 1587 $\mu$m | 600 $\mu$m |
| $a_\alpha$ | 3.0 mrad | 2.4 mrad | 2.4 mrad | 41 mrad | 20 mrad | 17 mrad |
| $a_\beta$ | 5.1 mrad | 4.3 mrad | 3.5 mrad | 25 mrad | 20 mrad | 10 mrad |
| $a_\gamma$ | 0.66 mrad | 0.49 mrad | 0.49 mrad | 0.75 mrad | 0.51 mrad | 0.25 mrad |

Table 4.1: *Alignment accuracy limits of Chi2AlignAlg with the presented setup and the selected track sample.*

In figure 4.28 the alignment parameter flow of all SCT modules from barrel layer 0 is shown. We observe a falling tendency of the alignment parameters $a_x$ to offset the 6 $\mu$m shift of the initial residual distribution.

The spread of alignment parameters for barrel layer 0 and thus the inferred alignment accuracy is significantly better than for the whole SCT barrel. In fact the alignment parameters of the SCT modules on barrel layer 3 set all the alignment accuracy limits for SCT barrel modules in table 4.1. We attribute this to the fact that the hits on barrel layer 3 are the last hits on the refitted tracks and to calculate the unbiased input for alignment (residuals, residual errors and residual derivatives) involves a real extrapolation, not an interpolation between hits like for the SCT modules on the other barrel layers. The same argument is true for the outer most disks of the two end-caps. In general we can say that the 95% CL values in table 4.1 have contributions from statistical as well as systematical uncertainties. To further understand the nature of systematical effects is the subject of ongoing studies.

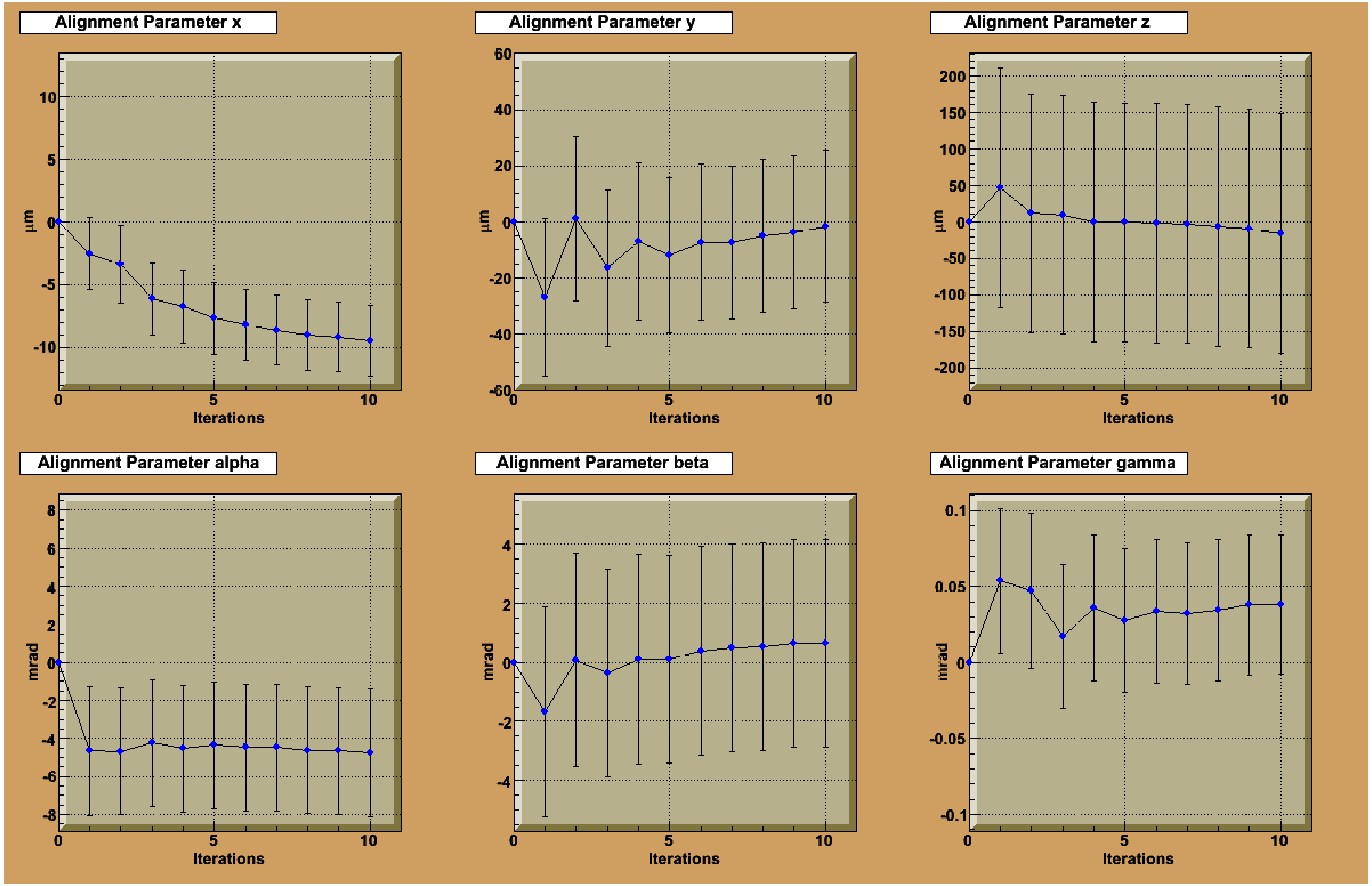

Figure 4.23: *Alignment parameter flow of module 2/2/2/2/10/1 (end-cap A disk 2) through ten iterations The module converges on a stable set of six alignment parameters.*

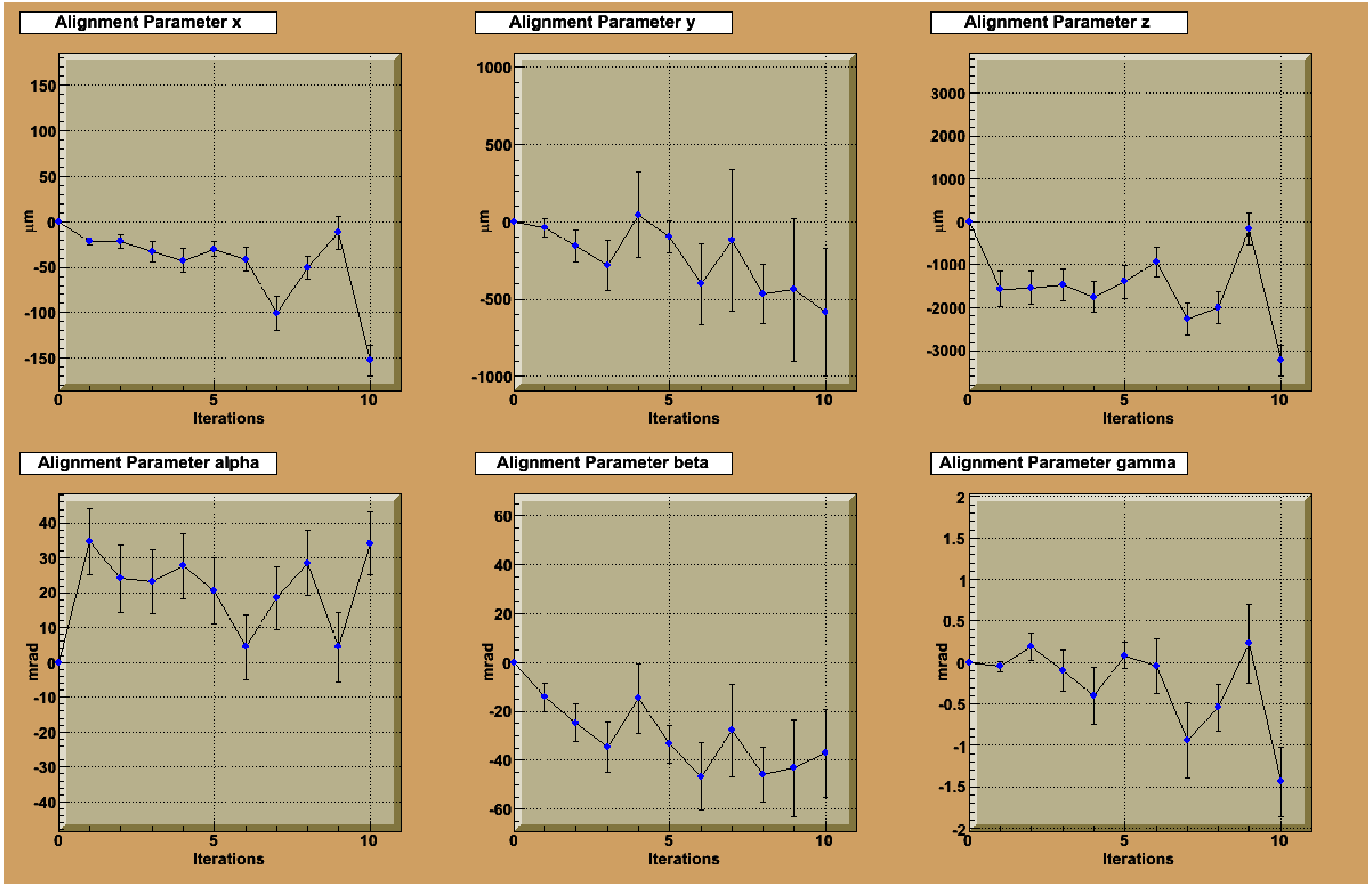

Figure 4.24: *Alignment parameter flow of module 2/2/2/8/51/0 (end-cap A disk 8) through ten iterations The convergence behavior of this module is erratic.*

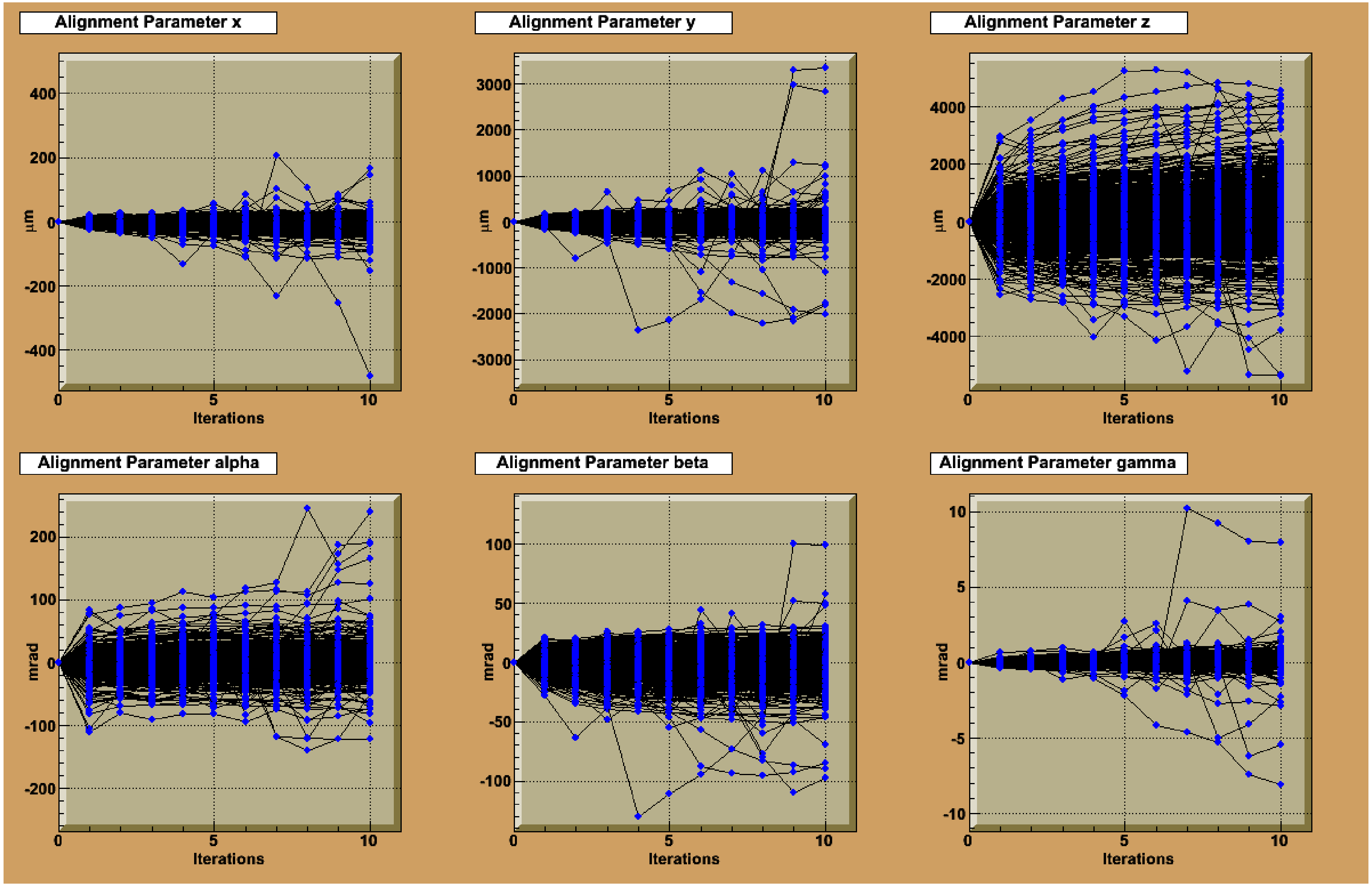

Figure 4.25: *Alignment parameter flow of all 988 SCT modules on end-cap A through ten iterations.*

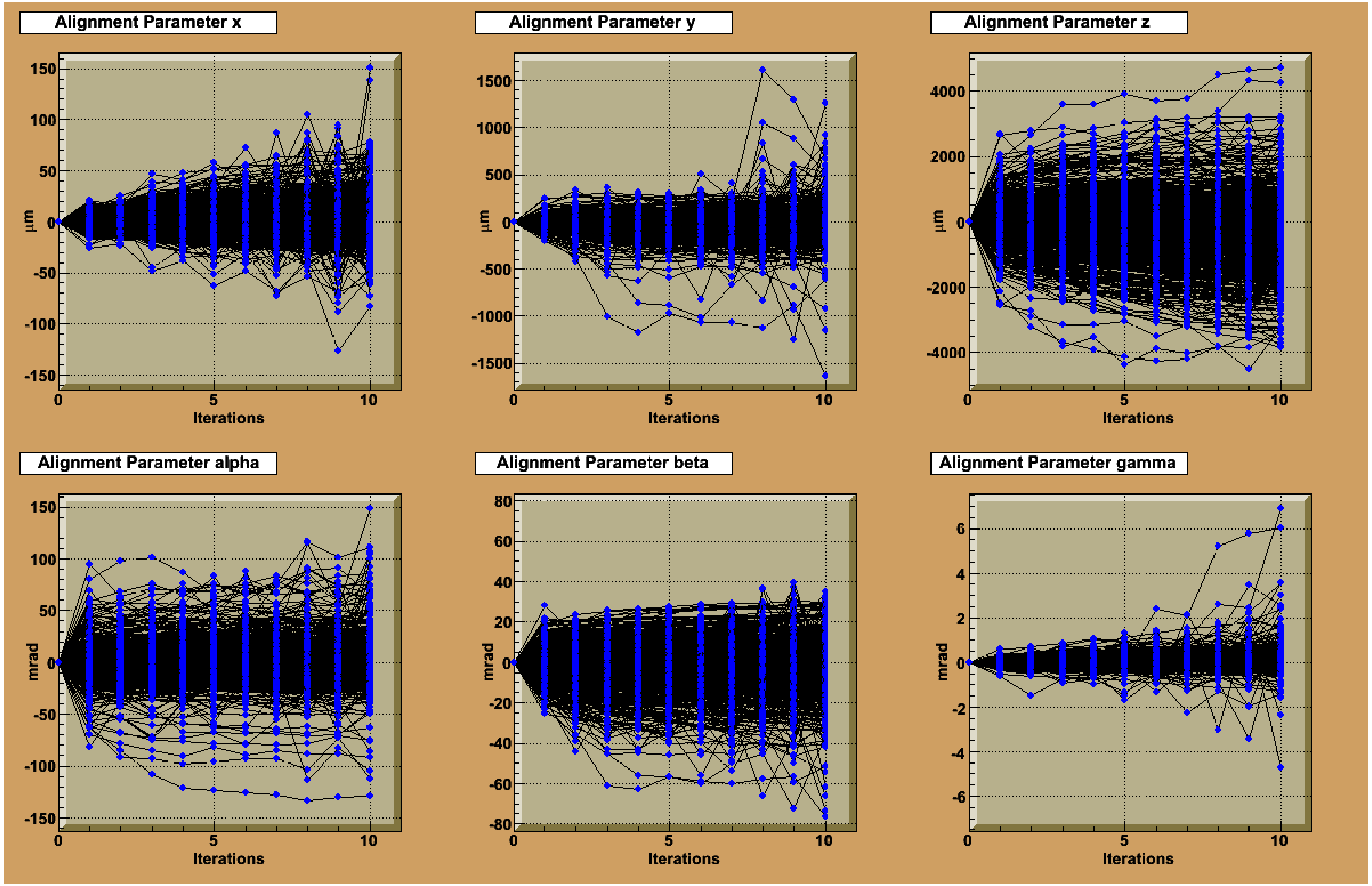

Figure 4.26: *Alignment parameter flow of all 988 SCT modules on end-cap C through ten iterations.*

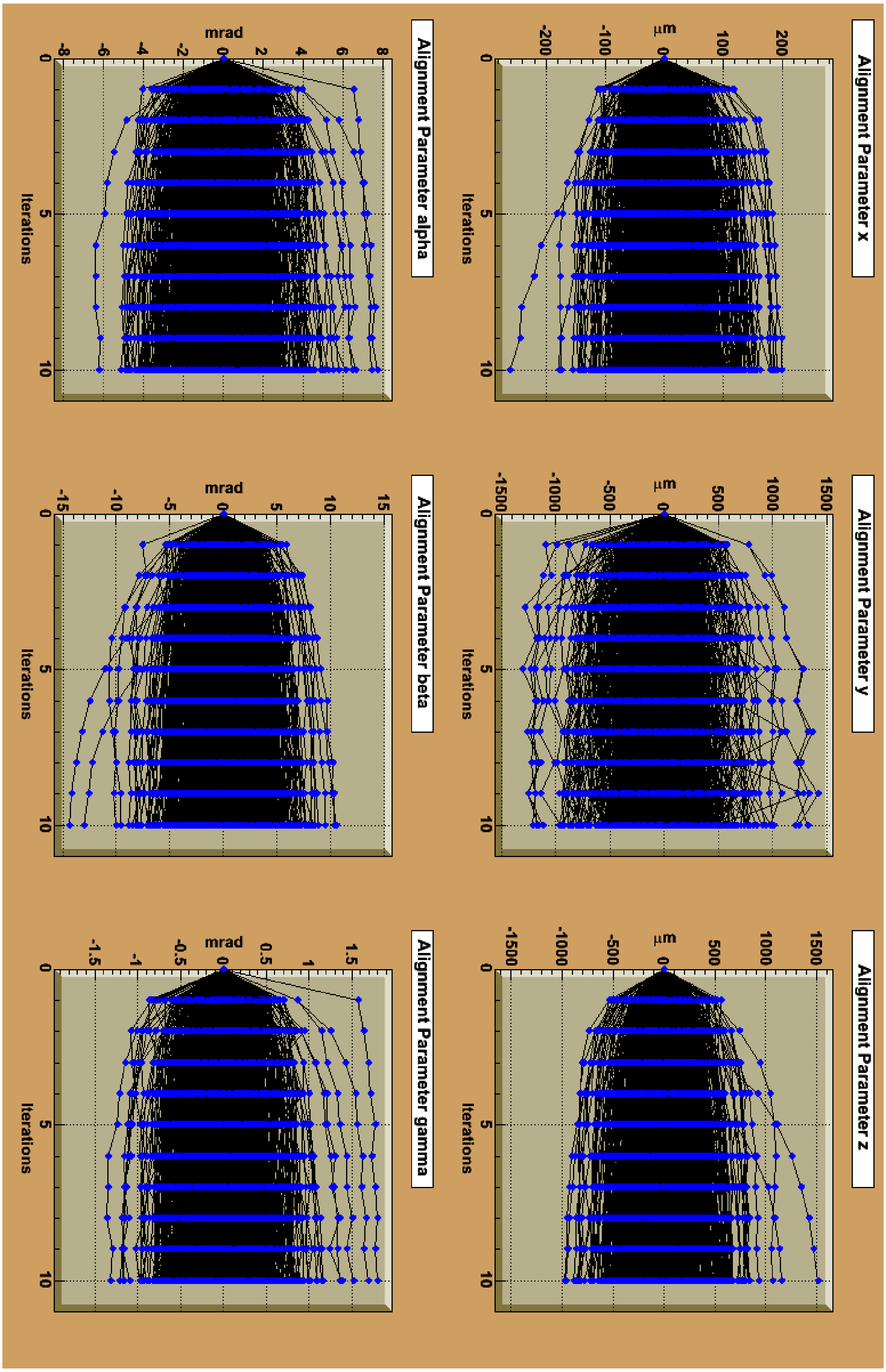

Figure 4.27: *Alignment parameter flow of all 2112 SCT barrel modules through ten iterations.*

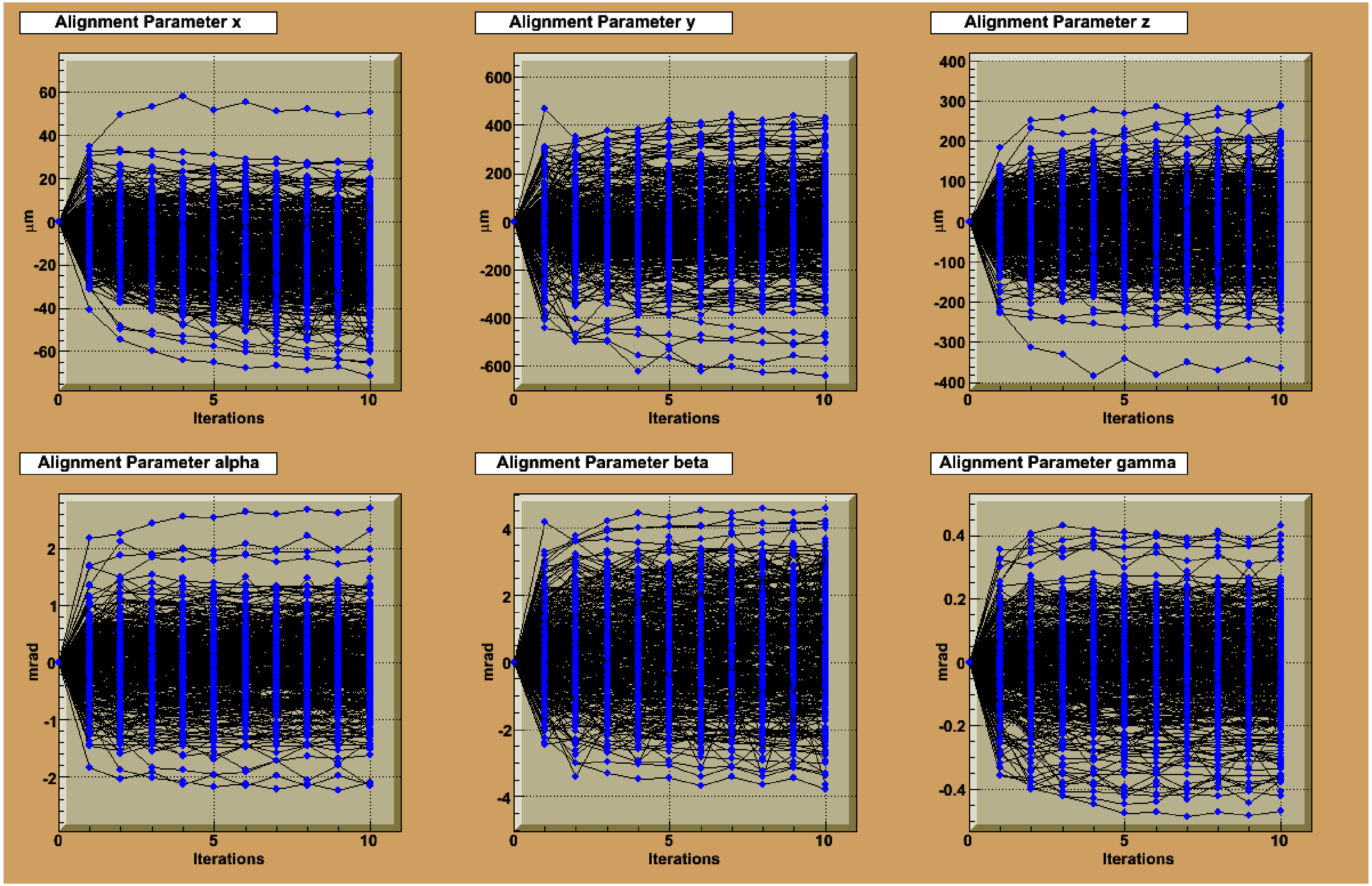

Figure 4.28: *Alignment parameter flow of all 384 modules on SCT barrel layer 0 through ten iterations.*

### 4.4.2 Misalignment setups

In the previous section we established the alignment accuracy of Chi2AlignAlg with nominal alignment with 95% confidence level. We expect that misalignments significantly below this accuracy are not recovered by Chi2AlignAlg. We conducted several tests to assess the ability of Chi2AlignAlg to recover different types of misalignment. We did two types of tests, single module misalignment where only 22 selected modules were misaligned and systematic misalignment where we misaligned all modules of end-cap A disk 3 or barrel layer 2 in a systematic way.

For systematic misalignments we used three different sets of misalignments for our tests. Details about the three sets of misalignments are listed in table 4.2.

| Systematic misalignment sets | | |
|---|---|---|
| Barrel layer 2 | MisalignmentSetLayer2_01 | $\Delta x = 50\ \mu m$ |
| | MisalignmentSetLayer2_02 | $\Delta y = 200\ \mu m$ |
| | MisalignmentSetLayer2_03 | $\Delta z = -200\ \mu m$ |
| Endcap A disk 3 | MisalignmentSetDiskA3_01 | $\Delta x = 50\ \mu m$ |
| | MisalignmentSetDiskA3_02 | $\Delta y = 200\ \mu m$ |
| | MisalignmentSetDiskA3_03 | $\Delta z = -200\ \mu m$ |

Table 4.2: *List of systematic misalignment sets. The misalignment shifts are with respect to the local coordinate frame of each module. For example a $\Delta x$ shift would correspond to a $\phi$ rotation of the whole layer or disk around the global $\hat{z}$-axis .*

The 22 modules selected for single wafer misalignment are listed in table 4.3. They were chosen among the 4088 SCT modules such that no track originating from the interaction point would hit more than one misaligned module. Eight different sets of misalignments were applied to the 22 modules. In table 4.4 the details of each set of misalignments are listed.

| Modules for single module misalignment | | | |
|---|---|---|---|
| Superstructures | Layer / Disk | Phi | Eta |
| Barrel | 0 | 0 | -1 |
| | 1 | 15 | 3 |
| | 2 | 8 | -4 |
| | 3 | 42 | 5 |
| Endcap A | 0 | 0 | 0 |
| | 1 | 30 | 2 |
| | 2 | 10 | 1 |
| | 3 | 45 | 0 |
| | 4 | 20 | 2 |
| | 5 | 5 | 1 |
| | 6 | 39 | 0 |
| | 7 | 15 | 1 |
| | 8 | 51 | 0 |
| Endcap C | 0 | 15 | 1 |
| | 1 | 2 | 0 |
| | 2 | 22 | 2 |
| | 3 | 17 | 1 |
| | 4 | 47 | 0 |
| | 5 | 27 | 2 |
| | 6 | 7 | 1 |
| | 7 | 28 | 0 |
| | 8 | 41 | 0 |

Table 4.3: *List of the 22 modules selected for single module misalignment. The classification is according to the ATLAS offline readout identifier scheme [35].*

| Misalignment sets for the 22 modules | |
|---|---|
| MisalignmentSet22_01 | $\Delta x = 50\ \mu m$ |
| MisalignmentSet22_02 | $\Delta y = 200\ \mu m$ |
| MisalignmentSet22_03 | $\Delta z = -200\ \mu m$ |
| MisalignmentSet22_04 | $\Delta x = 50\ \mu m$ |
| | $\Delta y = 200\ \mu m$ |
| MisalignmentSet22_05 | $\Delta x = 50\ \mu m$ |
| | $\Delta z = -200\ \mu m$ |
| MisalignmentSet22_06 | $\Delta y = 200\ \mu m$ |
| | $\Delta z = -200\ \mu m$ |
| MisalignmentSet22_07 | $\Delta \alpha = 5\ mrad$ |
| | $\Delta \gamma = 5\ mrad$ |
| MisalignmentSet22_08 | $\Delta x = 50\ \mu m$ |
| | $\Delta y = 200\ \mu m$ |
| | $\Delta z = -200\ \mu m$ |
| | $\Delta \alpha = 5\ mrad$ |
| | $\Delta \beta = -5\ mrad$ |
| | $\Delta \gamma = 5\ mrad$ |

Table 4.4: *List of the eight different sets of misalignment applied to the 22 SCT modules selected for single module misalignment.*

### 4.4.3 Tests with misaligned detector layout

For our tests with a misaligned SCT geometry we fed modified alignment parameters into the conditions database to serve as misalignments. The misalignments were applied to the selected track sample during the refit. As the tracks were originally simulated with nominal alignment we expect Chi2AlignAlg to correct for the misalignment applied during reconstruction and to recover the nominal alignment.

In figure 4.29 the alignment parameter flow with nominal initial alignment for the 22 modules selected for misalignment is shown. Figure 4.29 serves as a reference for our tests with misalignments. We observe that most modules converge on a stable set of alignment parameters and only few modules show erratic convergence behavior.

The alignment parameter flow of module 2/2/0/2/8/-4 (barrel layer 2) with the misalignment Set22_08 as initial alignment is shown in figure 4.30. We observe that this module converges on a stable set of alignment parameters and that the six misaligned degrees of freedom are corrected for. Nominal alignment is within the error estimate of all alignment parameters after ten iterations. We also observe that the initial alignment is recovered after ∼3 iterations and that subsequent iterations do not improve the alignment accuracy.

In figure 4.31 the alignment parameter flow of the four SCT barrel modules from the set of 22 misaligned modules with the misalignment Set22_08 as initial alignment is shown. We observe that the six misaligned degrees of freedom are recovered within the accuracy limits established for SCT barrel modules in section 4.4.1 (see table 4.1).

In figure 4.32 the alignment parameter flow of 17 of the 18 SCT end-cap modules from the set of 22 misaligned modules with the misalignment Set22_08 as initial alignment is shown. Module 2/2/2/8/51/0 (end-cap A disk 8) is not shown in figure 4.32, but discussed separately below. In figure 4.32 we observe that for most of the modules the six misaligned degrees of freedom are recovered within the accuracy limits established in section 4.4.1 (see table 4.1). The same modules that showed erratic convergence behavior in figure 4.29 do so in figure 4.32 as well.

In figure 4.33 the alignment parameter flow of SCT module 2/2/2/8/51/0 with the misalignment Set22_08 as initial alignment is shown. For comparison the alignment parameter flow of this module with nominal initial alignment is shown in figure 4.24. By comparing figures 4.24 and 4.33 we observe that the choice of initial alignment considerably influences the behavior of the alignment parameter flow during later iterations. This dependency on initial conditions is an indication that the $\chi^2$ of this module does not converge on the global minimum in the high-dimensional parameter space of the hit sample of this module. $\chi^2$ converges on local minima, depending on the initial conditions. This is an indication of the "pathological" nature of the parameter space, i.e. of the hit sample.

We also observe that the erratic nature of the alignment flow builds up during subsequent iterations and is not dampened. As we already observed in figure 4.30 that a few iterations are sufficient for a module with a stable convergence behavior to correct for applied misalignments a cut on the number of iterations seems plausible. This is shown in figure 4.34 for the alignment parameter flow of all 22 modules with the misalignment Set22_08 as initial alignment.

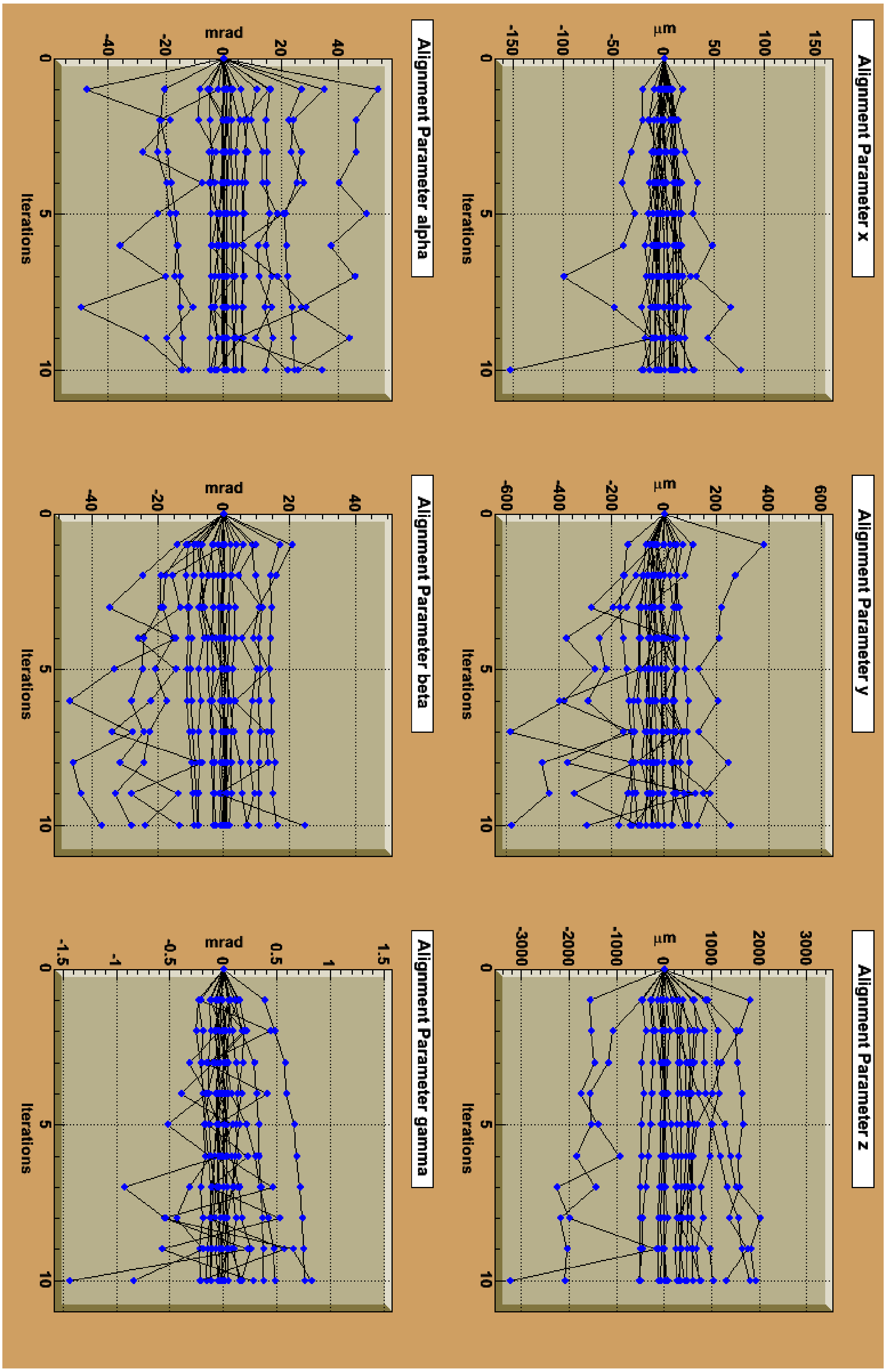

Figure 4.29: *Alignment parameter flow through ten iterations with nominal initial alignment of the 22 modules selected for single module misalignment.*

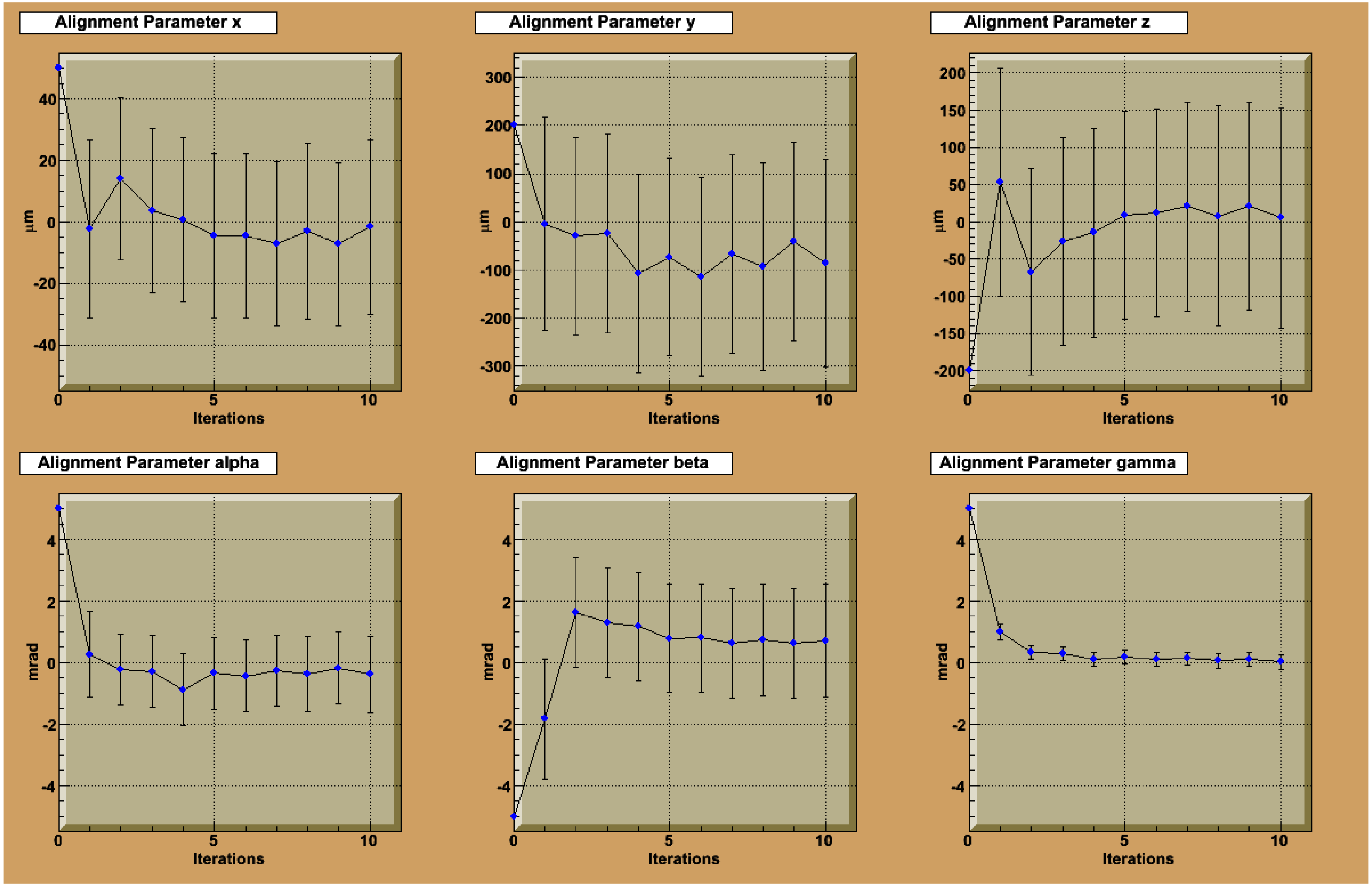

Figure 4.30: *Alignment parameter flow through ten iterations of module 2/2/0/2/8/-4 (barrel layer 2) with initial misalignment Set22_08 (see table 4.4).*

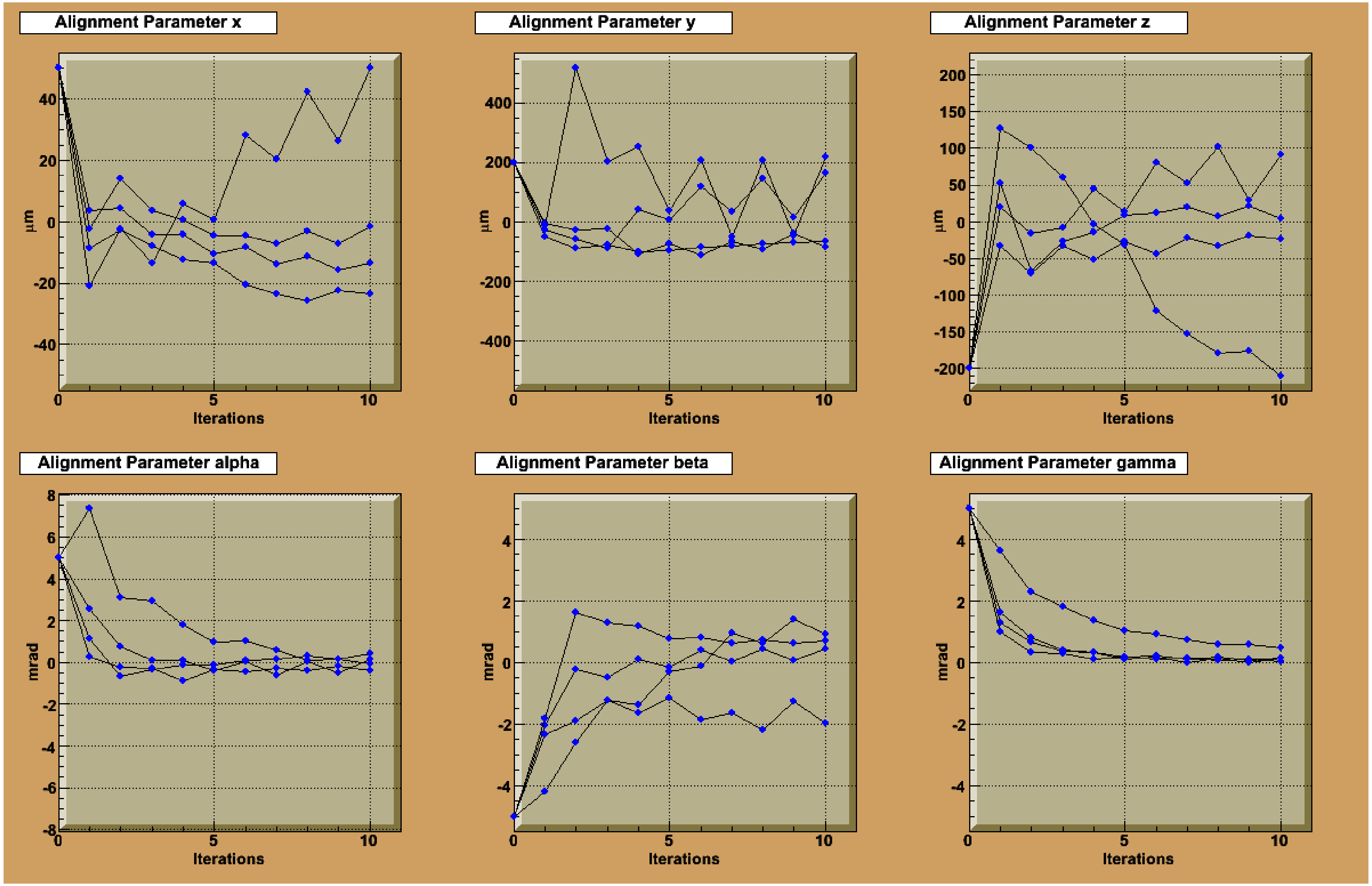

Figure 4.31: *Alignment parameter flow through ten iterations of the four SCT barrel modules from the set of 22 modules. Initial alignment was Set22_08 (see table 4.4).*

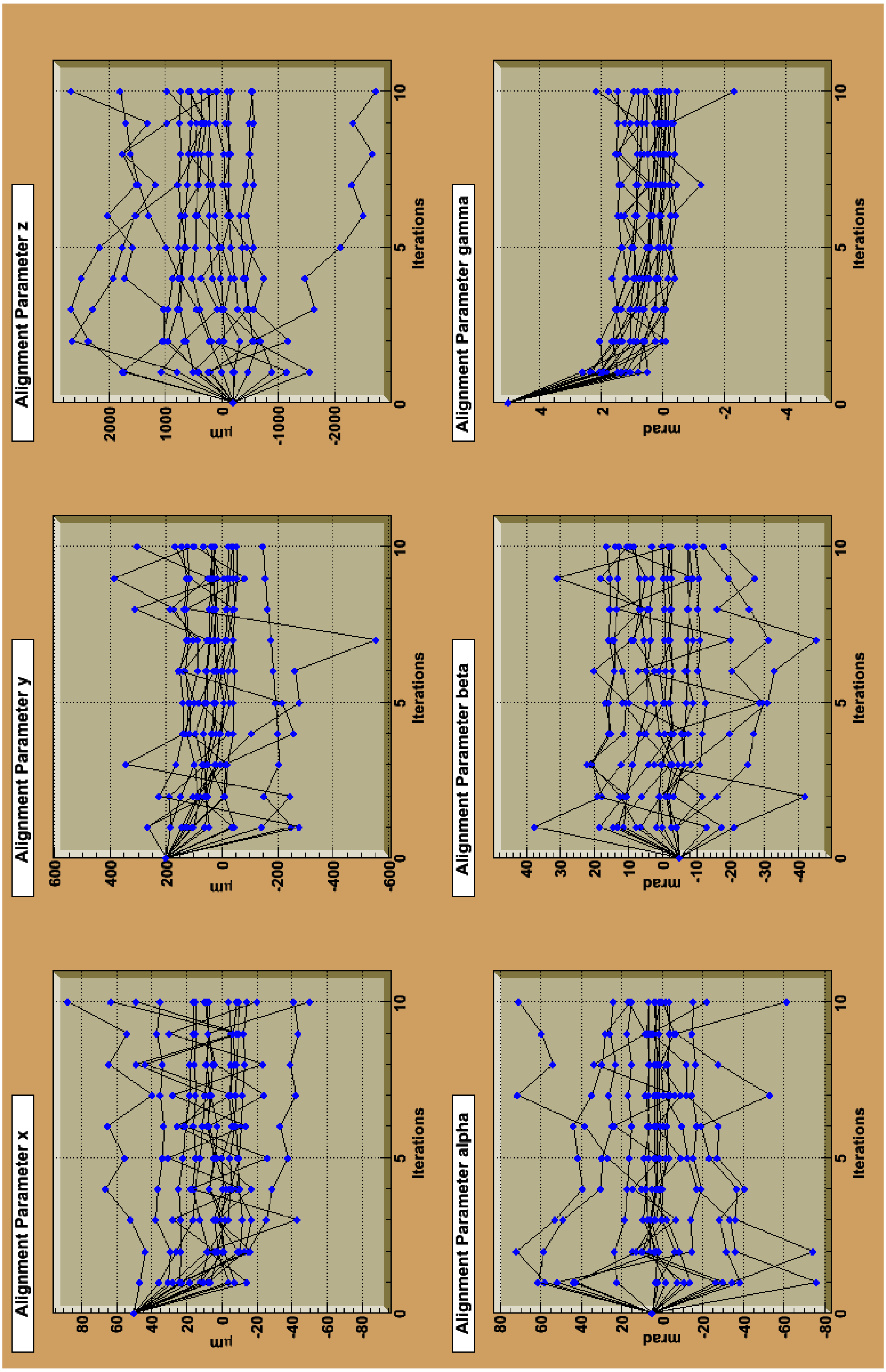

Figure 4.32: *Alignment parameter flow through ten iterations of 17 SCT end-cap modules from the set of 22 modules. Initial alignment was Set22_08 (see table 4.4).*

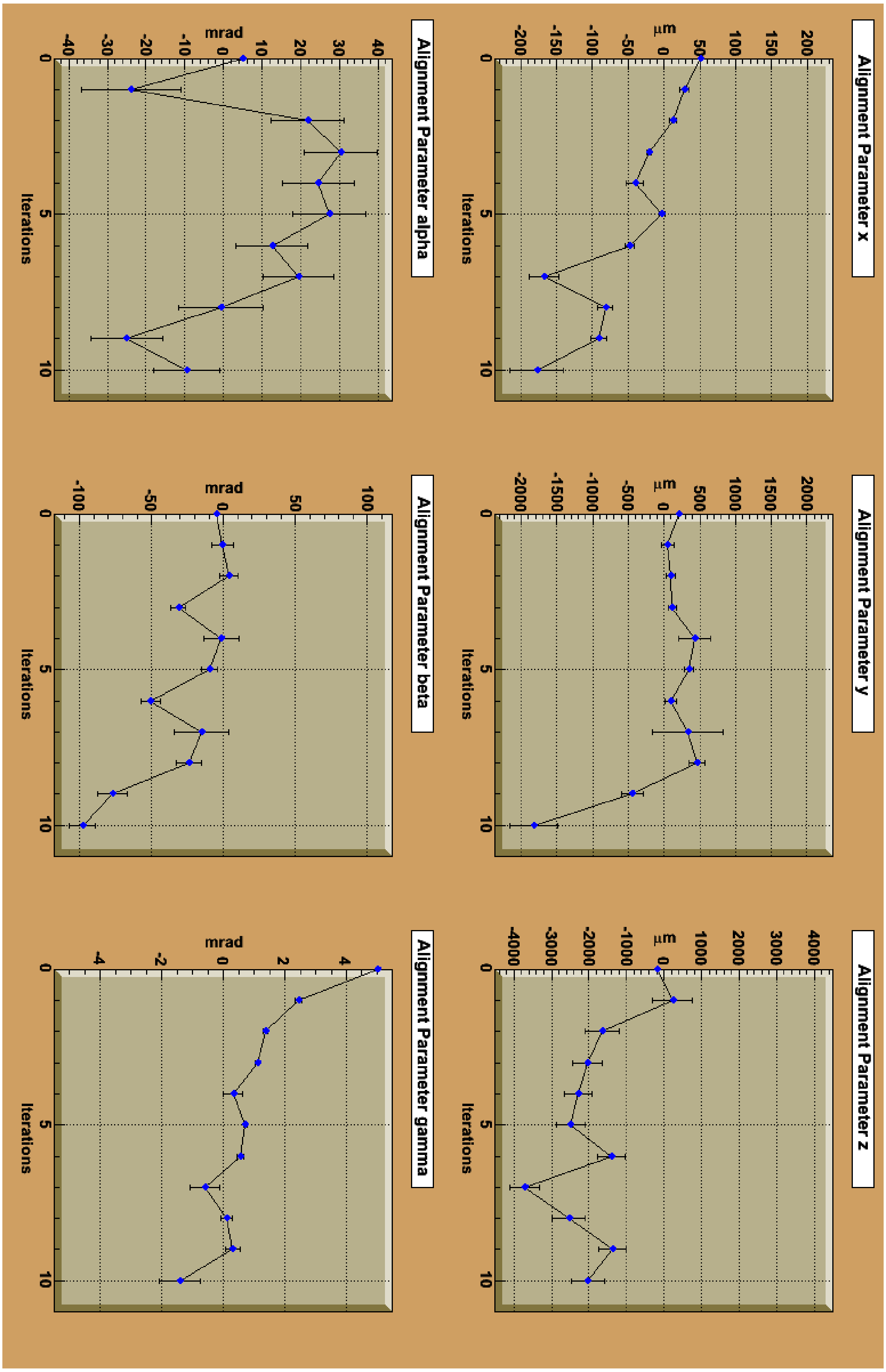

Figure 4.33: *Alignment parameter flow through ten iterations of module 2/2/2/8/51/0 (end-cap A disk 8) with initial misalignment Set22_08 (see table 4.4).*

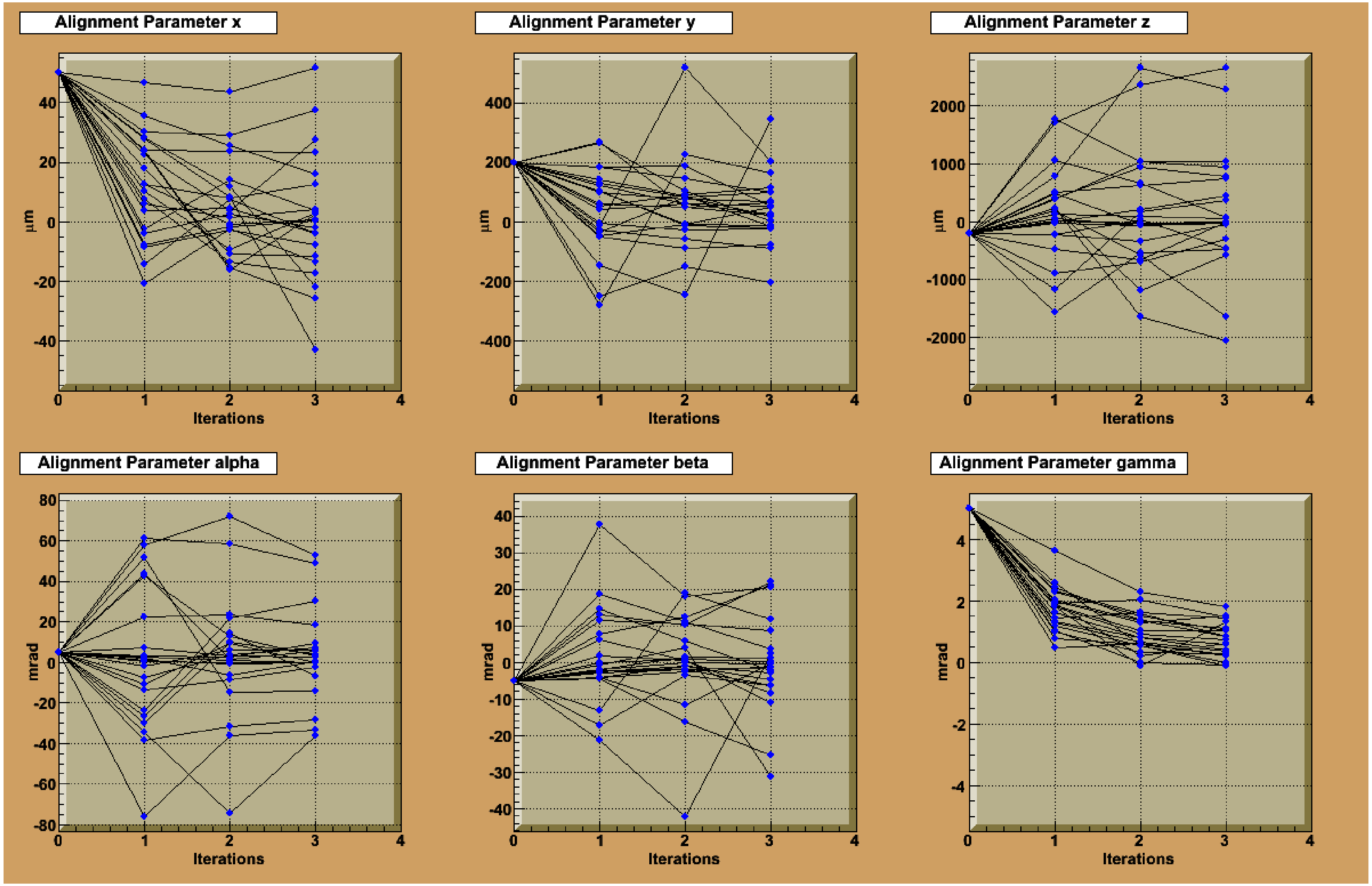

Figure 4.34: *Alignment parameter flow through three iterations of the 22 modules selected for single module misalignment. Initial alignment was Set22_08 (see table 4.4).*

The second type of test we did was to systematically misalign all SCT modules on barrel layer 2 or end-cap A disk 3. The goal of these tests was to check if the performance of Chi2AlignAlg with misaligned global degrees of freedom of a whole superstructure (like a $\phi$-rotation, or a global $\hat{z}$-shift) significantly differs from the performance with nominal alignment or local misalignment.

In figure 4.35 the alignment parameter flow of all SCT modules on barrel layer 2 with nominal initial alignment is shown. Figure 4.35 serves as a reference for figure 4.36 where the alignment parameter flow of the same modules is shown with the misalignment SetLayer2_01 as initial alignment. The alignment parameter flows of these modules with misalignment SetLayer2_02 and SetLayer2_03 as initial alignment are shown in appendix B. By comparing figures 4.35 and 4.36 we observe that the applied misalignment is corrected for within the accuracy established for nominal alignment. This confirms the results of the tests with single module misalignment.

In figure 4.37 the alignment parameter flow of all modules from end-cap A disk 3 with nominal initial alignment is shown. Again figure 4.37 serves as a reference figure 4.38 where the alignment parameter flow of the same modules with the misalignment SetDiskA3_01 as initial misalignment is shown. In appendix B the alignment parameter flows for SetDiskA3_02 and SetDiskA3_03 are shown.

In figure 4.37 we observe a rising tendency of the flow of the alignment parameter $a_x$ as the individual alignment flow threads do not spread out evenly around the initial nominal alignment 0. This behavior is mirrored in figure 4.38. The alignment parameter flow does not settle on nominal alignment but converges on a similar alignment as in figure 4.37. It is not clear where this offset along the local x-coordinate (corresponding to a global $\phi$-rotation of end-cap A disk 3) comes from, as the residual distribution of end-cap A disk 3 is centered at 0 through all iterations. Again we confirm that the performance of Chi2AlignAlg with nominal alignment is the limiting factor for the tests with systematically misaligned SCT geometry.

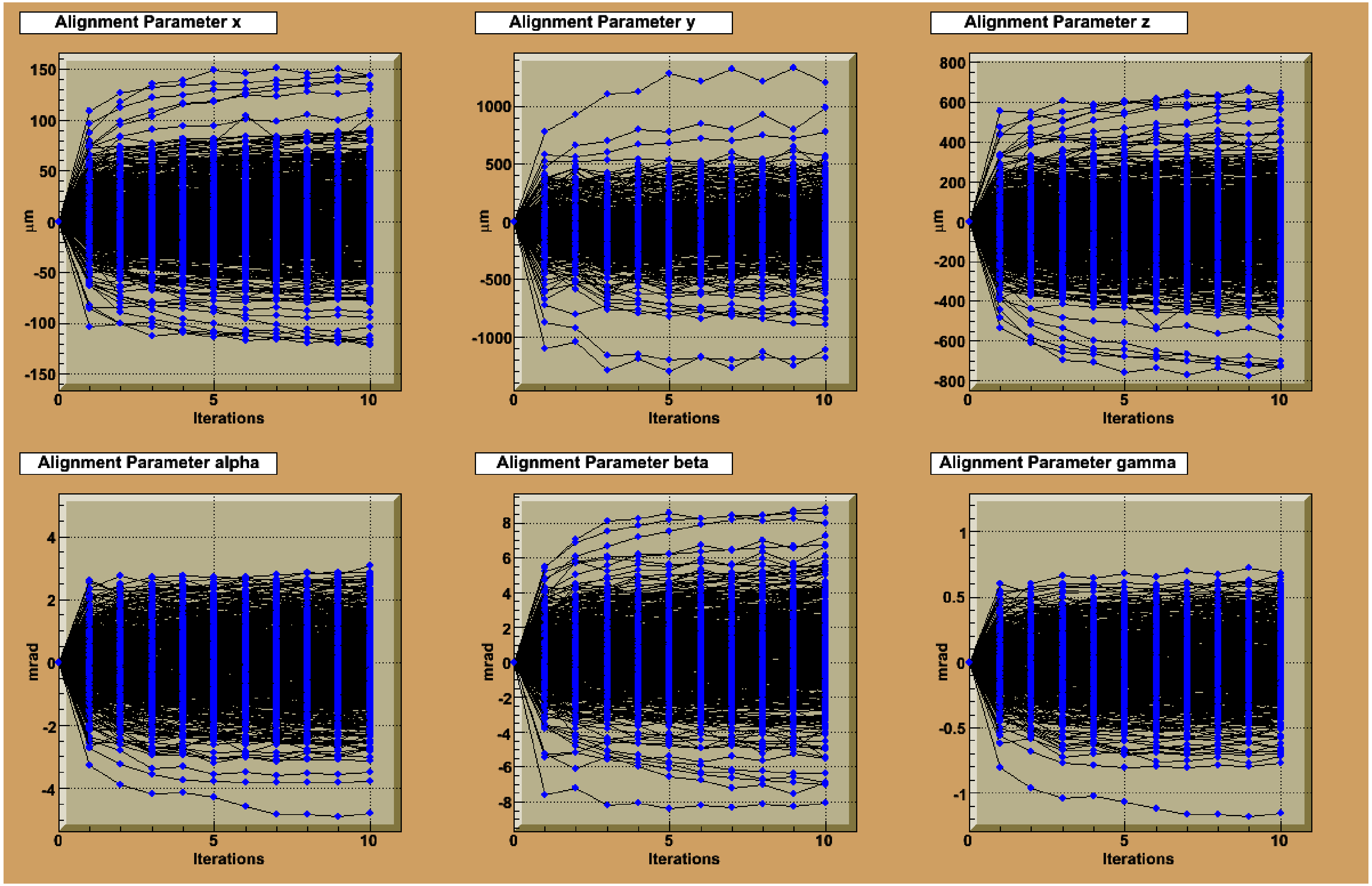

Figure 4.35: *Alignment parameter flow through ten iterations with nominal initial alignment of all modules on barrel layer 2.*

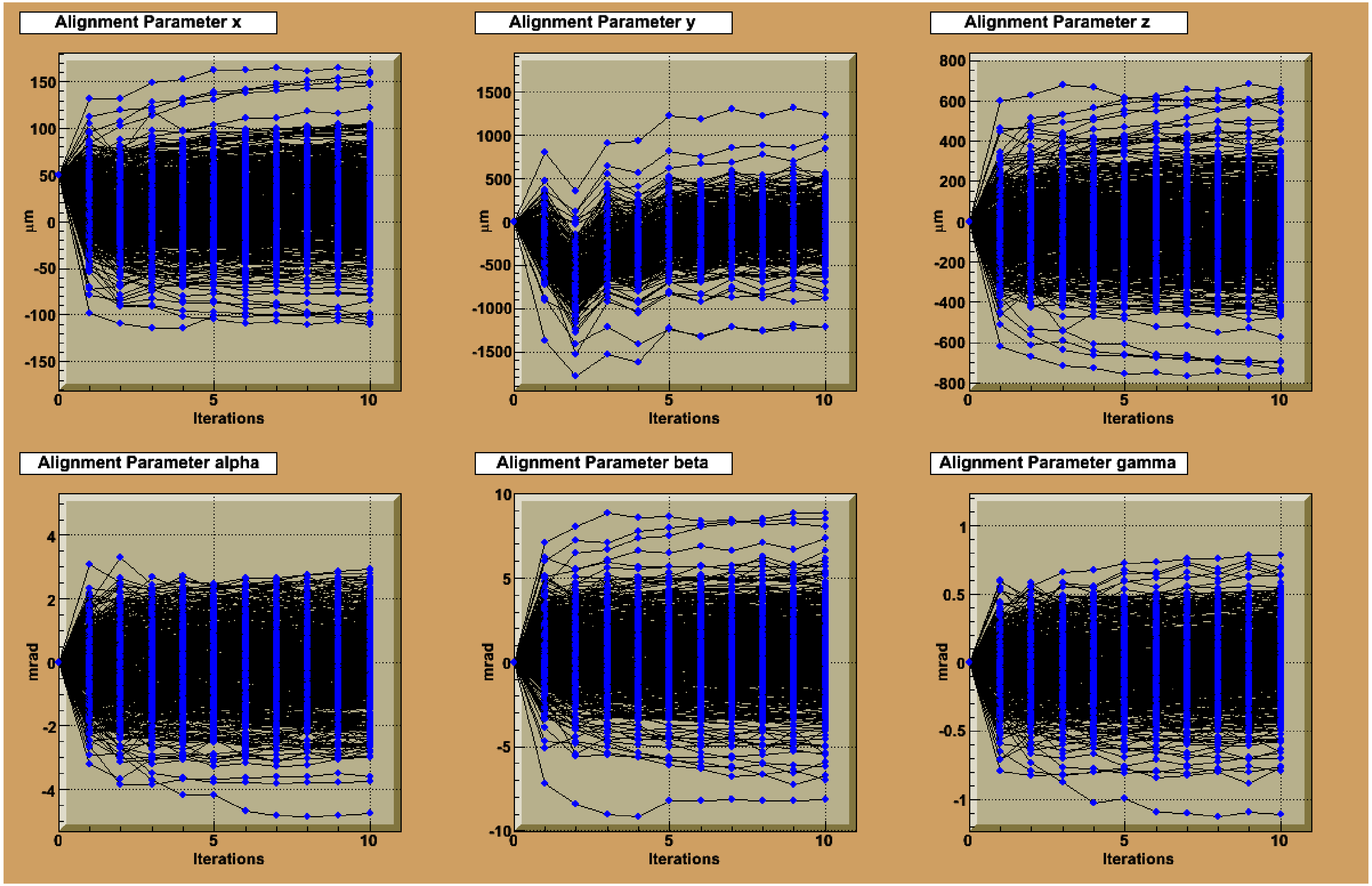

Figure 4.36: *Alignment parameter flow through ten iterations of all modules on barrel layer 2 with systematic initial misalignment. Initial alignment was SetLayer2_01 (see table 4.2).*

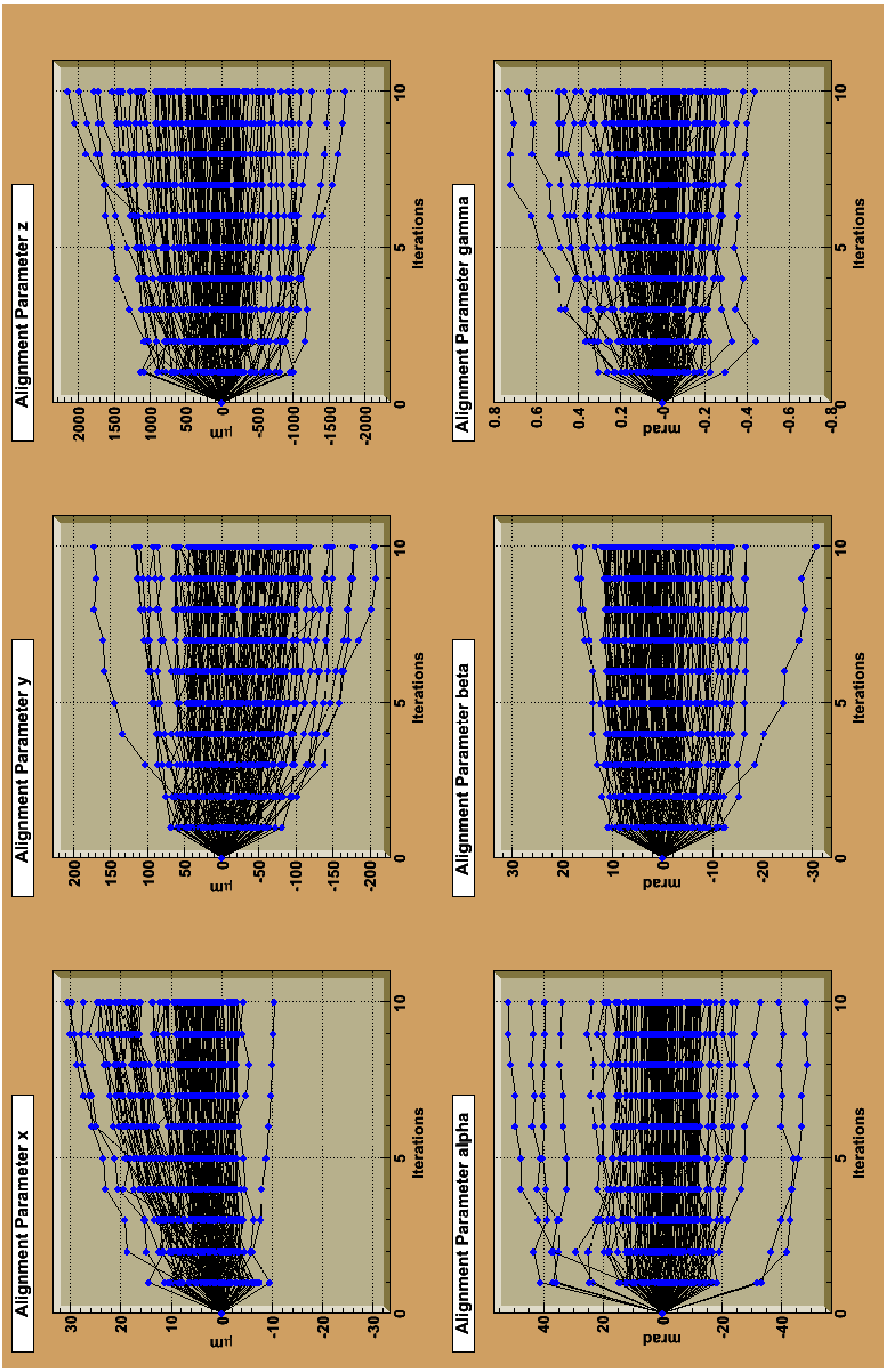

Figure 4.37: *Alignment parameter flow through ten iterations with nominal initial alignment of all modules on end-cap A disk 3.*

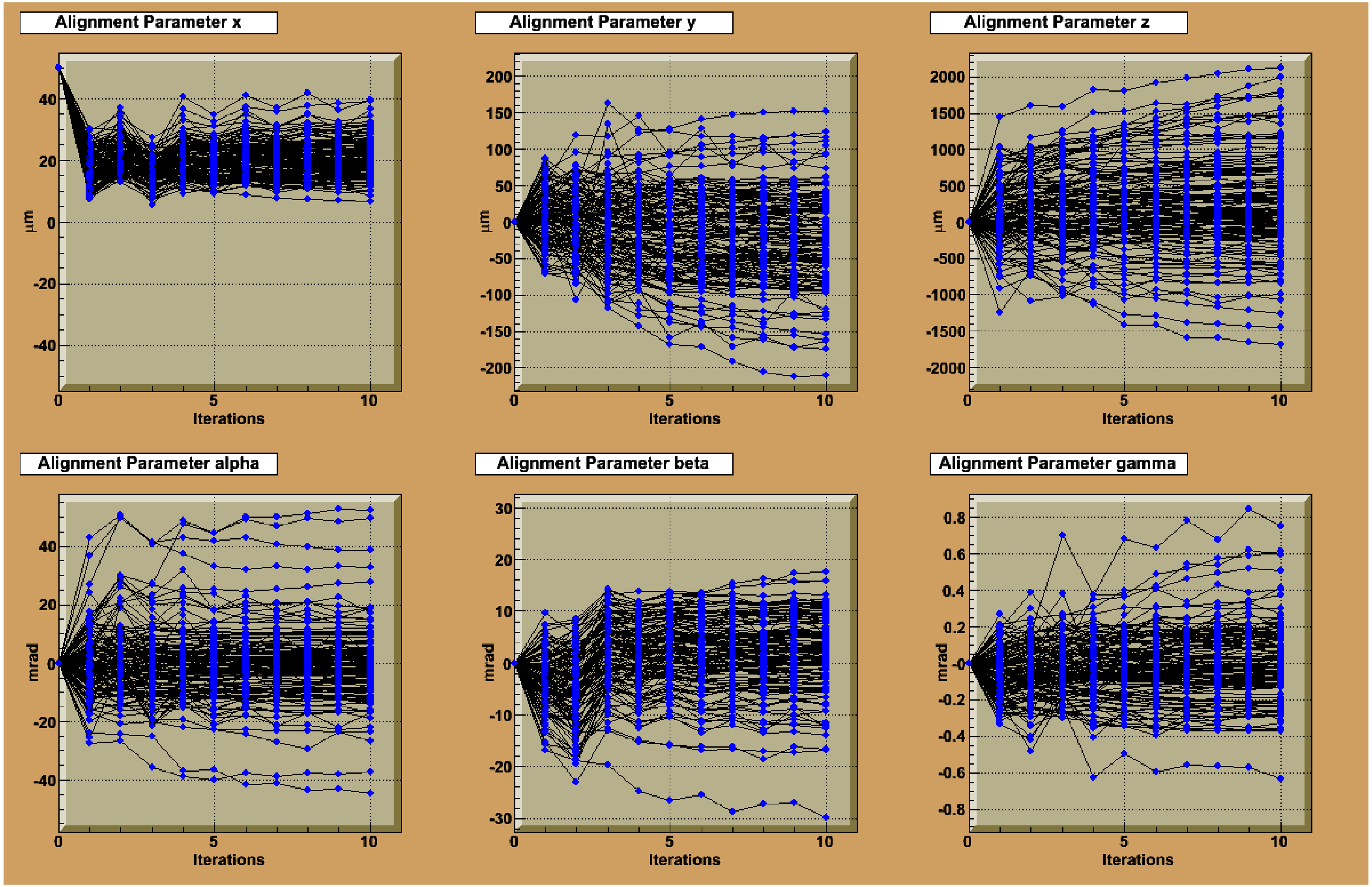

Figure 4.38: *Alignment parameter flow through ten iterations of all modules on end-cap A disk 3 with systematic initial misalignment. Initial alignment was SetDiskA3_01 (see table 4.2).*

## 4.5 Discussion

In section 1.5 and in chapter 2 we stated that track based detector alignment can improve the as-built alignment precision (see table 1.3) and is supposed to meet the required alignment precision (see table 1.4). Towards this end we successfully developed and implemented a track based alignment algorithm in the ATLAS analysis framework Athena. Within Athena our alignment algorithm is called Chi2AlignAlg and is an official part of the framework and located in package InnerDetector/InDetAlignment/SiRobustAlign. Chi2AlignAlg runs iteratively as shown in figure 2.2.

We tested the performance of Chi2AlignAlg in Athena release 10.5.0 and with the version of Chi2AlignAlg from package tag SiRobustAlign-00-00-24. For our tests we used a selected sample of 130k simulated single pion tracks. Our 95% CL estimates of the achievable alignment accuracy limits are listed in table 4.1. For two of the three translational degrees of freedom of SCT barrel modules we can reach an alignment accuracy comparable to the as-built alignment precision. For SCT end-cap modules we can improve the alignment accuracy of the most sensitive coordinate compared to the as-built alignment precision. None of the alignment accuracy limits listed in table 4.1 meets the required alignment accuracy (see table 1.4).

As we outlined in this chapter, we consider that this failure to meet the required alignment accuracy is not an intrinsic flaw of Chi2AlignAlg. The internal machinery of Chi2AlignAlg relies on comparatively new features of Athena that do not perform optimally yet. This results in a degradation of the quality of the input parameters, like outliers in the residual distributions, reduced track fit efficiency or wrong track fit error estimation. The degraded input propagates through the alignment machinery of Chi2AlignAlg and degrades the achievable alignment accuracy. Also the statistical error limits the achievable accuracy of all alignment parameters. To reduce the statistical error by a factor of 10 would require a track sample 100 times as large as the selected track sample, i.e. about 10 million tracks.

Athena is still rapidly evolving and additional functionality is constantly added to the framework. To be compatible with current releases and to profit from improved or added functionality Chi2AlignAlg has to change as well. For the purpose of this thesis we needed to make a snapshot of the status of Chi2AlignAlg and its capabilities and shortcomings at a certain stage. The performance of Chi2AlignAlg from tag SiRobustAlign-00-00-24 with Athena release 10.5.0 is not optimal. Some of the issues we raised here have been solved with release 11.0.0 or in the nightly builds of bug fix release 11.0.1. And for sure new problems have been introduced.

Chi2AlignAlg is not finished by a long way. Iterations within one Athena job, proper handling of cluster hits and usage of superior tracking tools are on the way to be implemented. Chi2AlignAlg is currently extended to include pixel alignment as well as SCT alignment. We only brushed the topic briefly in section 4.3 but the TrackCollection Chi2AlignAlg retrieves from StoreGate is the point where a track selection tool could assist Chi2AlignAlg. Only a basic track selection tool is already available and refinements to it are under development. The impact of track selection on alignment is under investigation.

# Chapter 5

# Conclusions

Detector alignment is an essential link in the chain from detector construction to final physics analysis. To fully exploit the physics potential of LHC the ATLAS detector must be very well understood. This implies among other things the need for very good calibration constants and alignment parameters.

Alignment precision of the 4088 SCT modules in the ATLAS Inner Detector must be better than the intrinsic resolution of 16 $\mu$m × 580 $\mu$m, perpendicular and along the readout strips, otherwise fitted track parameters will be dominated by systematic alignment uncertainties instead of limited statistics due to the finite number of hits on a track. We developed an iterative local $\chi^2$ algorithm for alignment of the ATLAS SCT subdetector, where "local" means that alignment corrections are derived on a module-by-module basis and correlations are only brought into play through refitting of track parameters during iterations.

We presented an in-depth derivation of the underlying mathematical concepts of our alignment approach in chapter 2. A generic formula for the track-based derivation of alignment parameters was obtained:

$$\Delta\vec{a}_k = -\left(\sum_{tracks} \frac{2}{\sigma^2_{i\,k}} \left(\frac{\partial r_{i\,k}(\vec{a}_k)}{d\vec{a}_{k\,0}}\right)\left(\frac{\partial r_{i\,k}(\vec{a}_k)}{d\vec{a}_{k\,0}}\right)^T\right)^{-1} \cdot \left(\sum_{tracks} \frac{2}{\sigma^2_{i\,k}} \left(\frac{\partial r_{i\,k}(\vec{a}_k)}{d\vec{a}_{k\,0}}\right) r_{i\,k}(\vec{a}_{k\,0})\right) \tag{5.1}$$

Equation 5.1 is generic in a sense that it imposes no constrains on the underlying tracking model, residual definition or dimensionality of the alignment problem in question.

For a proof of principle we developed a prototype program in ROOT as a testbed for various alignment scenarios in a controlled environment. Valuable lessons were learned there – e.g. on error estimation and the choice of residual definition – and the results propagated into the ATLAS software framework Athena.

The main part of this thesis was the successful development and implementation of an iterative alignment algorithm for the ATLAS SCT subdetector. We started development with Athena release 9.0.3 and Chi2AlignAlg is now an official part of the ATLAS software framework since Athena release 11.0.0. The current Event Data Model (EDM) of ATLAS and recent developments in the tracking realm of Athena allow a fast and efficient iteration cycle as pattern recognition for track finding is separated from track fitting.

Chi2AlignAlg is a ready-to-use tool for alignment of the ATLAS SCT subdetector and we presented first tests and their results in chapter 4. We used a selected sample of 130k single pion tracks that was simulated with a perfectly aligned ATLAS detector geometry for these tests. We observed that a very firm selection of input tracks is necessary to ensure stability and convergence of our iterative alignment algorithm. The two selection criteria we employed on the single pion sample were a cut on the minimum track momentum and track isolation, i.e. rejection of events with secondary tracks from hadronic interaction. Various studies are still ongoing to investigate the impact of track selection on track-based alignment.

We were able to estimate with 95% CL the alignment accuracy limits for the present setup of Athena and Chi2AlignAlg. With the limited statistics of the selected track sample the alignment accuracy limits are comparable to the as-built alignment precision but do not meet the required alignment precision yet. Our tests with misaligned detector geometry showed that the applied misalignments are recovered within the estimated alignment accuracy limits.

Improvements of the tracking tools, database handling tools and geometry description used by Chi2AlignAlg will directly improve the performance of our algorithm. Nonetheless there are various possibilities for intrinsic performance improvements. For example the analytical calculation of derivatives could replace the current numerical one. As well there is still the task of integrating the alignment of the ATLAS pixel detector into Chi2AlignAlg.

Furthermore, due to the flexibility gained by working within Athena it is possible to test this approach with a real detector setup and measured data, namely the ATLAS Combined Testbeam (CTB). A crosscheck with existing CTB alignment is then possible. Also in the upcoming combined TRT barrel and SCT barrel cosmics run Chi2AlignAlg will be tested with data. The migration from tests with simulated particle tracks to alignment runs with measured data is a vital step to improve Chi2AlignAlg and to be prepared for day 1 when LHC will start operation and ATLAS will measure the first collision data.

# Appendix A

# Additional calculations

## A.1 Detailed derivation of alignment equation

We left out some details concerning the derivation of equation (2.5). A derivation with more intermediate steps is presented here. We will make use of the following identity:

$$\frac{d(\vec{x}^{\,T}\cdot A\cdot\vec{x})}{d\vec{x}} = A^T\cdot\vec{x} + A\cdot\vec{x} = 2A\cdot\vec{x} \quad \text{(for } A \text{ symmetric)} \tag{A.1}$$

$$\begin{aligned}
&\sum_{tracks}\frac{d}{d\vec{a}_0}\left(\vec{r_i}^{\,T}(\vec{a})\cdot V_i^{-1}\cdot\vec{r}_i(\vec{a})\right) + \sum_{tracks}\frac{d^2}{d\vec{a}_0^2}\left(\vec{r_i}^{\,T}(\vec{a})\cdot V_i^{-1}\cdot\vec{r}_i(\vec{a})\right)\Delta\vec{a} = \\
&= \sum_{tracks}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot 2V_i^{-1}\cdot\vec{r}_i(\vec{a}_0) + \sum_{tracks}\frac{d}{d\vec{a}_0}\left(\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot 2V_i^{-1}\cdot\vec{r}_i(\vec{a}_0)\right)^T\Delta\vec{a} = \\
&= \sum_{tracks}\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot 2V_i^{-1}\cdot\vec{r}_i(\vec{a}_0) + \\
&\qquad + \left[\sum_{tracks}\left(\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot 2V_i^{-1}\cdot\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)^T\right) + \sum_{tracks}\left(\vec{r_i}^{\,T}(\vec{a})\cdot 2V_i^{-1}\cdot\underbrace{\frac{d^2\vec{r}_i(\vec{a})}{d\vec{a}_0^2}}_{=0}\right)\right]\Delta\vec{a} \\
&= \sum_{tracks}2\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot V_i^{-1}\cdot\vec{r}_i(\vec{a}_0) + \left(\sum_{tracks}2\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)\cdot V_i^{-1}\cdot\left(\frac{d\vec{r}_i(\vec{a})}{d\vec{a}_0}\right)^T\right)\Delta\vec{a}
\end{aligned} \tag{A.2}$$

In A.2 we used the $\frac{d}{d\vec{a}_0} = \frac{d}{d\vec{a}}\Big|_{\vec{a}=\vec{a}_0}$ shorthand notation from chapter 2 again. $\frac{d^2\vec{r}_i(\vec{a})}{d\vec{a}_0^2} = 0$ is the case if $\vec{r}$ is a linear function of $\vec{a}$. This assumption of a linear dependence is at the heart of every linear least square fit [44].

## A.2 Variance of top hat distribution

The variance of an arbitrary distribution $f(x)$ is defined as

$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 \tag{A.3}$$

With $f(x)$ defined as a top hat function as follows

$$f(x) = \begin{cases} \frac{1}{a} & x = [0, a] \\ 0 & else \end{cases} \tag{A.4}$$

This leads to

$$\langle x^2 \rangle = \frac{\int x^2 f(x)\, dx}{\int f(x)\, dx} = \int\limits_0^a x^2 \frac{1}{a}\, dx = \frac{1}{3} a^2 \tag{A.5}$$

and

$$\langle x \rangle = \frac{\int x f(x)\, dx}{\int f(x)\, dx} = \int\limits_0^a x \frac{1}{a}\, dx = \frac{1}{2} a \tag{A.6}$$

And thus to

$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 = \frac{a^2}{12} \quad \Rightarrow \quad \sigma = \frac{a}{\sqrt{12}} \tag{A.7}$$

# Appendix B

# Additional plots

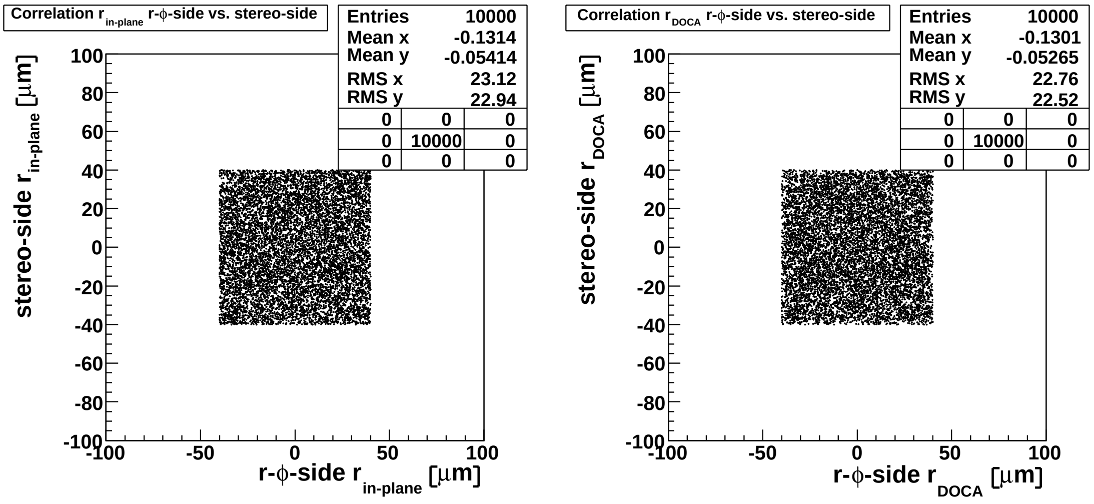

Figure B.1: *Correlation between residual measurement of r-$\phi$-side (x-Axis) and stereo-side (y-Axis) of the SCT module. The left side shows the correlation for $r_{in-plane}$ and the right side for the $r_{DOCA}$. One can clearly see that for both residual definitions the r-$\phi$- and stereo-side measurement are uncorrelated.*

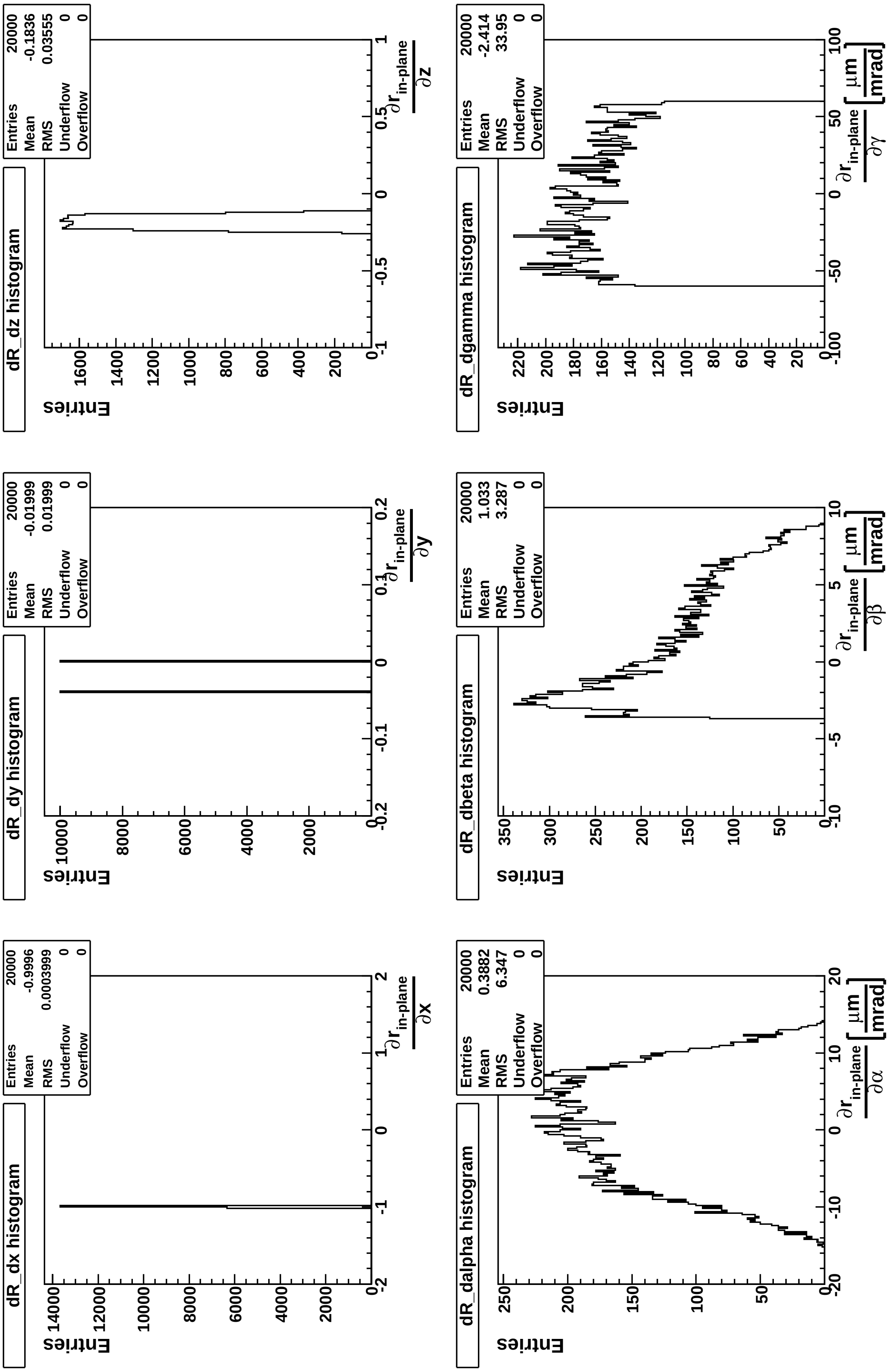

Figure B.2: *Distribution of* $\frac{\partial r_{in-plane}}{\partial \alpha}$ *for 10000 hits on both sides of the SCT module with gaussian shaped* $r_{in-plane}$ *distribution.*

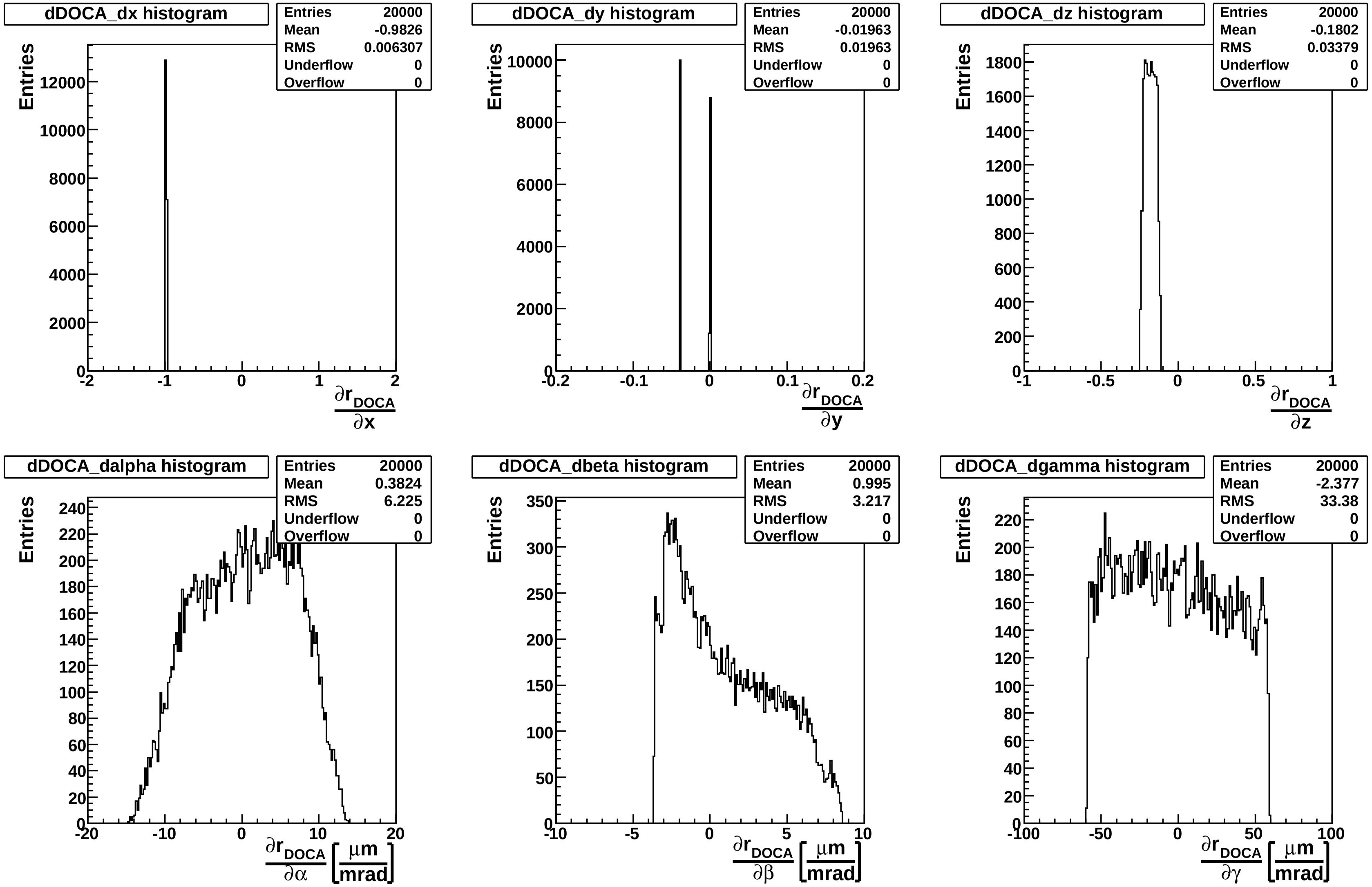

Figure B.3: *Distribution of* $\frac{\partial r_{DOCA}}{\partial a}$ *for 10000 hits on both sides of the SCT module with gaussian shaped* $r_{DOCA}$ *distribution.*

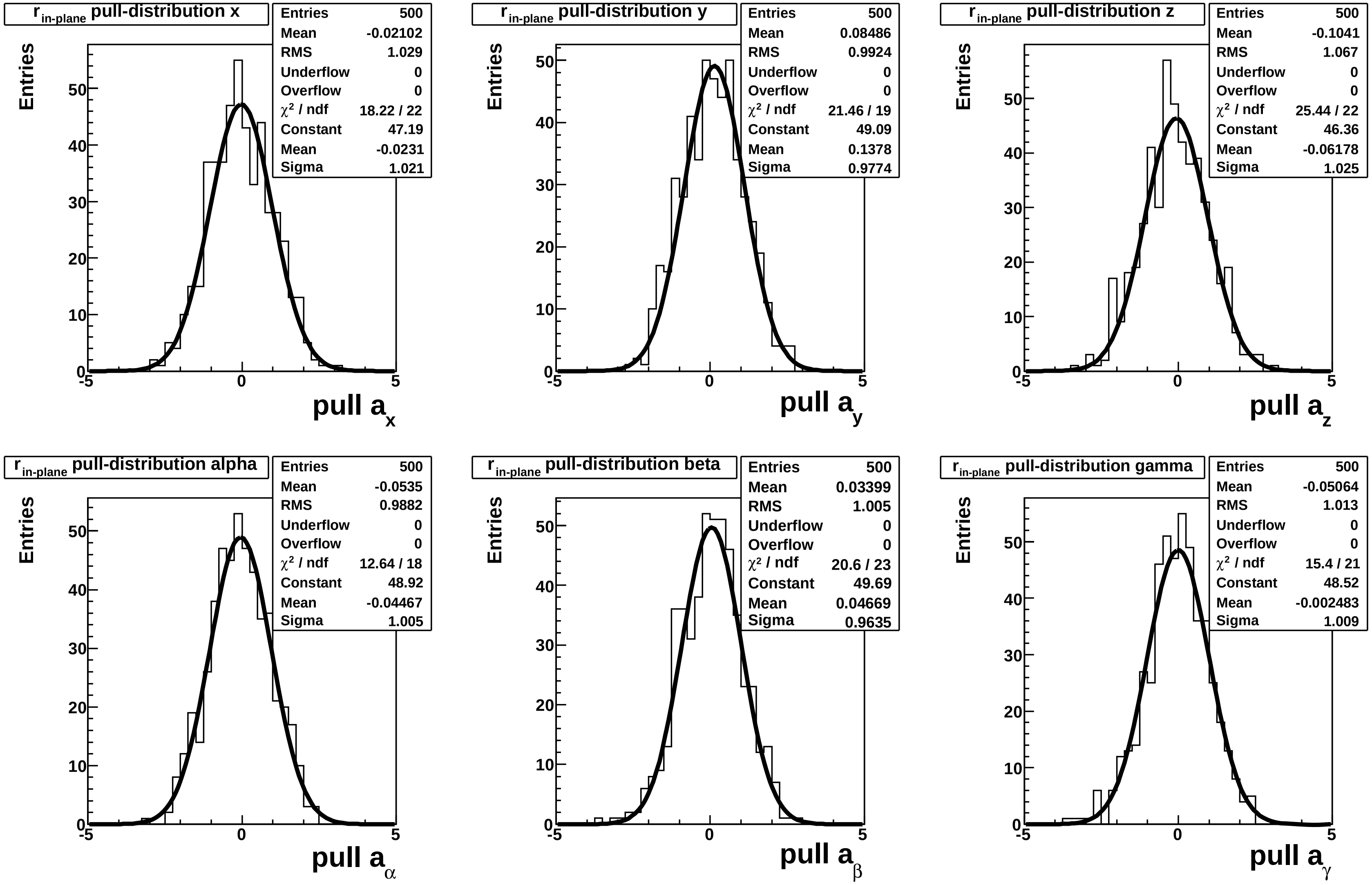

Figure B.4: *Pull distributions for the six alignment parameters with gaussian shaped* $r_{in-plane}$ *distribution.*

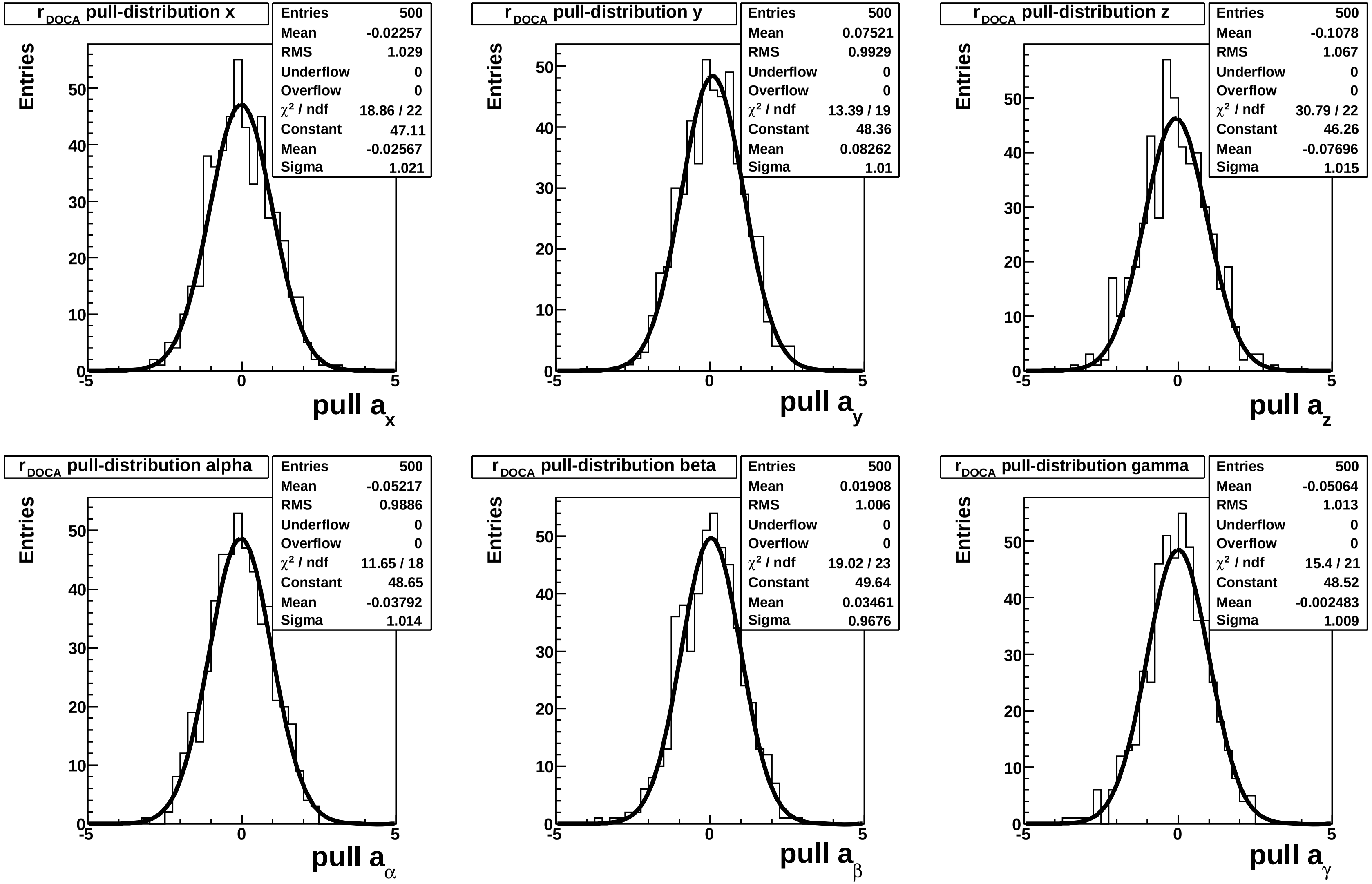

Figure B.5: *Pull distributions for the six alignment parameters with gaussian shaped* $r_{DOCA}$ *distribution.*

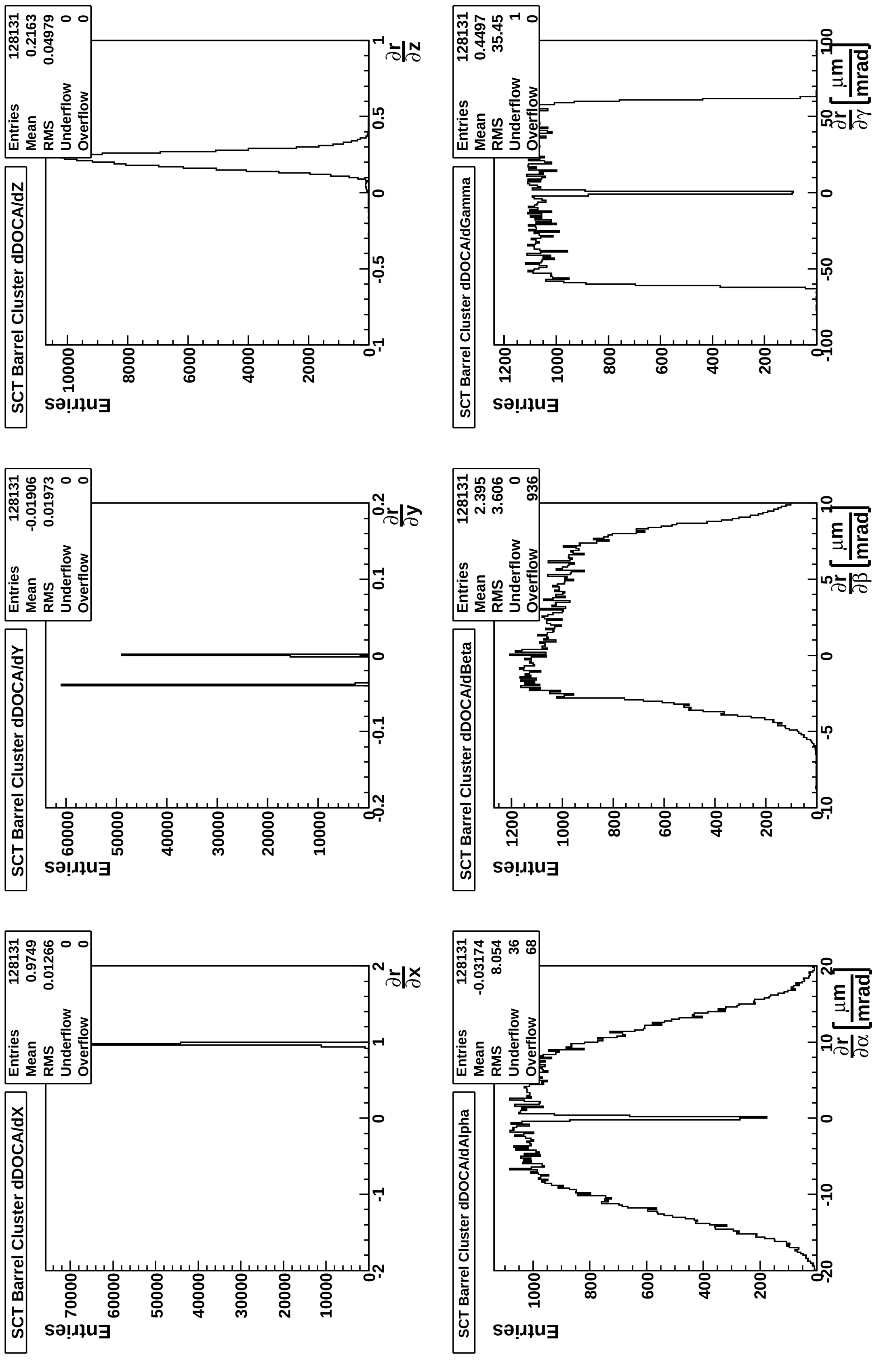

Figure B.6: *Distributions of the residual derivatives* $\frac{\partial r_{DOCA}}{\partial a_i}$ *of cluster hits for all SCT barrel modules The coordinates (x,y,z) denote the local coordinate axes. (α,β,γ) denote rotation angles around these axes.*

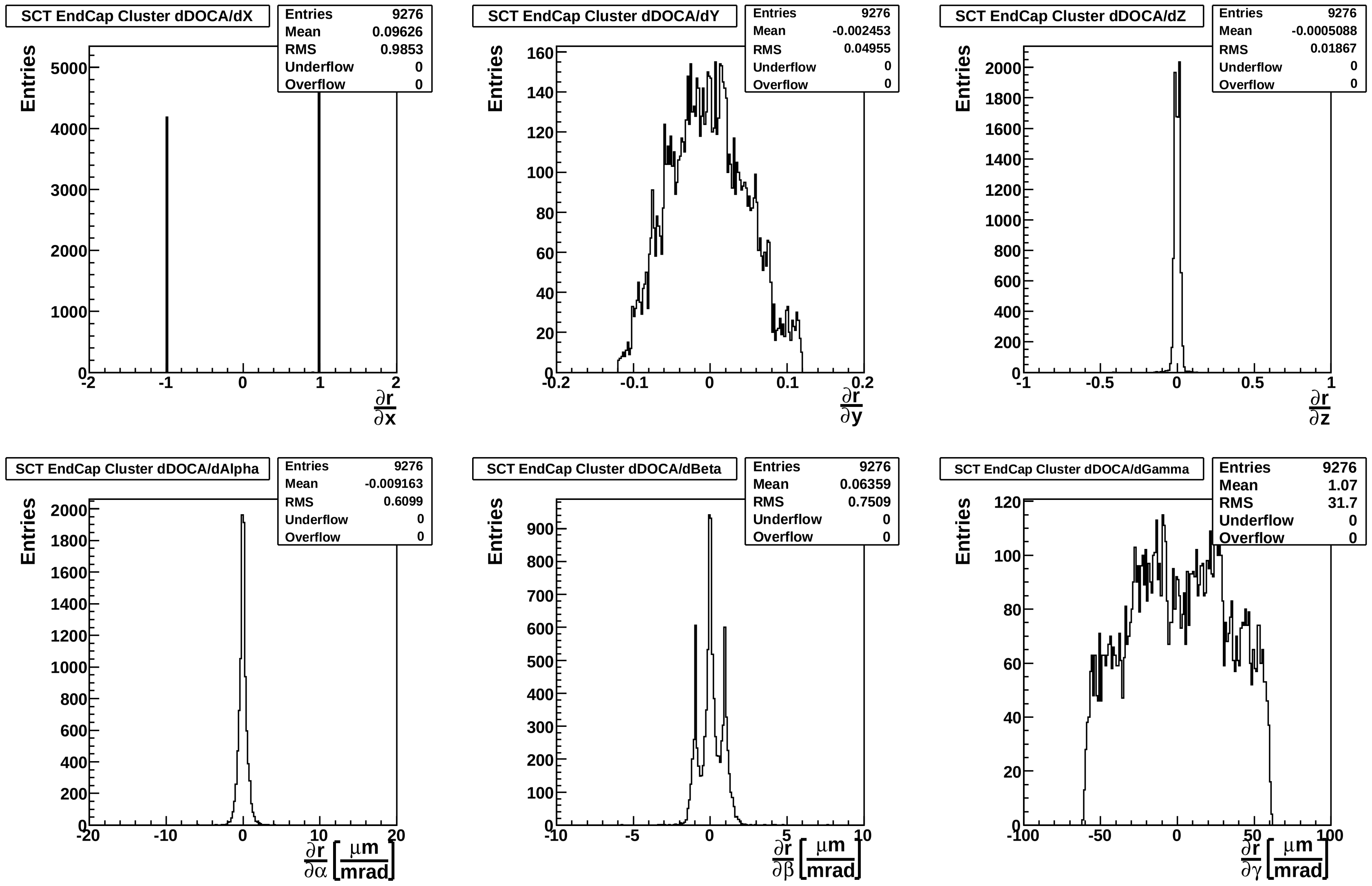

Figure B.7: *Distributions of the residual derivatives* $\frac{\partial r_{DOCA}}{\partial a_i}$ *of cluster hits for all SCT end-cap modules The coordinates (x,y,z) denote the local coordinate axes.* $(\alpha,\beta,\gamma)$ *denote rotation angles around these axes.*

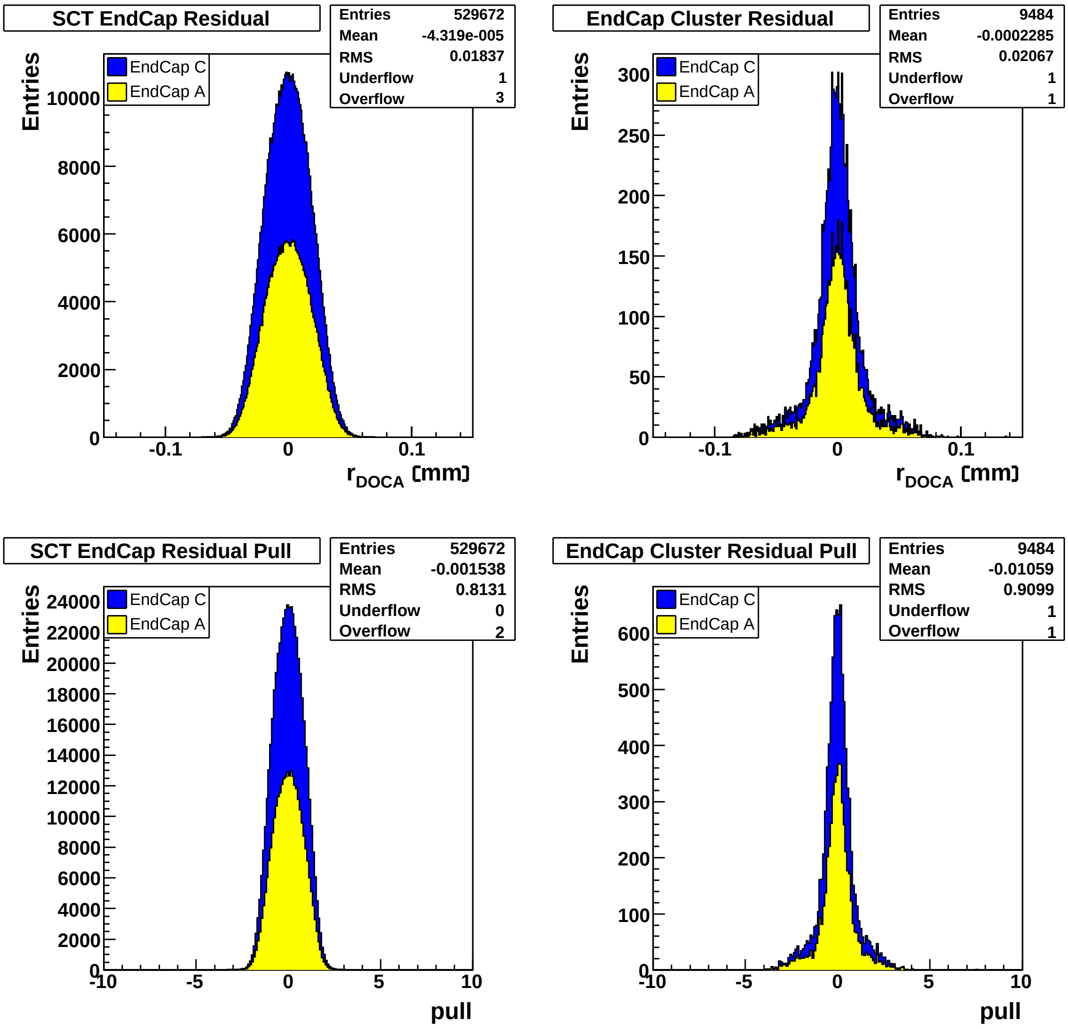

Figure B.8: *Residual and pull distributions of single strip hits and cluster hits on all SCT end-cap modules with xKalman tracks.*

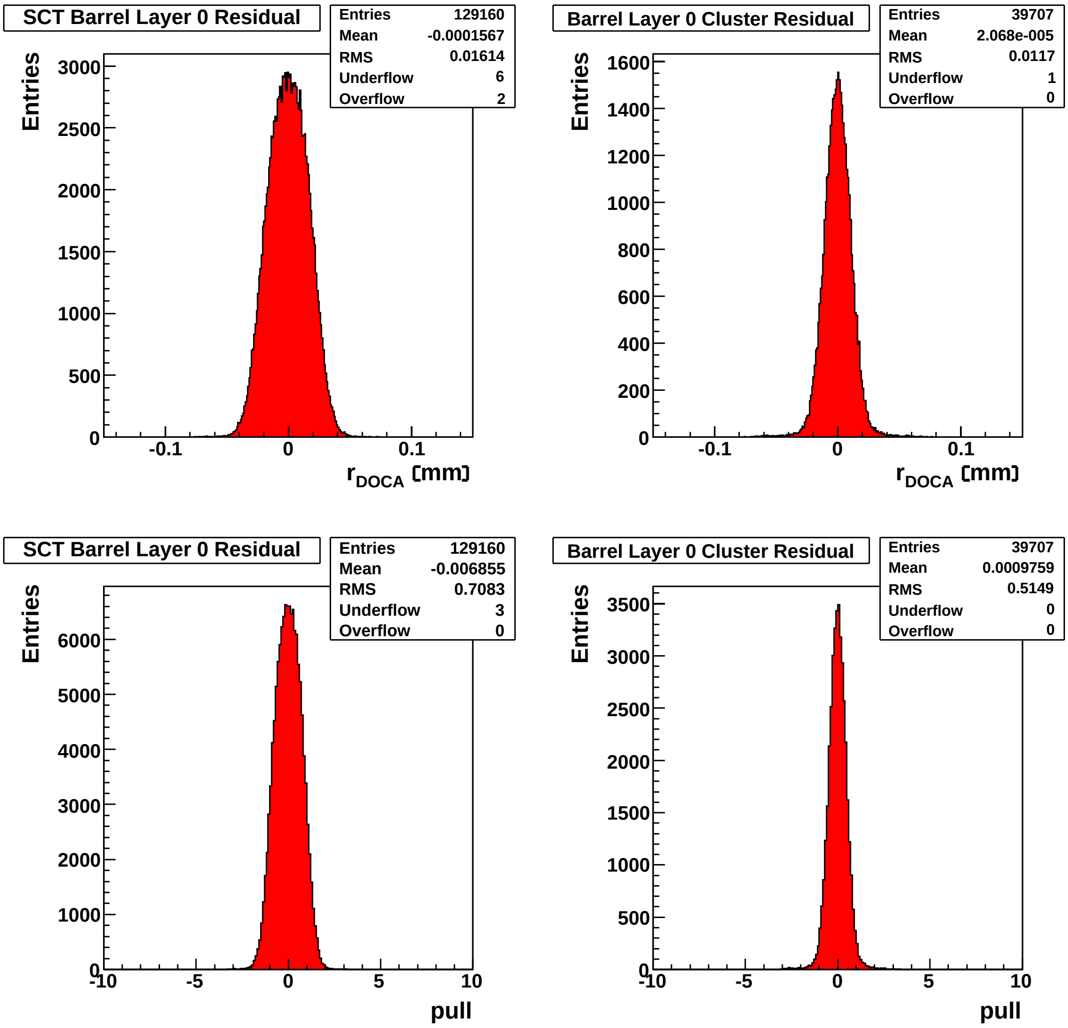

Figure B.9: *Residual and pull distributions of single strip hits and cluster hits on all SCT modules of barrel layer 0 with xKalman tracks. The distribution is centered at zero.*

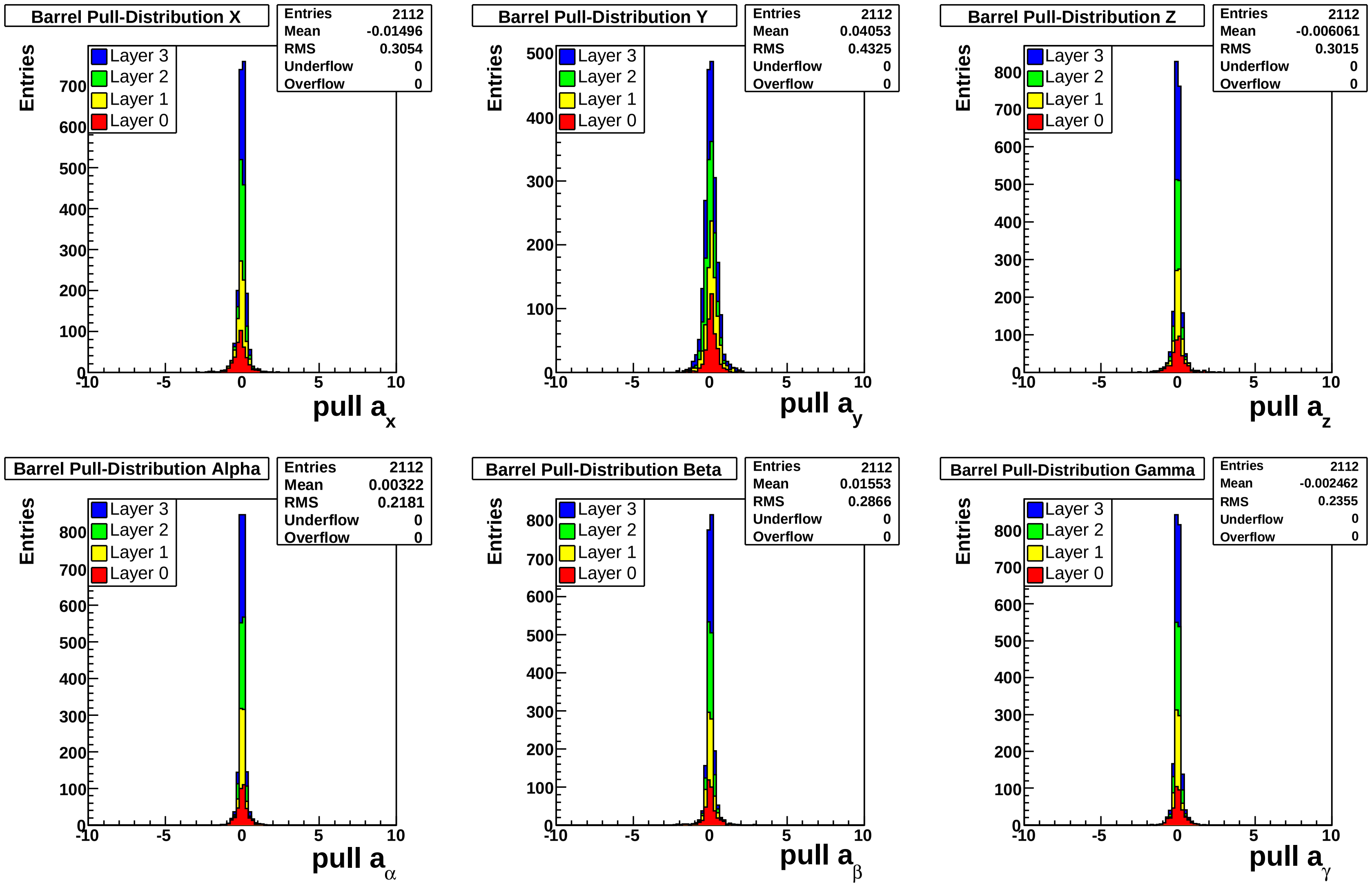

Figure B.10: *Differential pull distributions from iteration 5 for all six degrees of freedom of all SCT barrel modules.*

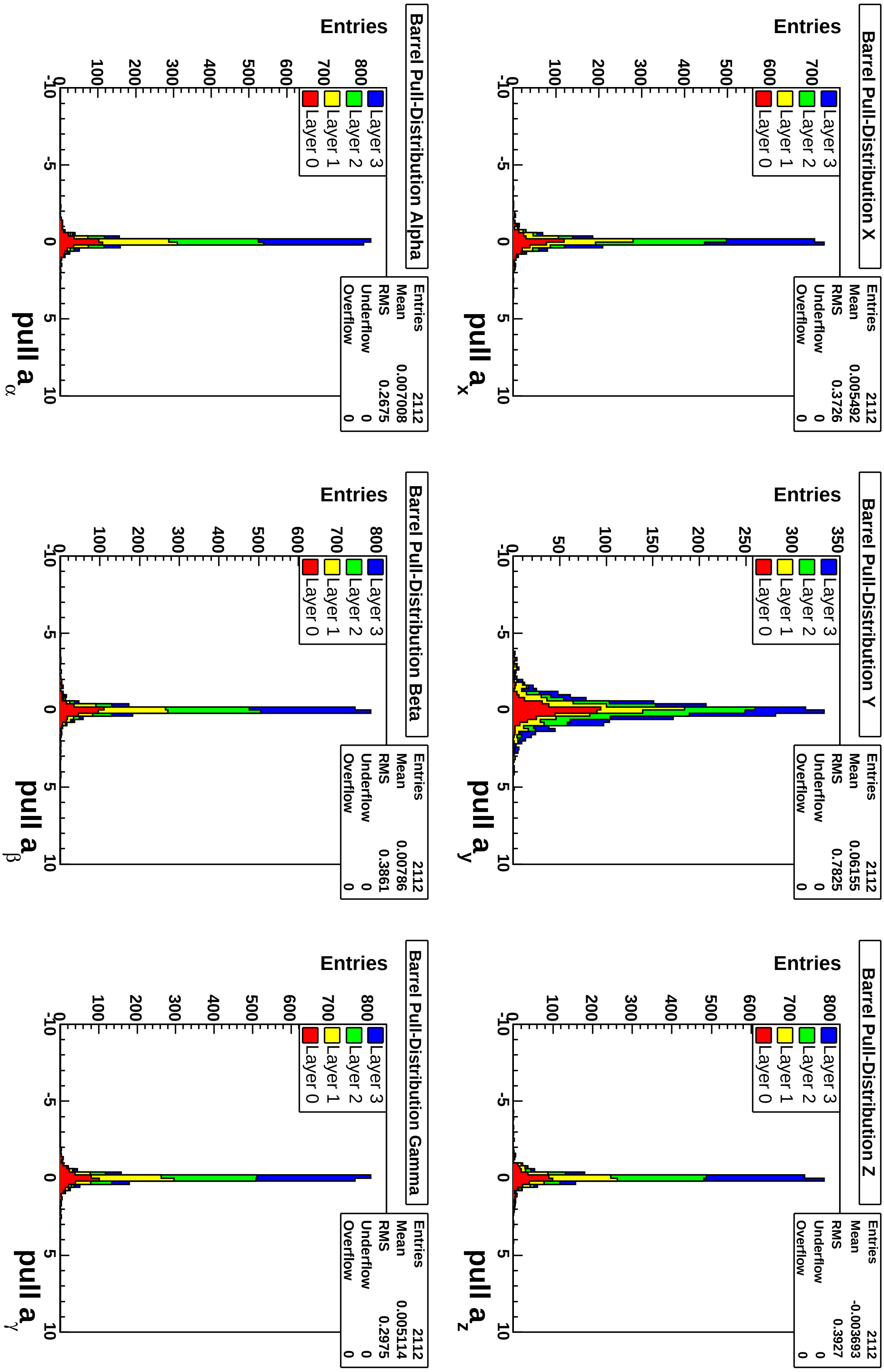

Figure B.11: *Differential pull distributions from iteration 10 for all six degrees of freedom of all SCT barrel modules.*

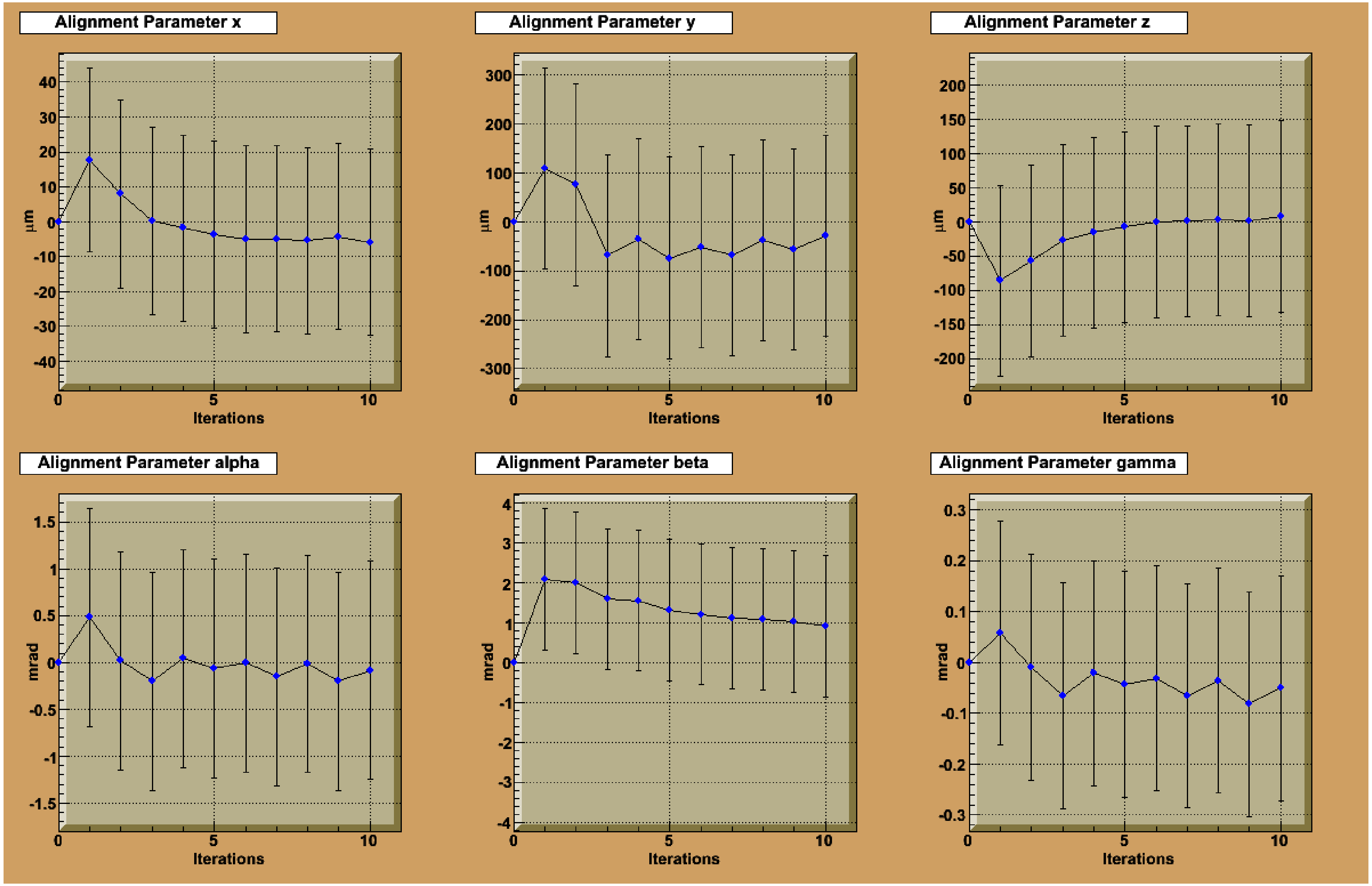

Figure B.12: *Alignment parameter flow of module 2/2/0/2/8/-4 (barrel layer 2) through ten iterations The module converges on a stable set of six alignment parameters.*

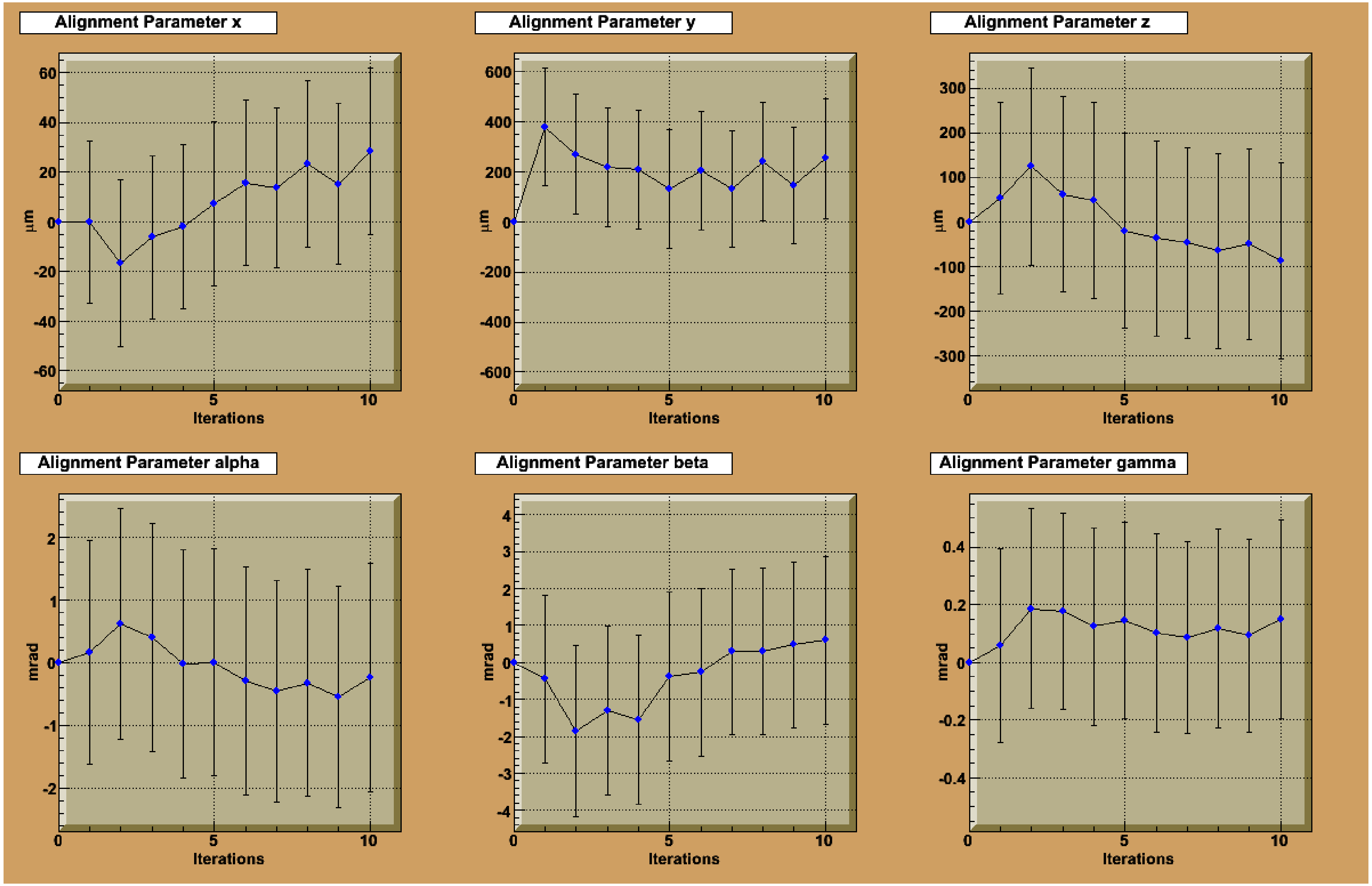

Figure B.13: *Alignment parameter flow of module 2/2/0/3/42/5 (barrel layer 3) through ten iterations The module converges on a stable set of six alignment parameters.*

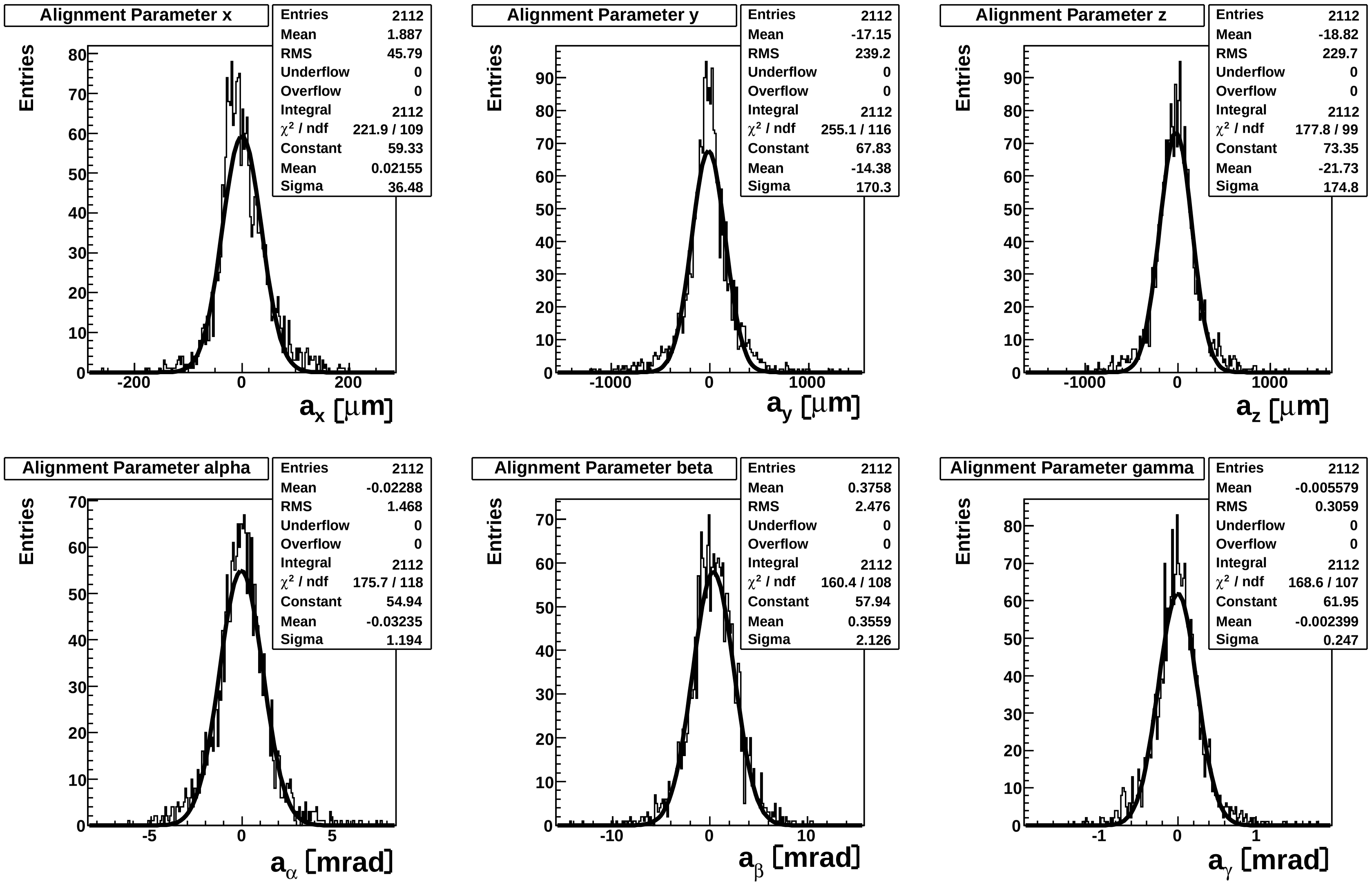

Figure B.14: *Alignment parameter distributions from iteration 10 for all six degrees of freedom of all SCT barrel modules.*

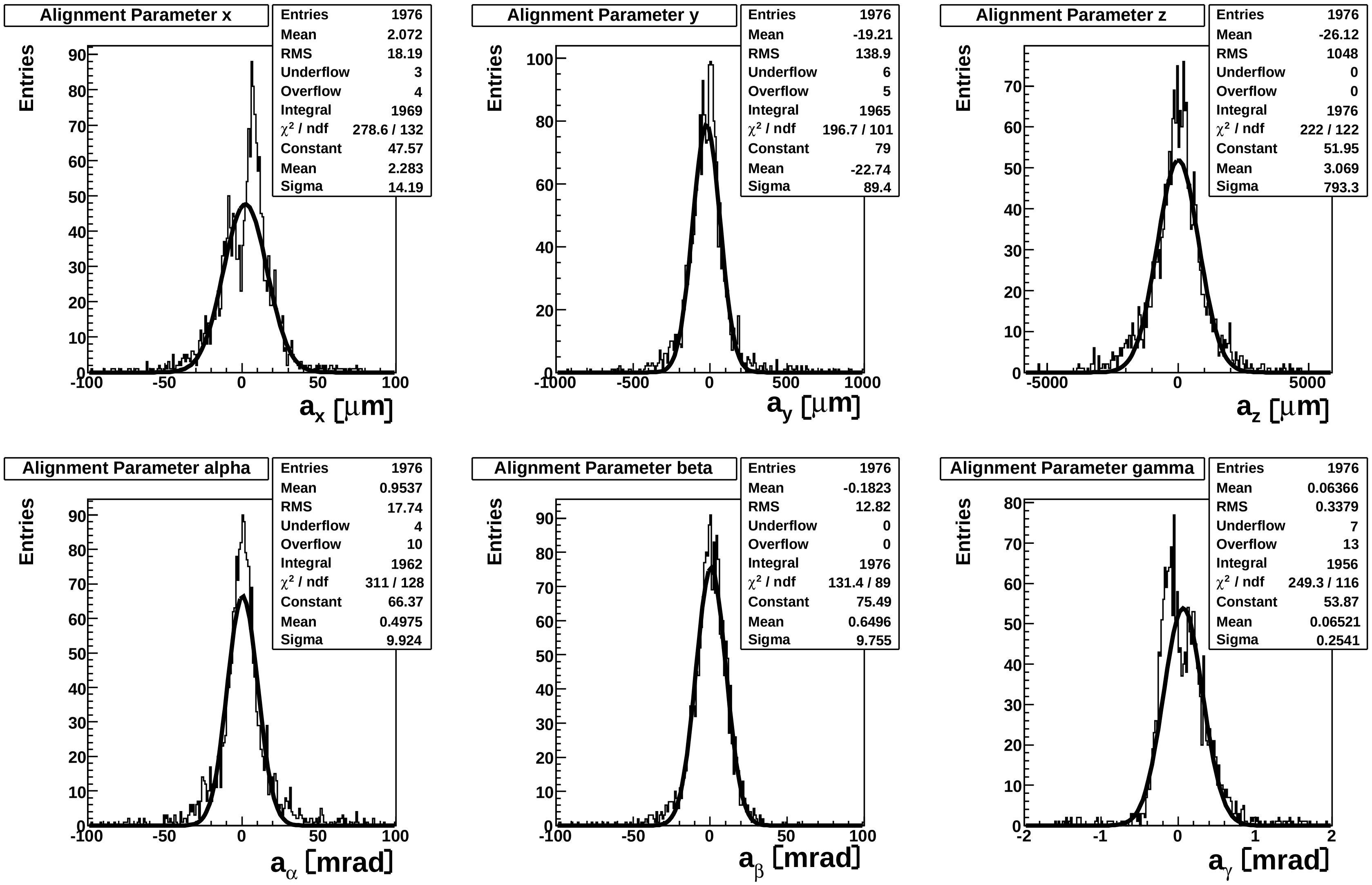

Figure B.15: *Alignment parameter distributions from iteration 10 for all six degrees of freedom of all SCT end-cap modules*

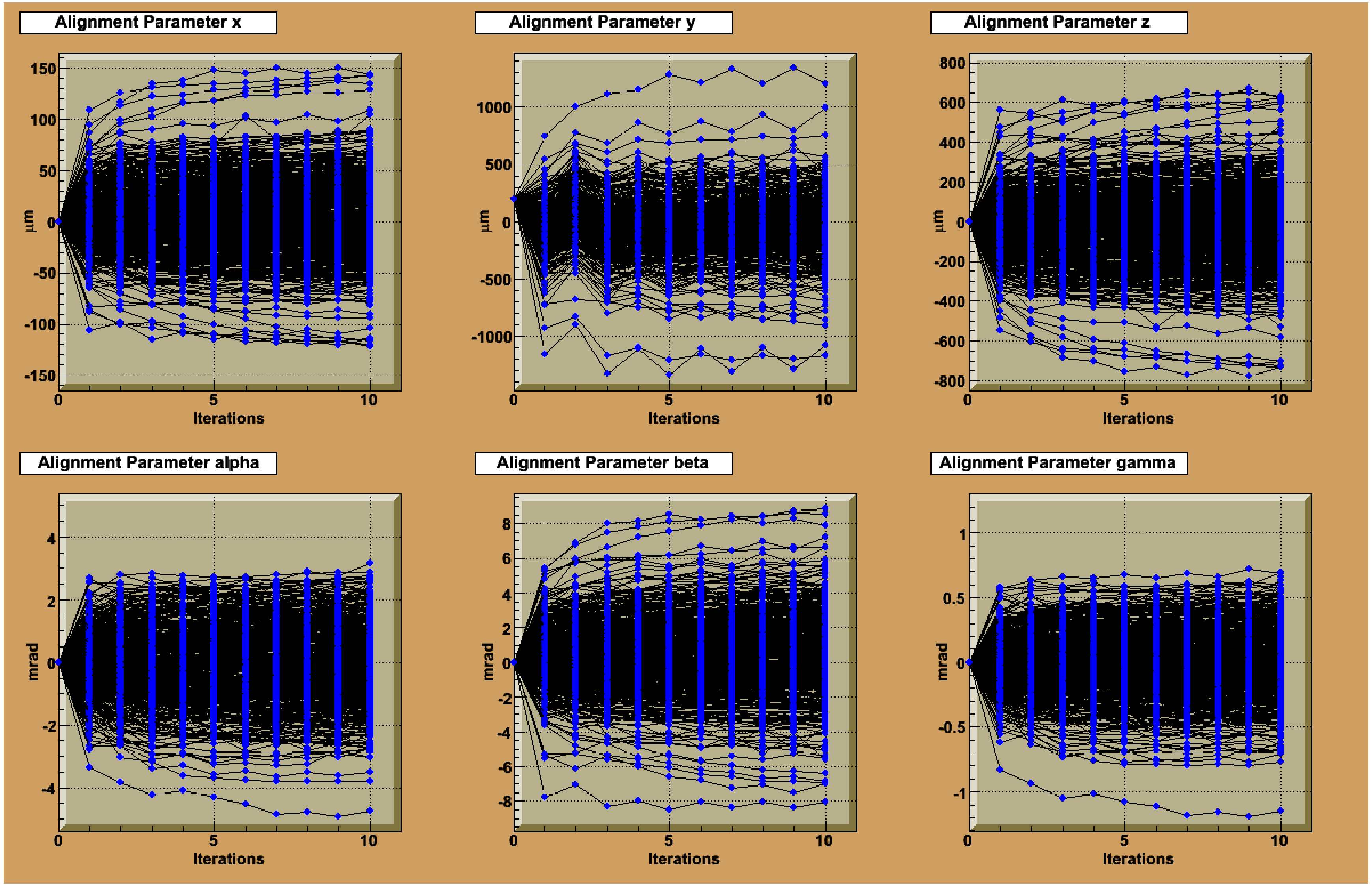

Figure B.16: *Alignment parameter flow through ten iterations of all modules on barrel layer 2 with systematic initial misalignment. Initial alignment was SetLayer2_02 (see table 4.2).*

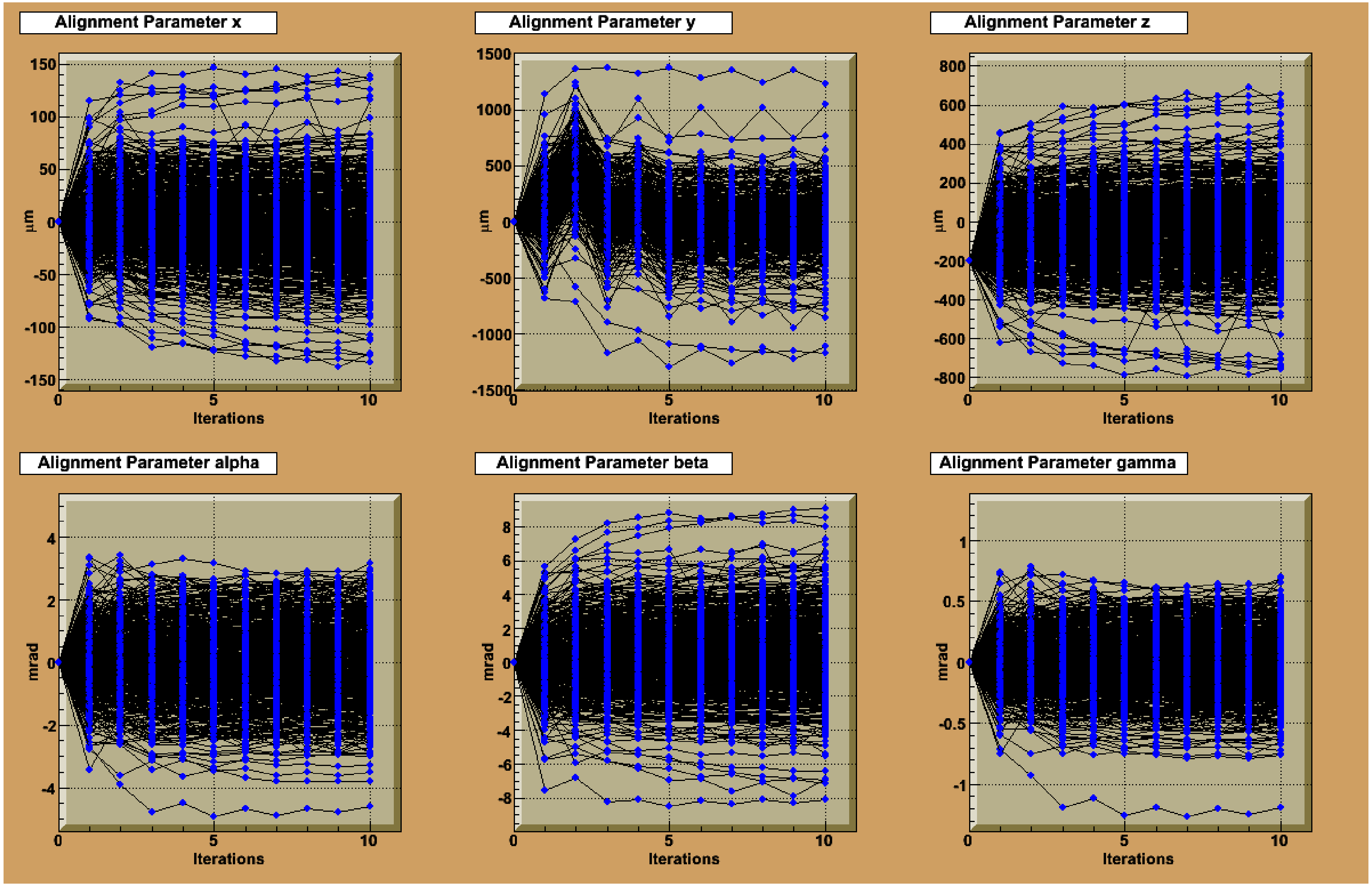

Figure B.17: *Alignment parameter flow through ten iterations of all modules on barrel layer 2 with systematic initial misalignment. Initial alignment was SetLayer2_03 (see table 4.2).*

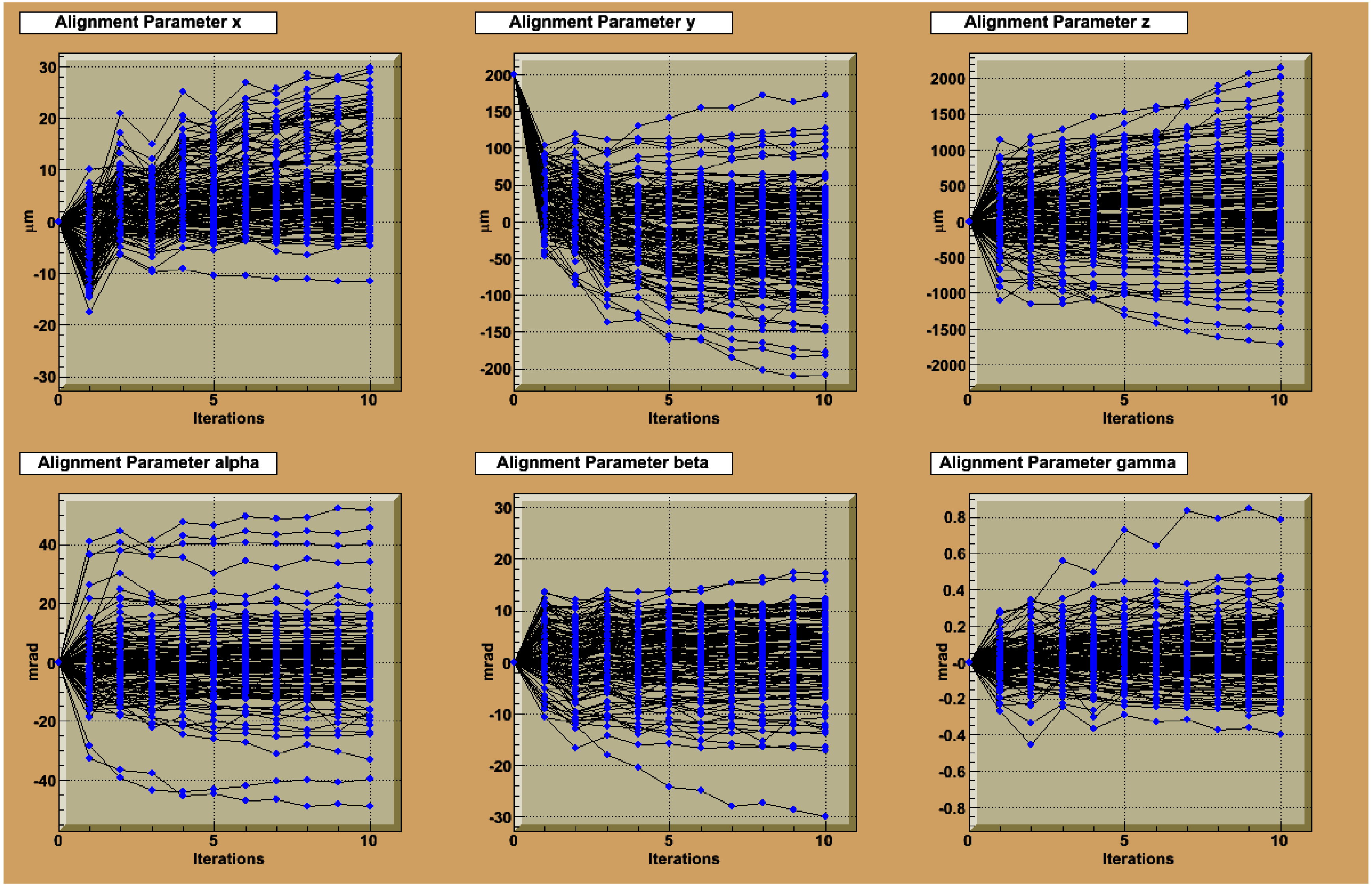

Figure B.18: *Alignment parameter flow through ten iterations of all modules on end-cap A disk 3 with systematic initial misalignment SetDiskA3_02 (see table 4.2).*

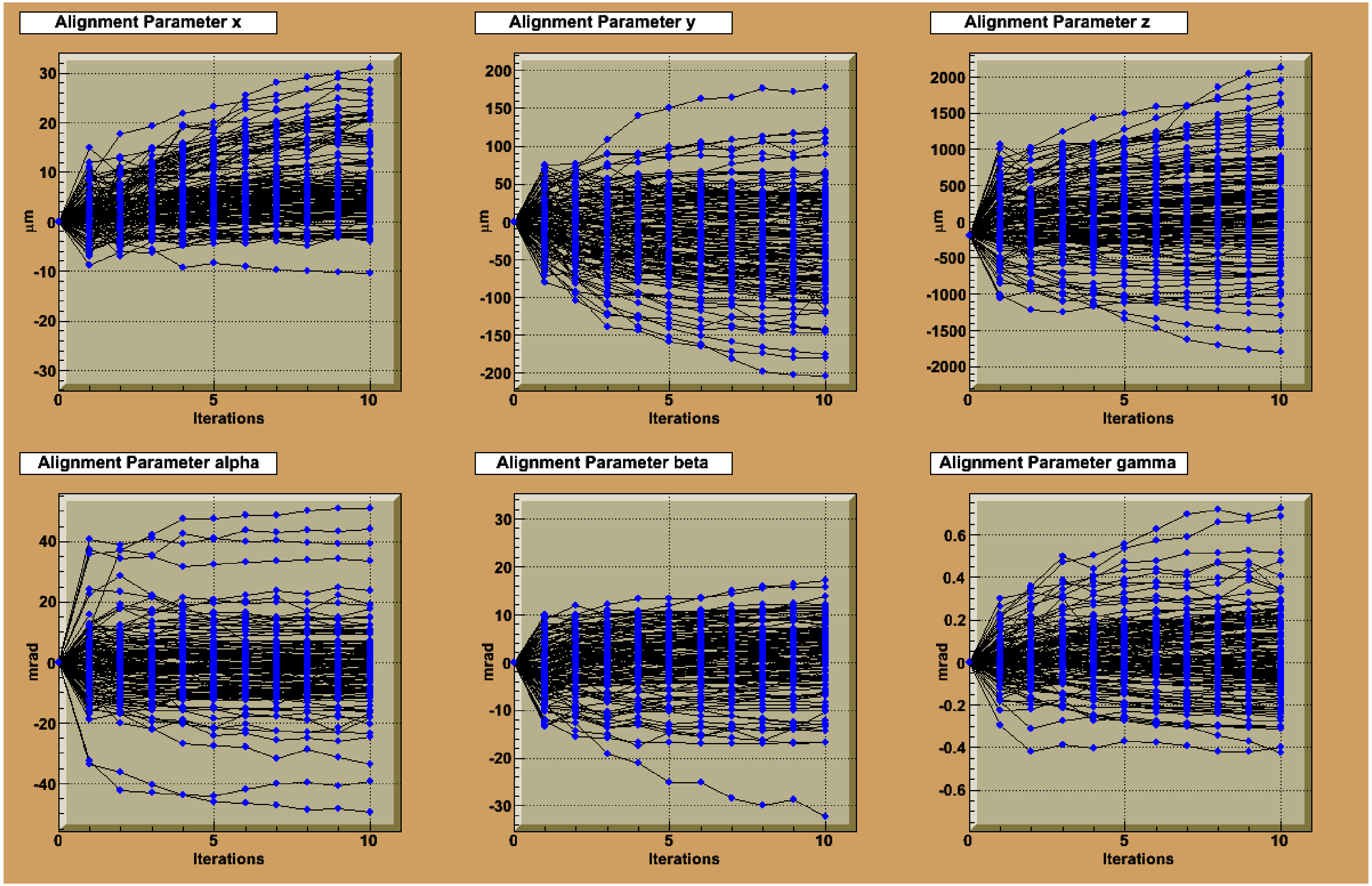

Figure B.19: *Alignment parameter flow through ten iterations of all modules on end-cap A disk 3 with systematic initial misalignment SetDiskA3_03 (see table 4.2).*

# List of Figures

# List of Tables

# Bibliography

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

[2] S. Glashow, *Partial Symmetries of Weak Interactions*, Nucl. Phys. **22** (1961) 579.

[3] S. Weinberg, *A Model of Leptons*, Phys. Rev. Lett. **19** (1967) 1264.

[4] A. Salam, Elementary particle theory, p. 367, Almqvist and Wiksell, Stockholm, 1968.

[5] H. Fritzsch and M. Gell-Mann, Proc. XVI Int. Conf. on High Energy Physics (Fermilab 1972) , Jackson, J.D. and Roberts, A.

[6] M. Gell-Mann, *A schematic model of baryons and mesons*, Phys. Rev. Lett. **8** (1964) 214.

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

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

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

[10] C. Rembser, *LHC - Machine and Detectors*, `http://maria-laach.physik.uni-siegen.de/2005/Folien/Rembser/ML2005_Rembser_01.ppt`, Lecture at 37. Herbstschule für Hochenergiephysik 2005, Maria Laach.

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

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

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

[14] The TOTEM Collaboration, *TOTEM: Technical Design Report - Total Cross Section, Elastic Scattering and Diffraction Dissociation at the Large Hadron Collider at CERN*, CERN-LHCC-2004-002.

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

[16] The ATLAS Collaboration, *ATLAS Detector and physics performance: Technical design report*, CERN-LHCC-99-15.

[17] J. Ellis, *Beyond the Standard Model for hillwalkers*, (1998), CERN-PH-TH-98-329, hep-ph/9812235.

[18] S. Eidelman et al. (Particle Data Group Collab.), *Review of Particle Physics*, Physics Letters B **592** (2004) 1+.

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

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

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

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

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

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

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

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

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

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

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

[30] 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.

[31] J. Schieck, *Production of endcap modules for the ATLAS semiconductor tracker*, Nuclear Instruments and Methods in Physics Research **A552** (2005) 208.

[32] R. Nisius, *End-cap Modules for the ATLAS SCT*, Nuclear Instruments and Methods in Physics Research **A530** (2004) 44.

[33] Campabadal, F. et al., *Beam tests of ATLAS SCT silicon strip detector modules*, Nuclear Instruments and Methods in Physics Research **A538** (2005) 384.

[34] *ATLAS Detector Description Database*, `http://atlas-php.web.cern.ch/atlas-php/DDDB/`.

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

[36] S. Gibson, *SCT Survey Data*, `http://agenda.cern.ch/askArchive.php?base=agenda&categ=a055395&id=a055395s2t9/transparencies`, Presentation given at Inner Detector Performance Workshop October 2005, CERN.

[37] Terada, S. et al., *Design and development of a work robot to place ATLAS SCT modules onto barrel cylinders*, Nuclear Instruments and Methods in Physics Research **A541** (2005) 144.

[38] R. Hawkings and P. Jez, *Initial alignment of the ATLAS SCT and pixel detectors and its effect on track-finding*, ATLAS internal Note, ATL-INDET-INT-2005-002 (2005).

[39] S. Haywood, *Offline Alignment & Calibration of the Inner Detector*, ATLAS internal Note, ATL-INDET-2000-005 (2000).

[40] R. Frühwirth et al., *Data Analysis Techniques for High-Energy Physics*, Cambridge University Press, 2nd edition, 2000.

[41] *ATLAS Tracking Event Data Model*, `http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/Tracking/TrkDoc/EDM/`, ATLAS Note in preparation.

[42] 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).

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

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

[45] R. Brun et al., *ROOT Users Guide 4.04*, 2005.

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

[47] S. Brandt, *Datenanalyse mit statistischen Methoden und Computerprogrammen*, Bibliographisches Institut, 2nd edition, 1981.

[48] M. Feindt, *Einführung in die Datenanalyse*, `http://www-ekp.physik.uni-karlsruhe.de/~feindt/MariaLaach.pdf`, Lecture at 36. Herbstschule für Hochenergiephysik 2004, Maria Laach.

[49] The ATLAS Collaboration, *ATLAS Computing: Technical Design Report*, CERN-LHCC-2005-022.

[50] *ATLAS CVS repository*, `http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/InnerDetector/InDetAlignment/SiRobustAlign/`, Location of SiRobustAlign in the CVS repository.

[51] *ATLAS tag collector*, `http://atlastagcollector.in2p3.fr/`.

[52] S. Menke, *Single Particle Files from Post-Rome Production*, `http://menke.home.cern.ch/menke/cgi-bin/hec/postrome.sh`.

[53] F. Akesson, *Track Reconstruction with xKalman*, `https://uimon.cern.ch/twiki/bin/view/Atlas/InDetXKalman`, ATLAS Computing Wiki.

[54] K. Assamagan et al., *Final Report of the ATLAS AOD/ESD Definition Task Force*, ATLAS internal Note, ATLAS-SOFT-2004-006 (2004).

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

[56] A. Salzburger, *TrkExtrapolation*, `https://uimon.cern.ch/twiki/bin/view/Atlas/TrkExtrapolation`, ATLAS Computing Wiki.

[57] A. Salzburger, *The Track Extrapolation package in the new ATLAS Tracking Realm*, `http://indico.cern.ch/contributionDisplay.py?contribId=109&sessionId=24&confId=0`, Proceedings CHEP 04.

[58] W. Liebig, *TrkMeasurementUpdator*, `https://uimon.cern.ch/twiki/bin/view/Atlas/TrkMeasurementUpdator`, ATLAS Computing Wiki.

[59] A. Salzburger, private communication.

[60] W. Liebig, private communication.

[61] E. Moyse, private communication.

[62] T. Cornelissen, *Savannah bug report*, `http://savannah.cern.ch/bugs/?func=detailitem&item_id=10865`, known software bug of release 10.5.0.

[63] R. Barlow, *Statistics: a guide to the use of statistical methodes in the physical sciences*, John Wiley & Sons, 1989.