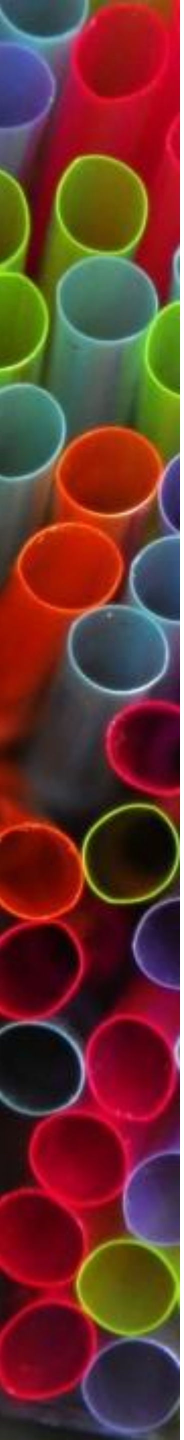


Straw Tube Tracker

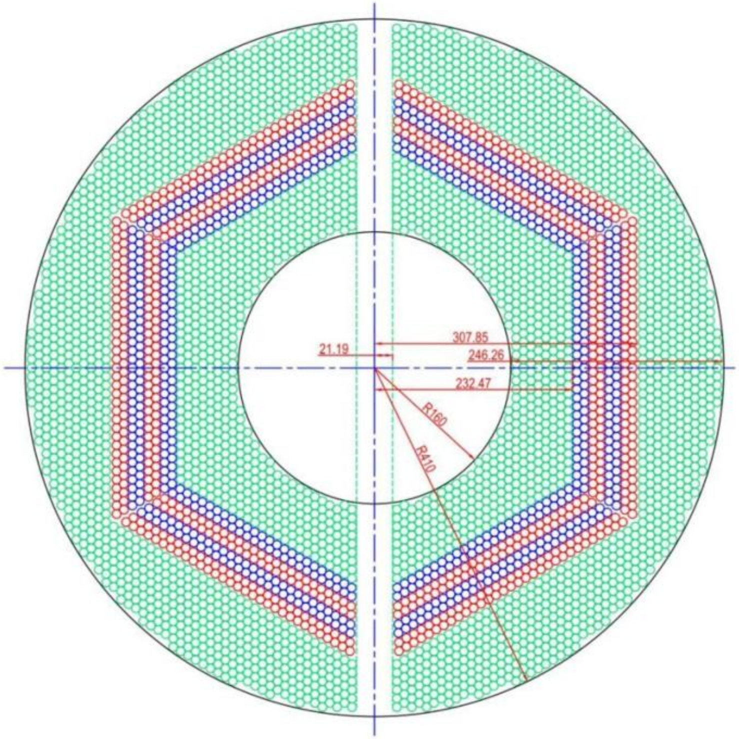
Lia Lavezzi

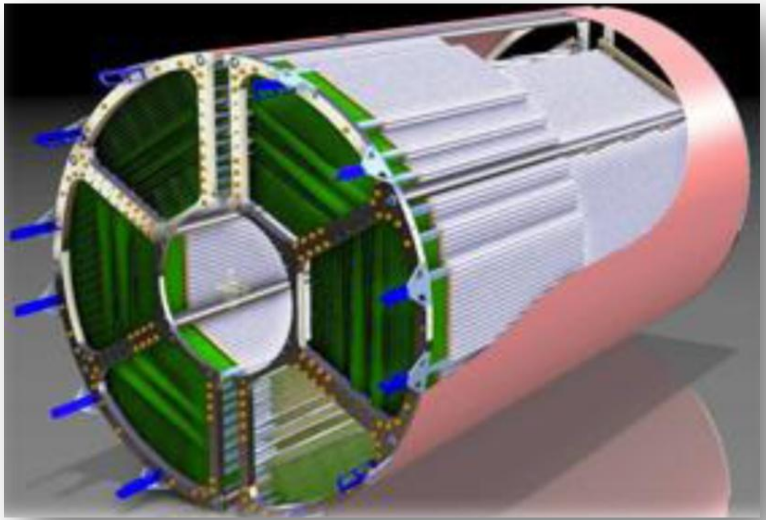
INFN Pavia

PANDA Computing Week

Torino, 23 – 27 July 2012

Geometry





Positioning
- inner radius = 150 mm
- outer radius = 420 mm
- length = 1500 mm

Layout
- ≈ 5000 tubes
- in 23-27 planar double layers:
  - 15-19 axial (green)
  - 4 skewed, ±2.89° (blue/red)

Materials
- Al-mylar, 27μm thick, Ø10μm
- 20 μm wire (W/Re, gold plated)
- Ar/CO_2 90/10%, 2 bar
- X/X_0 = 1.2 %

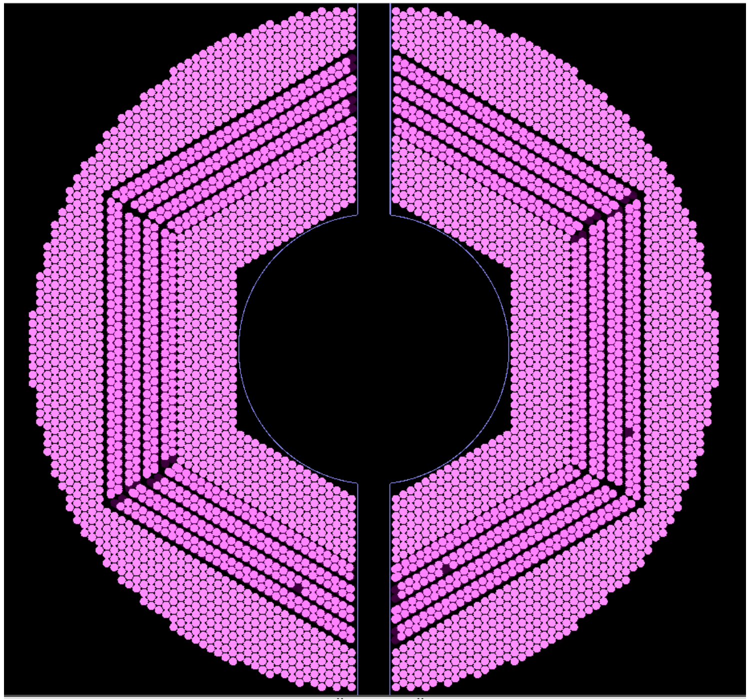
Resolution
- $\sigma_{r\phi} \approx$ 150 μm
- $\sigma_z \approx$ some mm
- $\sigma_E/E \approx$ 8%
- $\sigma_p/p \approx$ 2% @ 1GeV/c, 2T

Simulation

Simulation

the directory which contains the STT software is **`pandaroot/stt`**

❖ The detector description is contained in `PndStt`
❖ The MCpoint is `PndSttPoint` (in **`pnddata/SttData`**)



❖ the geometry is described in an ASCII file:
`straws_skewed_blocks_35cm_pipe.geo`

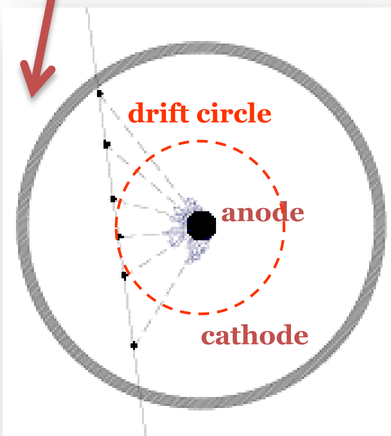
It contains the description of
❖ the tubes: mylar coating + ArCO2(90/10%) gas + wire
❖ the inner and outer supports
❖ no other passive elements

Digitization

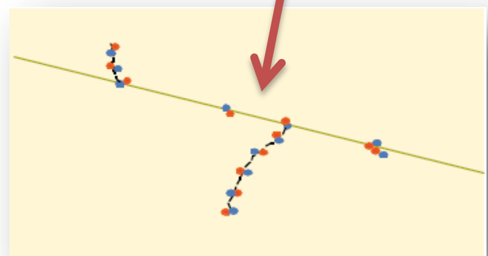
Digitization

*The single straw tube is a **drift** tube: it gives a drift time → drift radius*

1. Passage of charged particle

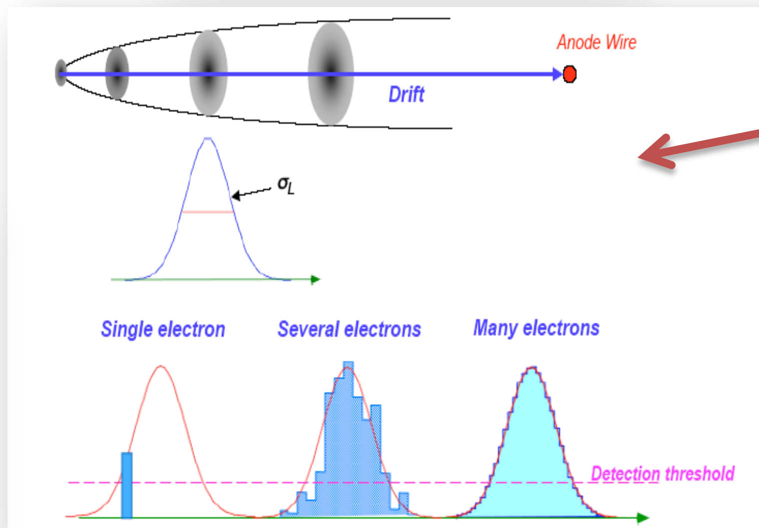


2. Ionization: couple e⁻/ion



3. Drift

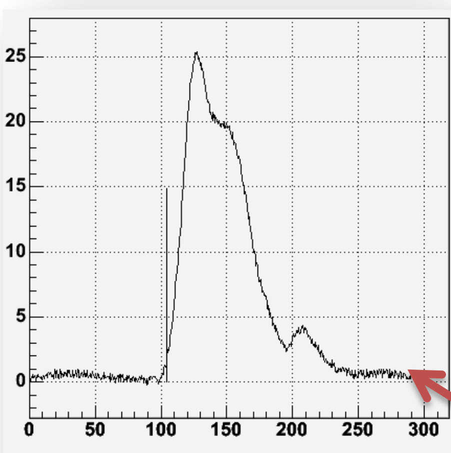
4. Diffusion



5. Avalanche

Polya, gain ~ 10^4

6. Signal



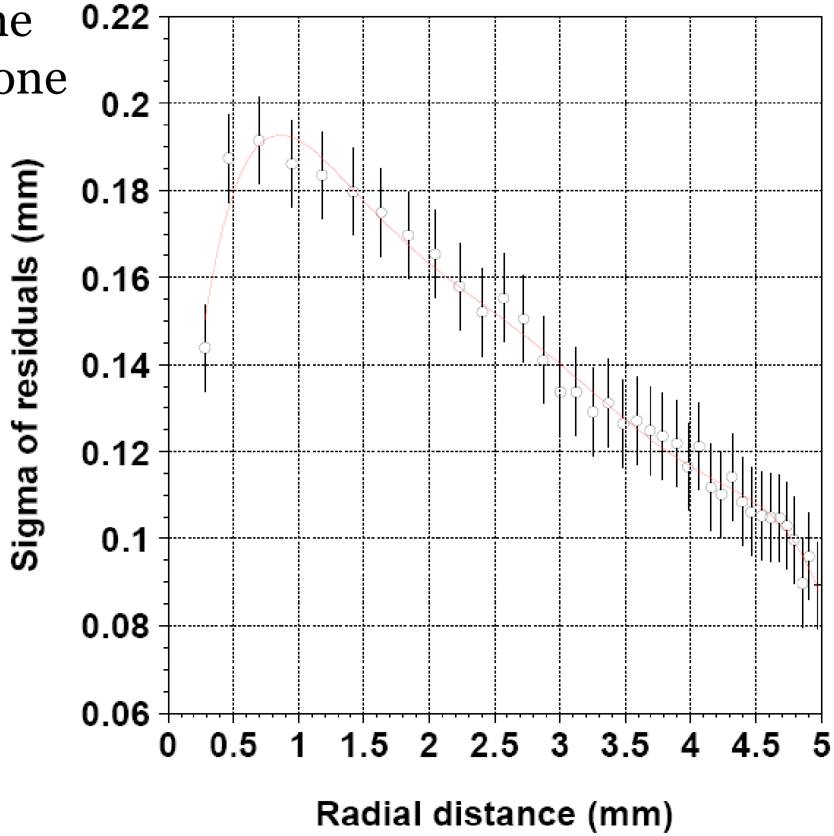
Three different digitization tasks are available:

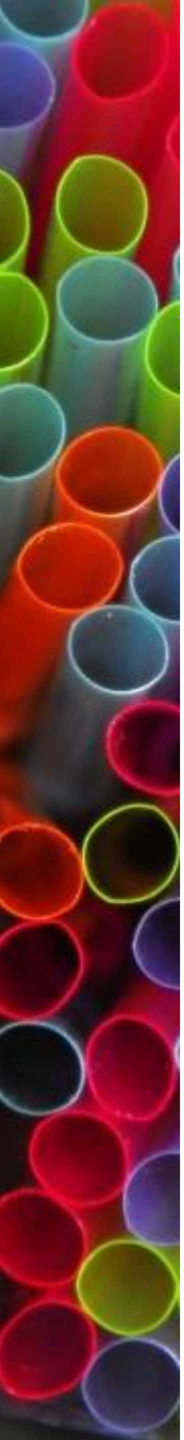
❖ `PndSttHitProducerIdeal`: as the name says, it contains ideal digitization

❖ `PndSttHitProducerRealFull`: it contains the **full simulation** of the detector response. It simulates the physical effect of the formation of the electron/ion couples due to ionization, the drift of the electrons towards the anode wire and the avalanche formation.
The signal on the wire is simulated, the drift time is computed and the isochrone is reconstructed

❖ `PndSttHitProducerRealFast`:
• the **isochrone** radius is sampled from a gaussian with σ from the resolution curve and μ=true radius

• the **energy loss** is modelled on the one coming from the experimental evidence





Mapper

The mapper

In the code several information about the geometry are necessary:
- *the digitization needs: entrance and exit local coordinates*
- *the reconstruction needs: tube position in space, wire direction, tube length*

The geometrical information is written to the parameters file

Two objects have been built to retrieve them:
- The `PndSttTube`, which contains all the geometrical information
- The `PndSttMapCreator`, which associates each PndSttTube to a tubeID and vice versa and allows to retrieve the tube geometrical information from it.

In each Task, in the Init() function, the MapCreator reads the parameters and fills the TClonesArray of PndSttTube.

PndGeoSttPar

```
PndSttMapCreator *mapper =
              new PndSttMapCreator(fSttParameters);
fTubeArray= mapper->FillTubeArray();
```

Now you can get the right tube via the tubeID:

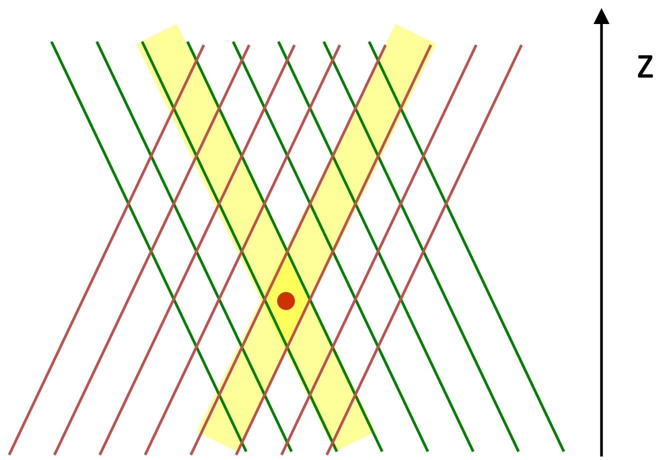
```
PndSttHit *hit = (PndSttHit*) fHitArray->At(i);
PndSttTube *tube =
       (PndSttTube*) fTubeArray->At(hit->GetTubeID());
```

Reconstruction

Reconstruction

❖ The inputs to the reconstruction are:
- ❖ **which** tube fired: its position and orientation in space
- ❖ the **drift radius**

❖ The z coordinate reconstruction is accomplished thanks to the skew tubes



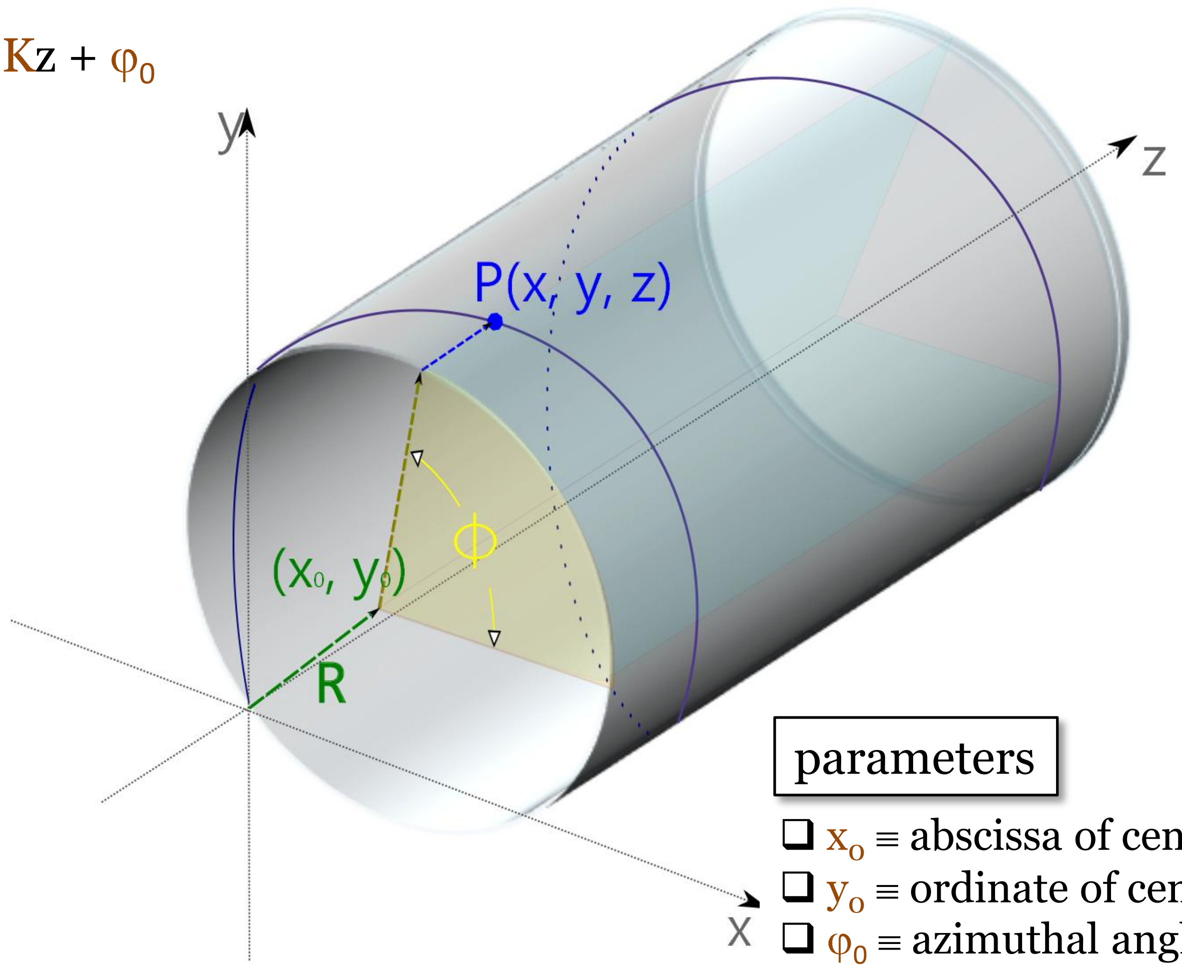
❖ The track finding of the STT alone hits is contained in the track finder class `PndSttTrackFinderReal`, called by the task `PndSttFindTracks`

```
PndSttTrackFinderReal* trkFin = new PndSttTrackFinderReal(0);

PndSttFindTracks* findTracks = new PndSttFindTracks("Track
Finder", "FairTask", trkFin, iVerbose);
findTracks->AddHitCollectionName("STTHit", "STTPoint");
fRun->AddTask(findTracks);
```

Helix parametrization

- $x - x_0 = R\cos(Kz + \varphi_0)$
- $y - y_0 = R\sin(Kz + \varphi_0)$
- $\varphi = Kz + \varphi_0$



parameters

- $x_0 \equiv$ abscissa of center of cylinder
- $y_0 \equiv$ ordinate of center of cylinder
- $\varphi_0 \equiv$ azimuthal angle at z = 0
- $R \equiv$ radius of cylinder
- $K \equiv$ rate of increase of φ

Reconstruction

Track finding in STT alone

The reconstruction is an iterative procedure

• It starts from the outer tubes and finds cluster of firing tubes in the xy plane with a vicinity criterion

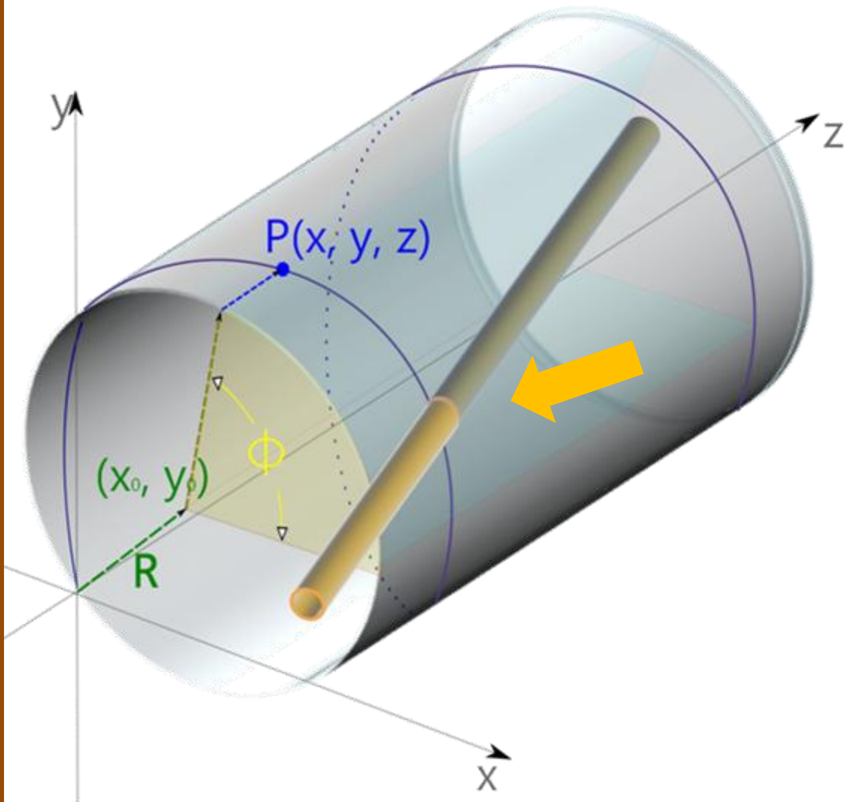
• It uses a coformal transformation to transform the circular tracks in the xy plane to straight lines in the conformal frame

• Once a cluster is identified, it is fitted with a straight line in the conformal plane and new, near tubes are associated to it

Contraint: the tracks are supposed to be primary tracks, i.e. pass in (0, 0, 0)

• After the track is fully described in the xy plane, the z coordinate is reconstructed

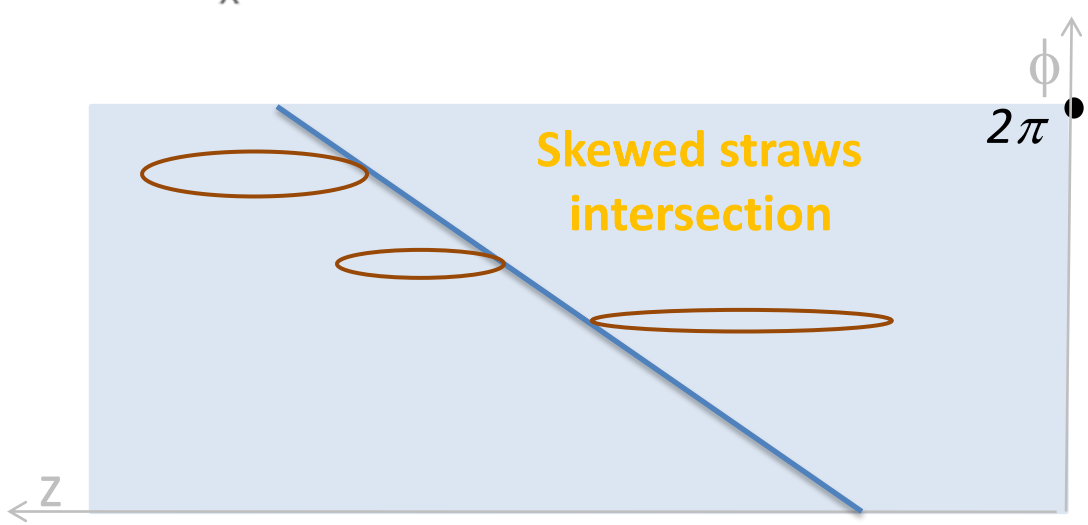
z reconstruction



- a skewed straw tube intersects the cylinder on which the trajectory lies where the arrow indicates

- Due to the geometry and small inclination (3°) of the skewed straws the tangency of the straight line essentially occurs at the edges of the ellipses

- the extremities are fitted with a straight line



Particle identification

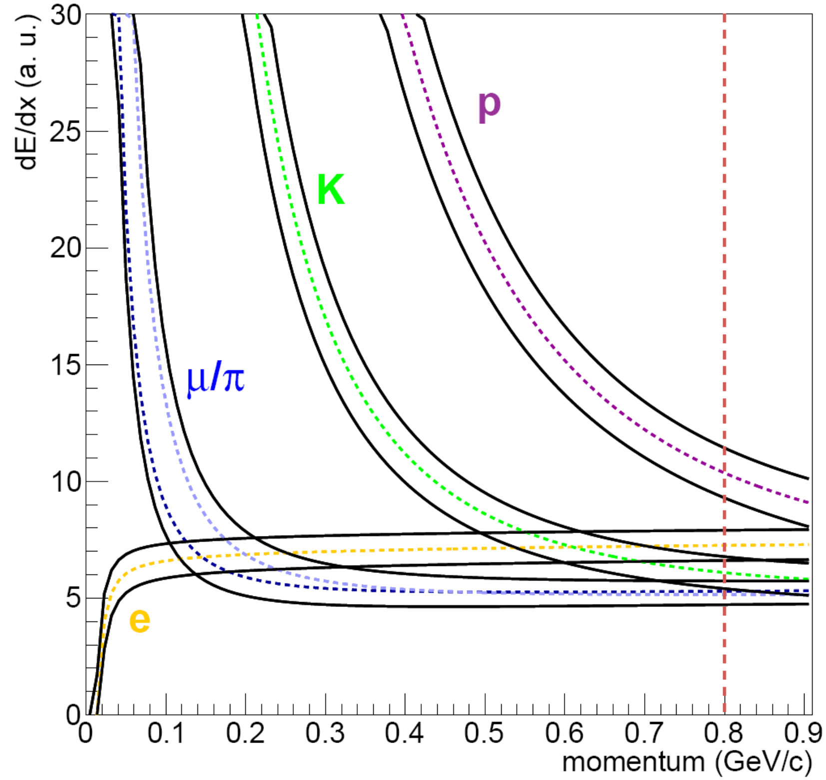
Particle identification

○ dE/dx vs momentum @p < 0.8 GeV/c can help to separate particle types
○ the truncated mean method is used to obtain dE/dx values.

• Particles of different types (e, μ, K, p) are simulated at various momenta in the range [0.05, 0.8] GeV/c

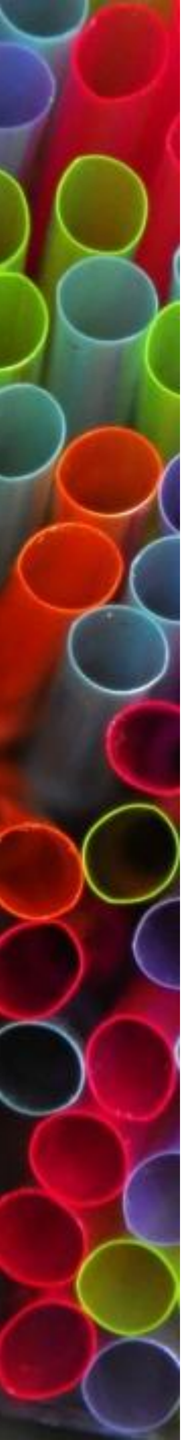
• For each track the reconstructed momentum and the mean dE/dx is calculated with the truncated mean method at 70%

• The simulation of the single tube has been tuned on the real data taken @ FZJ, where a resolution around 8% was shown



CUT1: if < 5 hits remain after the truncation, the dE/dx is set to 0 and the pid fails.
CUT2: p > 0.8 GeV/c → no pid possible.

```
PndPidSttAssociatorTask *aStt= new PndPidSttAssociatorTask();
fRun->AddTask(aStt);
```



Open Issues

Workpackages

STT Workpackages

- Implementation of electronics following the real project of Cracow

- Implementation of a new geometry:
  - Up-to-date
  - With passive
  - Easy to change (CAD → ROOT geometry description)
  - Fast at runtime

- New parametrization with double gaussians for dE/dx to take into account the long tail in the truncated mean distribution

- Implement the STT part of code for the time based simulation