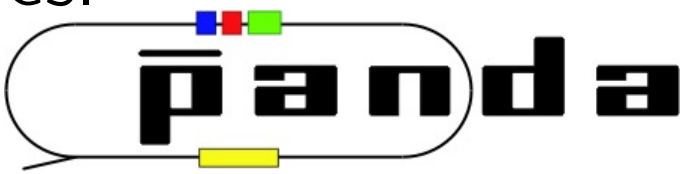
Reconstructing Longitudinal Track Parameters with the STT

Walter Ikegami Andersson

Uppsala University
on behalf of the $\overline{\text{P}}$ANDA collaboration

$\overline{\text{P}}$ANDA Tracking Workshop
September 18-19, 2018
GSI



Outline

- Introduction to STT and STTCellTrackFinder
- Method 1: Hough Transformation
  Described in STT design report, used in Secondary Track Finder
  by L. Lavezzi
- Method 2: Combinatorics, "Climbing Tree"
  Used in alternative method developed at WASA-at-COSY
  by M. Jacewicz
- Summary & Outlook

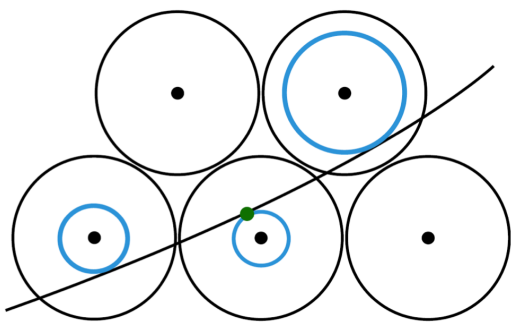
The $\overline{\text{P}}$ANDA Straw Tube Tracker

STT specifications	
Total straws	4636
Axial layers	15-19
Stereo layers	8
Stereo angle	$\pm$ 2.9 deg

Numbers taken from
STT design report

Isochrone radius

Radial distance from track to wire



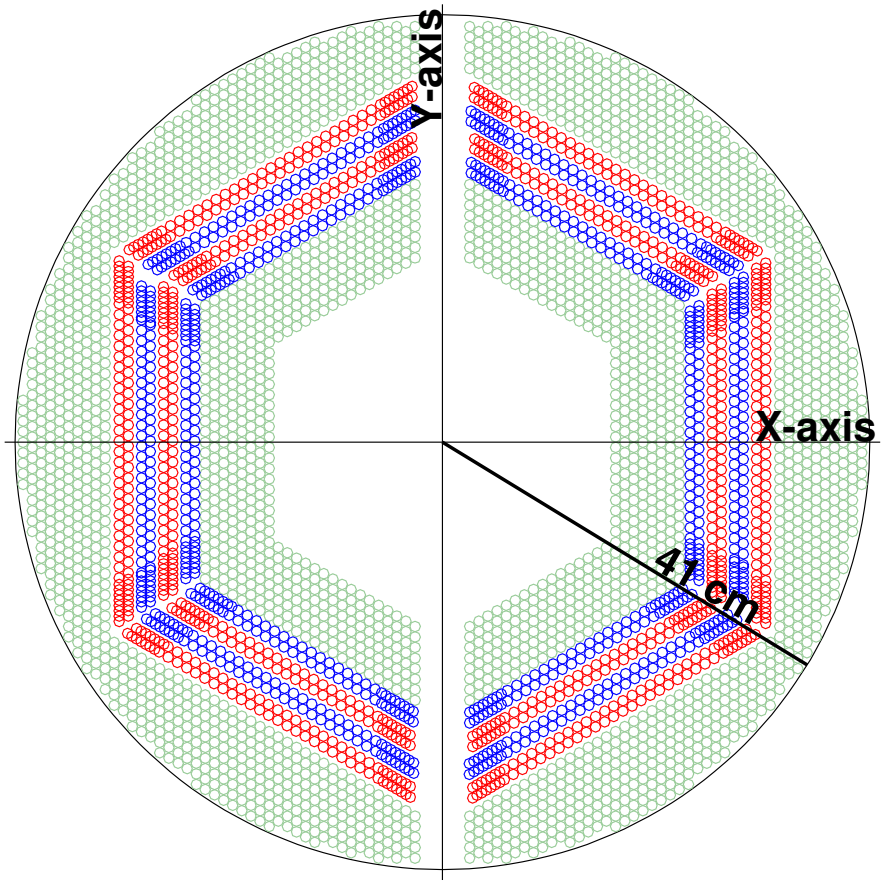


Figure: Cross sectional view of STT
Green - parallel straw
Red, blue - skewed straw

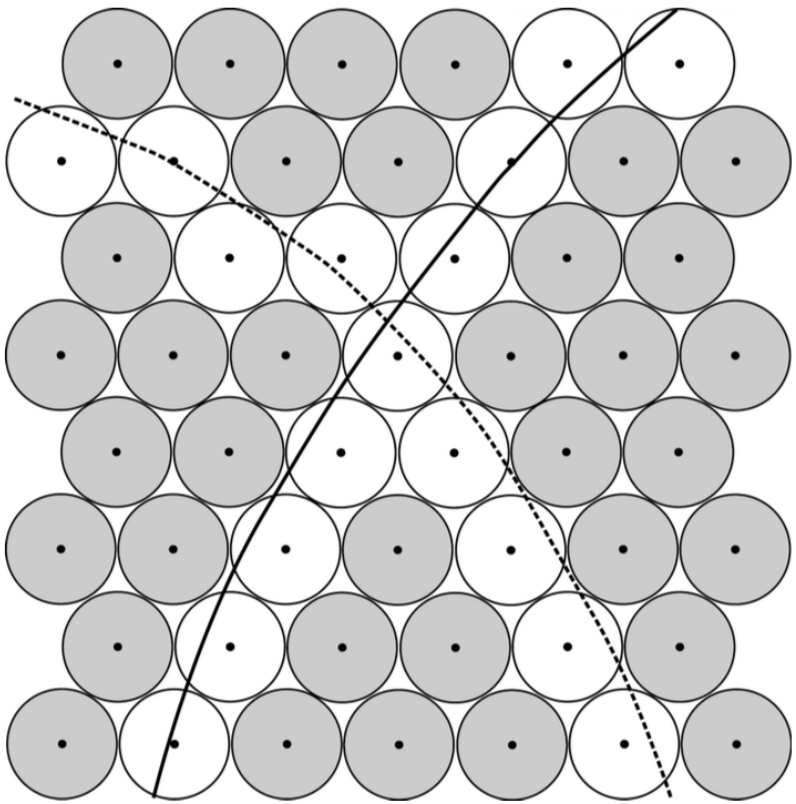
Tracking algorithm dedicated for STT

Track reconstruction algorithm using only STT.
(J. Schumann, Forschungszentrum Jülich)

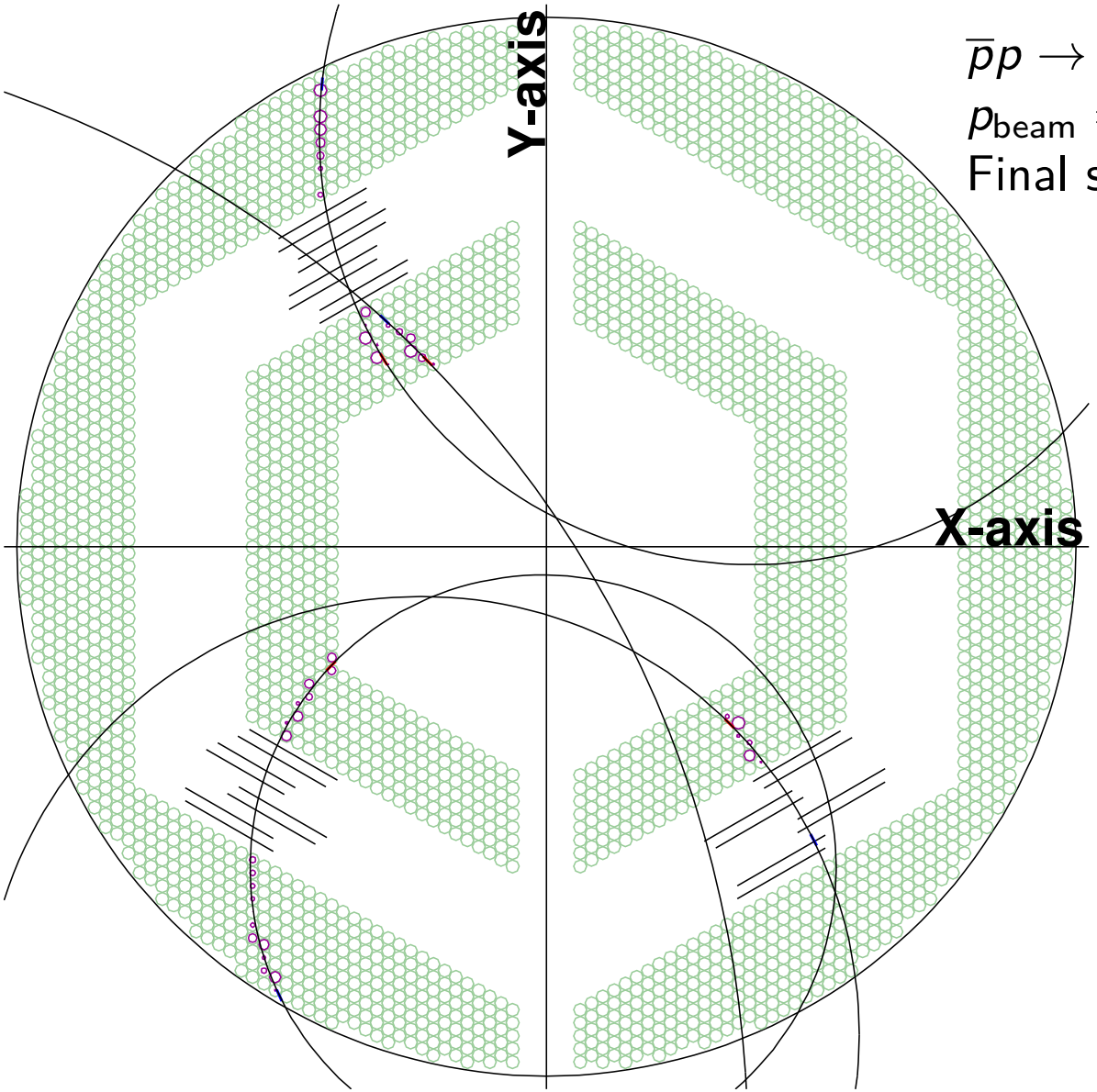
1. Cluster hits in parallel straws into tracklets (neighboring relations)
2. Refined circle fit using isochrones
3. Assign skewed straw hits to track

Output: circle for each track in xy-plane

Must include skewed straws to reconstruct p_z

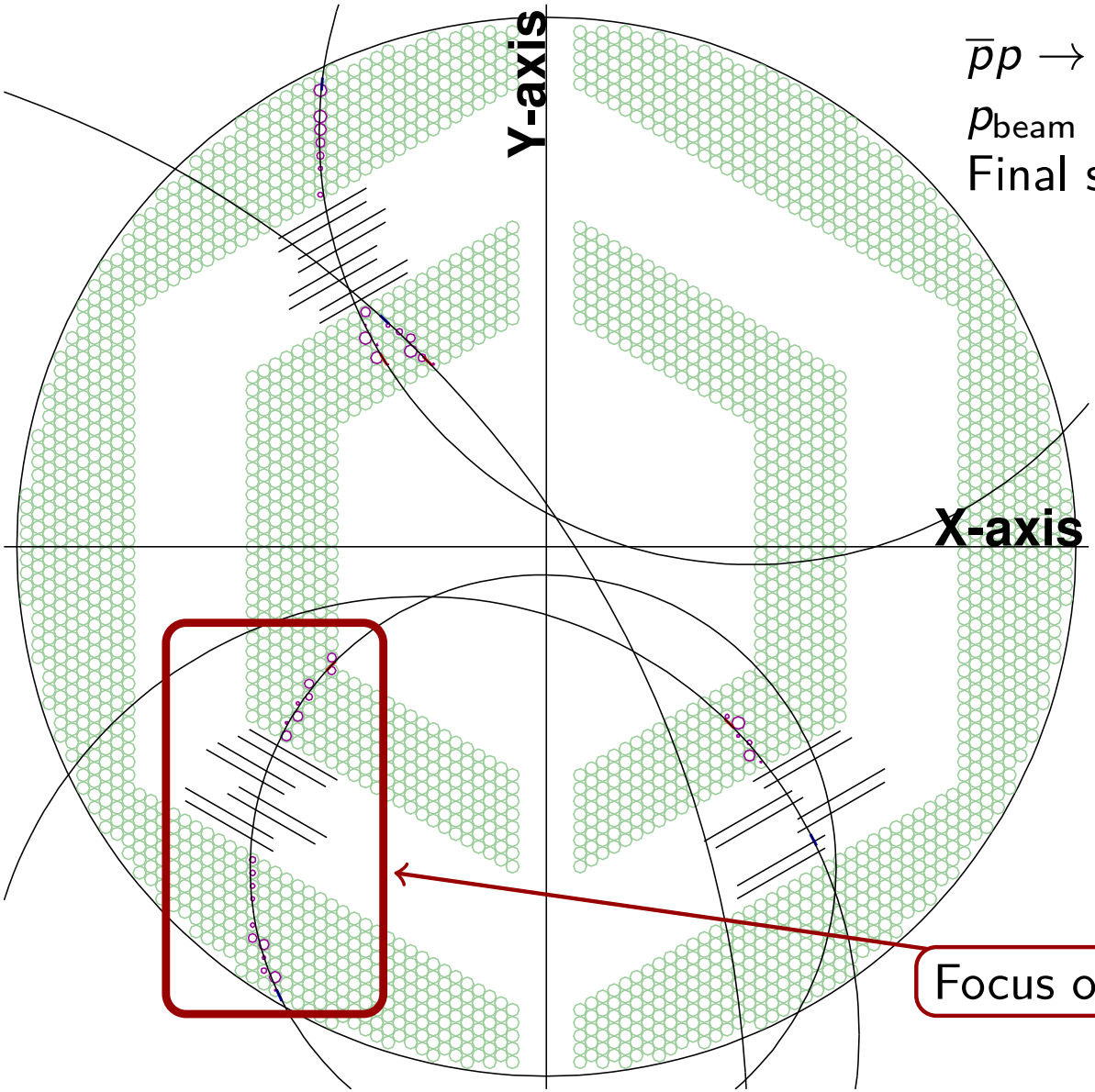


Longitudinal position from skewed straws



$\overline{p}p \rightarrow \overline{\Lambda}\Lambda$ generated at
$p_{\text{beam}} = 1.64$ GeV/c
Final state: $\overline{p}\pi^+ p\pi^-$

Longitudinal position from skewed straws



$\overline{p}p \to \overline{\Lambda}\Lambda$ generated at
$p_{\text{beam}} = 1.64$ GeV/c
Final state: $\overline{p}\pi^+ p\pi^-$

Longitudinal position from skewed straws

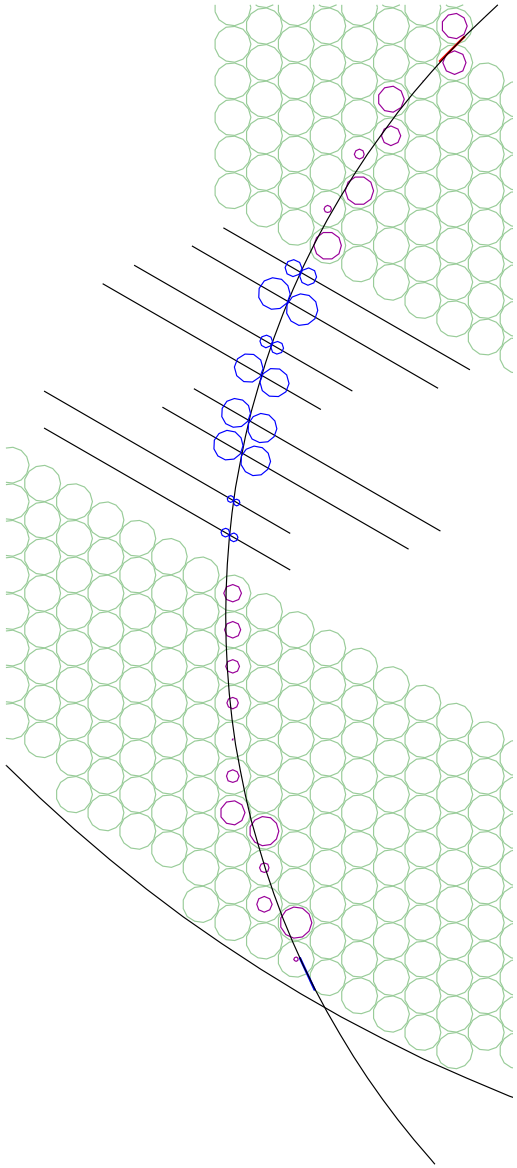
The method:

1. Extract isochrone radius in skewed straw
2. Center of isochrone gives $z-$position
3. Generate all possible isochrone positions
4. Calculate (z, ϕ)

Ambiguity: Each straw gives two possible (z, ϕ)

Solve ambiguity

Use Hough transform or combinatoric method to reject fake positions



Method 1: Hough transform

Find geometric shapes in images.

- Helix trajectory $\rightarrow$ straight line in $z - \phi$ space
- Line parameters in xy-plane, slope k and intercept m
  - $y(x) = kx + m$

Problem: The intercept parameter m unbound.

Hesse normal form

$$r = x\cos\theta + y\sin\theta$$

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right)$$

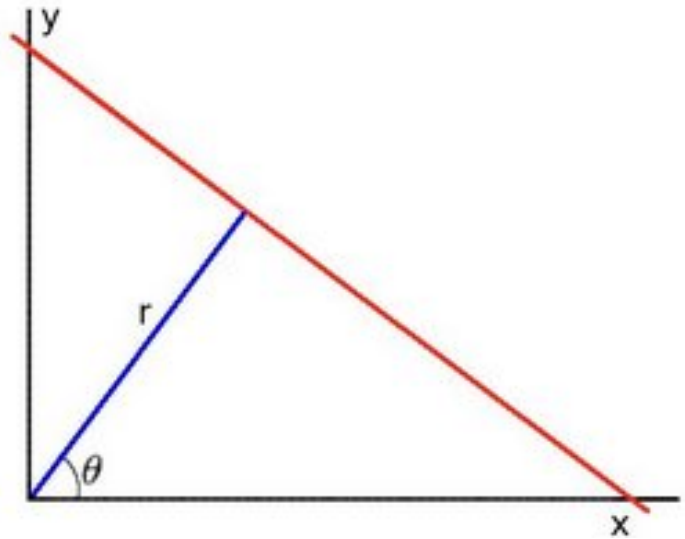
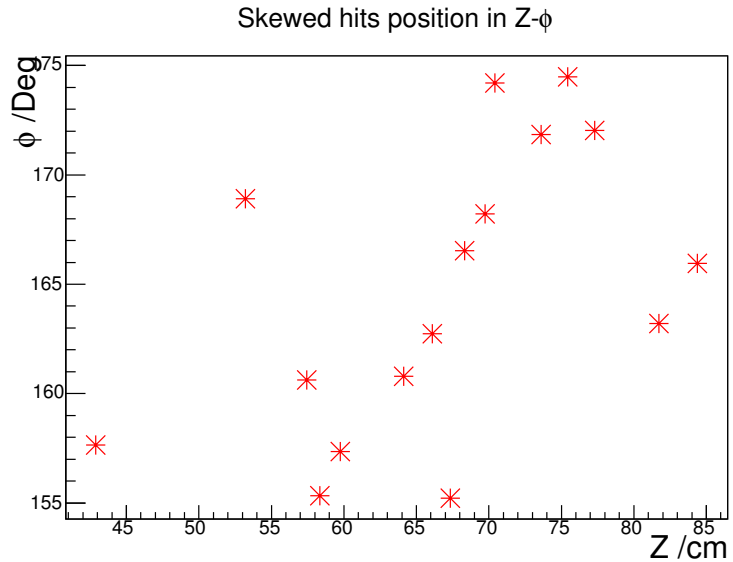


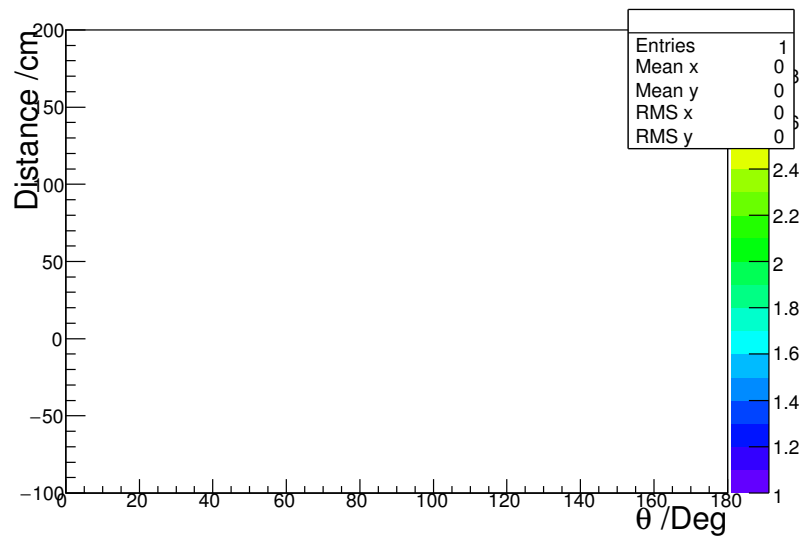
Figure: Blue line perpendicular to red line and crosses the origin

Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space

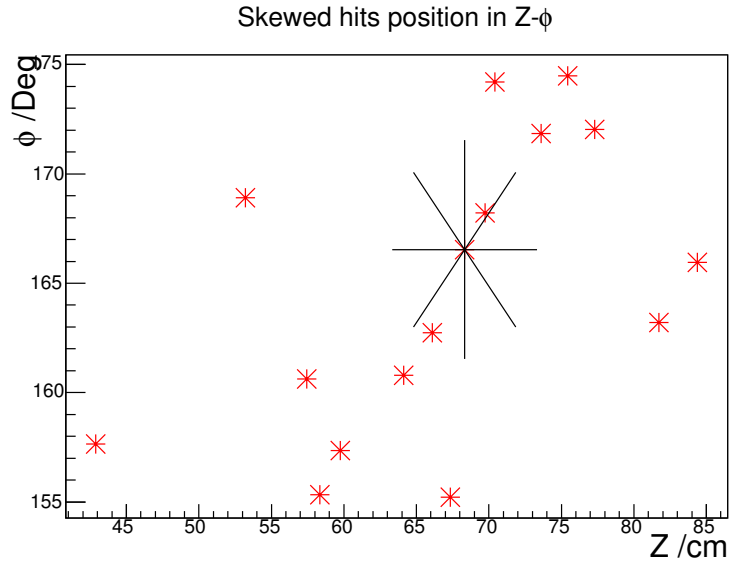


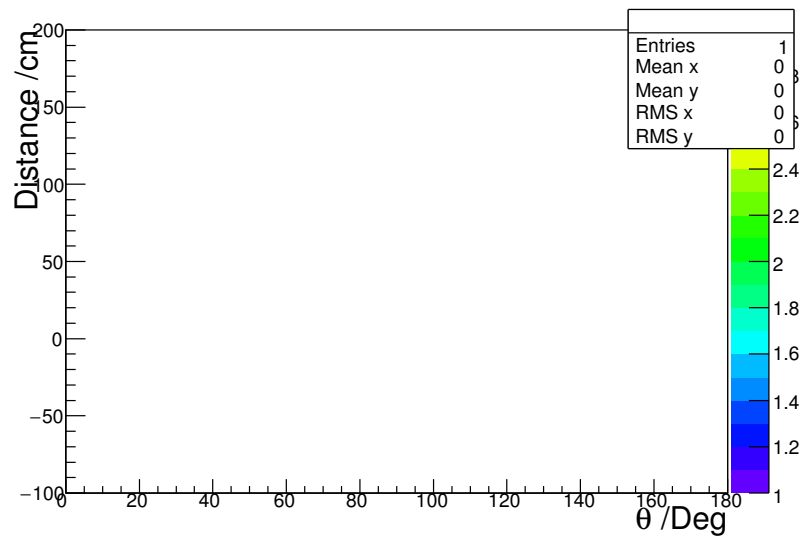


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines

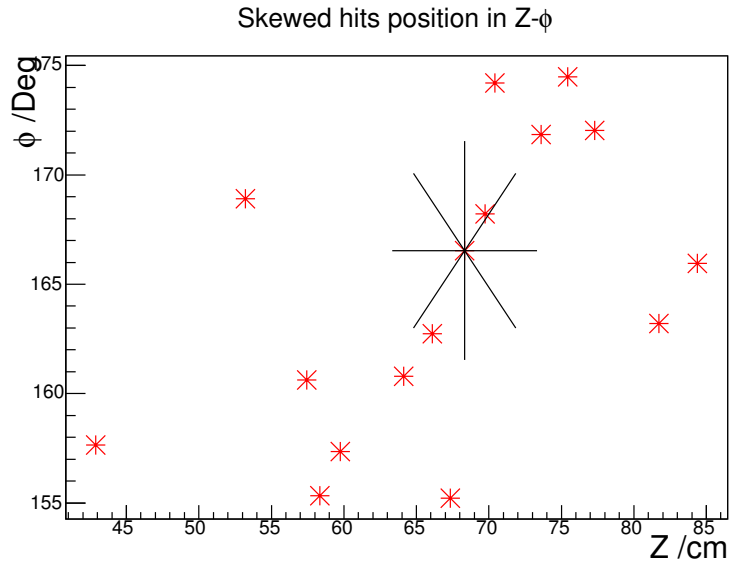


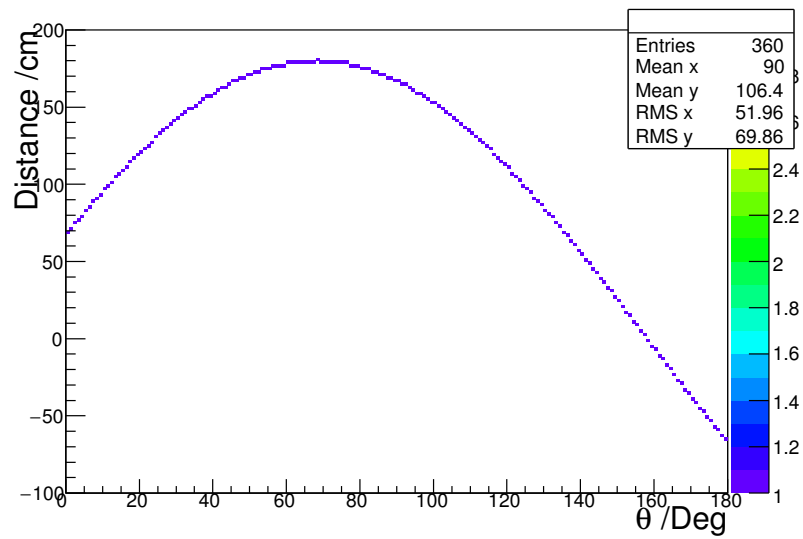


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space

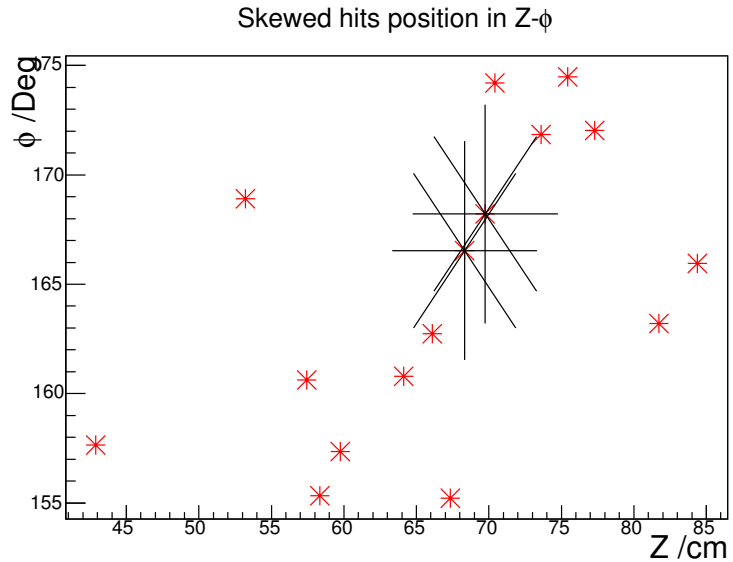


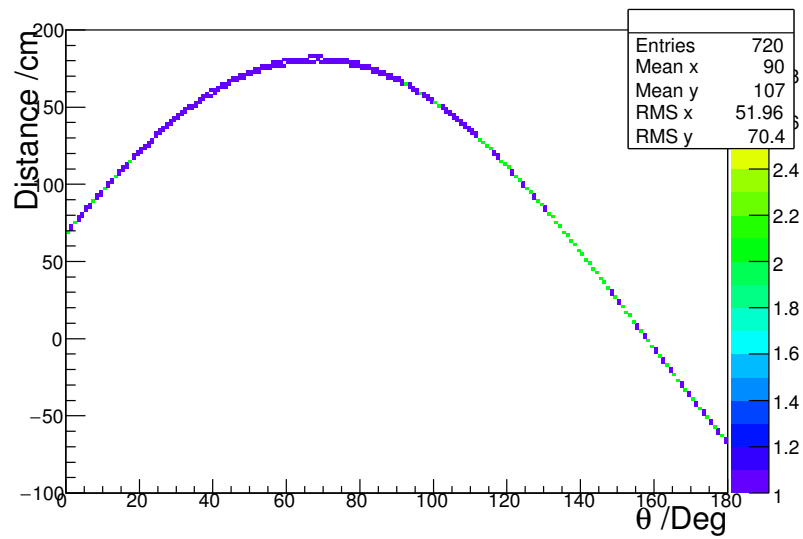


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points

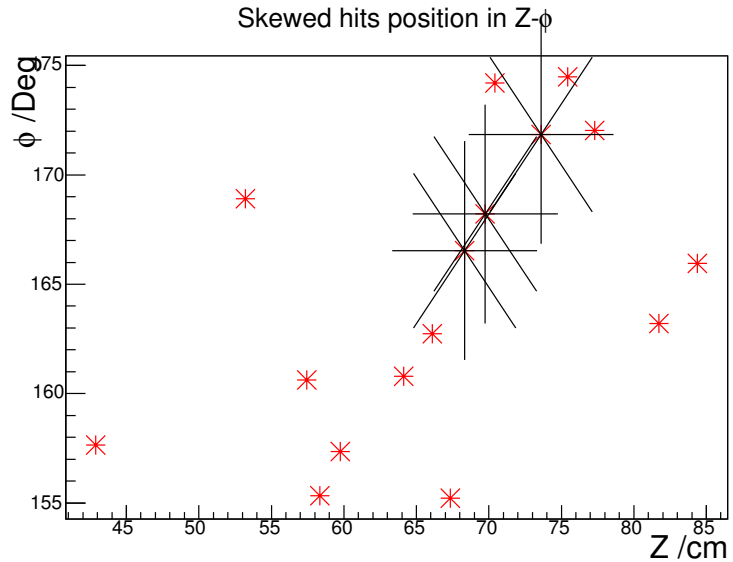


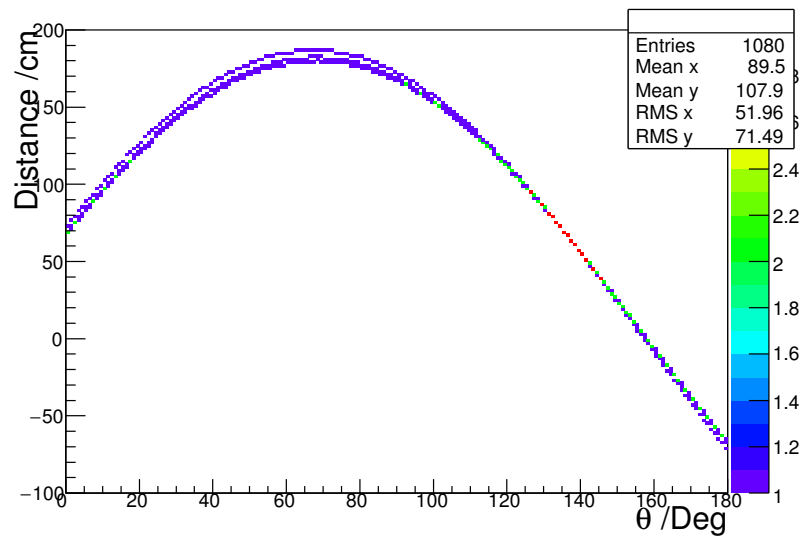


Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points



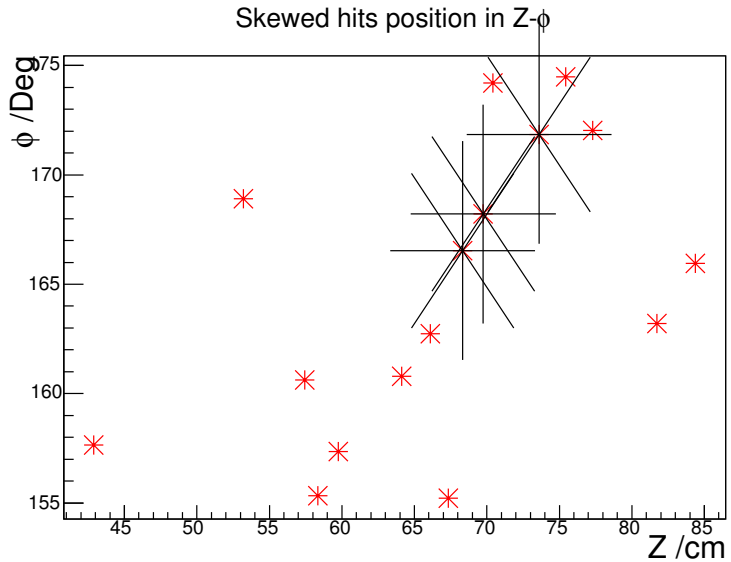


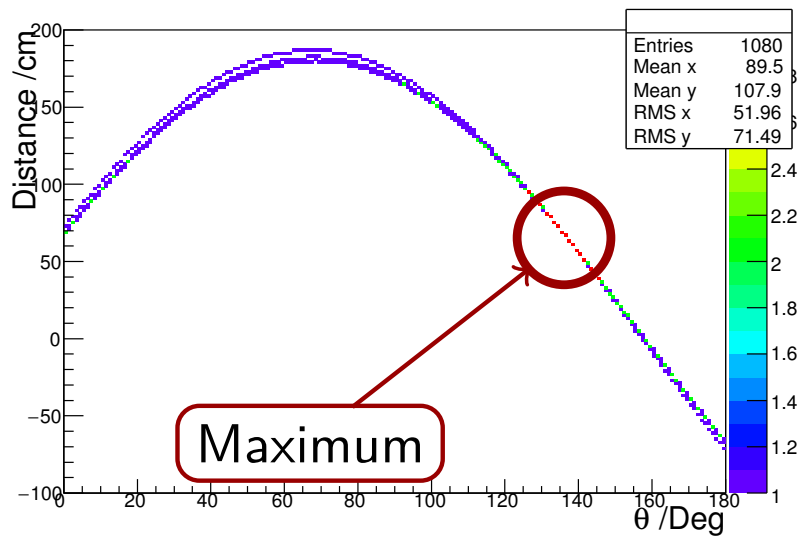
Method 1: Hough transform

The method:

1. Isochrone centers in $z - \phi$ space
2. Generate set of all lines
3. Parameters $\rightarrow$ accumulator space
4. Repeat for all points
5. Voting procedure $\rightarrow$ true line

True line found in maximum!





Method 1: Hough transform - our track

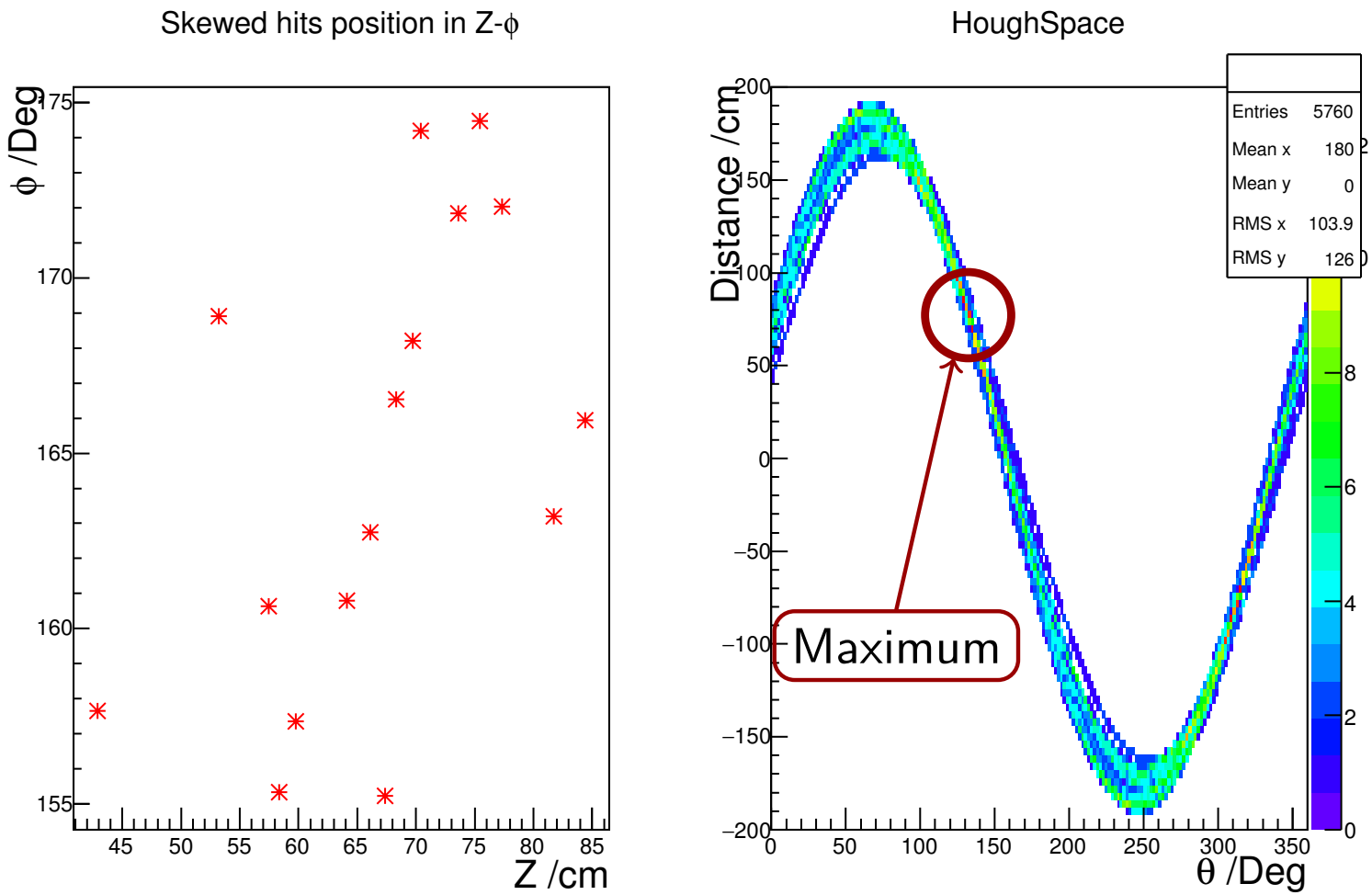


Figure: 360 lines generated for each data point in steps of 1° in θ

Method 1: Extracting helix angle

The method:

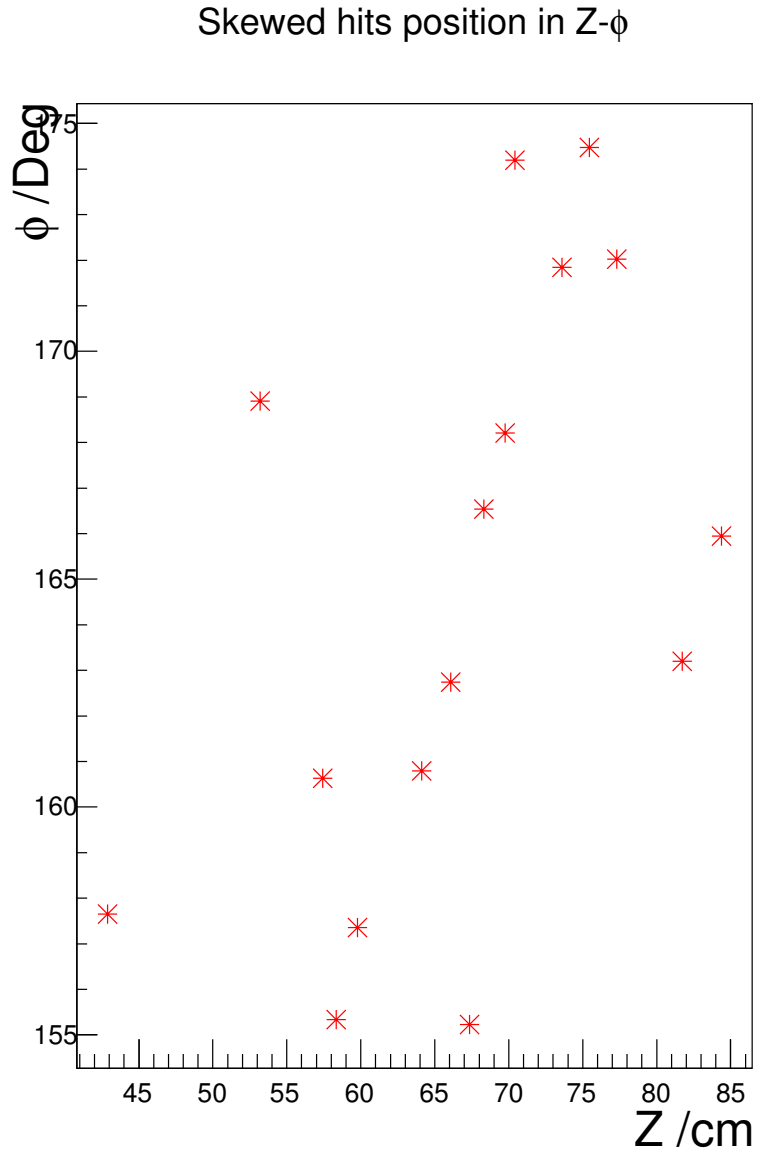
1. Calculate point of closest approach (POCA) from hits to true line
2. Accept hit with smallest POCA
3. Straight line fit with selected (z, ϕ) coordinates

Finish

The slope of the fitted line yields the helix angle. z_0 and p_z can now be extracted!

Method 2: Combinatorics

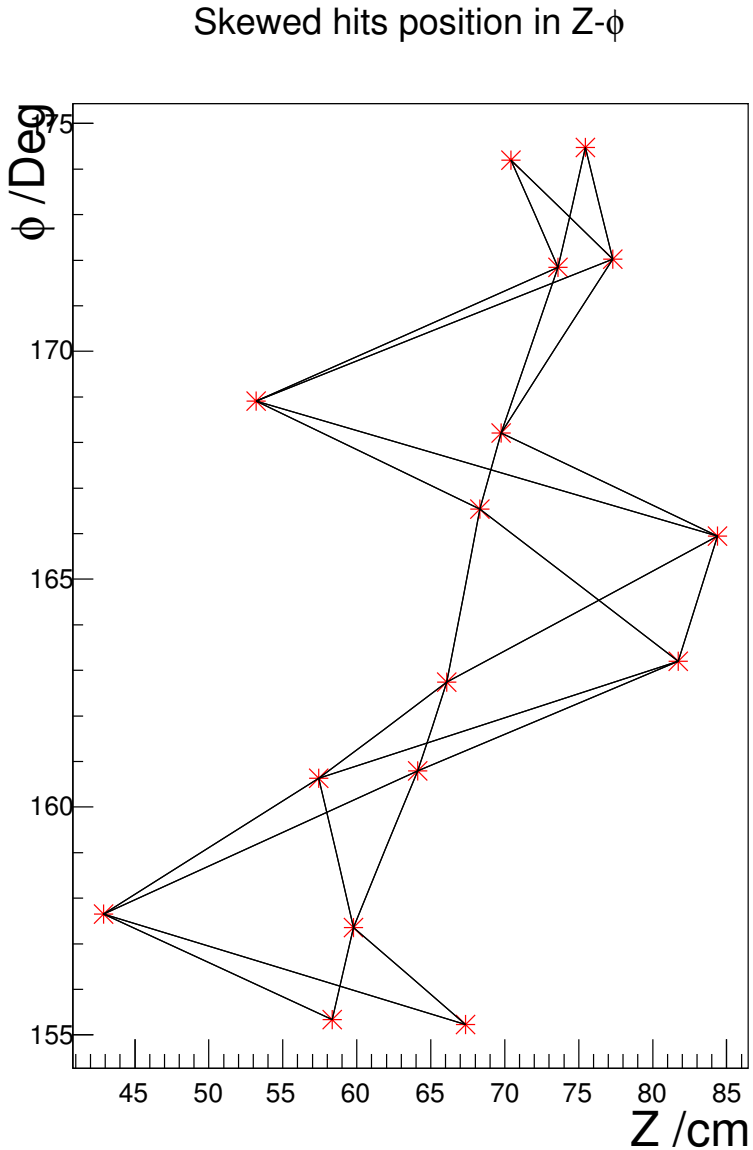
The method:



Method 2: Combinatorics

The method:

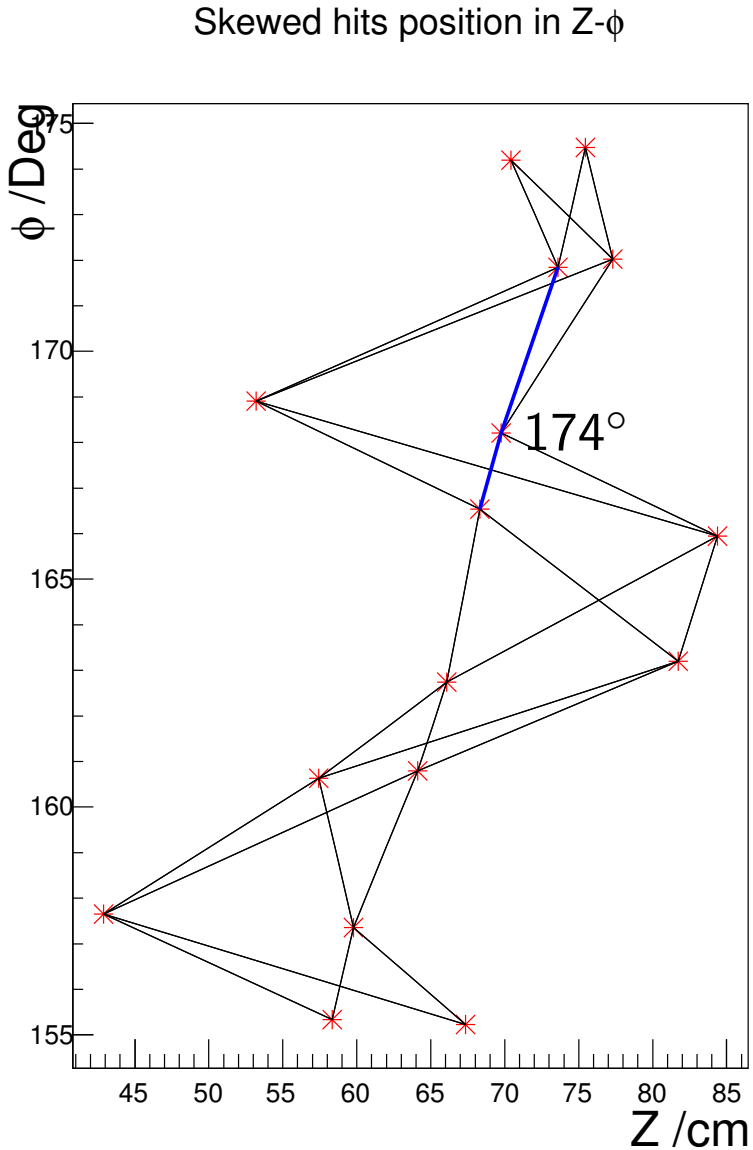
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws



Method 2: Combinatorics

The method:

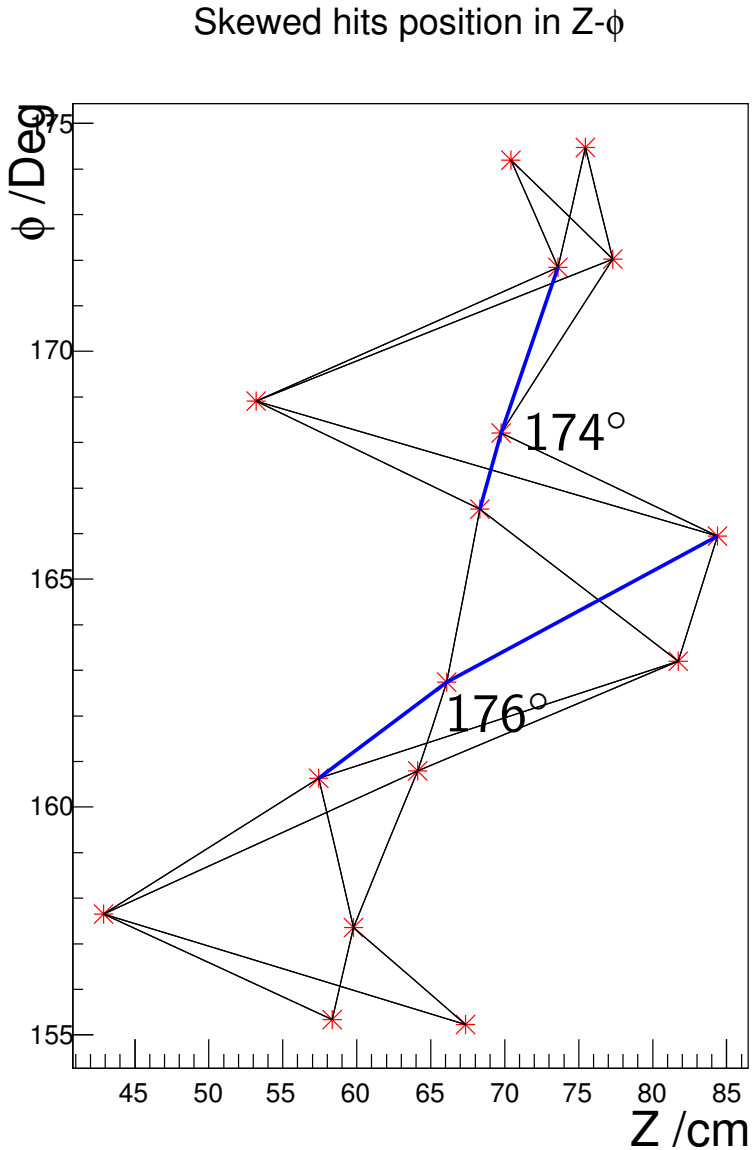
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines



Method 2: Combinatorics

The method:

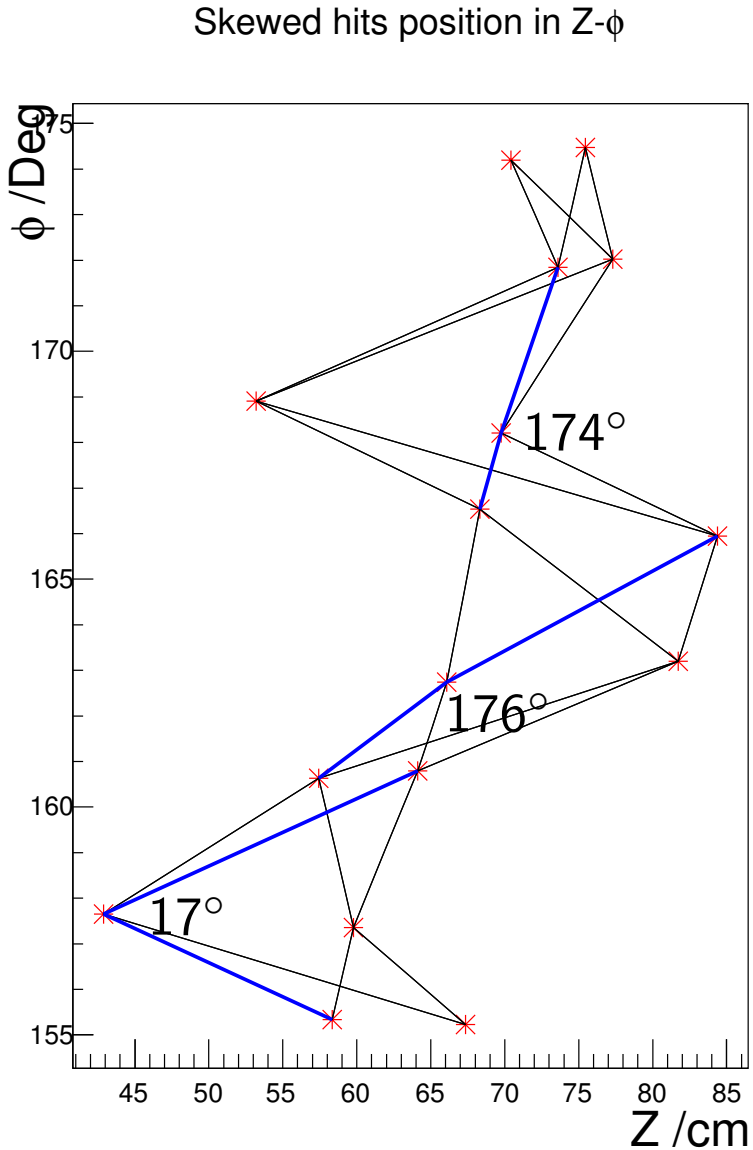
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines



Method 2: Combinatorics

The method:

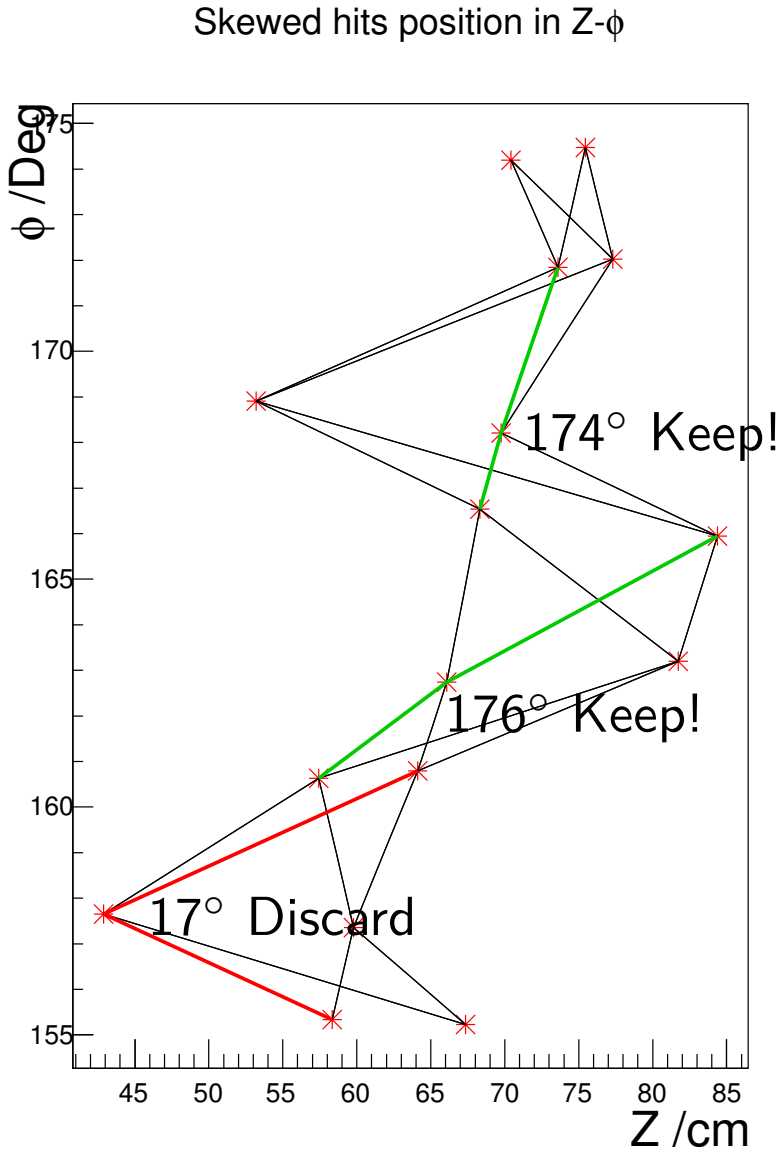
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines



Method 2: Combinatorics

The method:

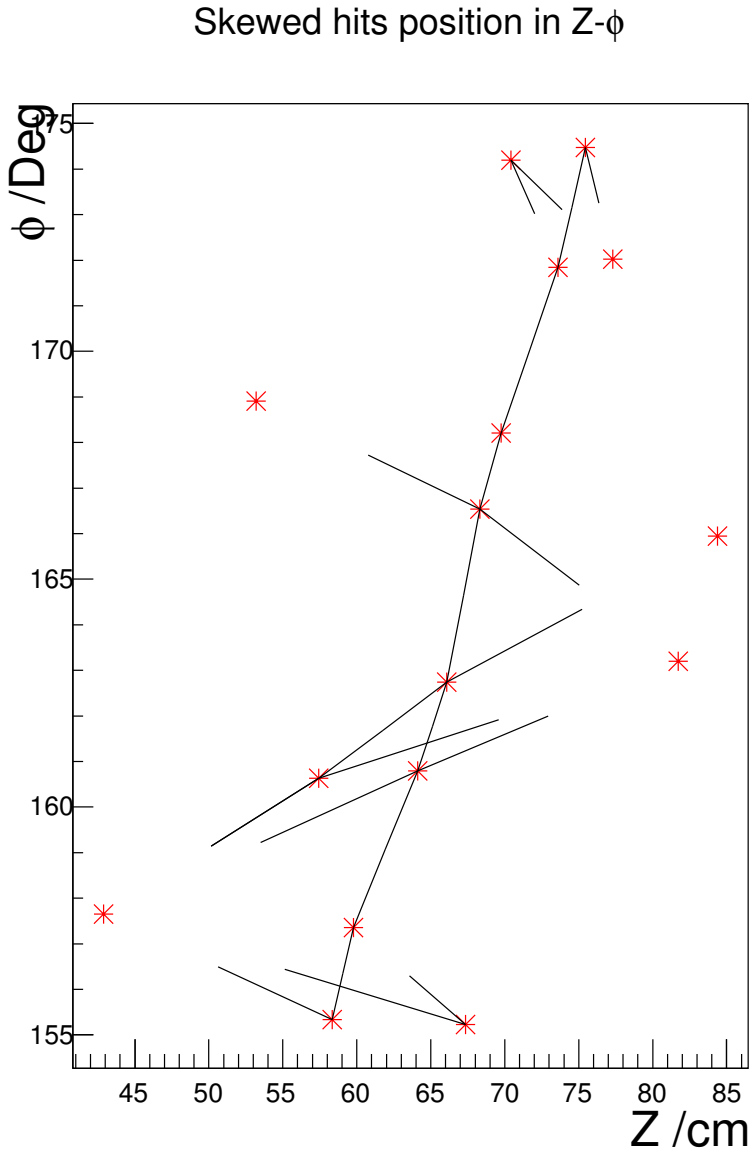
1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 160°$ $\rightarrow$ reduces number of combinations



Method 2: Combinatorics

The method:

1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 160°$ $\rightarrow$ reduces number of combinations

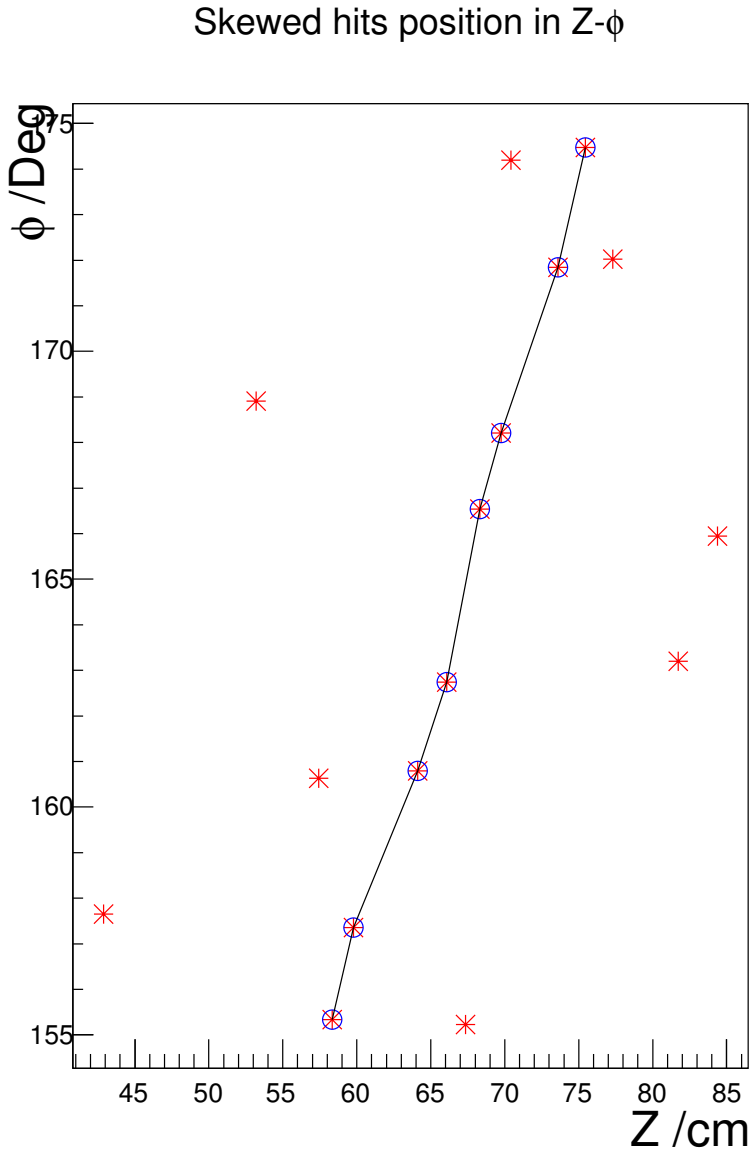


Method 2: Combinatorics

The method:

1. Calculate all lines between (z, ϕ) points in neighboring skewed straws
2. Calculate angle between all possible neighboring lines
3. Ignore paths where $\theta < 160°$ $\rightarrow$ reduces number of combinations
4. Choose path with $\min(\sum \theta_i - 180°)$

Hits in final path chosen as true hits



PzFinder - Code structure

- PndSttSkewStrawPzFinderTask.cxx
  - PndTrack - Standard $\overline{\text{P}}$ANDA track object
  - PndTrackCand - PndSttHits belonging to track
  - PndRiemannTrack - Riemann circle parameters to track
- PndSttSkewStrawPzFinder.cxx
  - MoveSkewedHitstoCircle
    - Calculates all possible (z, ϕ) in skewed straw
  - HoughTrueIsoFinder
    - Fills accumulator space, find maximum, rejects fake hits with POCA
  - LineCombiIsoFinder
    - Generates lines, calculates angles, find best path
  - PzLineFitExtract
    - Simple line fit to true (z, ϕ) hits and extracts helix angle
- PndSttSkewStrawPzFinderAnaTask.cxx
  - Task for analysing and drawing output

Reconstruction macro

```
PndSttCellTrackFinderTask *TrackFinder = new PndSttCellTrackFinderTask();
TrackFinder->SetPersistence(kTRUE);
TrackFinder->SetAnalyseSteps(kTRUE);
TrackFinder->SetVerbose(0);
fRun->AddTask(TrackFinder);

PndSttSkewStrawPzFinderTask *PzFinder = new PndSttSkewStrawPzFinderTask();
PzFinder->StoreData(kTRUE);
fRun->AddTask(PzFinder);

PndSttSkewStrawPzFinderAnalysisTask *PzAna = new
     PndSttSkewStrawPzFinderAnalysisTask();
fRun->AddTask(PzAna);

PndMCTrackAssociator* trackMC = new PndMCTrackAssociator();
trackMC->SetTrackInBranchName("FinalTrack");
trackMC->SetTrackOutBranchName("SttMvdGemTrackID");
trackMC->SetPersistence(kFALSE);
fRun->AddTask(trackMC);

PndRecoKalmanTask* recoKalman = new PndRecoKalmanTask();
recoKalman->SetTrackInBranchName("FinalTrack");
recoKalman->SetTrackInIDBranchName("SttMvdGemTrackID");
recoKalman->SetTrackOutBranchName("SttMvdGemGenTrack");
recoKalman->SetBusyCut(50); // CHECK to be tuned
recoKalman->SetTrackRep(0); // 0 Geane (default), 1 RK
recoKalman->SetPropagateToIP(kFALSE);
fRun->AddTask(recoKalman);
```

Summary and Outlook

- Two methods for reconstructing longitudinal track components developed
- Benchmark needed
  - Selection purity of left-right ambiguity
  - Longitudinal momentum resolution
- PndSttHit objects does not store left-right ambiguity information
- Need to develop custom QA task for benchmarking

Thank you for your attention!

Backup