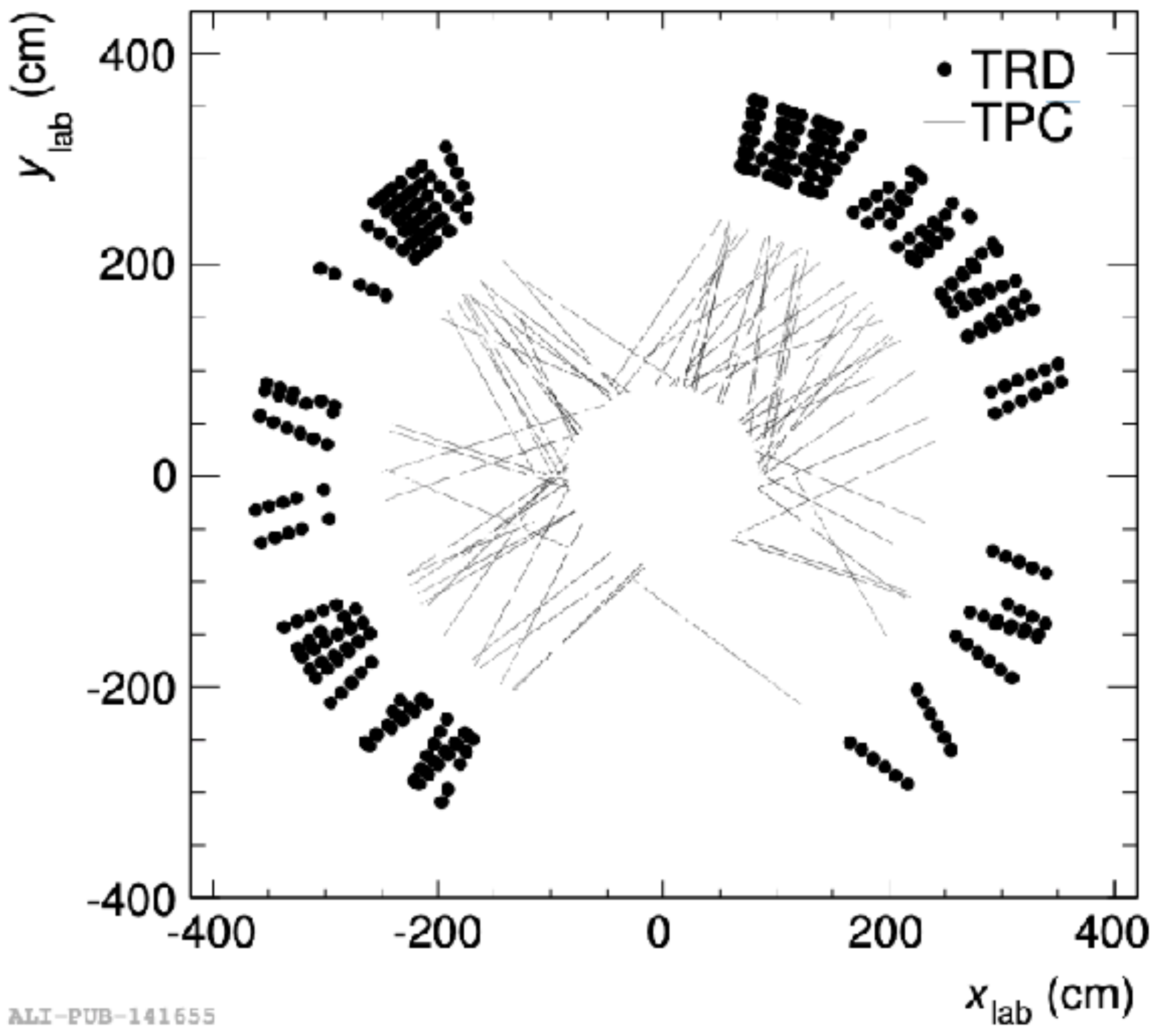
MOST2 offline tracking reconstruction and alignment

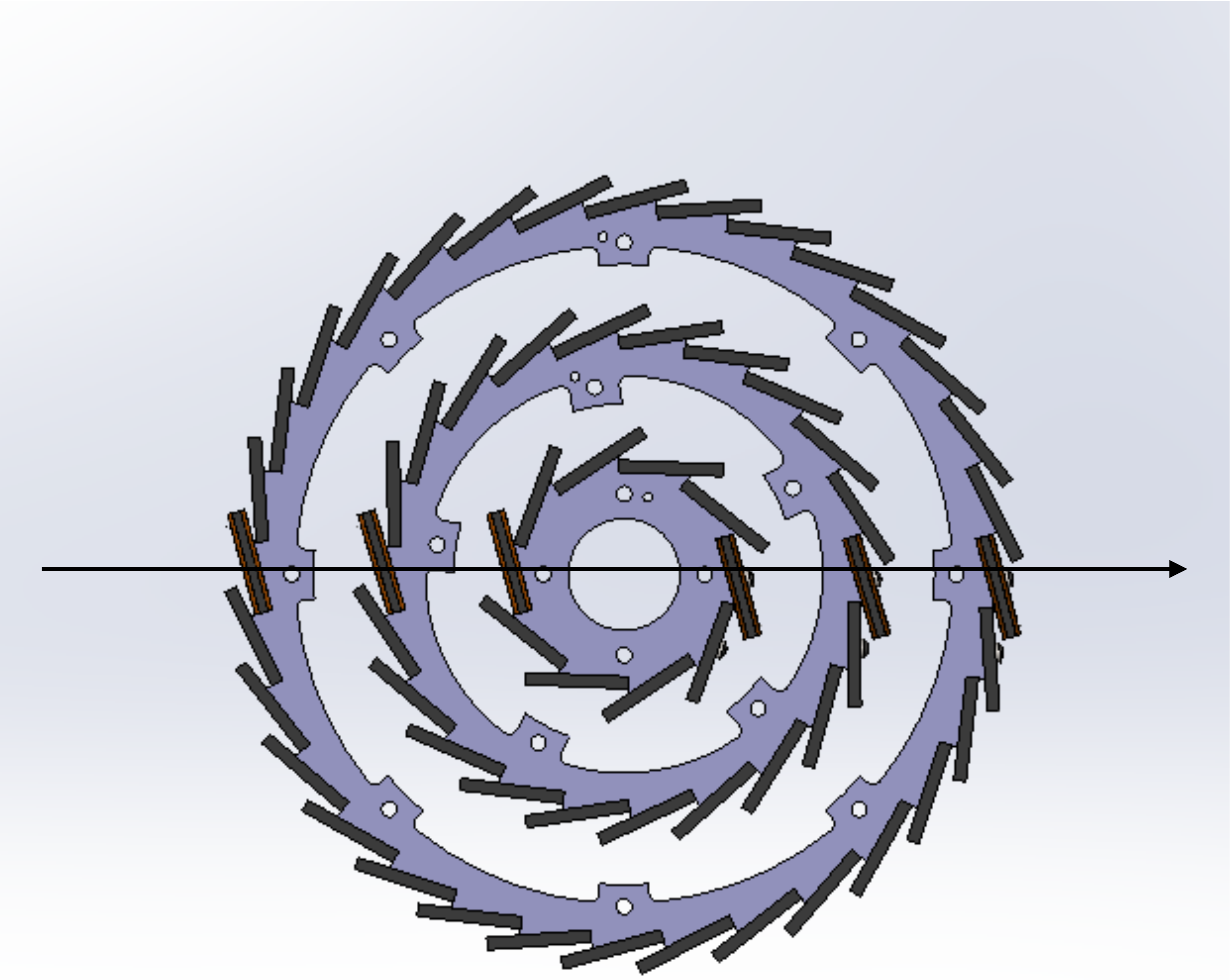
Shuqi Li, Hao Zeng, Gang Li, Linghui Wu, Zhijun Liang (IHEP)
Yiming Hu (NJU)

Introduction

- measuring spatial resolution
- event track reconstruction
  - No magnetic field
  - Straight line fit
- track alignment
  - None-ideal detector
  - correction for the detector position
  - determining free degree for local and global detector parameters



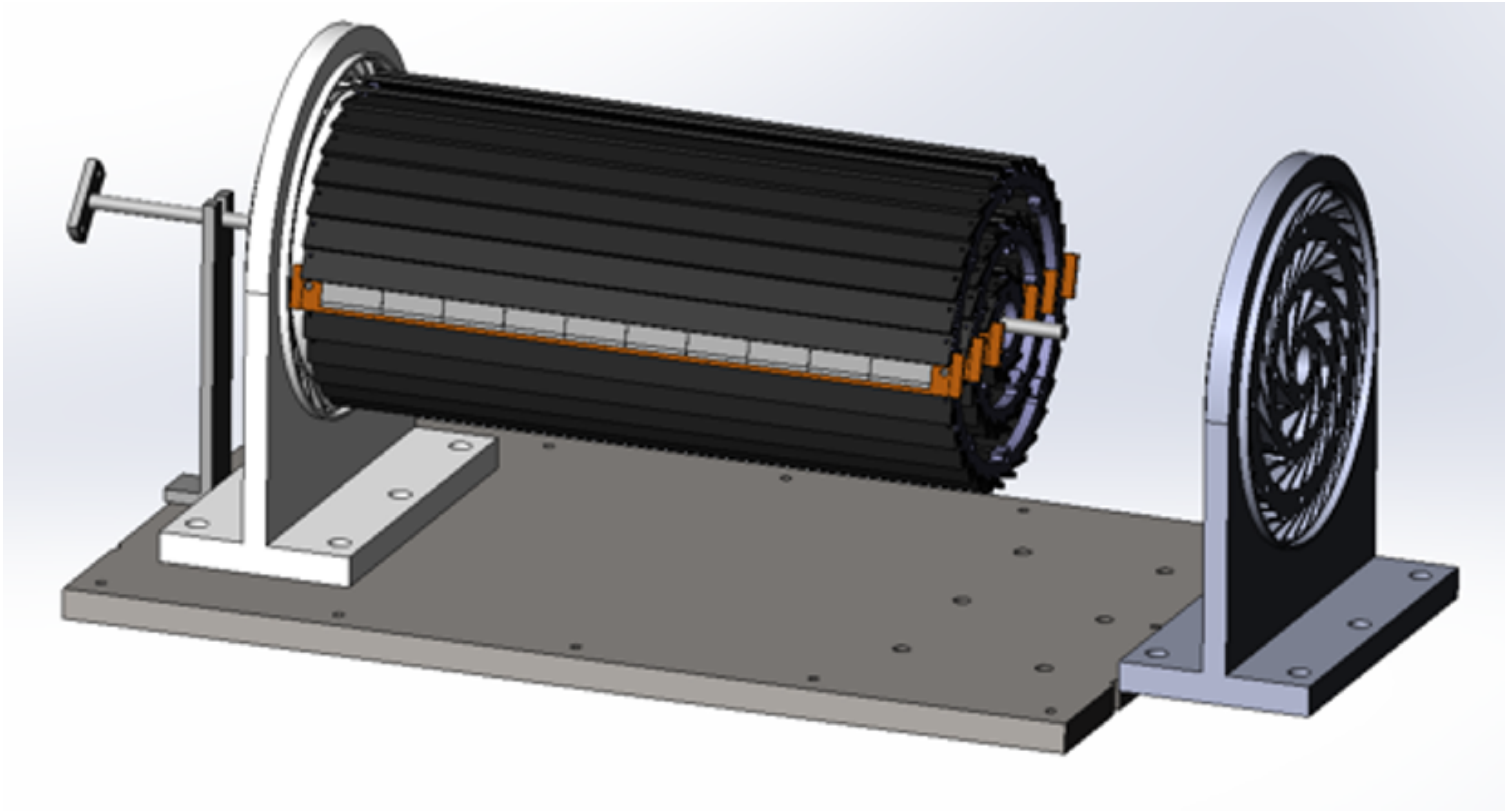
Electron Beam

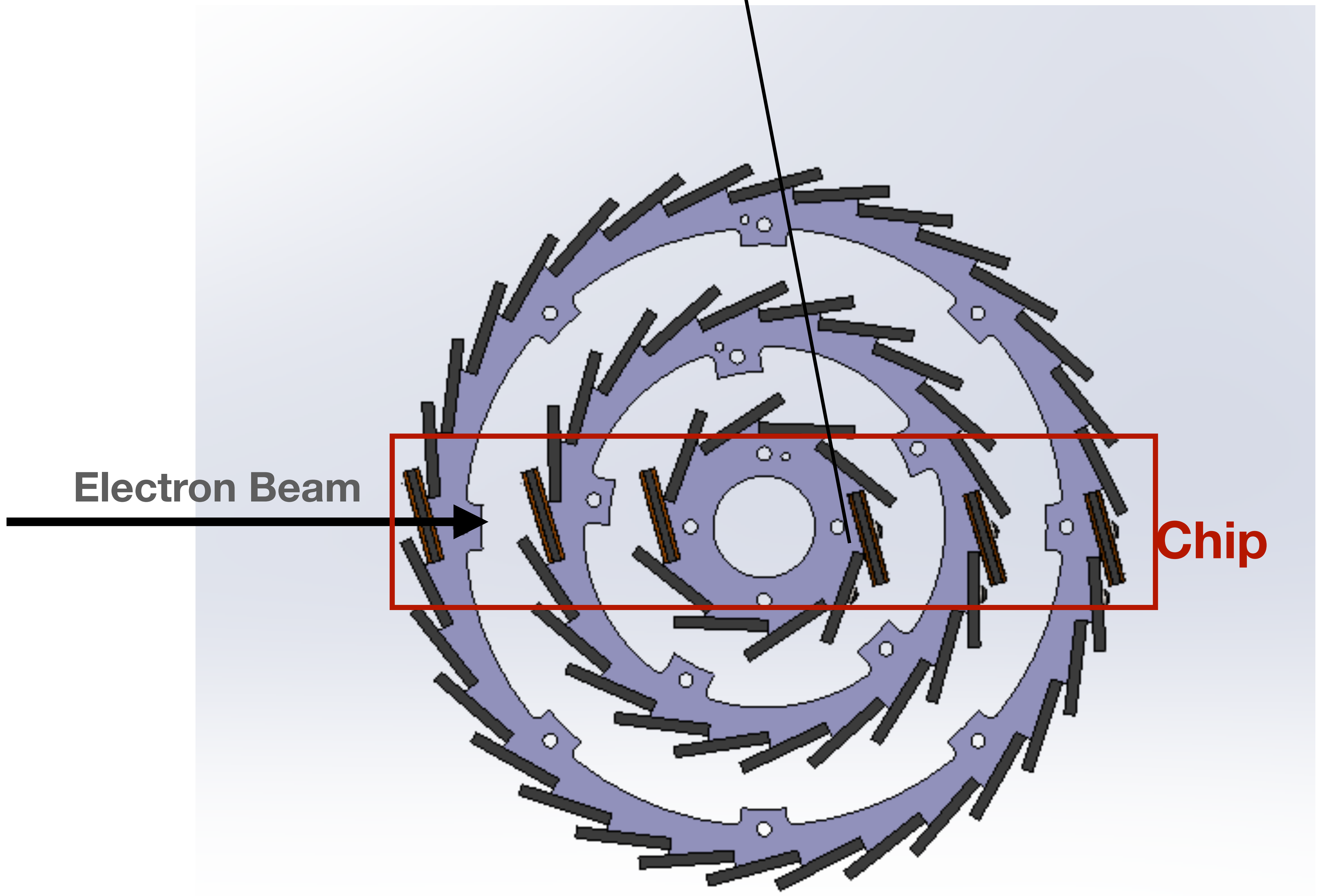


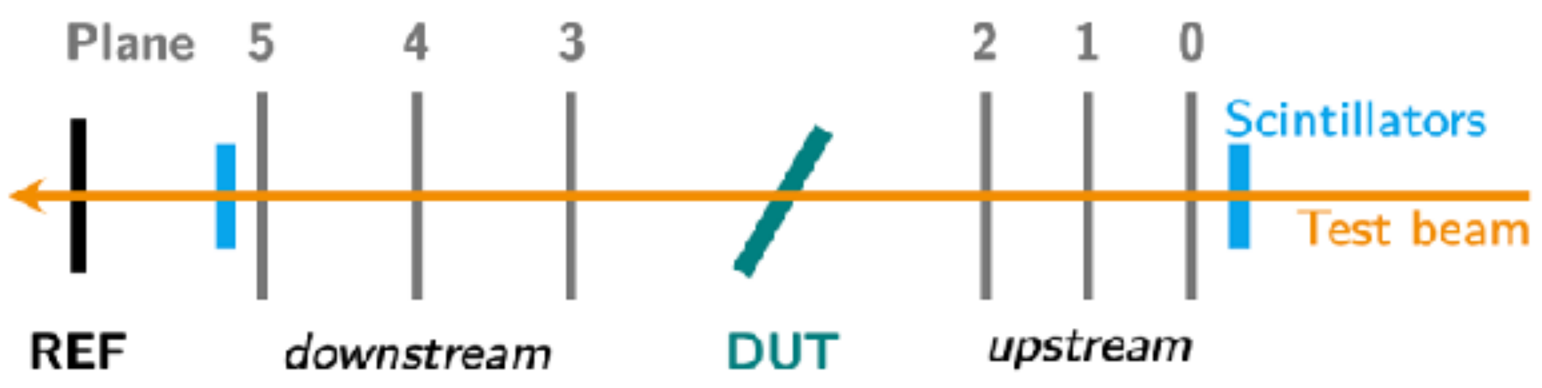
Geometry

- Prototype

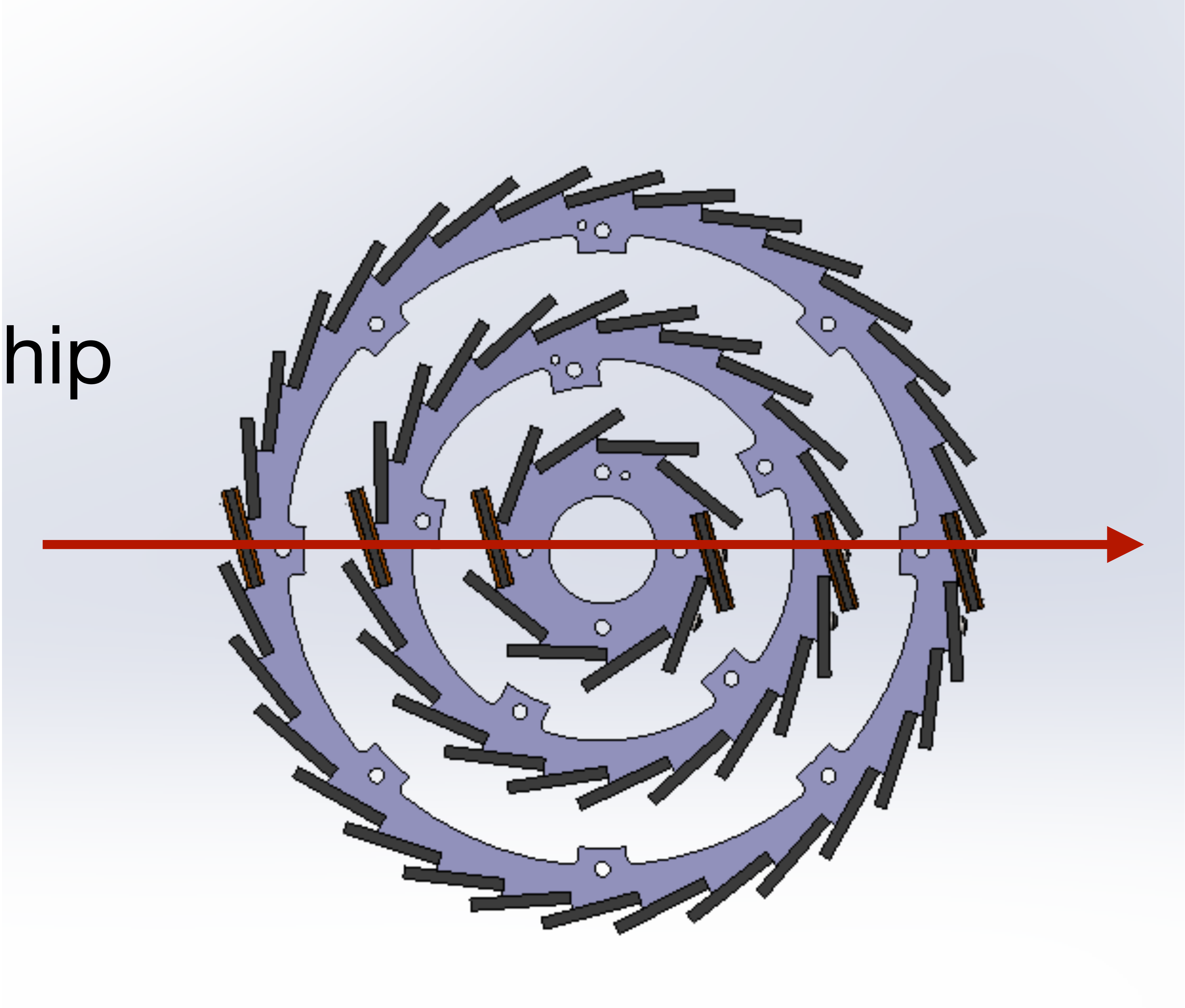
a medium chip will not participant track reconstruction
the residual plot of fit position and measured position decide the spatial resolution







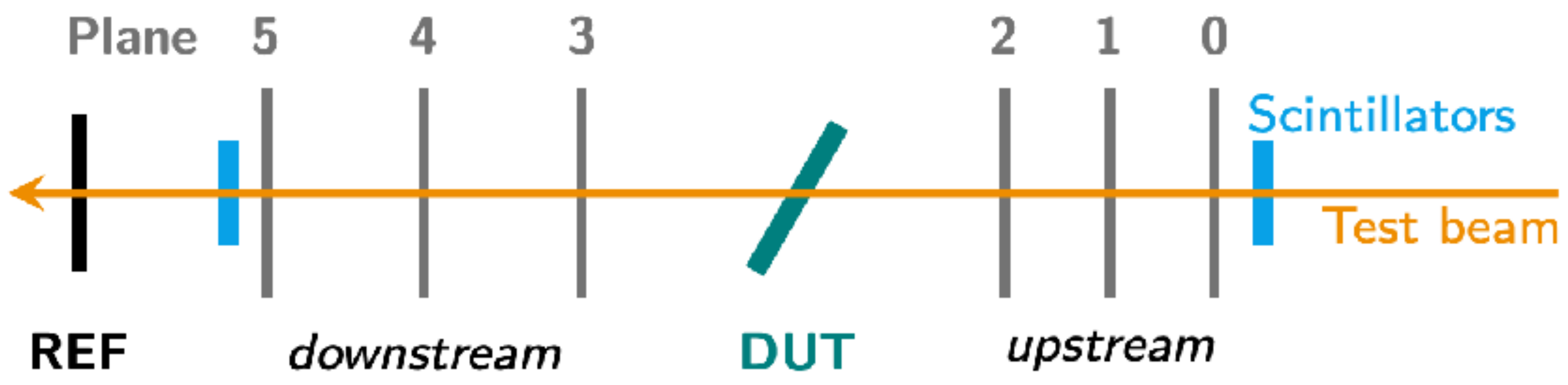
track reconstruction



- every hit or cluster position in corresponding detector chip
- straight line with little shift due to multi-scattering
- method: Least squares fitting

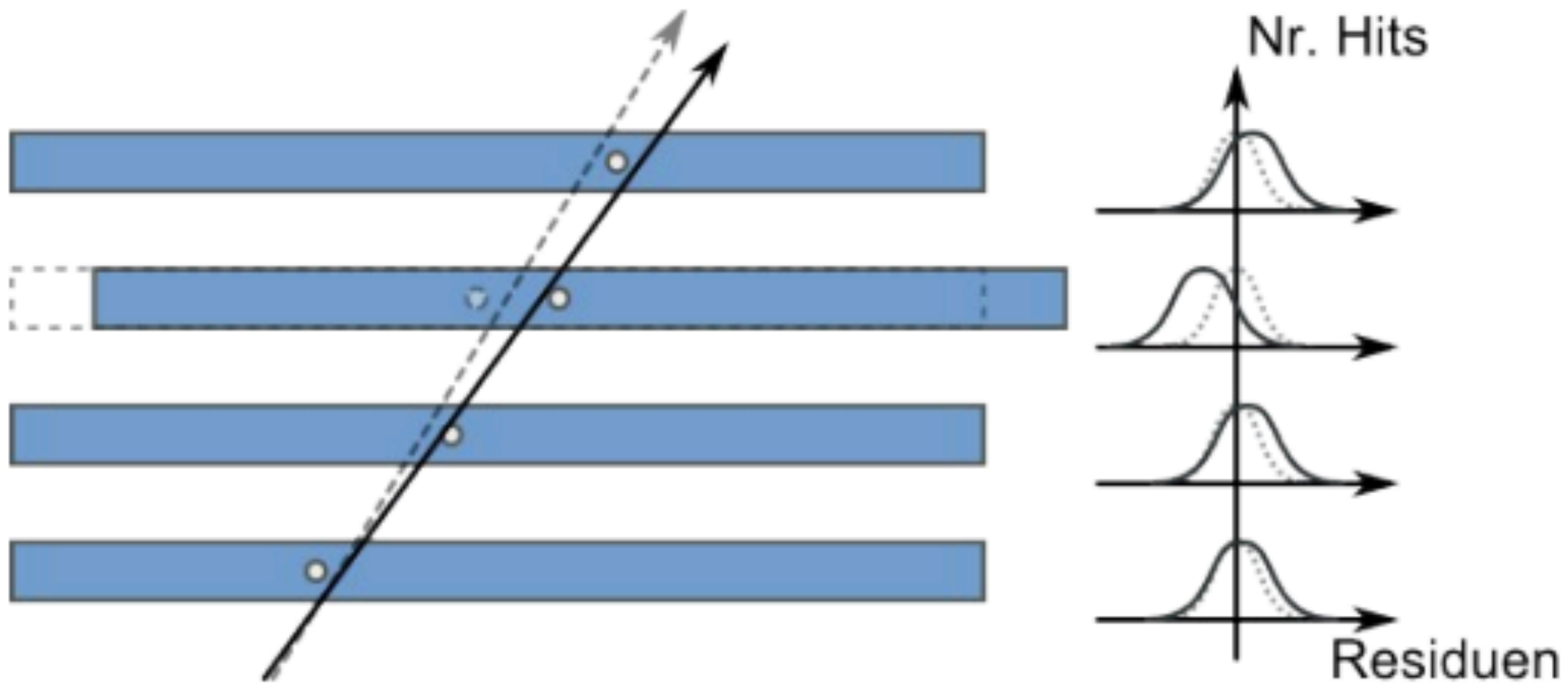
$$\chi^2 = \sum_l \frac{(y_l - f_l(\alpha))^2}{\sigma_l^2},$$

- device under test determining the space resolution



track alignment

Residual: distance of hit with intersection point of track in a module.

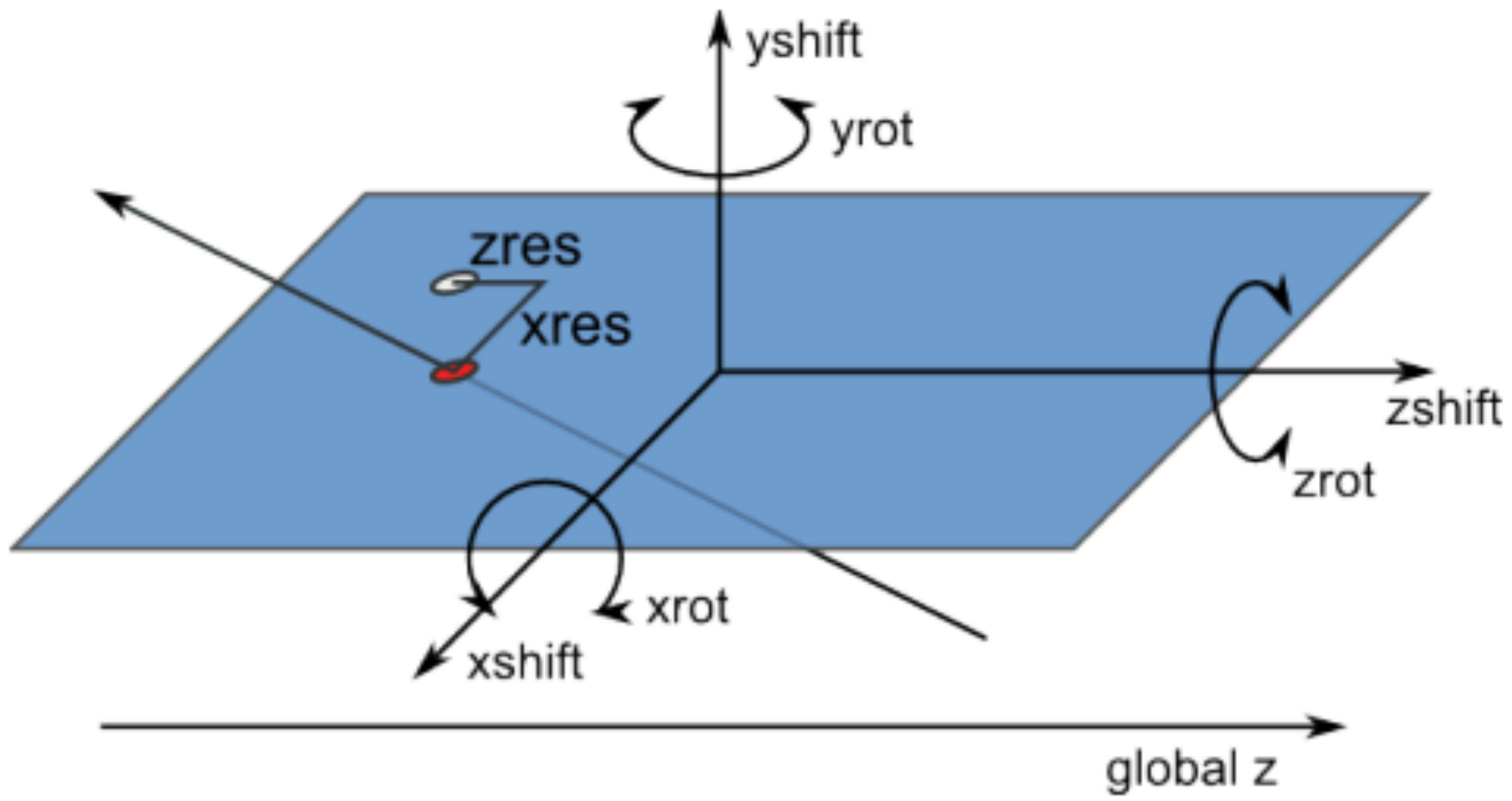


Ideal: Exact positions of modules are known.

➡ Residuals follow a gaussian distribution

In reality: Known positions of modules are ±100 µm

➡ Residual is shifted because hit position is shifted.



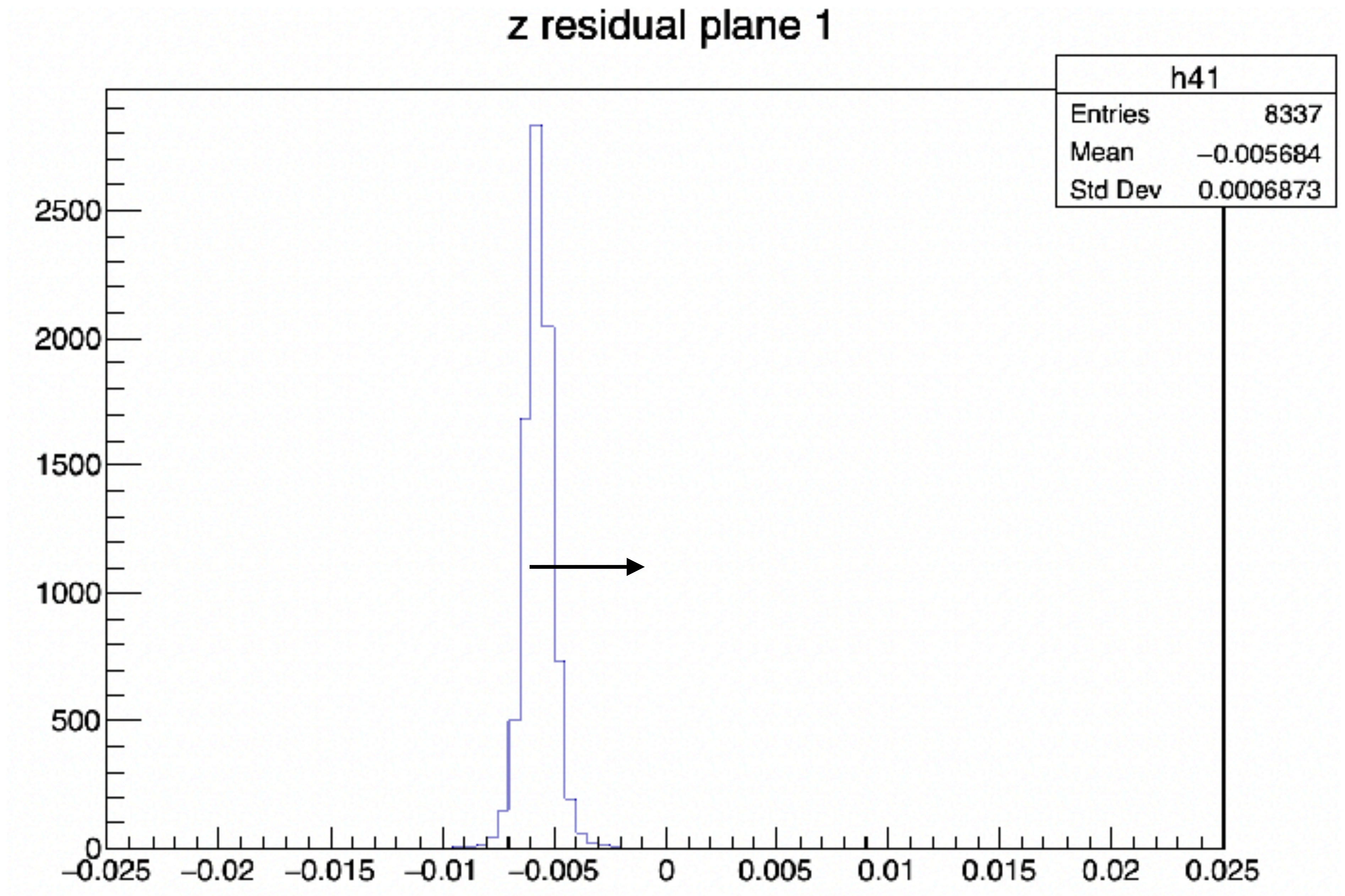
Solution: calculate correction parameters for each module

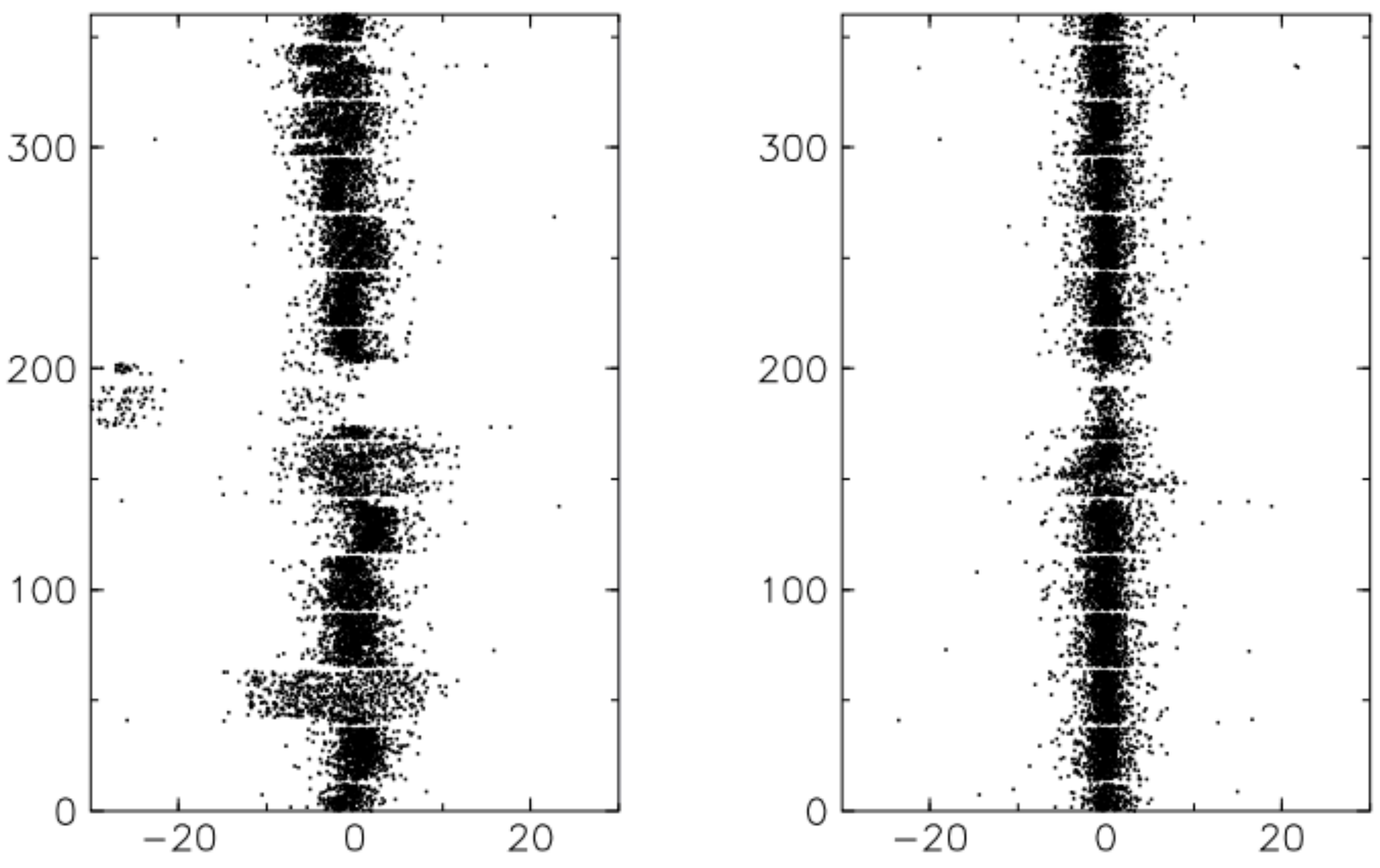
Translation 3 parameters
Rotation 3 parameters
...

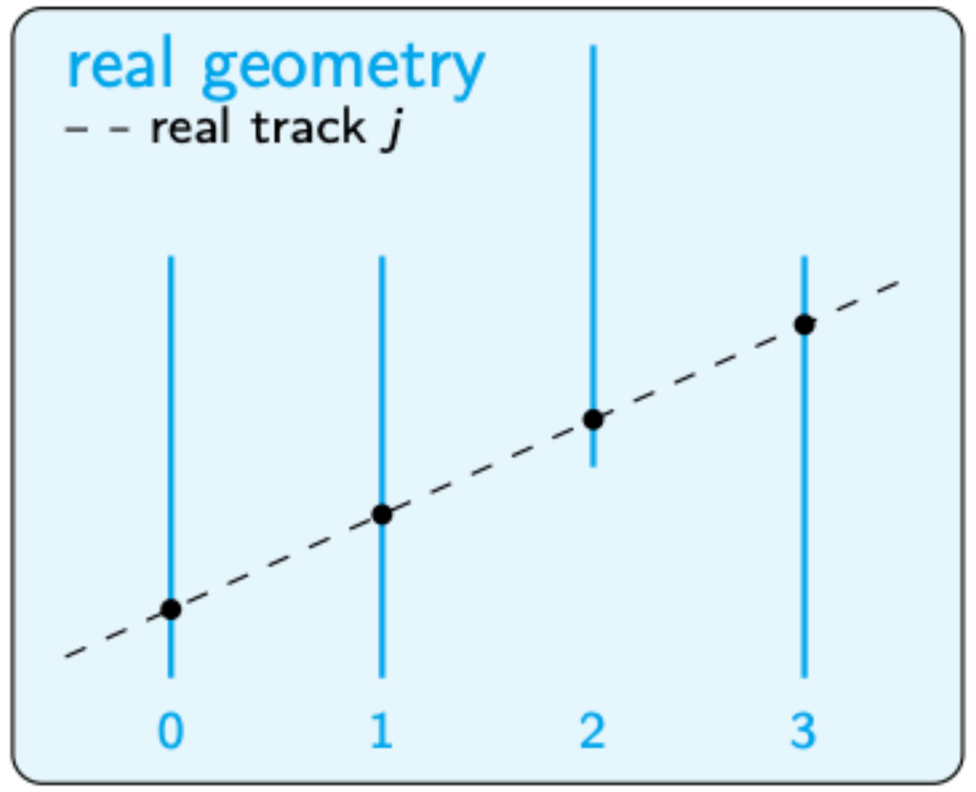
by using residuals of a large number of tracks.

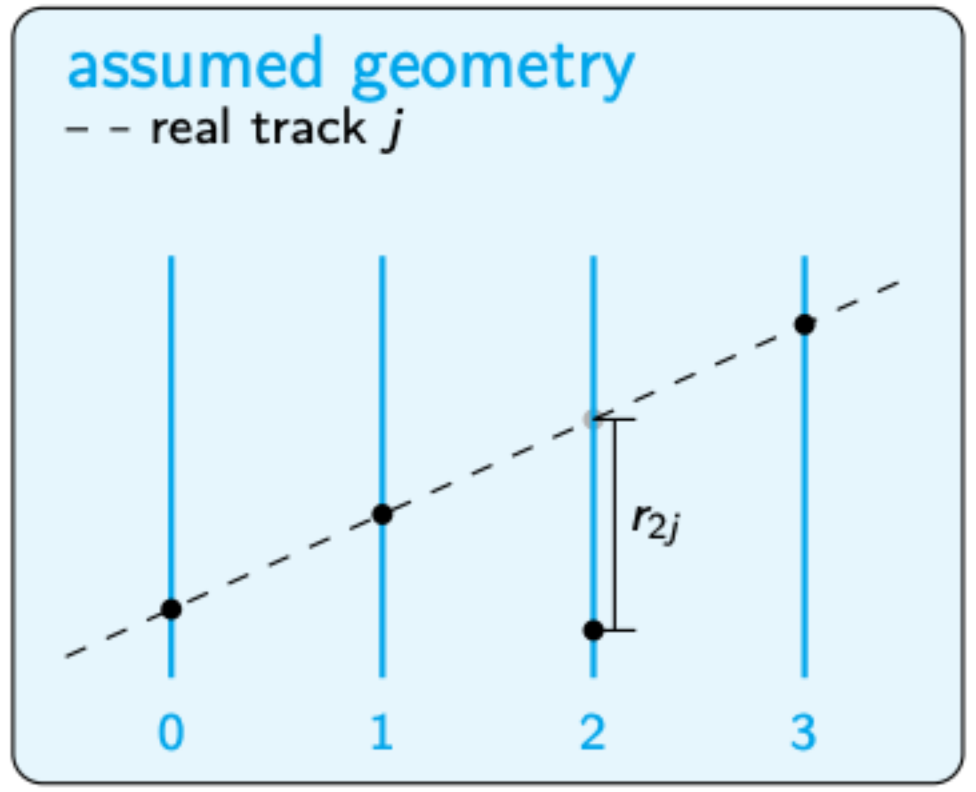
track alignment

- determining the alignment parameters (free degree) that can influence spatial resolution
- residual plot
- enough reconstructed tracks are necessary
- simultaneous fit for alignment parameters and track parameters









Summary

- preparing pesudo-data for track reconstruction using Geant4 simulation
- write the Least squares fitting for track reconstruction
- need to determine which alignment parameters (~4 local free degree for every chip) can influence spatial resolution according simulation results
- write alignment code (may be interface to Millepede II)
- estimation how much statistics can fit the alignment parameters
- eventually finish a standalone framework for tracking reconstruction and alignment

backup

track alignment

Minimize: $\chi^2 = \sum_{i \in tracks} \vec{r}_i^{\,T} V_i^{-1} \vec{r}_i$

V : covariance matrix

the **residual** $\vec{r}_i(\vec{p}, \vec{q}_i)$ is a function of

alignment parameters $\vec{p}$ and of

the track parameters $\vec{q}_i$

$$\Rightarrow \frac{d\chi^2(\vec{p})}{d\vec{p}} = 0 \qquad (1)$$

For the sake of notation the dependence of χ^2 on $\vec{q}_i$ will not be written here.

$$\chi^2(\vec{p}) = \chi^2(\vec{p}_0) + \left.\frac{d\chi^2(\vec{p})}{d\vec{p}}\right|_{\vec{p}=\vec{p}_0} (\vec{p}-\vec{p}_0) \qquad (2)$$

Rewrite as a Taylor expansion.

Linear least square minimization:
expand up to first order

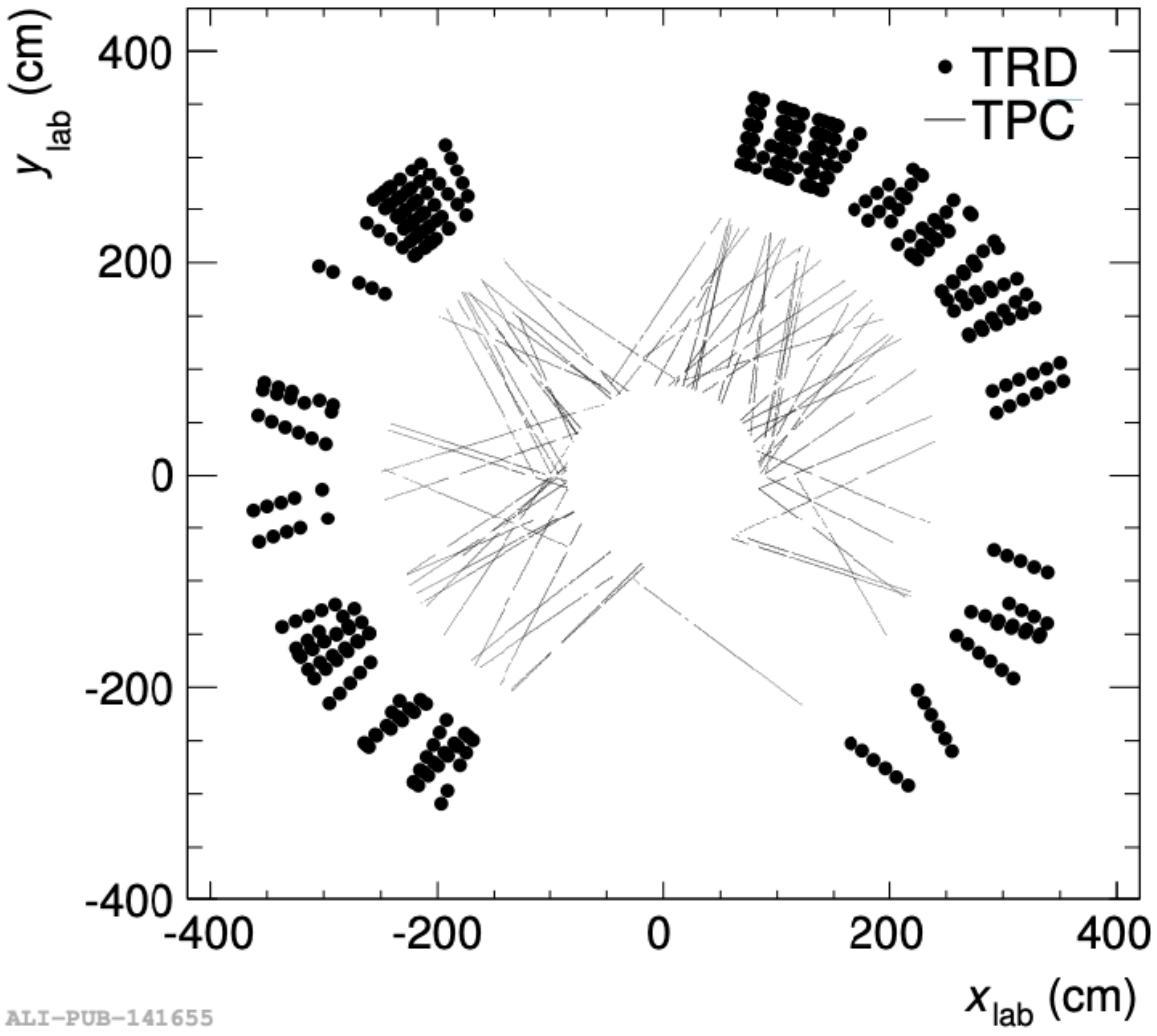
$\vec{p}_0$ is the vector of initial alignment parameters

(1) in (2) with $\Delta\vec{p} = (\vec{p}-\vec{p}_0)$ yields:

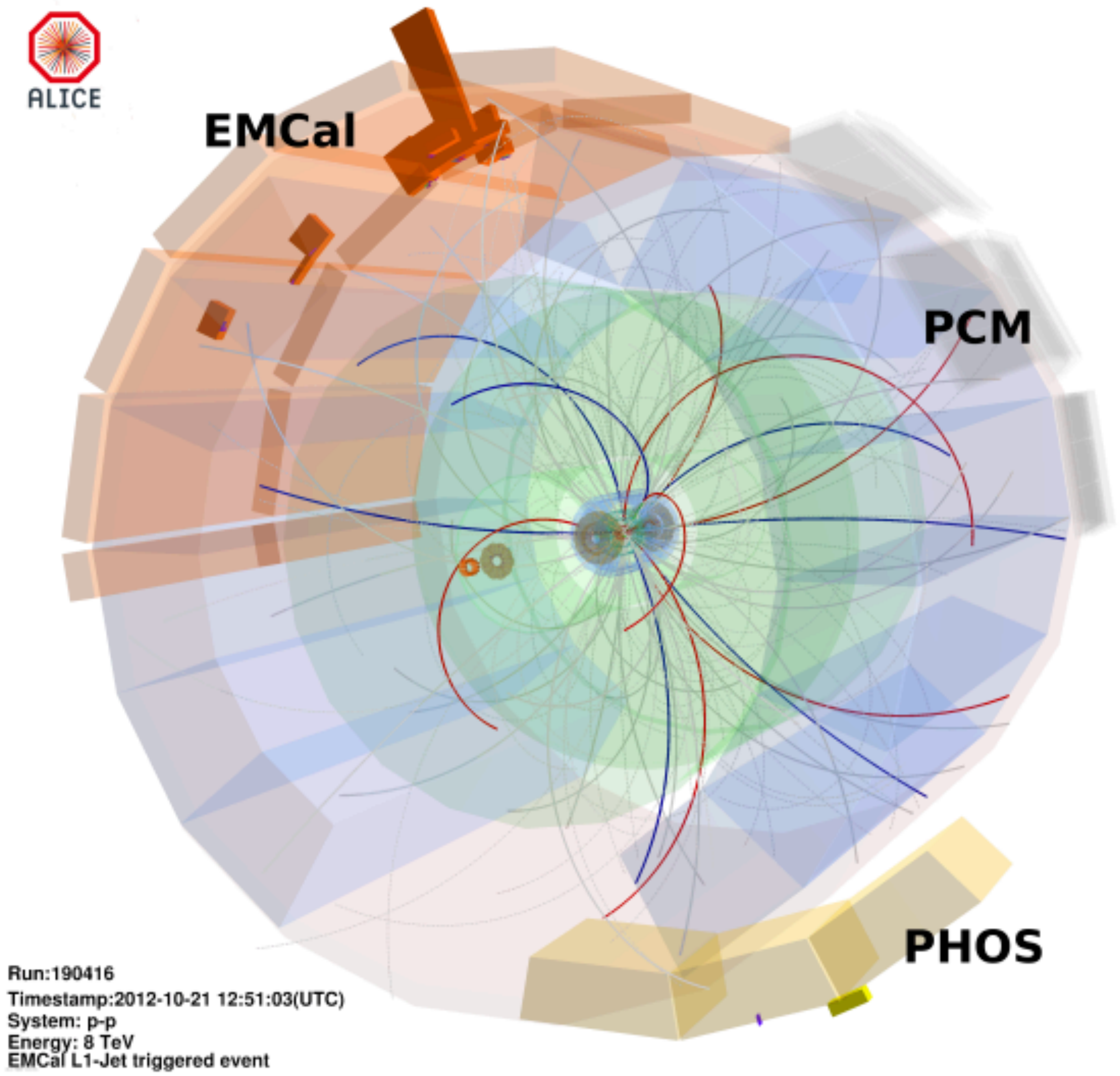
$$\boxed{\begin{gathered}\underbrace{(J^T V_i^{-1} J)}_{C}\ \Delta\vec{p} = \underbrace{J^T V_i^{-1}\ \vec{r}_i(\vec{p}_0)}_{b} \\ C\ \Delta\vec{p} = \vec{b}\end{gathered}}$$

J: Jacobi Matrix

Task: Invert the Matrix C to find alignment corrections $\Delta\vec{p}$



Cosmic muons without magnetic field (ALICE)



Homogeneous magnetic field - helix shape

- Straight line
- Parametrisation ($\vec{x}(t) = \vec{x_0}(t) + \vec{v_0}t$)

- helix
- $x = x_0 + rcos(\omega t + \phi_0)$
- $y = y_0 + rsin(\omega t + \phi_0)$
- $z = v_z t$

Assuming

- linear state change (by matrix multiplication)
- all state information can be characterised completely by their state vector and its covariance matrix
- measurements are unbiased and can be completely described by the measurement covariance matrix
- processed affecting the track state are unbiased and completely described by the process covariance matrix
- different measurements and process noise are all independent

The Kalman filter

Ingredients

- The state vector $\vec{s}$ and its covariance matrix $\overrightarrow{S}$
- The measurement vector $\overrightarrow{m}$ in the coordinate system of measurement
- A matrix $\overrightarrow{F}$ transforming the state vector $\vec{s}$ from last measurement to next measurement
- A matrix $\overrightarrow{H}$ transforming from the state vector $\vec{s}$ into the measurement space $\overrightarrow{m}$
- The covariance matrix for the process noise Q (multi-scattering...)
- The covariance matrix for the measurement uncertainty $\overrightarrow{V}$

Ingredients

- knowing the state $\vec{s}_{k-1|k-1}$ at the position k of the previous measurement
- First step: update state and covariance matrix to current position including process noise

state vector: $\vec{s}_{k|k-1} = F_{k-1}\vec{s}_{k-1|k-1}$

covariance matrix: $S_{k|k-1} = F_{k-1}S_{k-1|k-1}F_{k-1}^T + Q$

- Second step: taking this as the prior, update state vector and covariance matrix with information from next measurement

Kalman gain matrix: $K_k = S_{k|k-1}H_k^T(V_k + H_k S_{k|k-1}H_k^T)^{-1}$

updating state vector: $\vec{s}_{k|k} = \vec{s}_{k|k-1} + K_k(\boxed{\vec{m}} - H_k\vec{s}_{k|k-1})$

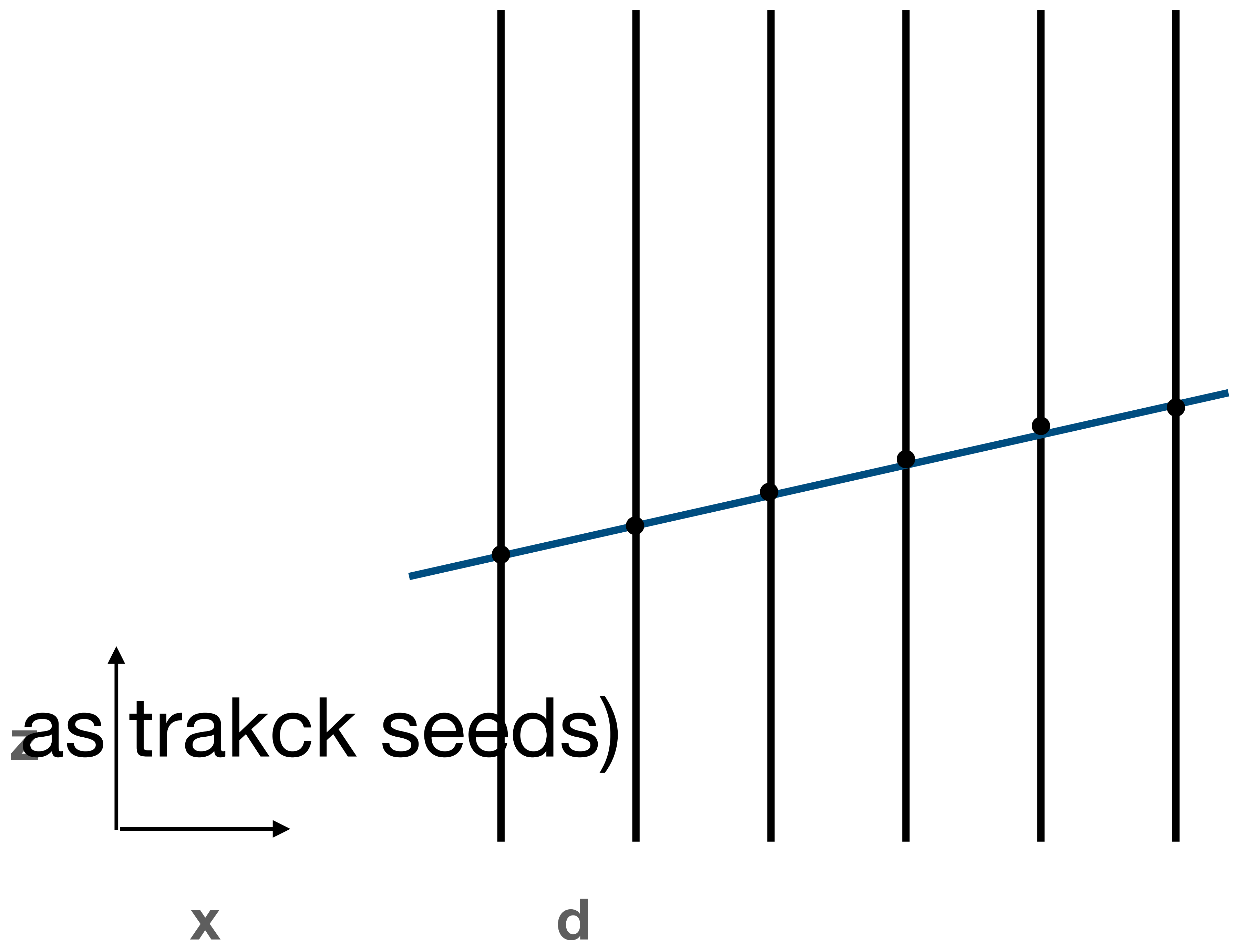
updating covariance matrix: $S_{k|k} = (1 - K_k H_k)S_{k|k-1}$

tracking in two dimensions

- track state vector $\vec{s} = (x, y, dx/dz, dy/dz, q/p)$
- simplest example, state vector $\vec{z}_{k-1} = (z, dz/dx)_{k-1}$
- propagator matrix $F_k = \begin{pmatrix} 1 & d \\ 0 & 1 \end{pmatrix}$
- considering multi-scattering $Q = \begin{pmatrix} \theta_{MS}^2 d^2 & \theta_{MS}^2 d \\ \theta_{MS}^2 d & \theta_{MS}^2 \end{pmatrix}$
- prediction k-th position track vector (two outer layers as trakck seeds)

$$z_{k|k-1} = F_k z_{k-1|k-1} = \begin{pmatrix} 1 & d \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z_{k-1} \\ z'_{k-1} \end{pmatrix}$$

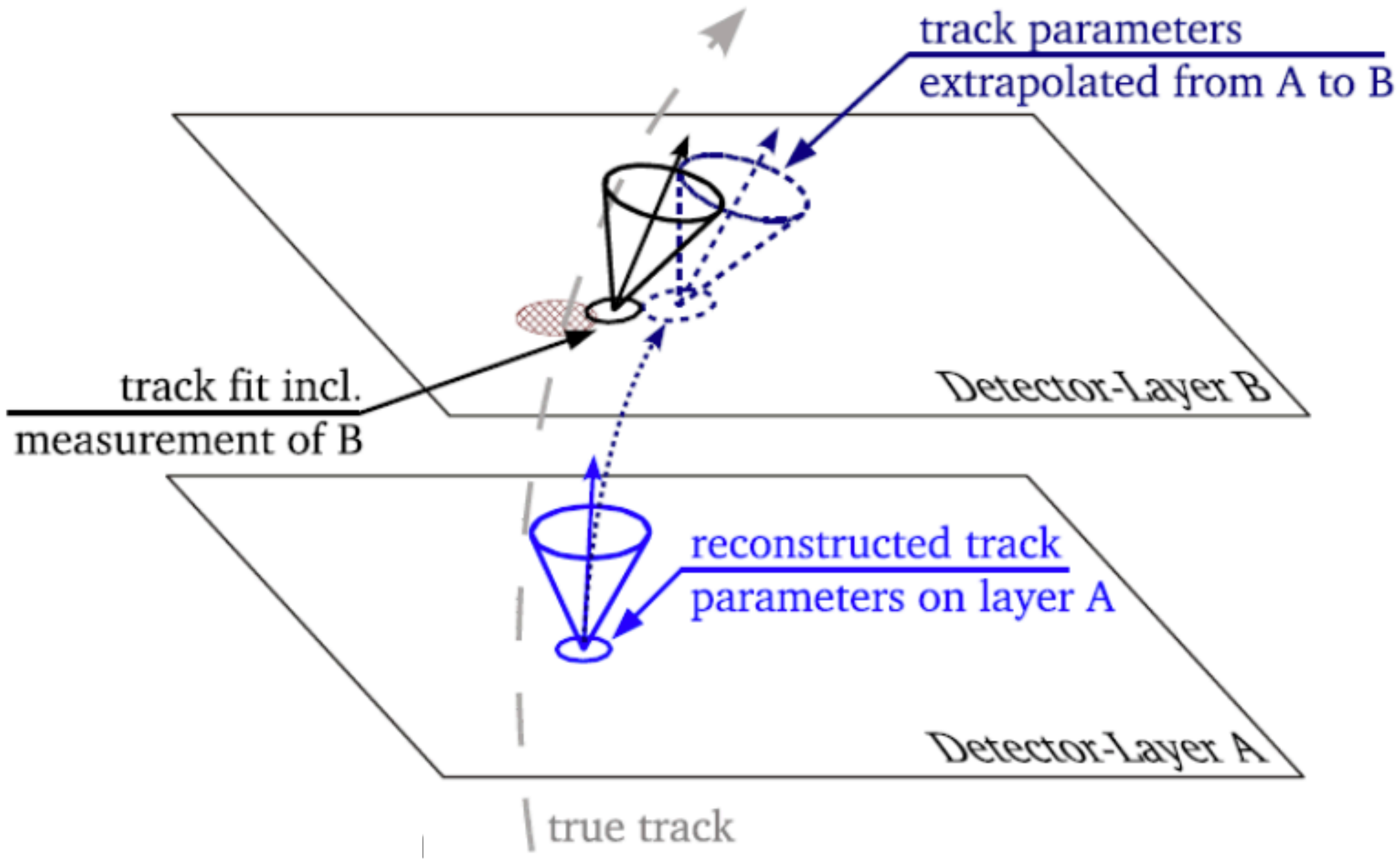
$$C_{k|k-1} = F_k C_{k-1} F_k^T + Q$$



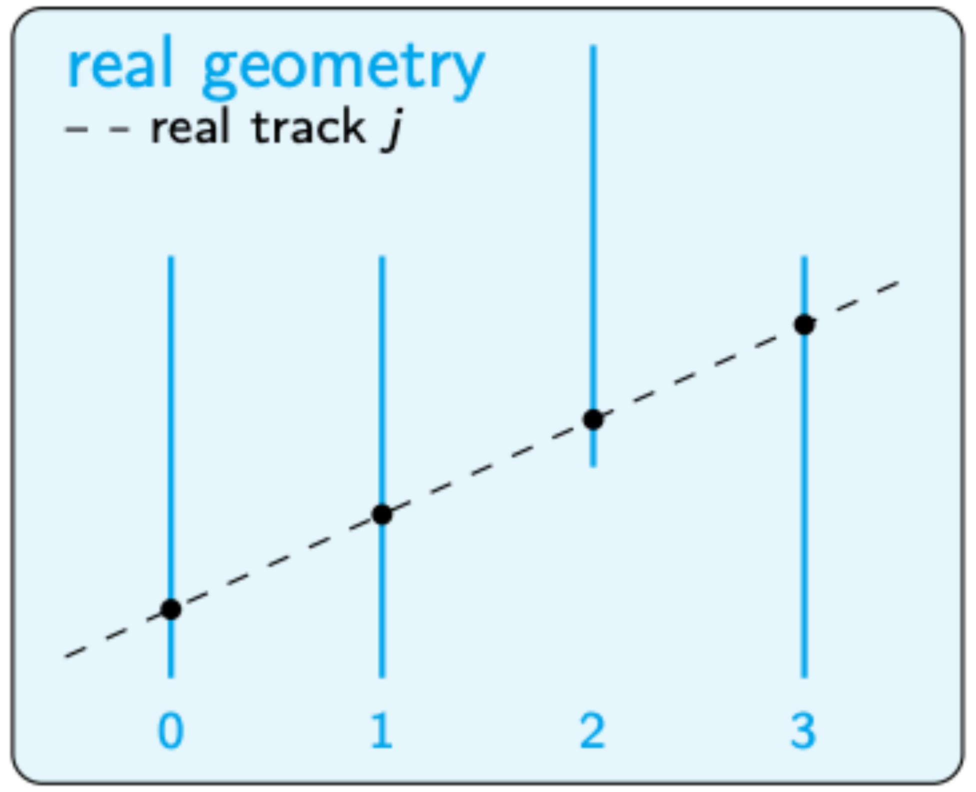
- updating the state vector and covarinance using Kalman gain matrix taking k-th measurement point into account

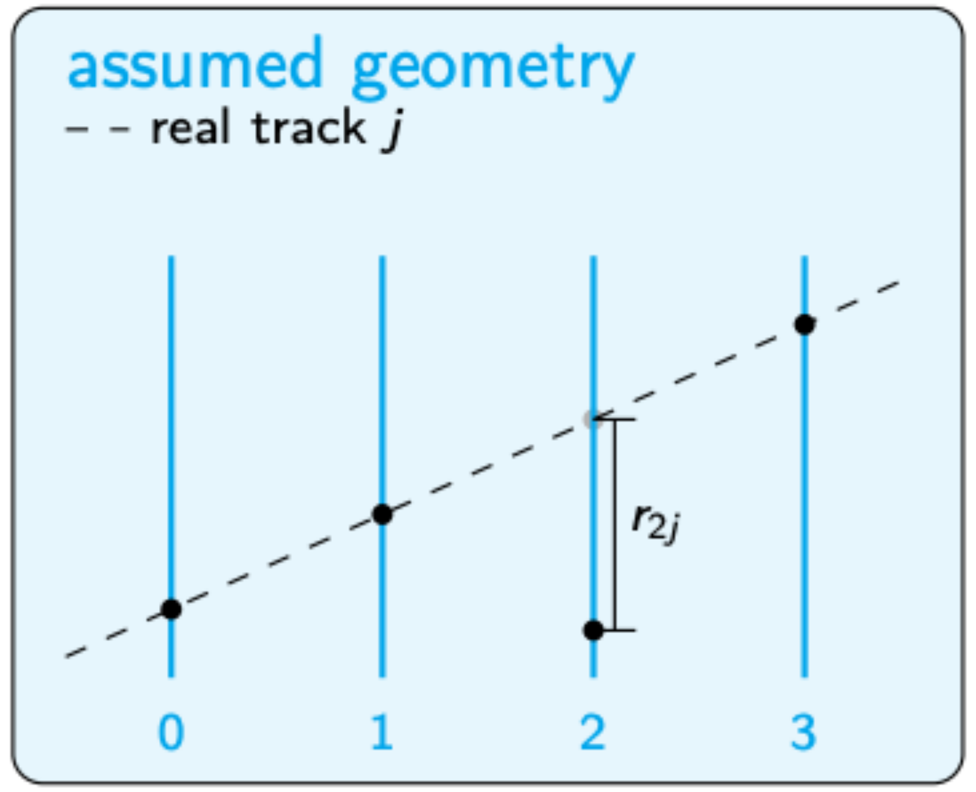
- the residual r, the difference between a measurement m and its prediction by the track state

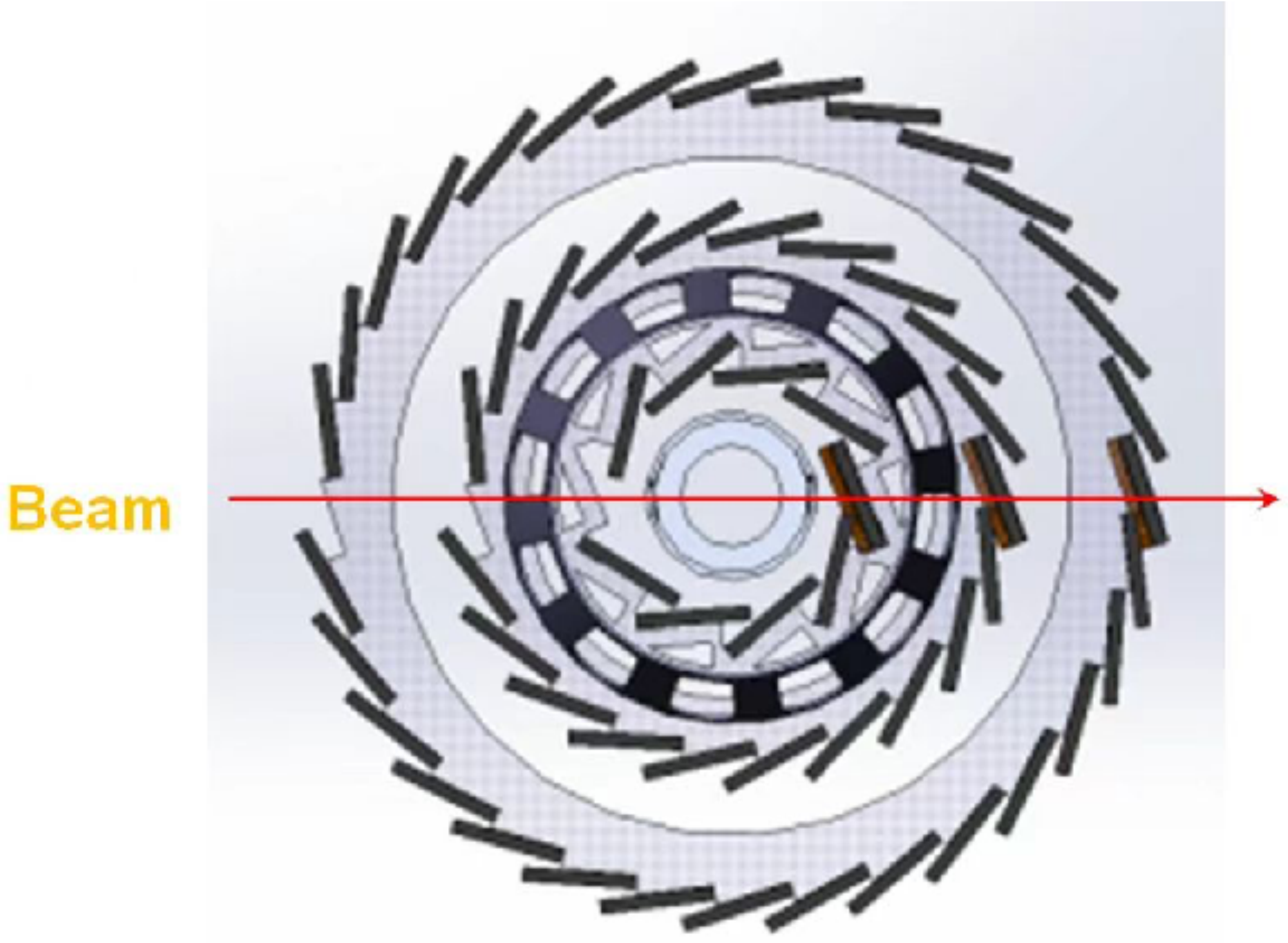
$$r_k = (1 - H_k K_k) r_k^{k-1}, \; R_k = (1 - H_k K_k) V_k, \; \chi^2 = r^T R^{-1} r$$



Global alignment with tracks





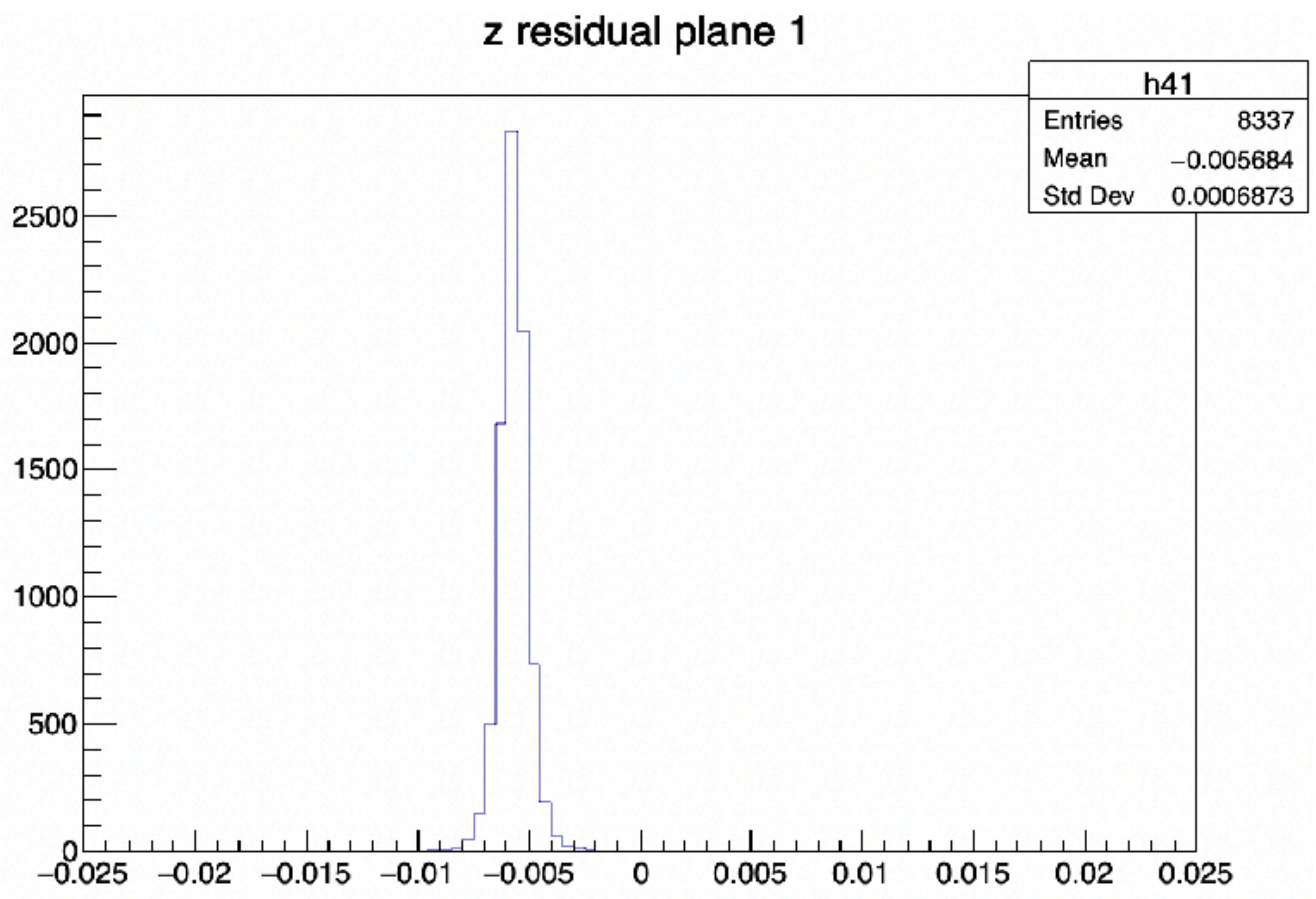


Idea: track-hit residuals r_{ij} between predicted and measured hit positions as a measure of misalignment

Tracks correlate alignment and track parameters

Approach: Minimize $\chi^2 = \sum_j^{\text{tracks}} \sum_i^{\text{hits}} \frac{r_{ij}^2}{\sigma_{ij}^2}$ (here: uncorrelated uncertainties σ_{ij}) for many tracks to determine all alignment parameters

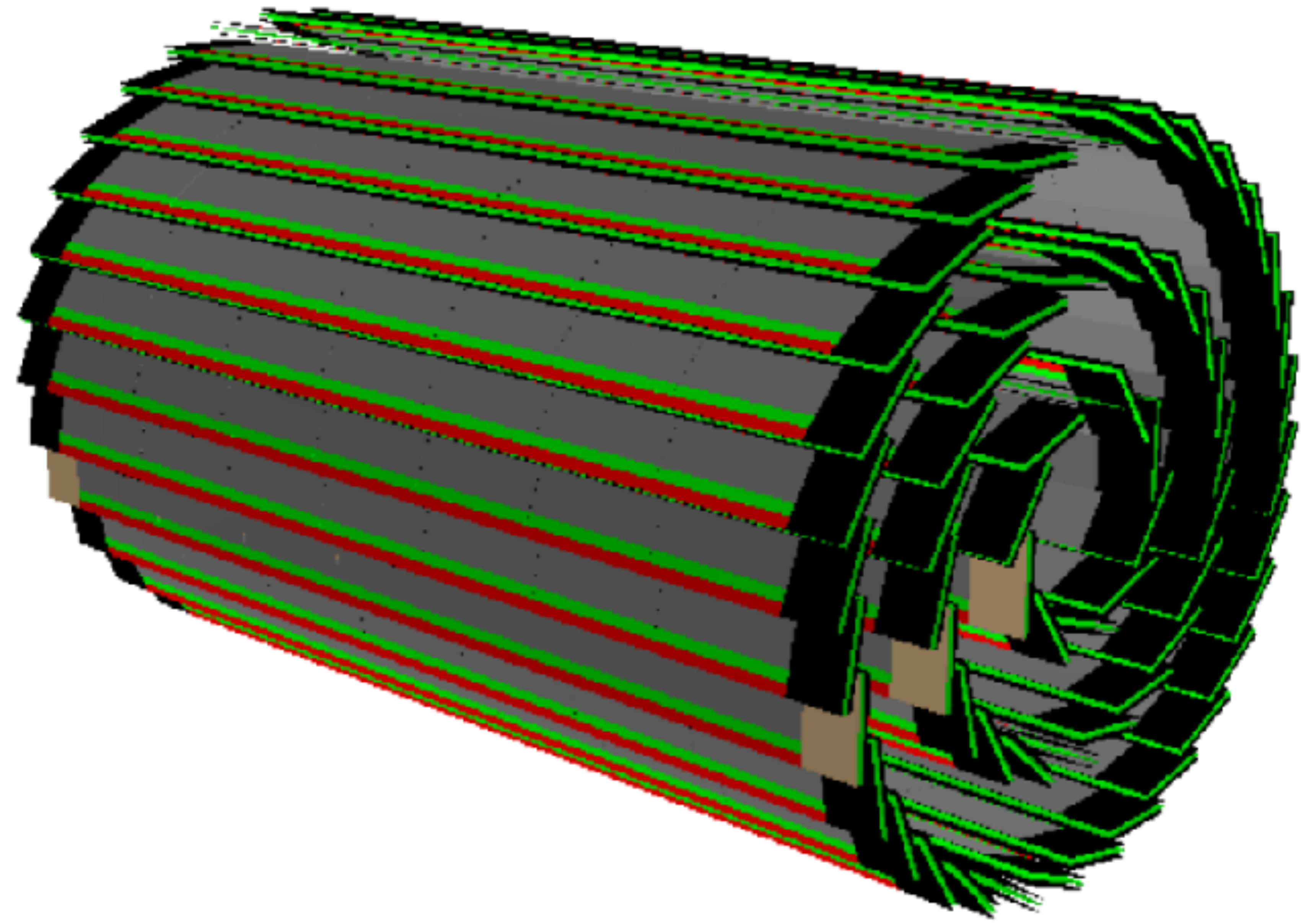
```
// random shifts in z
misalign_z[0] = 0.;
misalign_z[1] = -0.008;
misalign_z[2] = 0.;
misalign_z[3] = 0.;
misalign_z[4] = 0.0014;
misalign_z[5] = 0.;
```

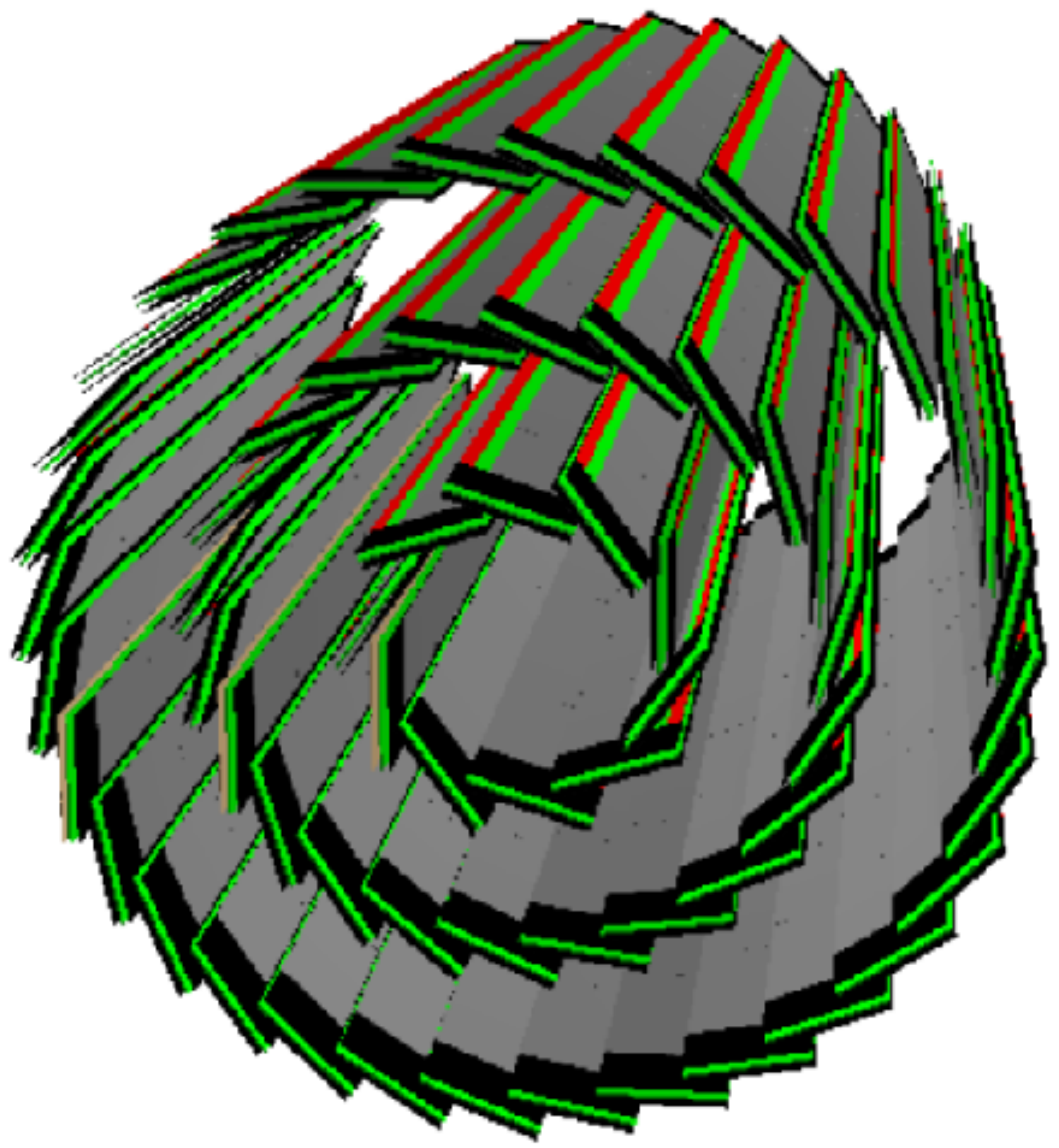


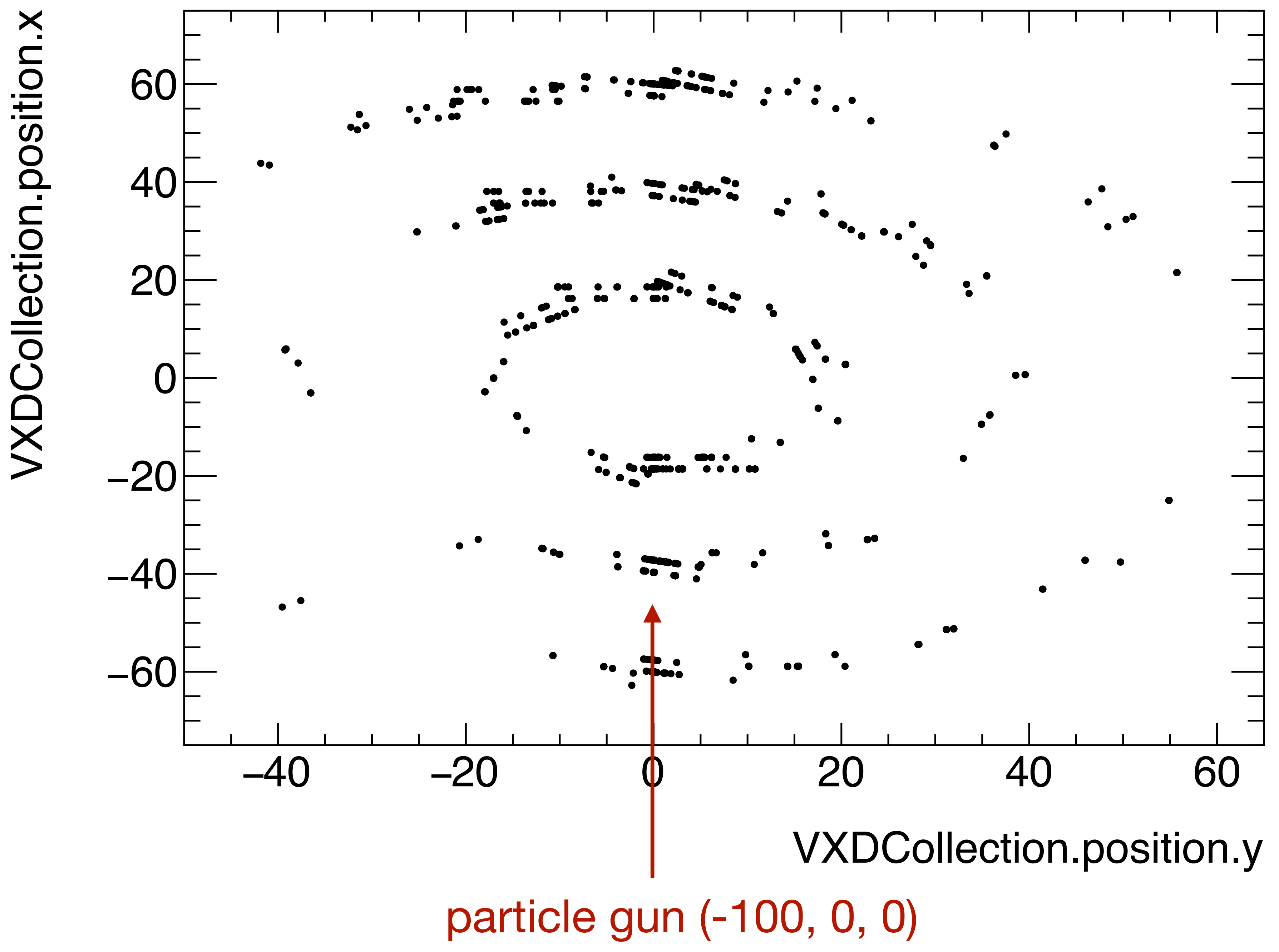

```
Alignment corrections in z:
plane 0 Original Misalignment 0 Correction 0.0031165 Alignment error -0.0031165 +- 7.57121e-05
plane 1 Original Misalignment -0.008 Correction -0.00568443 Alignment error -0.00231557 +- 9.21077e-05
plane 2 Original Misalignment 0 Correction 0.00149632 Alignment error -0.00149632 +- 9.92951e-05
plane 3 Original Misalignment 0 Correction 0.00068776 Alignment error -0.00068776 +- 9.92942e-05
plane 4 Original Misalignment 0.0014 Correction 0.00129078 Alignment error 0.000109223 +- 9.21051e-05
plane 5 Original Misalignment 0 Correction -0.000906882 Alignment error 0.000906882 +- 7.57067e-05
```

$$\chi^2 = (\overline{m} - H\overline{x})^T V^{-1} (\overline{m} - H\overline{x})$$

$$\overline{x} = (H^T V^{-1} H)^{-1} H^T V^{-1} \overline{m}$$







VXDCollection.position.x
60
40
20
0
−20
−40
−60
−40
−20
0
20
40
60
VXDCollection.position.y
particle gun (-100, 0, 0)