Progress Report on Inner Detector Alignment

Roland Härtel

ATLAS Meeting

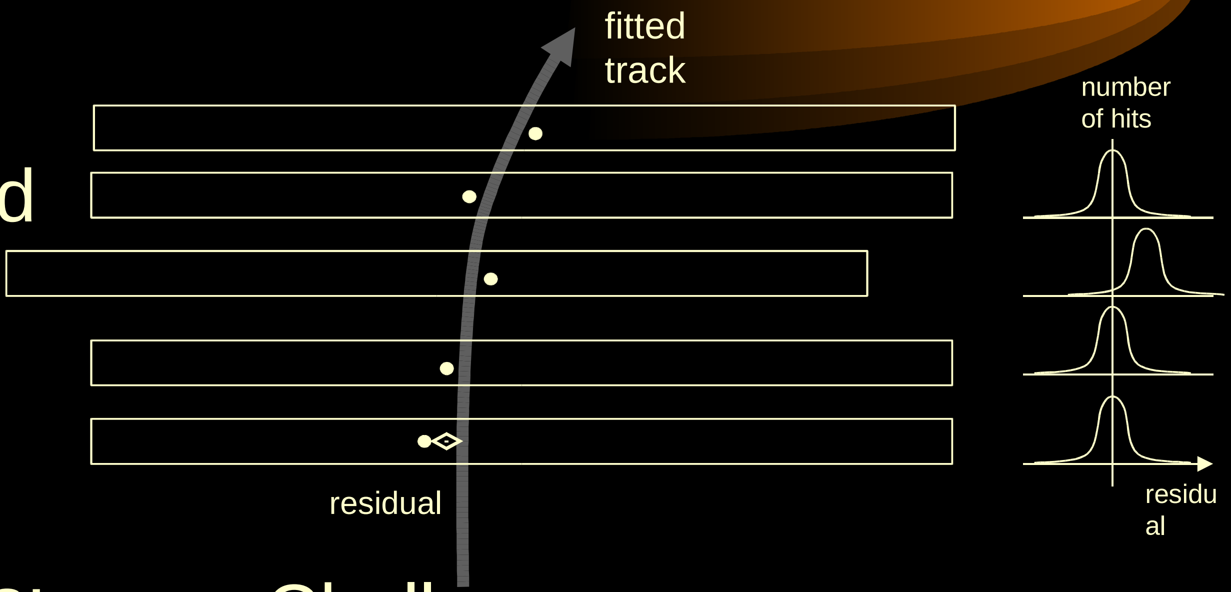
May 2nd, 2005

Content

- Overview
- Mechanics
- Current Status
- Next Steps

Software Alignment using Tracks

Use residual between fitted track and hit position



Different approaches:

Global Chi^2

Robust Algorithm

Robust Chi^2

Challenge:

alignment accuracy ~ 1 micron

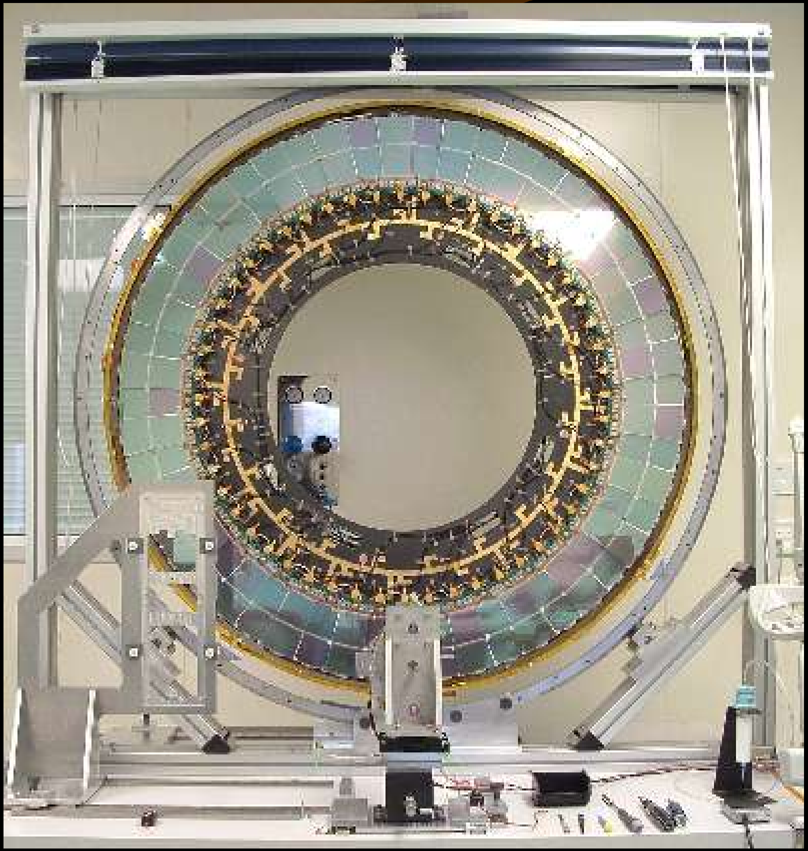
(Offline Alignment & Calibration of the Inner Detector - Working document)

Overview Global Chi^2 *(RAL)*

- Global fit of alignment parameters to hit information using Chi^2-minimization
- All correlations and constrains (multiple scattering, common vertex) contained in one huge matrix (36k x 36k)
- Alignment of SCT and Pixel, so far Barrel only
- Standalone code, migration to Athena ongoing
- Dedicated computing cluster for matrix inversion

Overview Robust Alignment *(Oxford)*

- Iterative method that aligns full Barrels + Barrel modules in rφ and z using overlaps
- Histogramming method fully prototyped
- So far "ntuple-level" analysis
- Ongoing work to migrate to Athena
- Extending the method to the EndCaps



Robust Chi^2 *(MPI)*

- Alternative automated approach recently started by MPI Munich
- Use the same iterative method as Robust Alignment but replace histogramming by:
- Local Chi^2 minimisation, correlations taken into account in the improvement of the track parameters through the iterations
- Full development will be continued in the Athena framework

Calculation of Alignment Parameters

$$(1)\quad \chi_0^2 = \Sigma\left(\frac{r}{\sigma}\right)^2$$

Definition of χ^2-function

$$(2)\quad \frac{\partial\chi^2}{\partial a_i} = 0$$

Look for minimum of χ^2-function

$$(3)\quad \chi^2 = \chi_0^2 + \frac{\partial\chi_0^2}{\partial a_i}\Delta a_i$$

Taylor-series expansion

$$(2)\rightarrow(3)\quad \frac{\partial\chi^2}{\partial a_j} = 0 = \frac{\partial\chi_0^2}{\partial a_j} + \frac{\partial^2\chi_0^2}{\partial a_j\partial a_i}\Delta a_i = \Sigma\left(\frac{2}{\sigma^2}r\frac{\partial r}{\partial a_i}\right) + \Sigma\left(\frac{2}{\sigma^2}\frac{\partial r}{\partial a_j}\frac{\partial r}{\partial a_i}\right)\Delta a_i$$

1st-order-approximation

Calculation of Alignment Parameters

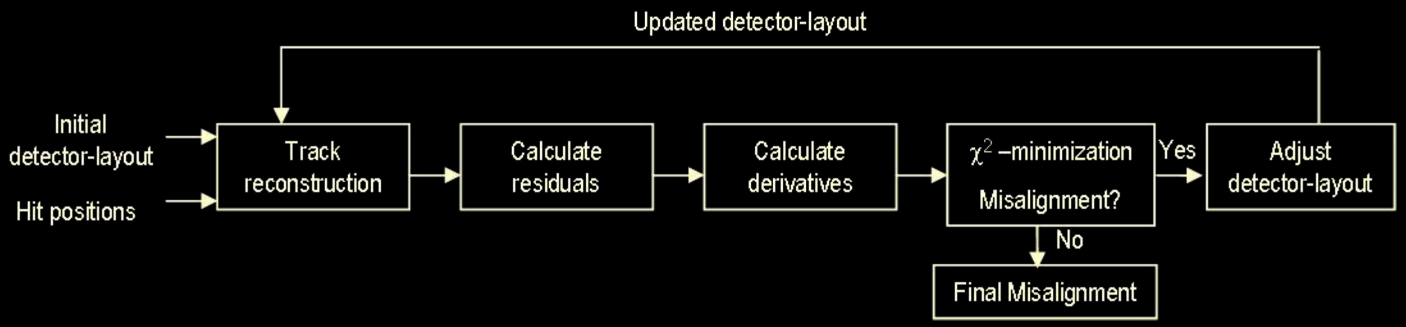
- Set of linear equations

  $\vec{b} + \Lambda\vec{a} = 0$

- Solve for $\vec{a}$
- 6 x 6 symmetric matrix, use efficient and numerically stable algorithm

Iterative Algorithm

- Determine 3D-residual (distance of closest approach) and derivatives wrt 6 alignment parameters for each module
- Complete geometry information contained in residual and derivatives
- Linearized Chi^2-minimization to calculate most likely set of alignment parameter

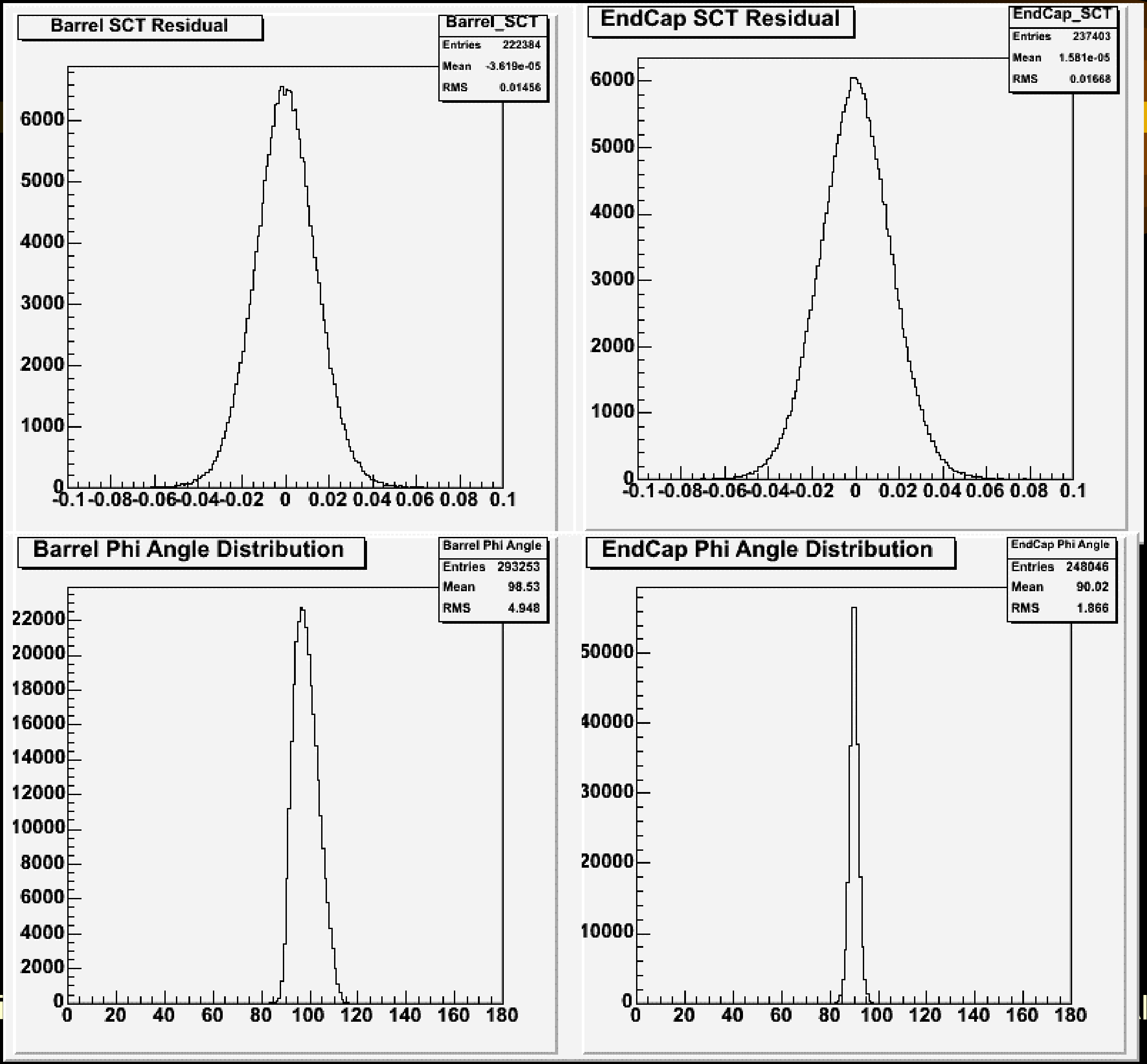


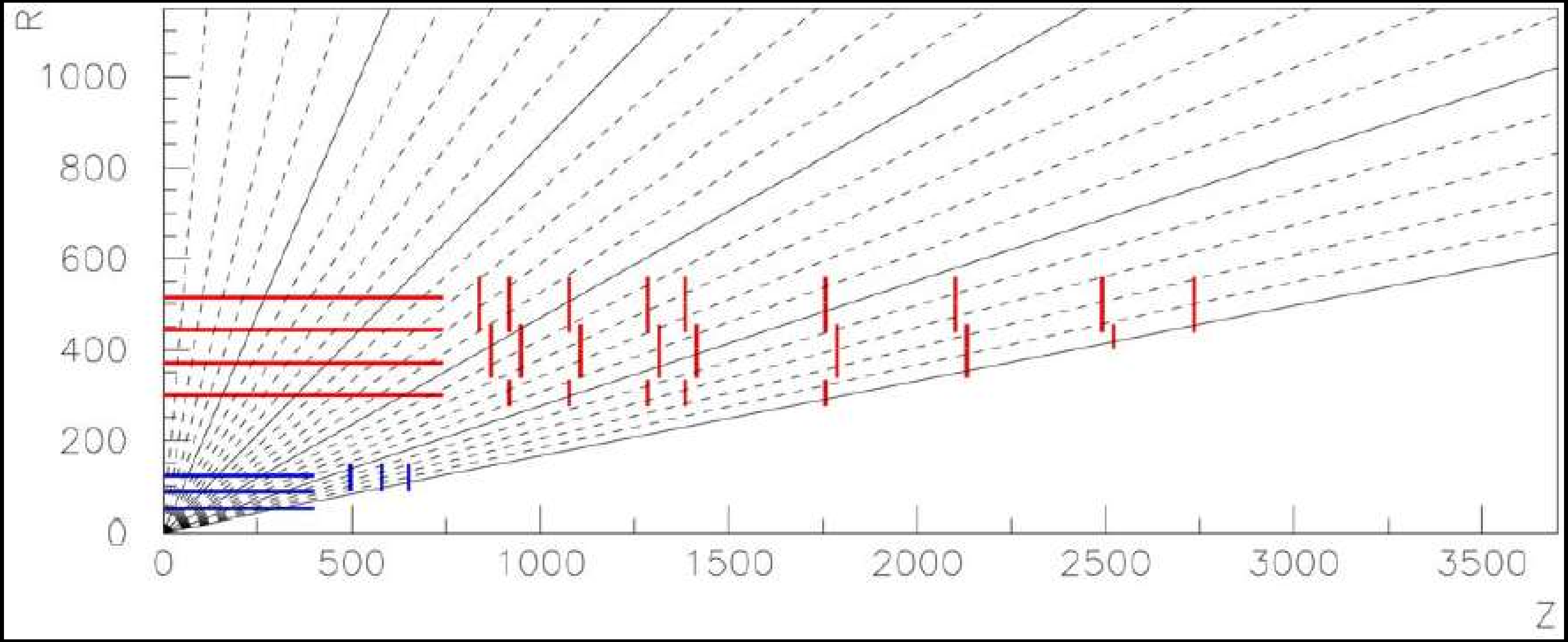
Current Status

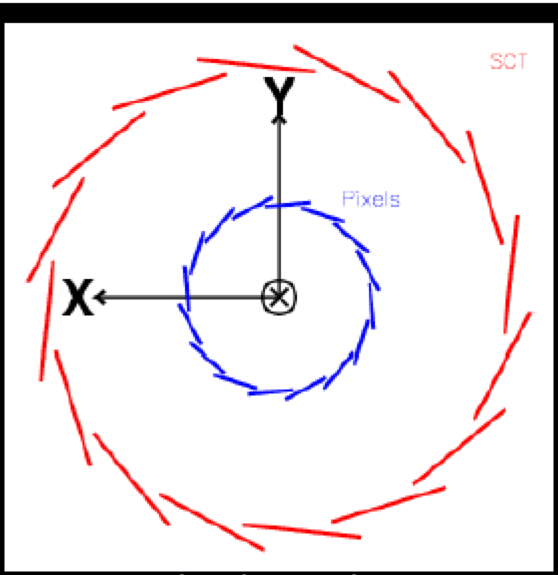
- Chi2Align Code in cvs repository (InnerDetector/InDetAlignment/SiRobustAlign)
- Runs with reconstructed Tracks from InDetRecExample or from ESD
- Uses Tools from Athena (GeoModel, Tracking)
- 0th iteration works   convince yourself

fineprint: InDetRecExample Reco with XKalman, Rome-Initial Layout,
10000 events Z ee, ~ 130k Tracks, Output to ESD
Alignment: 5000 events read from ESD, ~ 65k Tracks

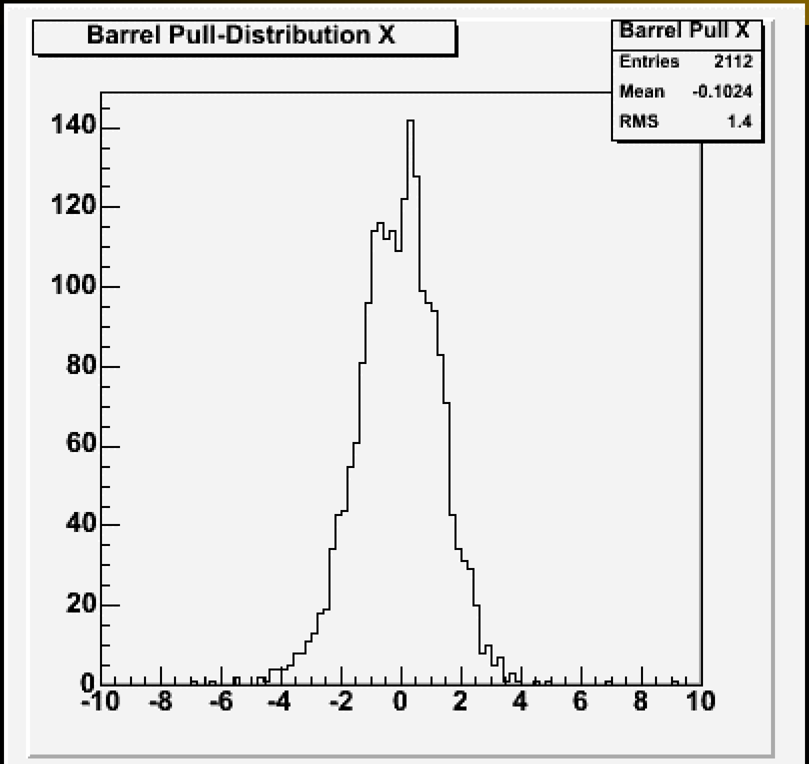
Residual Plots

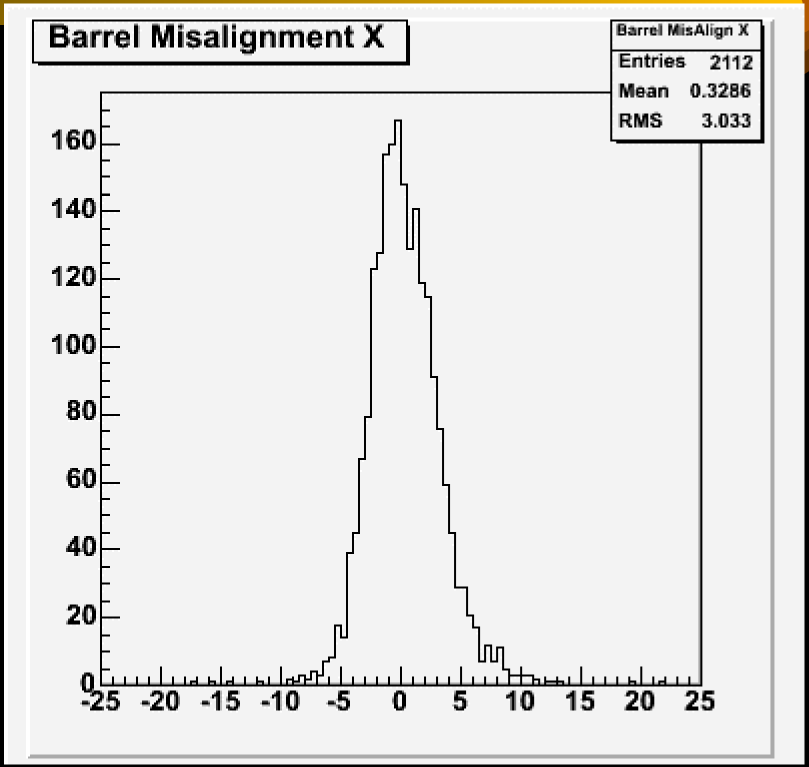



R
1000
800
600
400
200
0
0
500
1000
1500
2000
2500
3000
3500
Z

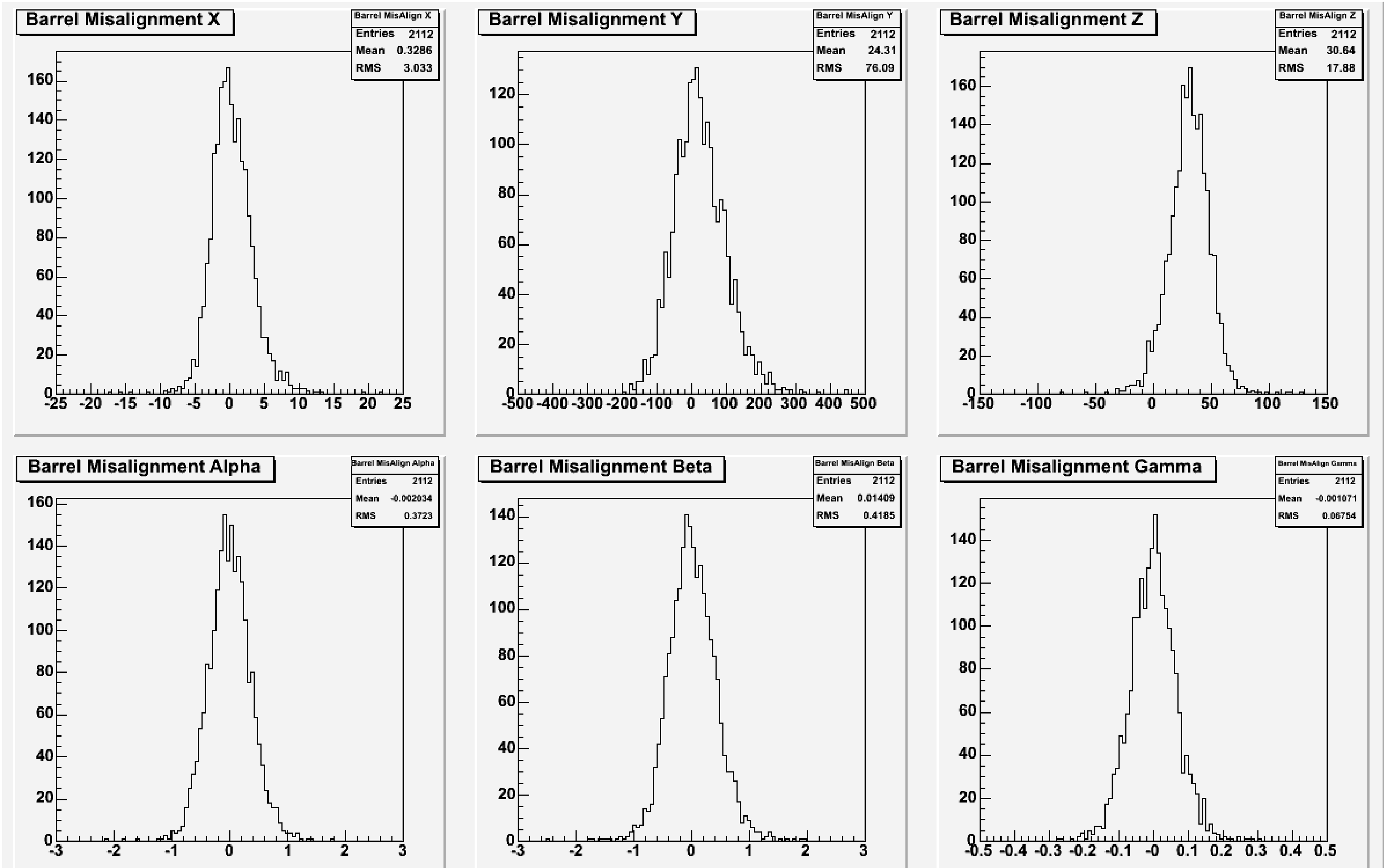

SCT
Y
Pixels
X

Barrel Pull and Misalignment x

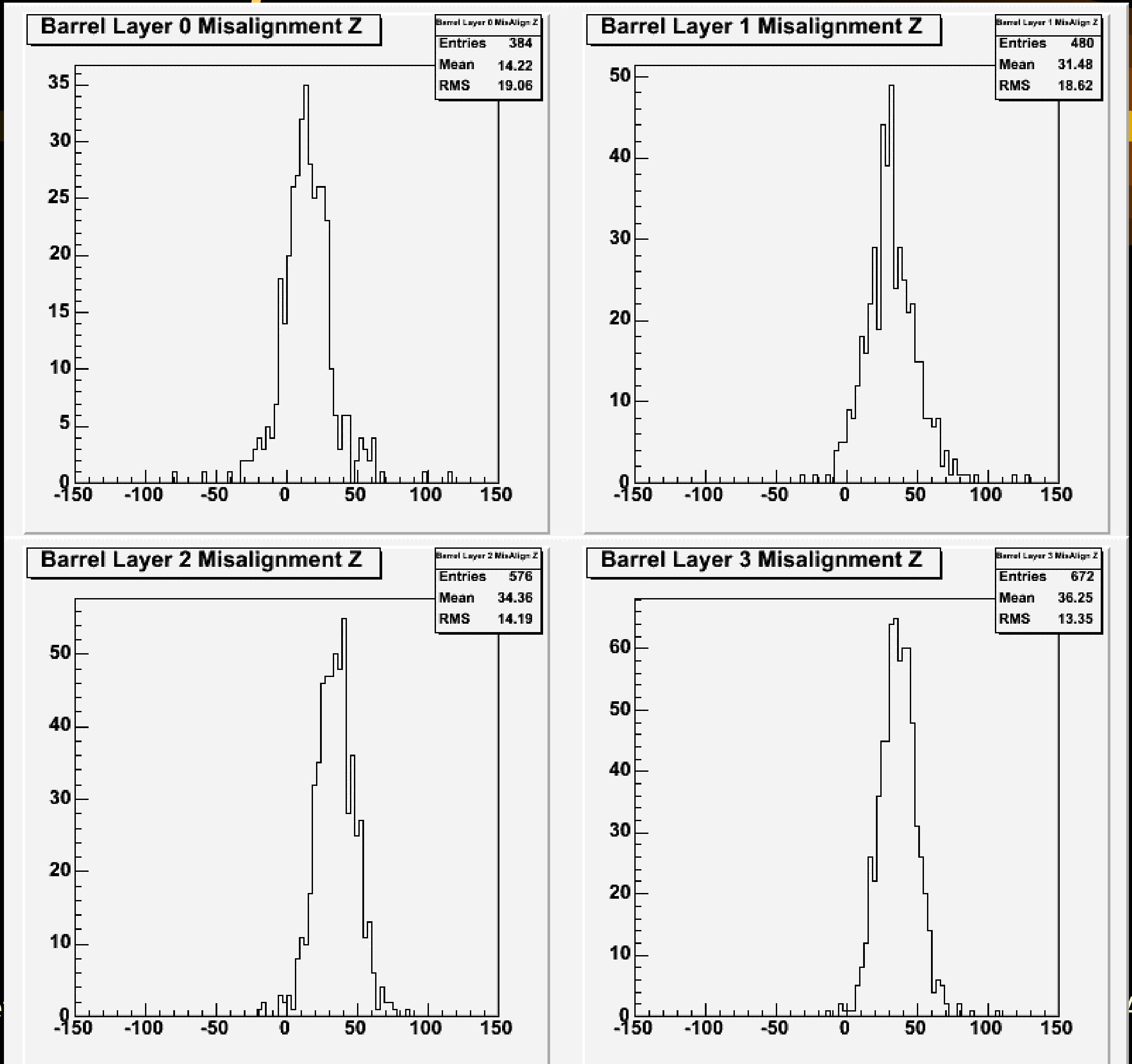




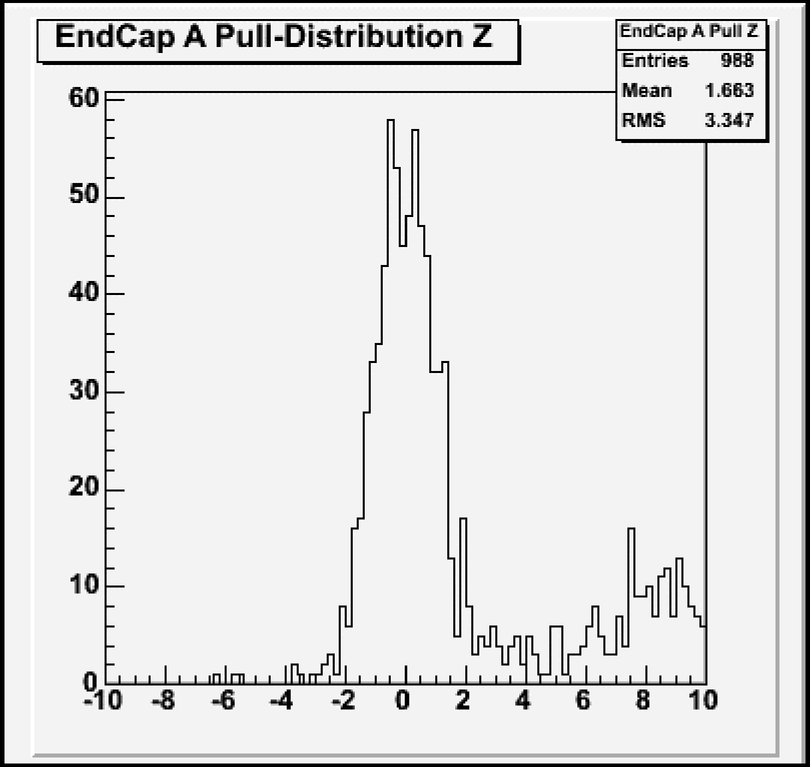
Barrel Misalignment all parameters

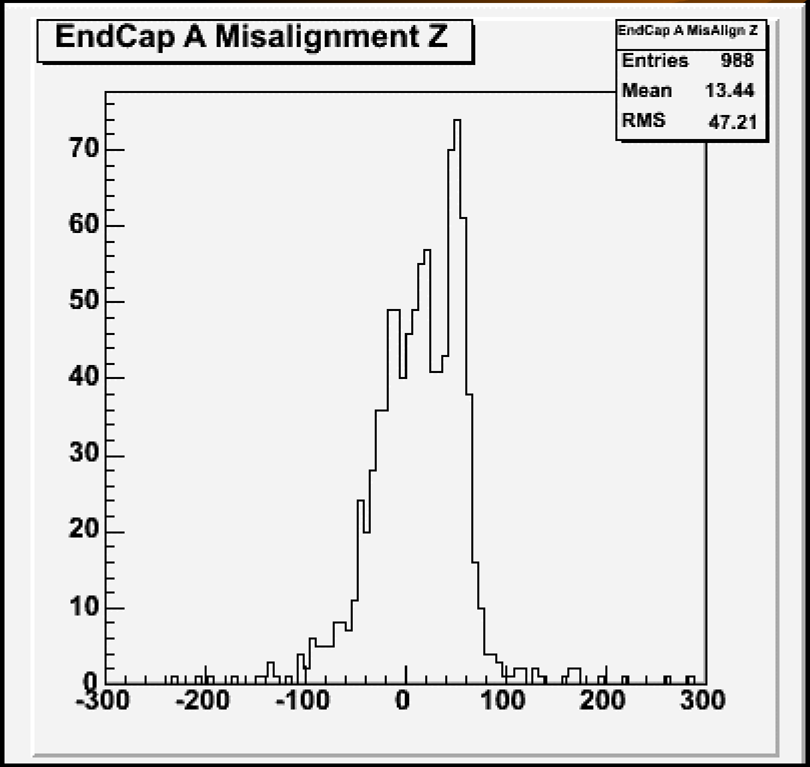


Misalignment z Substructure

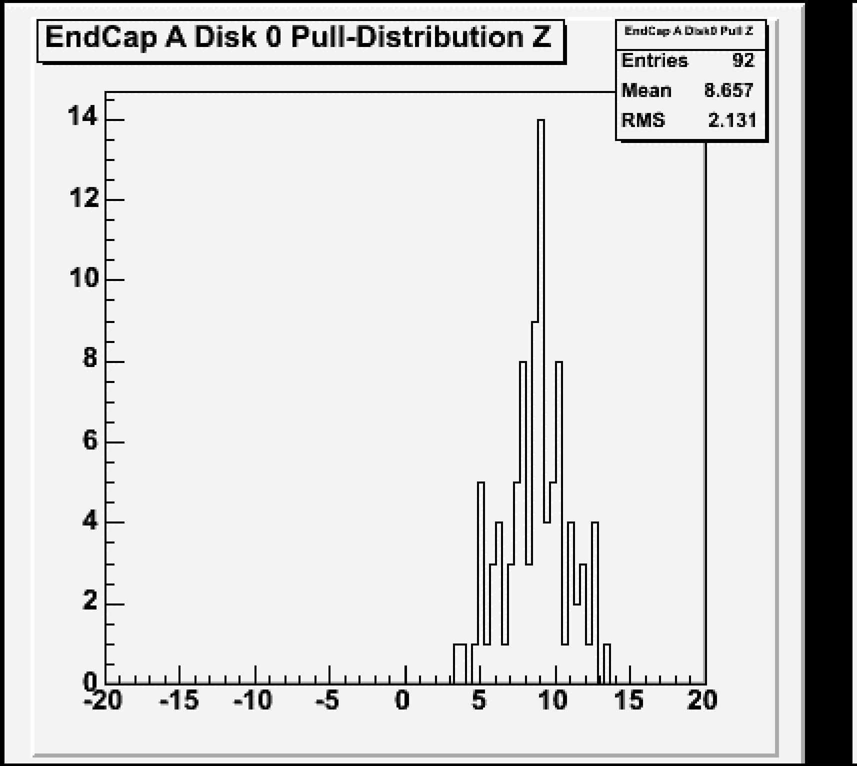


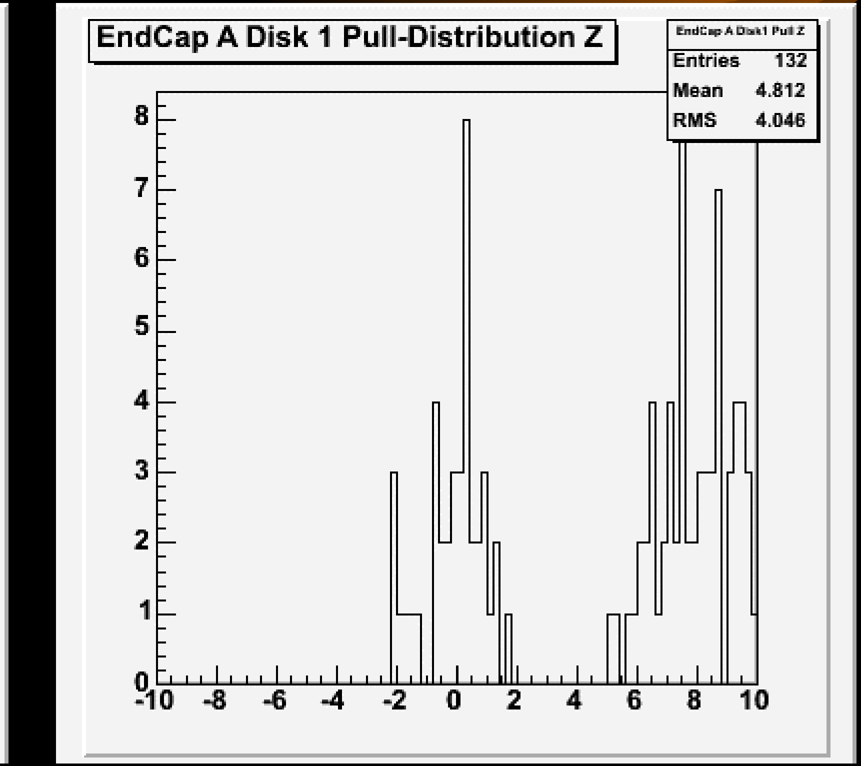
EndCap A Pull and Misalignment z



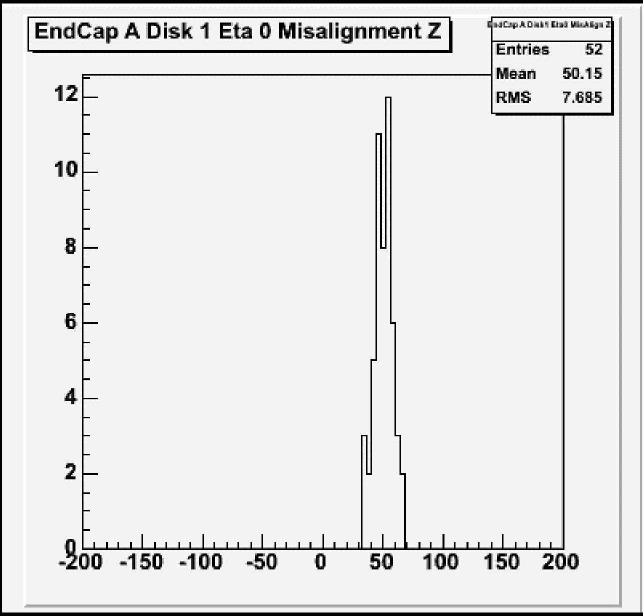


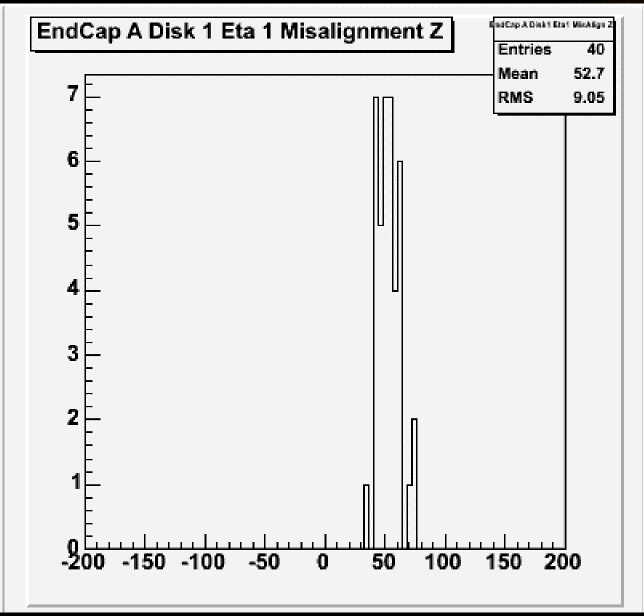
Pull z Substructure

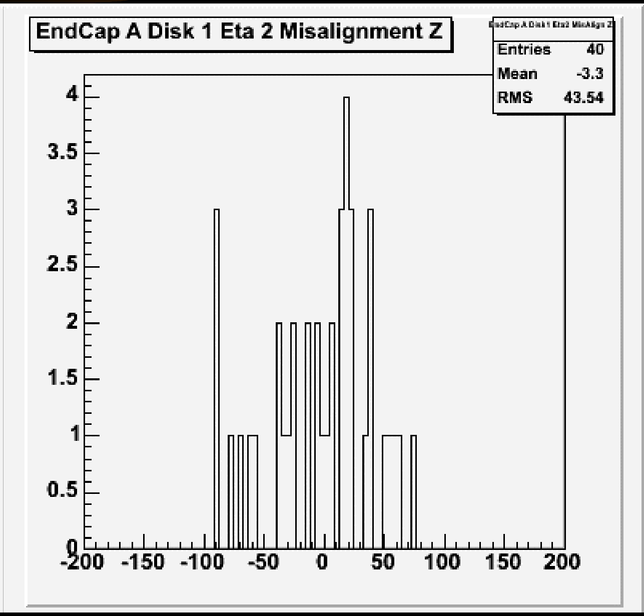




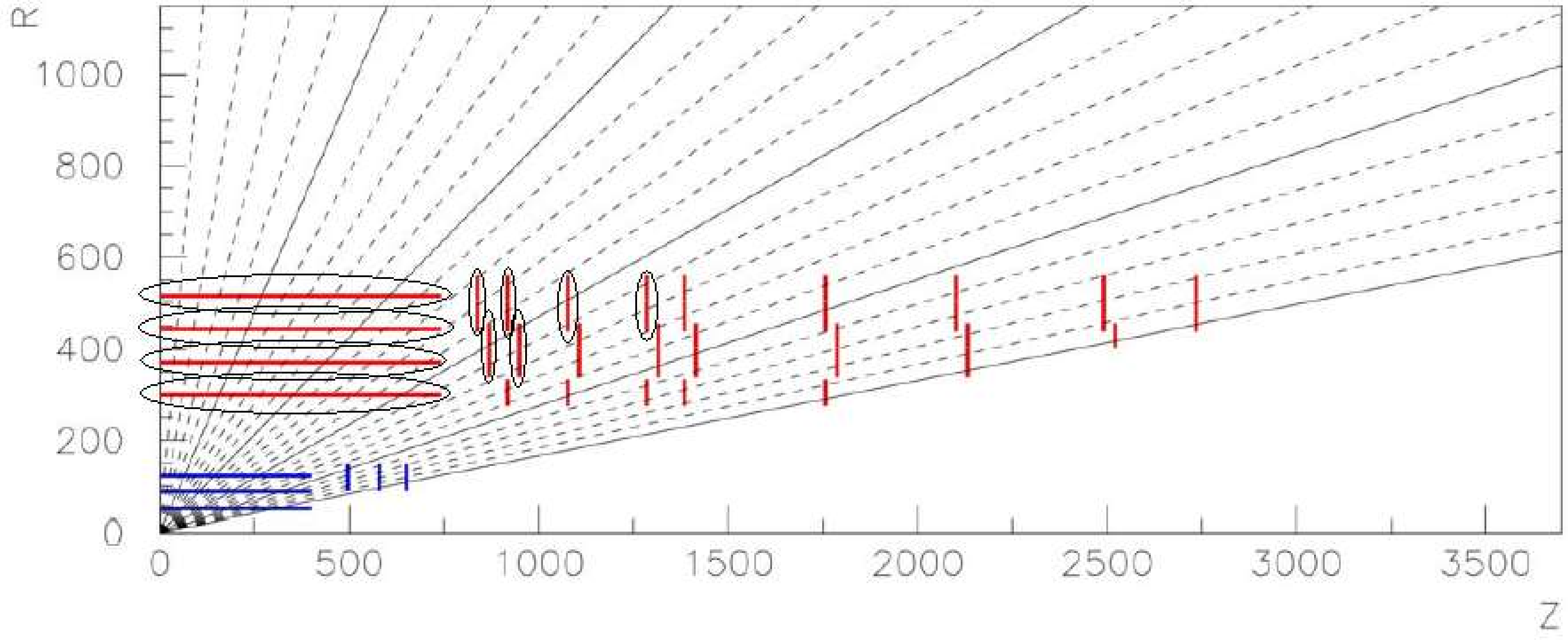
Misalignment z Substructure Disk 1







Geometry Layout



Next steps

- Use Silicon only tracks
- Use Barrel / EndCap only Tracks
- Use unbiased residual
- Get tracking Error

Catch several birds with one stone:
Track Refit

Refit Trk::Track
with
TrkKalmanFitter

Next steps

- Run with misaligned geometry
- Iterate

Use DB and IoVDBSvc to temporarily store alignment parameters

- Run on real data (CTB, SR1 Cosmics)

Conclusion

- Robust Chi^2 runs and produces first results
- Robust Chi^2 extensively uses tracking tools and geometry description (helpful to debug both)
- So far the only approach to produce EndCap alignment results