Pixel TPC simulation and reconstruction

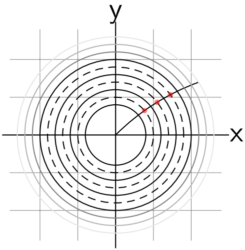
Kees Ligtenberg

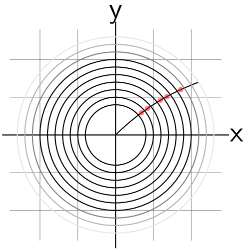
Nikhef lepton collider meeting

27 March 2017

Outline

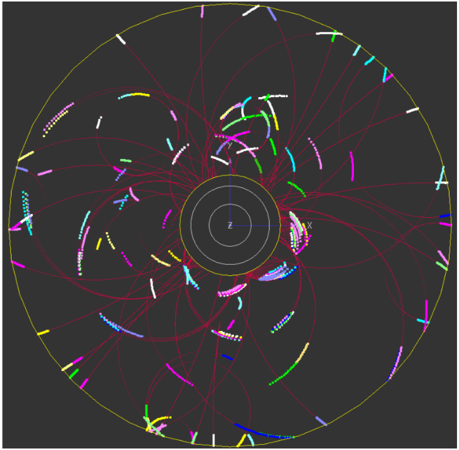
Simulation of pad hits compared to pixel hits





Pad hits	Pixel hits
6 mm $\times$ 1 mm	0.55μm $\times$0.55μm
Exactly one hit per layer	Multiple or no hits per layer
22 electrons per hit	1 electron per hit
Only diffusion in ϕ and z	Diffusion in r, ϕ and z

Track finding for pads using Clupatra

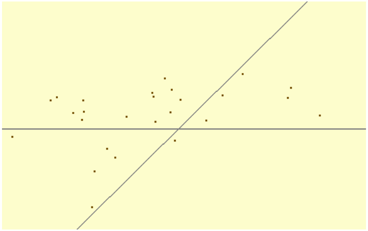


Frank Gaede, DESY, LCWS 2011

1. Repeatedly find small track segments
   - Uses nearest neighbour clustering by distance
2. Fit track to cluster
   - uses first, middle and last hit to initialise track parameters
3. Extend track inwards (and outwards)
   - Uses Kalman filter (Kaltest) in MarlinTrk, see track fitting

Issues when applying pad-track-finding to pixel-hits

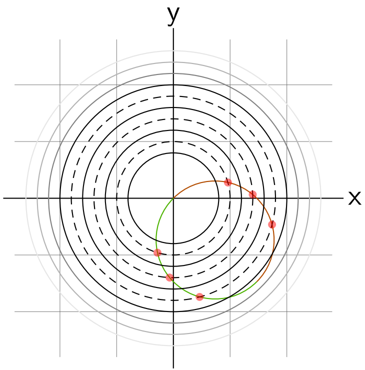
- Computational complexity of nearest neighbour clustering scales as $\mathcal{O}(N^2)$
  Unsuitable for many thousands of pixel hits
- Initialise Kalman filter with first, middle and last hit
  3 hits do not fix the track tight enough, first hits can pull the track fit in the wrong direction



Track finding for pixel TPC

- Perform clustering by ϕ (Hough-transform like)
  - Fill histogram of hits by ϕ
  - Maximum bin is cluster with track candidate
  - construct a straight line from the detector center to the average position
  - Cut hits on distance from this line
  - initialise track fit with this line

Track fitting for pads



Track fit:

- Calculate intersection of helix with layer in coordinates (ϕ, z)
- Add closest hit to fit (if $\chi^2 < \chi^2_{\text{threshold}}$)

For curled (low momentum) tracks, cluster inward and outward parts separately and merge

Fit tracks by Extended Kalman filter

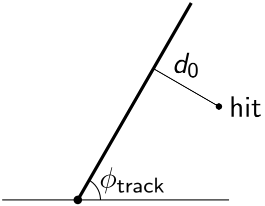
Fit track by an Extended Kalman Filter: a recursive fitting algorithm working in steps:

- Predict using state-propagator $\boldsymbol{a}_k^{k-1} = \boldsymbol{f}_k(\boldsymbol{a}_k)$
- Update with measurement $\boldsymbol{m}_k$ using state-to-measurement projector $\boldsymbol{h}_k(\boldsymbol{a}_k^{k-1})$
- ...

For pad track fitting

- $\boldsymbol{a}_k$ contains track parameters $(d_\rho, \phi_0, \kappa, d_z, \tan\lambda)$
- $\boldsymbol{m}_k$ is defined for pads as coordinates of a cylindrical surface (ϕ, z)
  Pixel hits are also smeared in the radial direction r direction
  Measurement vector has large error for tracks (almost) in z direction
  Difficult to extend towards multiple layers per element for pixels

Track fitting for pixel hits

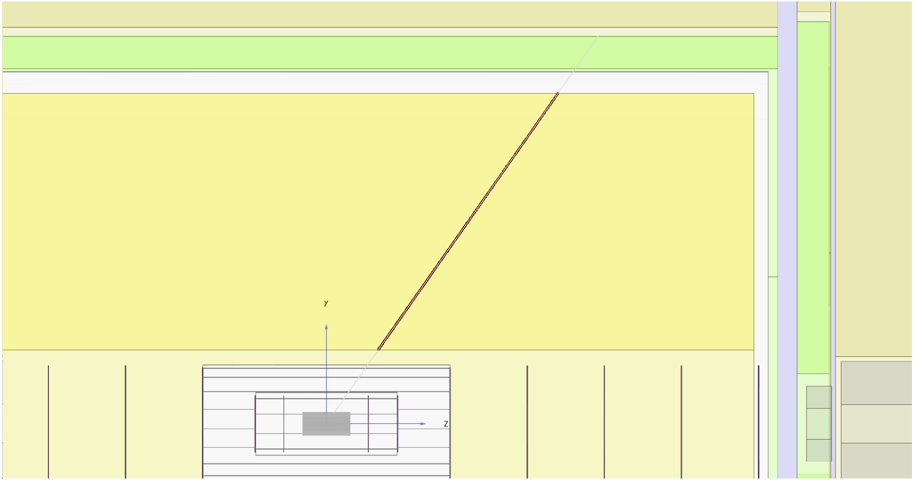


Define alternative measure with $\boldsymbol{m}_k$ as a function of $\boldsymbol{a}_k^{k-1}$

$$\boldsymbol{m}_k(\boldsymbol{a}_k^{k-1}) = \begin{pmatrix} d_0 \\ z \end{pmatrix} = \begin{pmatrix} \Delta x \sin(\phi_{\text{track}}) - \Delta y \cos(\phi_{\text{track}}) \\ z_{\text{hit}} + \tan\lambda(\Delta x \cos(\phi_{\text{track}}) + \Delta y \sin(\phi_{\text{track}}), \end{pmatrix}$$

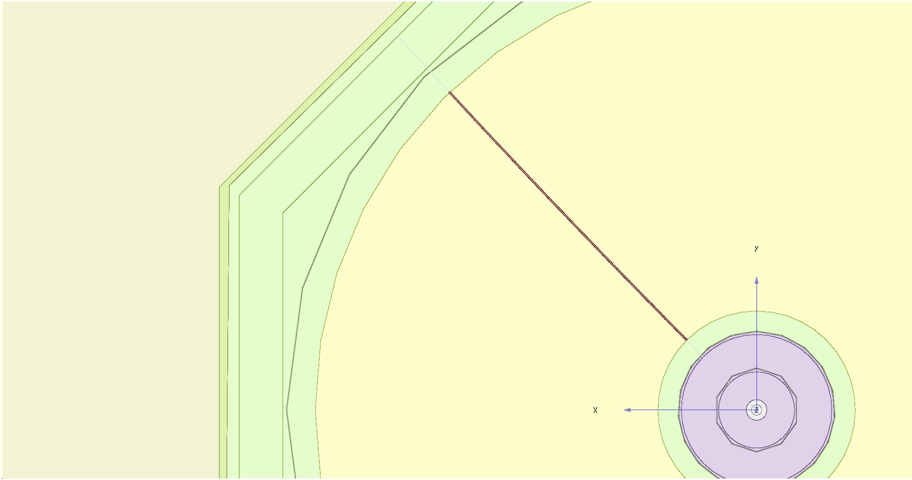
Fit of straight track

50 GeV muon



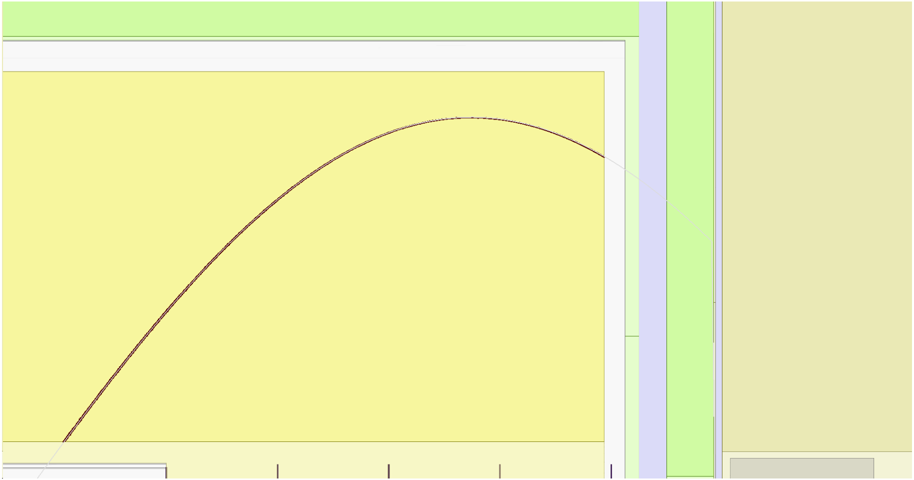
Fit of straight track

50 GeV muon



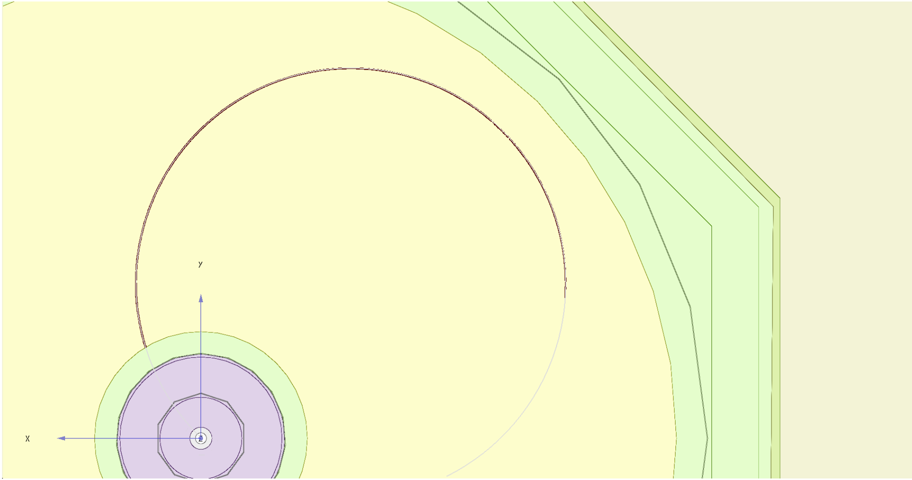
Fit of curled track

1 GeV muon



Fit of curled track

1 GeV muon



Conclusion

- Track finding was adapted for the larger number of pixel hits
- A less restrictive measurement vector was implemented for Kalman fitting
- Todo:
  - Implement energy loss and multiple-scattering

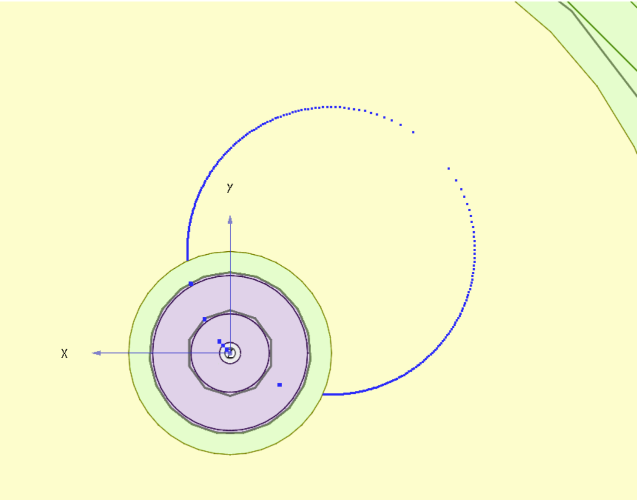
Extended Kalman filter

Recursive fitting algorithm to find state vector $\boldsymbol{a}_k$ and covariance $\boldsymbol{C}_k$ at site k from a series of measurements $\boldsymbol{m}_k$ by procedure:

- Predict
  - $\boldsymbol{a}_k^{k-1} = \boldsymbol{f}_{k-1}(\boldsymbol{a}_{k-1})$, where $\boldsymbol{f}_k(\boldsymbol{a}_k)$ is the state-propagator
  - $\boldsymbol{C}_k^{k-1} = \boldsymbol{F}_{k-1}\boldsymbol{C}_{k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1}$, where $\boldsymbol{F}_{k-1} = \frac{\partial \boldsymbol{f}_{k-1}}{\partial \boldsymbol{a}_{k-1}}$, and $\boldsymbol{Q}_k$ the covariance of the process noise
- Update
  - $\boldsymbol{a}_k = \boldsymbol{a}_k^{k-1} + \boldsymbol{K}_k\left(\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{a}_k^{k-1})\right)$, where $\boldsymbol{h}_k(\boldsymbol{a}_k)$ the projector, $\boldsymbol{K}_k = \boldsymbol{C}_k^{k-1}\boldsymbol{H}_k^T(\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_k^{k-1}\boldsymbol{H}_k^T)^{-1}$, $\boldsymbol{H}_k = \frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{a}_k^{k-1}}$, and $\boldsymbol{V}_k$ the coveriance of the measurement noise
  - $\boldsymbol{C}_k = \left((\boldsymbol{C}_k^{k-1})^{-1} + \boldsymbol{H}_k^T\boldsymbol{G}_k\boldsymbol{H}_k\right)^{-1}$, where $\boldsymbol{G}_k = (\boldsymbol{V}_k)^{-1}$
- (Smooth...)

See: Keisuke Fujii, Extended Kalman Filter, The AFCA-SIM-J Group

Pad simulation of 700 MeV muon



Hits are only at layer crossings
This was solved for pixel hits by adding an interpolator