Tracking in the IDEA chamber

Tassielli G.F. - *INFN* Lecce & Mathematics and Physics Dept., *University of Salento*

11th FCC-ee workshop, January 2019, CERN

Outline

Track reconstruction issues:

- Track fitting
  - general consideration and Kalman Filter
  - specific implementation aspects
- Track finding (Pattern Recognition)
  - general aspects
  - useful options for IDEA case
  - some details on current IDEA PR
- Summary

Track fitting – general consideration

Kalman Filter is a standard method for Track fitting in HEP (*alternatives exist but are still not good as KF for this problem*)

- 1960: R. Kalman, "A New Approach to Linear Filtering and Prediction Problems", **Trans. ASME (J. Basic Engineering), 82 D, 35-45, 1960**

One of the first applications: guiding Apollo 13 to the moon

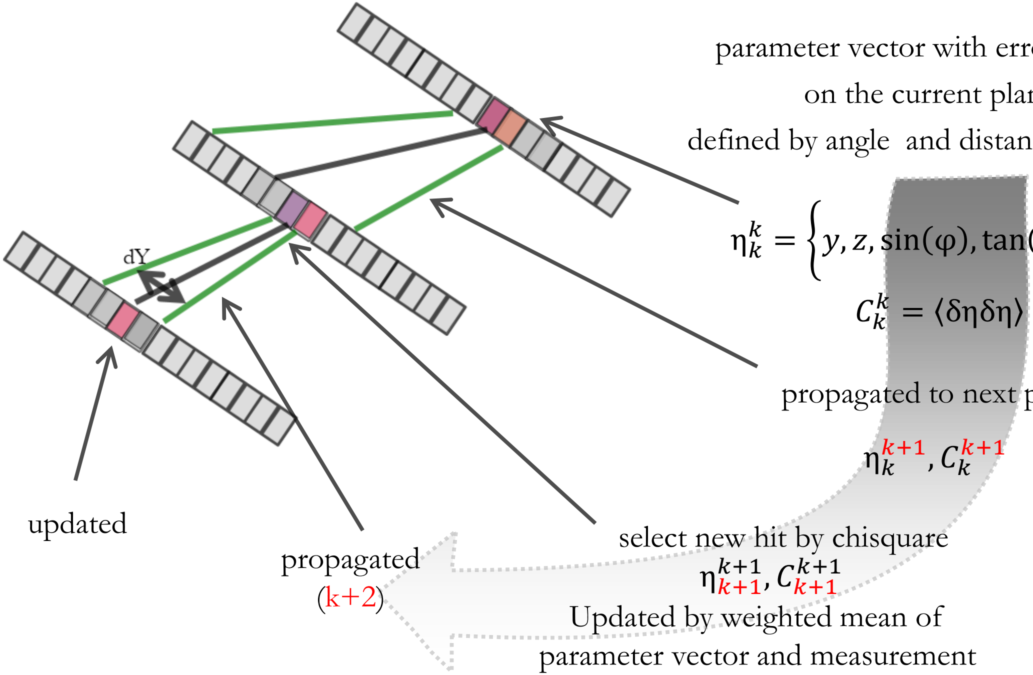
Now widely used: in just about every inertial navigation system(GPS, gyro systems), radar tracking

- First paper in HEP with equivalent equations:
  1984: P. Billoir, "Track Fitting With Multiple Scattering: A New Method," **NIM A (1984) 352**
- Classic author of Kalman Filter for HEP:
- 1987: R. Fruhwirth, "Application of Kalman filtering to track and vertex fitting", **NIM A 262 (1987) 444**

peculiarities:

- **recursive least-squares estimation;**
- **suitable for combined track finding and fitting;**
- **mathematically equivalent to least squares fit;**
- **avoids time-consuming large matrix inversion inherent in least-squares fits;**
- **straightforward to take into account material effects in extrapolation step.**

Basic Principle of Kalman Filter



parameter vector with error matrix
on the current plane
defined by angle and distance to origin

$$\eta_k^k = \left\{y, z, \sin(\varphi), \tan(\lambda), \frac{1}{pt}\right\}$$

$$C_k^k = \langle \delta\eta\delta\eta \rangle$$

propagated to next plane

$$\eta_k^{k+1}, C_k^{k+1}$$

select new hit by chisquare

$$\eta_{k+1}^{k+1}, C_{k+1}^{k+1}$$

Updated by weighted mean of
parameter vector and measurement

updated

propagated
(k+2)

Some matrix formalism underlie, but meaning is simple:
recursive usual χ^2 averaging

Prediction

$$\bar{p}_{k|k-1} = \mathbf{F}_k \bar{p}_{k-1|k-1}$$

$$\mathbf{C}_{k|k-1} = \mathbf{F}_k \mathbf{C}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{P}_k \mathbf{Q}_k \mathbf{P}_k^T$$

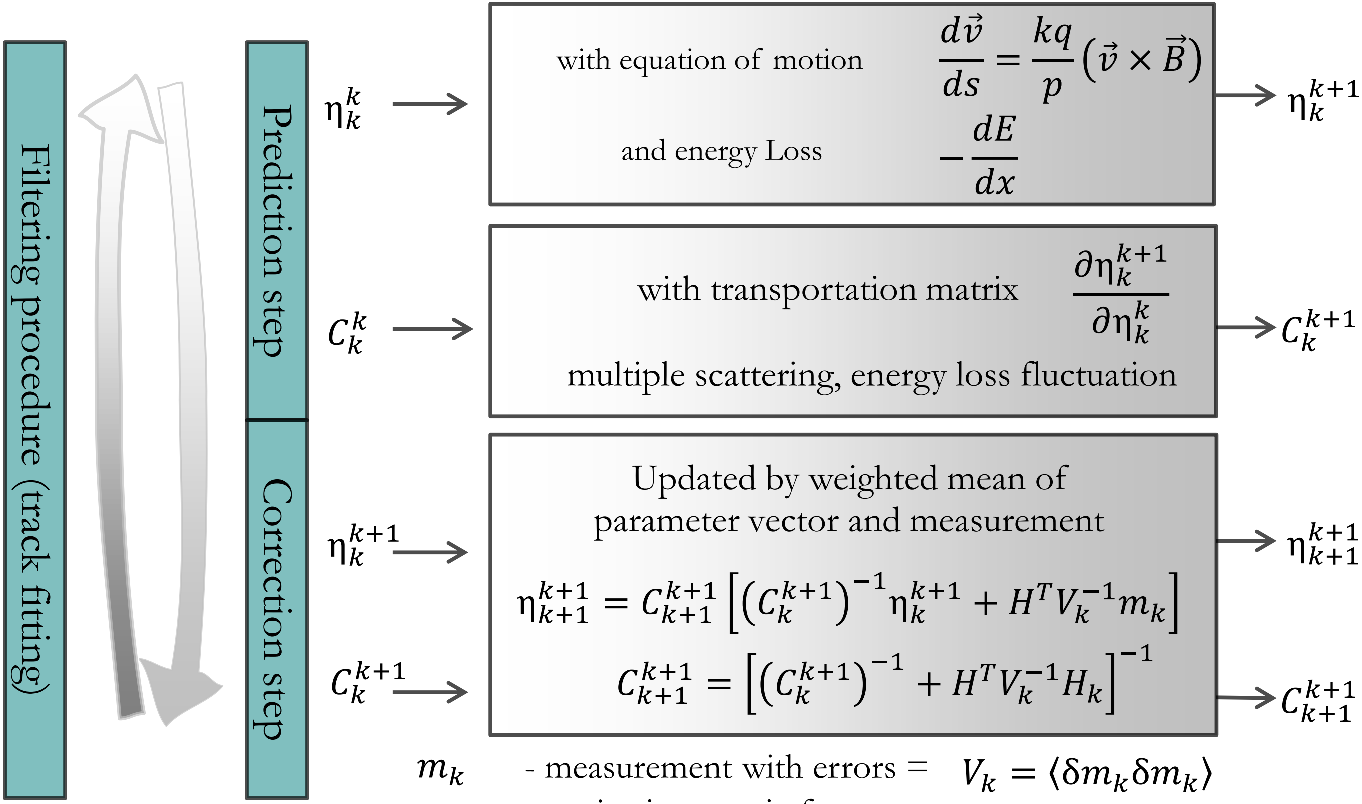
updated step

$$\bar{p}_{k|k} = \bar{p}_{k|k-1} + \mathbf{K}_k (\bar{x}_k - \mathbf{H}_k \bar{p}_{k|k-1})$$

$$\mathbf{K}_k = \mathbf{C}_{k|k-1} \mathbf{H}_k^T (\mathbf{V}_k + \mathbf{H}_k \mathbf{C}_{k|k-1} \mathbf{H}_k^T)^{-1}$$

$$\mathbf{C}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{C}_{k|k-1}$$

Basic Principle of Kalman Filter



Filtering procedure (track fitting)

Prediction step

Correction step

$\eta_k^k \rightarrow$ with equation of motion $\frac{d\vec{v}}{ds} = \frac{kq}{p}(\vec{v} \times \vec{B})$ and energy Loss $-\frac{dE}{dx}$ $\rightarrow \eta_k^{k+1}$

$C_k^k \rightarrow$ with transportation matrix $\frac{\partial \eta_k^{k+1}}{\partial \eta_k^k}$ multiple scattering, energy loss fluctuation $\rightarrow C_k^{k+1}$

$\eta_k^{k+1} \rightarrow$, $C_k^{k+1} \rightarrow$ Updated by weighted mean of parameter vector and measurement $\rightarrow \eta_{k+1}^{k+1}$, $\rightarrow C_{k+1}^{k+1}$

$$\eta_{k+1}^{k+1} = C_{k+1}^{k+1}\left[\left(C_k^{k+1}\right)^{-1}\eta_k^{k+1} + H^T V_k^{-1} m_k\right]$$

$$C_{k+1}^{k+1} = \left[\left(C_k^{k+1}\right)^{-1} + H^T V_k^{-1} H_k\right]^{-1}$$

m_k - measurement with errors = $V_k = \langle \delta m_k \delta m_k \rangle$

H - projection matrix from parameters to measurement

Variations of Kalman Filter

It can be some variations in implementation(most of them just matter of terminology for specific cases) or with extensions

SRKF – Square Root Kalman Filter:

Covariance matrix decompose in square root form
– can give numerical stability

Information Kalman Filter:

rewritten in form of inverse covariance matrix
- useful when some parameters can have infinite sigma

GSF – Gaussian-Sum Filter:

to deal with not gaussian fluctuations - instead of single Gaussian, pdfs modeled by mixture of Gaussians (implemented as a number of Kalman Filters run in parallel)

CKF - The Combinatorial Kalman Filter

Integrate track fitting and pattern recognition
– track splitted in case of few compatible hits

DAF – Deterministic Annealing Filter

On a same surface, several hits may compete for track with different weights
– good for outliers removal

Track fitting – specific implementation aspects

How to use?

Many software packages implement KFs and are available and ‘easy to use’:

- **genFit2**: https://github.com/GenFit/GenFit (arXiv:1410.3698 , NIN A620(2010)518–525) used by:
  - PANDA
  - Belle II
  - …
- **ACTS**: http://acts.web.cern.ch/ACTS/index.php
  - ATLAS
  - *FCC-sw*
- etc…

Track fitting – specific implementation aspects

What do we need to do?

- pass measurement points with their proper description
  - 3D (2D) point (pixel)
  - 1D point (strip)

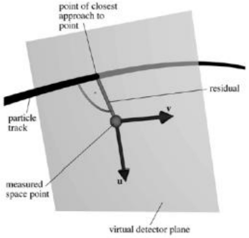


Fig. 2. Virtual detector plane (spanning vectors $\vec{u}$ and $\vec{v}$) for a space-point hit.

  - Drift distance

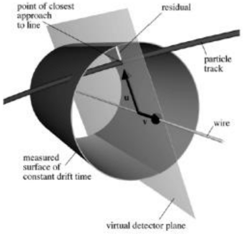


Fig. 3. Virtual detector plane (spanning vectors $\vec{u}$ and $\vec{v}$) for a wire-based drift detector.

- delivery a description of the material to allov the MS and ΔE evaluation
  - genFit2: GDML description
  - *ACTS: DD4Hep*

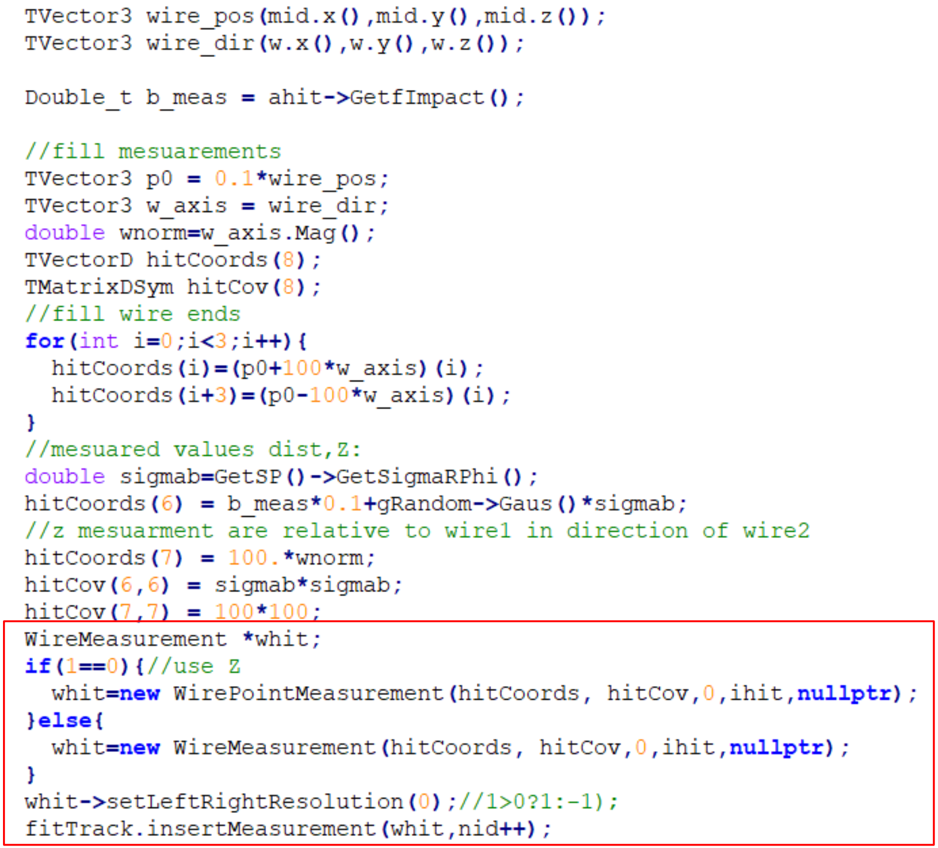
Track fitting – specific implementation aspects

genFit2 example, PIXEL:

```
TMatrixDSym hitpxlCov(2);
hitpxlCov.UnitMatrix();
hitpxlCov *= pxlResolution*pxlResolutionSndSd;

const CLHEP::Hep3Vector& lclmid   = fGeometry->GetPSHWROChanHandle()->GetROChanCenterLcl();

TVectorD hitCoords(2);
hitCoords[0] = 0.1*lclmid.x();
hitCoords[1] = 0.1*lclmid.z();
genfit::PlanarMeasurement* measurement = new genfit::PlanarMeasurement(hitCoords, hitpxlCov, 2, ihit, nullptr);

const CLHEP::Hep3Vector& ldmid    = fGeometry->GetPSHWROChanHandle()->GetROLayerMidPoint();
const CLHEP::Hep3Vector& ldFstSdDir    = fGeometry->GetPSHWROChanHandle()->GetLadderFstSdDir();
const CLHEP::Hep3Vector& ldSndSdDir    = fGeometry->GetPSHWROChanHandle()->GetLadderSndSdDir();
TVector3 ldMid(0.1*ldmid.x(),0.1*ldmid.y(),0.1*ldmid.z());
TVector3 udir(ldFstSdDir.x(),ldFstSdDir.y(),ldFstSdDir.z());
TVector3 vdir(ldSndSdDir.x(),ldSndSdDir.y(),ldSndSdDir.z());

measurement->setPlane(genfit::SharedPlanePtr(new genfit::DetPlane(ldMid, udir, vdir)), 0);
fitTrack.insertMeasurement(measurement,nid++);
```

Track fitting – specific implementation aspects

genFit2 example,
Drift distance:



```
TVector3 wire_pos(mid.x(),mid.y(),mid.z());
TVector3 wire_dir(w.x(),w.y(),w.z());

Double_t b_meas = ahit->GetfImpact();

//fill mesuarements
TVector3 p0 = 0.1*wire_pos;
TVector3 w_axis = wire_dir;
double wnorm=w_axis.Mag();
TVectorD hitCoords(8);
TMatrixDSym hitCov(8);
//fill wire ends
for(int i=0;i<3;i++){
  hitCoords(i)=(p0+100*w_axis)(i);
  hitCoords(i+3)=(p0-100*w_axis)(i);
}
//mesuared values dist,Z:
double sigmab=GetSP()->GetSigmaRPhi();
hitCoords(6) = b_meas*0.1+gRandom->Gaus()*sigmab;
//z mesuarment are relative to wire1 in direction of wire2
hitCoords(7) = 100.*wnorm;
hitCov(6,6) = sigmab*sigmab;
hitCov(7,7) = 100*100;
WireMeasurement *whit;
if(1==0){//use Z
  whit=new WirePointMeasurement(hitCoords, hitCov,0,ihit,nullptr);
}else{
  whit=new WireMeasurement(hitCoords, hitCov,0,ihit,nullptr);
}
whit->setLeftRightResolution(0);//1>0?1:-1);
fitTrack.insertMeasurement(whit,nid++);
```

Track finding – general aspects and useful options for IDEA case

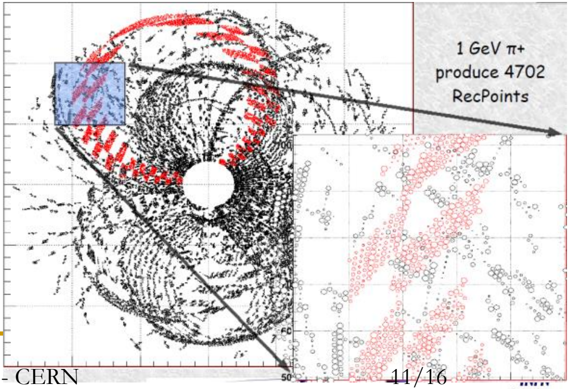
Track finding possible strategies: global vs local methods

- global methods
  - treat hits in all detector layers simultaneously
  - ‘find all’ tracks simultaneously
  - result independent of starting point or hits order

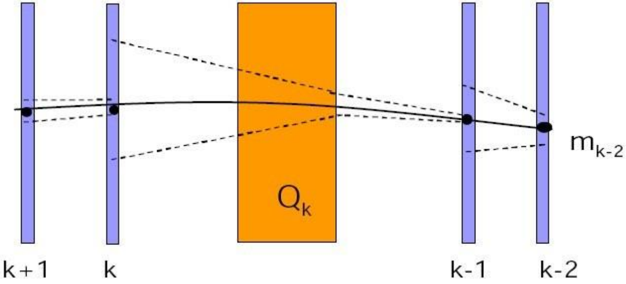
  examples: template matching, Hough transforms (conformal mapping), neural nets, cellular automation, ….
- local methods (‘track following’)
  - start with construction of track seeds
  - add hits by following each seed through detector layers
  - eventually improve seed after each hits

Stereo Drift Chamber peculiarities for PR:

- Left/Right single cell ambiguity
- Longitudinal position along the wire (in the transverse plane appear two separate circonpherences for the same track before applying a correction for the position along the wire)



Track finding - current IDEA PR (Local Method)



Follow track candidate iteratively through detection layers
<u>start</u> from an initial track segment (“seed”)

requires dedicated algorithm

<u>extrapolate:</u> estimate the expected track position in the next detection layer
<u>search:</u> look for hits within a window around the estimated track position
<u>update:</u> if a hit is found inside this search window, add it to the track candidate and update the track parameters
<u>iterate:</u> extrapolate the updated track candidate to the next detection layer

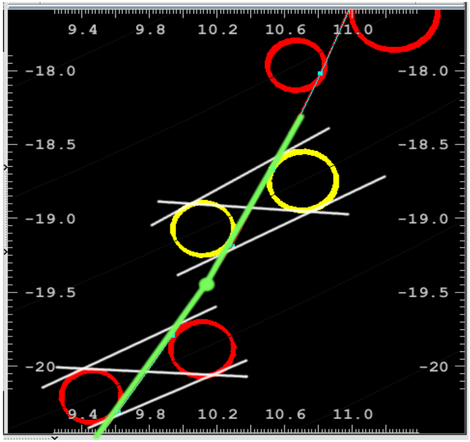
should be broad seeding: track reconstruction efficiency can depend on it, compromise between efficiency and CPU performance
allow for detector inefficiencies: if no hit is found in one layer, continue with the next layer; abandon the candidate if no hits are found in several consecutive layers
allow for combinatories: if more than one hit is found inside the search window, create a separate “branch” for each candidate; follow all branches concurrently

Track finding - current IDEA PR (Local Method) (DCH only)

Seeding from 2 pairs of hits (each pair on same layer) pointing at the origin

- 2 consecutive hits in same layer
  → 4=2x2(Left-Right) pairs with direction
- 2 pairs from nearest layers compatible:
  $|\Delta\cos(\varphi(\text{direction})-\varphi(\text{position}))|<0.2$,
  crossing Z inside DCH
- 1 pair with origin → Pt estimate
  (averaged over 2 pairs)
- Cross Point of 2 opposite stereo pairs give Z-coordinate (with Δφ correction from Pt)
- Pz = 0 at beginning

Z measurement give additional compatibility check between 2 hits and between 2 pairs



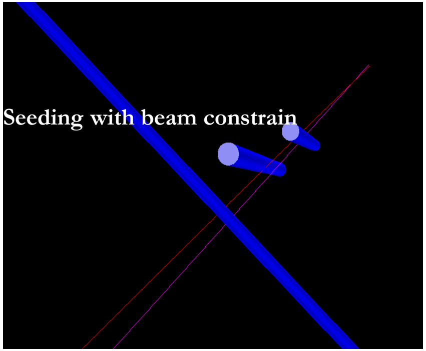
Red hits **projection at z=0 plane**
Yellow **rotated according to φ**

Combinatory low: 2 local compatibilities + 1 from opposite stereo view, but with direction angle check

Track finding - current IDEA PR (Local Method) (DCH only)

Seeding from 3 hits in different layers with origin constraint

- Take any 2 free hits from different stereo layers with a gap (4 or 6 layers)
- Cross Point of 2 wires give Z-coordinate (must be inside DCH volume)
- Select nearest free hits at middle (+-1) layer
- 2 hits from same stereo layer give initial angle in Rphi
- origin added with sigma Rphi~ 1mm Z ~ 1mm
- Seeds constructed for all 2x2x2=8 combination of Left-Right possibilities
- Checked that at -4 (+-1) layer are available free hits with $\chi^2 < 16$
- Extrapolate and assign any compatible hits (by χ^2) from last to first hits
- Refit segment to reduce beam constraint
- Check quality of track segment:
  - ❑ $\chi^2/NDF < 4$
  - ❑ number of hits found (>=7)
  - ❑ number of shared hits (<0.4Nfound)



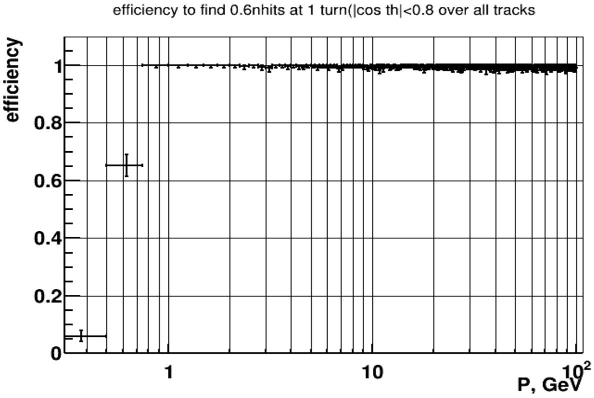
Combinatory high:
local compatibility over
different layers,
+ 1 from different stereo view

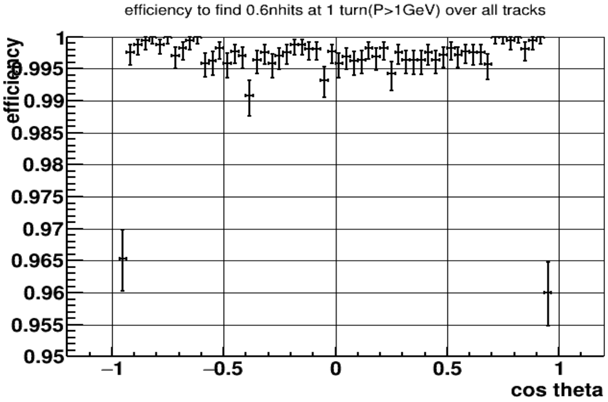
Track finding - current IDEA PR (Local Method) performance

10 μ's (0-100 GeV), DCH only (no longitudinal info used) with Z vtx preselection of seeds

eff ~ 99.5%
particle separation $\Delta\varphi_0$ ~ 0.005 rad

to be tested for secondary particles with vertex out of the SVX





Summary

- KF using genFit2 is implemented for the IDEA detector;
- KF in FCC-sw is available and could be interfaced to IDEA detector;
- Current PR for the IDEA detectors is developed using a local method approach;
- It reached a good performance but need to be tested with jets and with expected background and improvements are possible;
- It is available as an external software package;
- Different PR approach based on Global method (ex. Hough transform based) could be investigated.

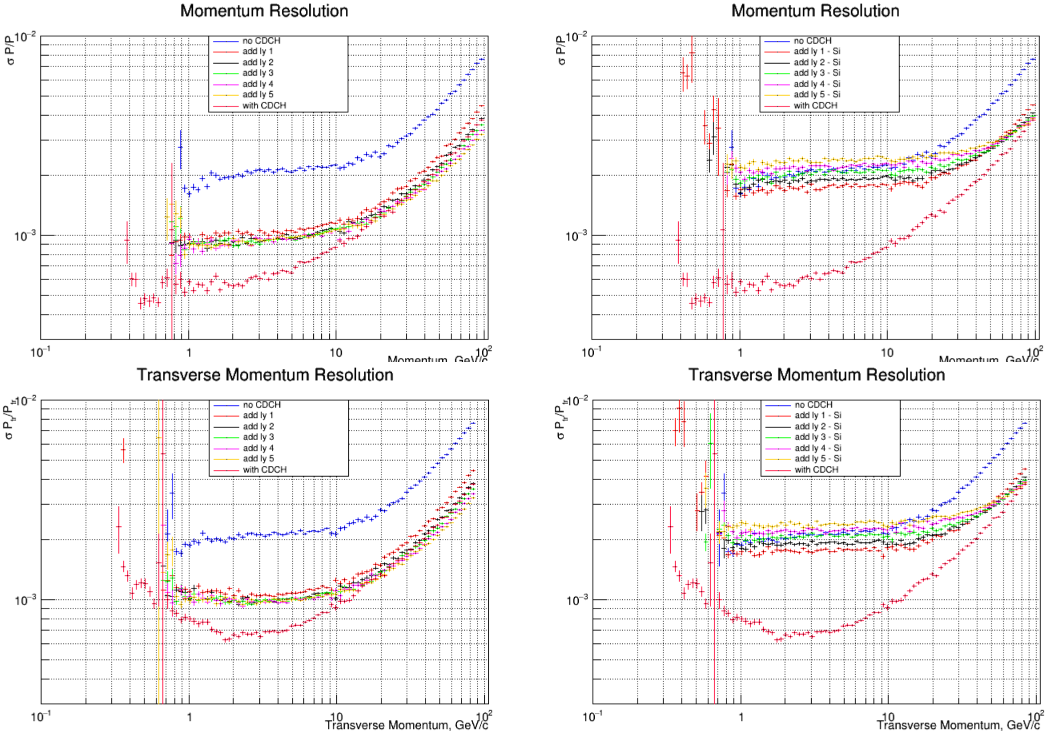
Thanks for your attention

Backup

IDEA – layout v1 – Expected tracking performance

Additional layer made of AIR:

Additional layer made of 450um Si:

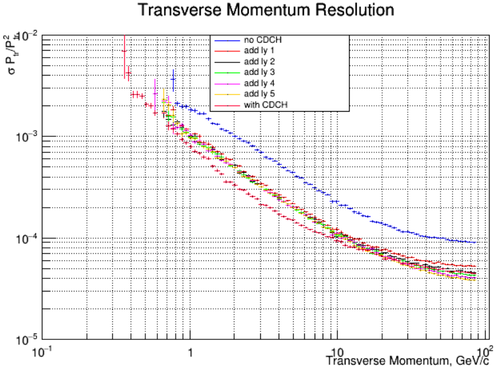


p

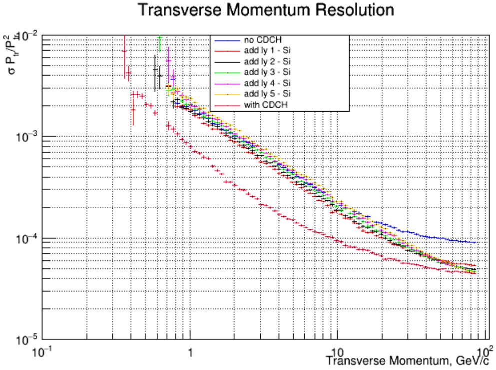
p_t

IDEA – layout v1 – Expected tracking performance

Additional layer made of AIR:

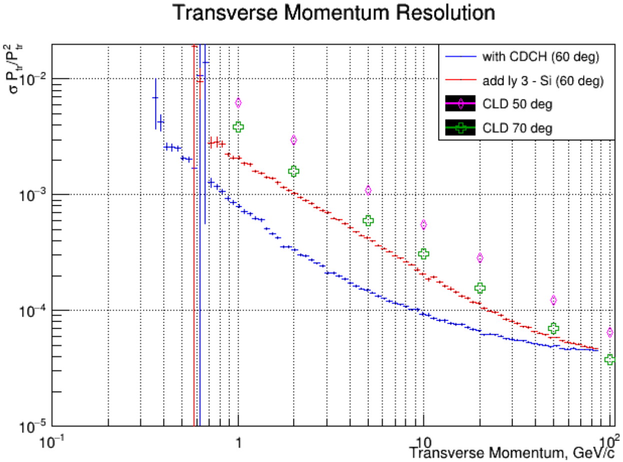


Additional layer made of 450um Si:



p_t

Comparison with CLD:

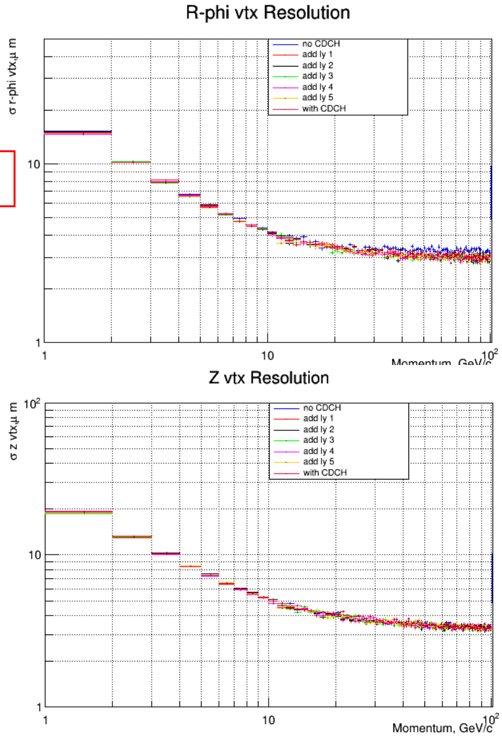


IDEA – layout v1 – Expected tracking performance

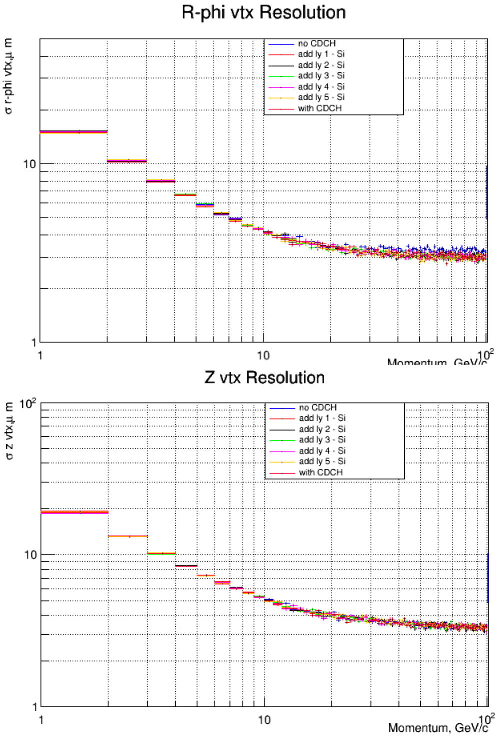
impact parameter

Z

Additional layer made of AIR:



Additional layer made of 450um Si:

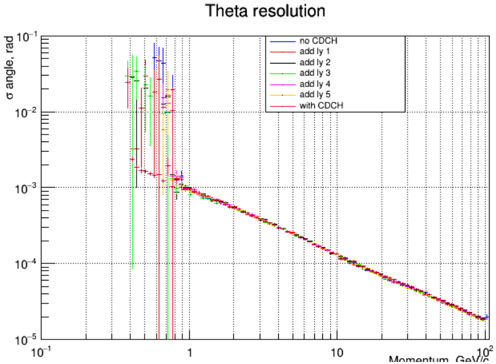


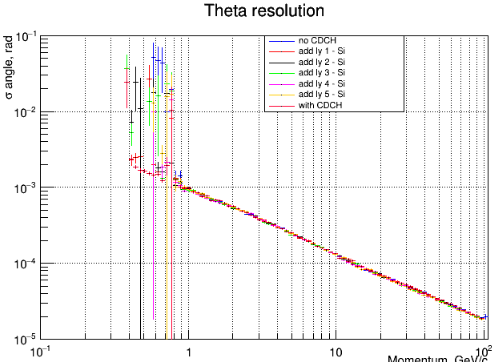
IDEA – layout v1 – Expected tracking performance

Additional layer made of AIR:

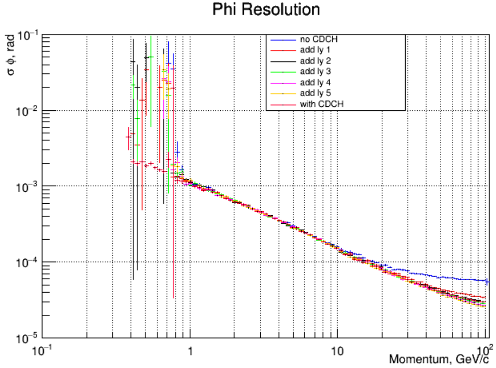
Additional layer made of 450um Si:

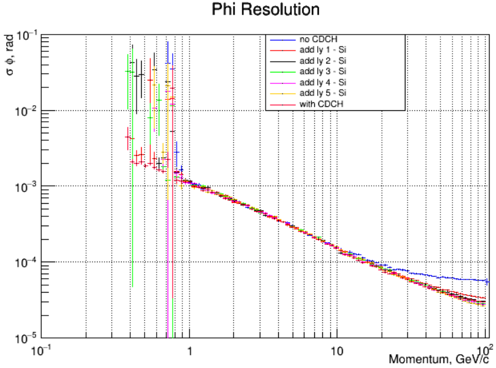
theta



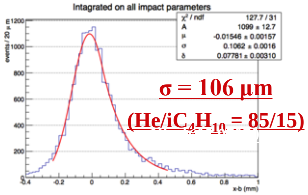


phi



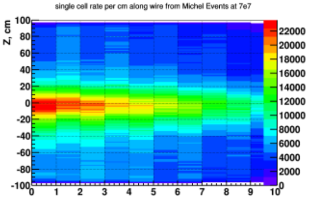


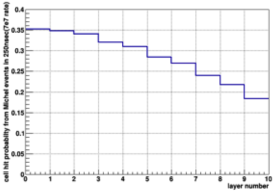
The MEG2 Drift Chamber Performance

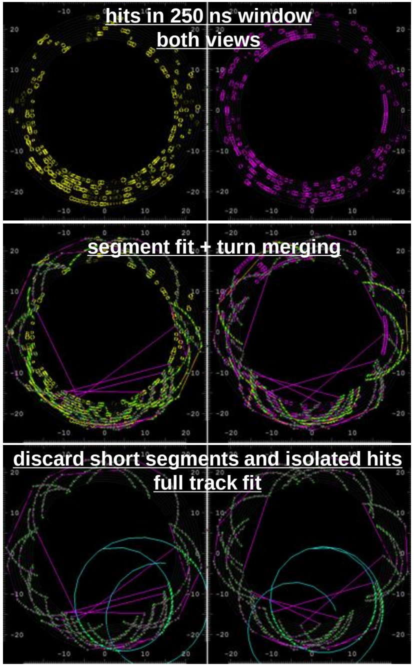


σ = 106 μm

(He/iC_4H_{10} = 85/15)





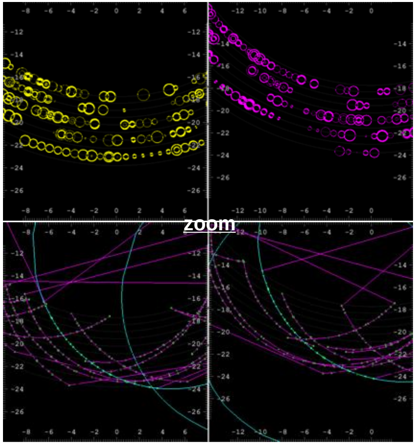


hits in 250 ns window
both views

segment fit + turn merging

discard short segments and isolated hits
full track fit

3D track finding and fit



zoom

signal track

michel tracks

Starting point: Hit simulation approaches

Common steps:

- reject the ionization acts releasing less that 10 eV in the gas or that has a G4Step length less than 5µm;
- assign the G4Step to the corresponding drift cell (resolve some geometric equations);
- evaluate the Distance of Closest Approach for each track crossing a cell;
- convert the DCA in time, smear it with the resolution (converted in time) and sum it to the signal propagation time along the wire and to the particle flight time;
- all the obtained hit times for a cell are time ordered;

Simple model for the hit creation:

- Constant drift velocity (ex. 2 cm/µs), B field effects neglected;
- spatial resolution, gaussian and constant with respect to the impact parameter, (~ 100µm);
- the hits with a time difference shorter than the maximum drift time (for the cell) are grouped together;
- (eventually evaluate the expected Number of Clusters);
- Create the hit needed by the reconstruction;

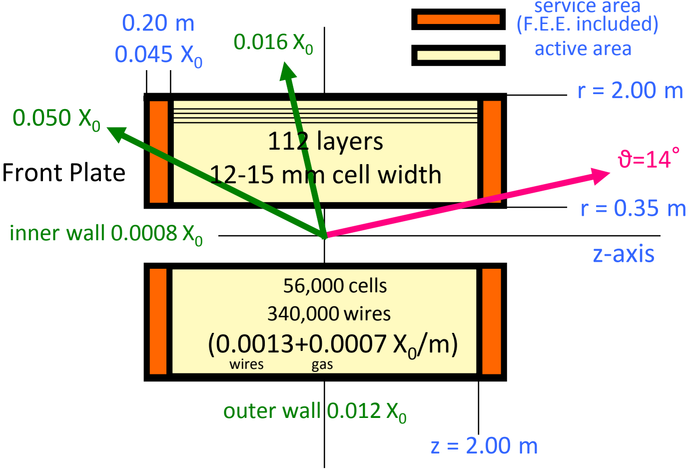
Detailed model (not yet implemented):

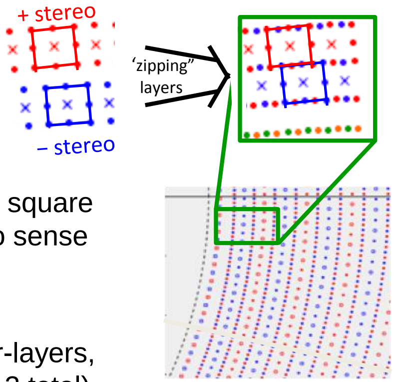
- *Use a realistic XT relation;*
- *Simulate in detail the Cluster generation and the Signal Waveforms;*
- *Analyze the waveform and extract the reconstructed Impact parameter and dN/dX and dE/dx.*

DCH Geometrical parameters

tracking efficiency ε ≈ 1
for ϑ > 14° (260 mrad)
97% solid angle

$0.016\ X_0$ to barrel calorimeter
$0.050\ X_0$ to end-cap calorimeter





- 12÷15 mm wide square cells 5 : 1 field to sense wires ratio
- 56,448 cells
- 14 co-axial super-layers, 8 layers each (112 total) in 24 equal azimuthal (15°) sectors ($N_i = 192 + (i - 1) \times 48$)
- alternating sign stereo angles ranging from 50 to 250 mrad