Tracking and Event Reconstruction

Veronique Ziegler

First CLAS12 Experiment Workshop

03/28/2017

Overview

- **DC Tracking**
  - Calibration constants included in tracking
    - T0-subtraction & Time-to-distance parameters extraction (Krishna Adhikari [UMiss], Mac Mestayer)
  - Geometry ccdb constants verified (the hard way!) & missing geometry methods implemented
    - R3 ministagger implemented in geometry (Andrey Kim [UConn])
  - New Kalman Filter
    - Overview
  - 5 out of 6 superlayer tracking
    - Algorithms
    - Results
  - Code release tags

- **SVT Tracking**
  - CVT tracking package under new clas12 git
  - Monitoring suite (Yuri Gotra)
  - Alignment code development using Millipede (Jerry Gilfoyle (U. Richmond)
  - Hits-on-track finding algorithm

- **Event Builder (Joseph Newton [ODU])**
  - Current implementation status
  - Bank outputs

❑DC Tracking

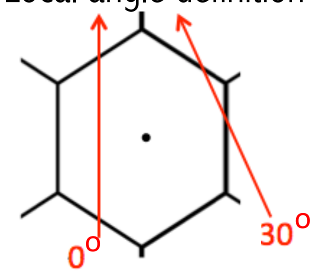
- Calibration constants included in tracking
  - T0-subtraction & Time-to-distance parameters extraction (Krishna Adhikari [UMiss])

Time-to-Distance Implementation in Reconstruction

- Calculation of time vs distance
  - **$\text{time} = x / v_0 + a.\hat{x}^{\,n} + b(\alpha).\,\hat{x}^{\,m}$**
    - alpha = local angle in deg.
    - v_0 = saturated drift velocity (cm/ns)
    - $\hat{x} = x / \text{dmax}$

Function parameters loaded from ccdb for each new run

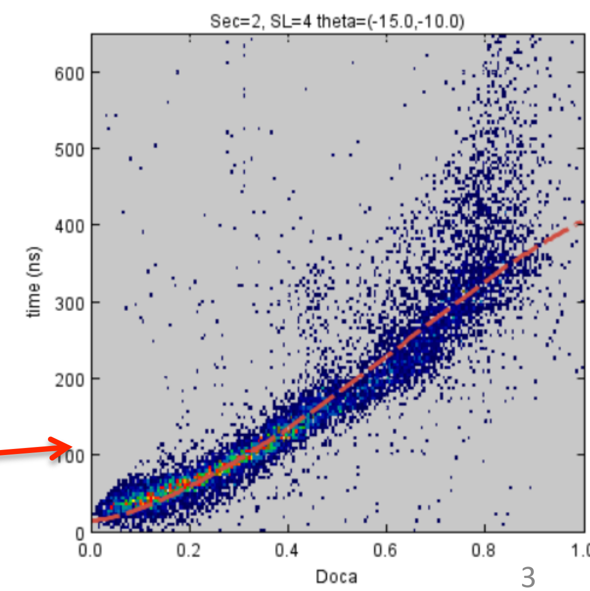
Local angle definition



→ **init:** For each sector fill static array *DISTFROMTIME[s][ibfield][icosalpha][tbin]* *[superlayer Idx, ibfield B bin, icosalpha cos(alpha) bin, tbin time bin]*

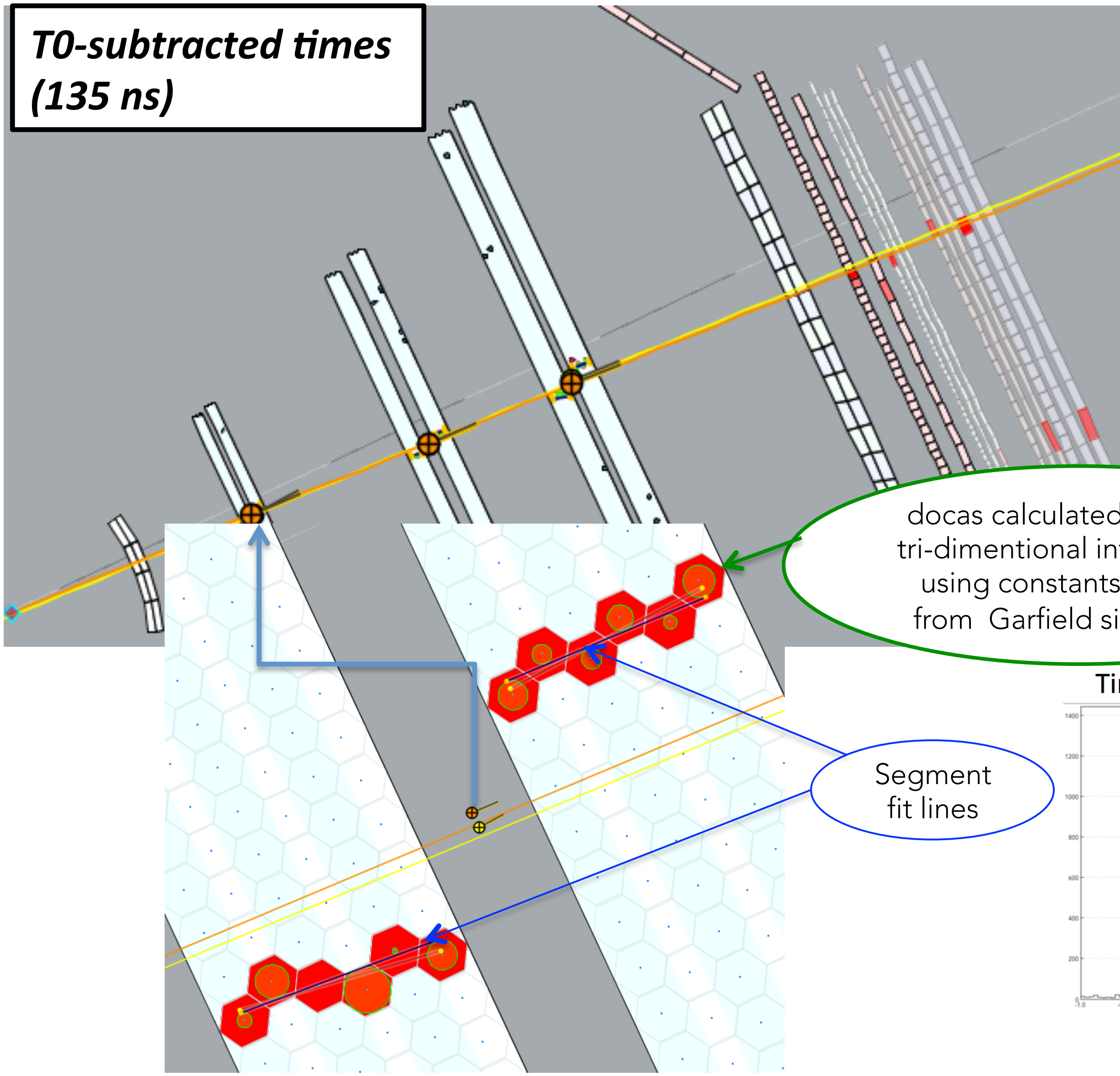
→ *For given time t, for hit in superlayer S, with loc angle α and B-field magnitude B, find corresponding B, t, and α bin intervals and interpolate in 3 dimensions.*

Parameters extracted from fits to doca vs time distributions in data [KPP run 758] (K. Adhikari)



Time-to-Distance Validation prior KPP

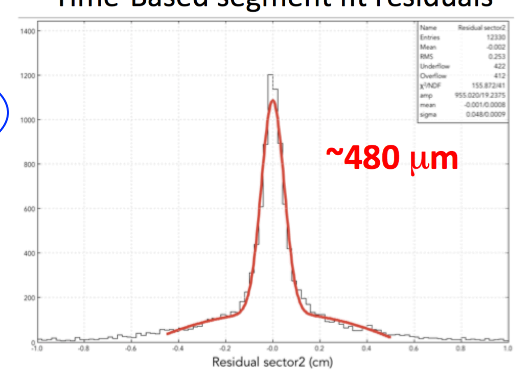
Using sub-sample of Run 686 (Cosmic data sample)



docas calculated from T2D tri-dimentional interpolation using constants (in ccdb) from Garfield simulations

Segment fit lines

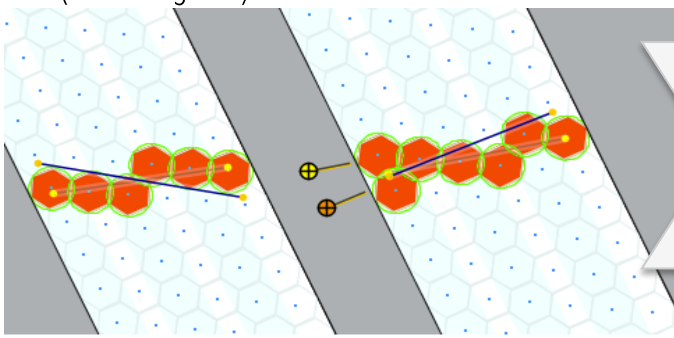
Time-Based segment fit residuals



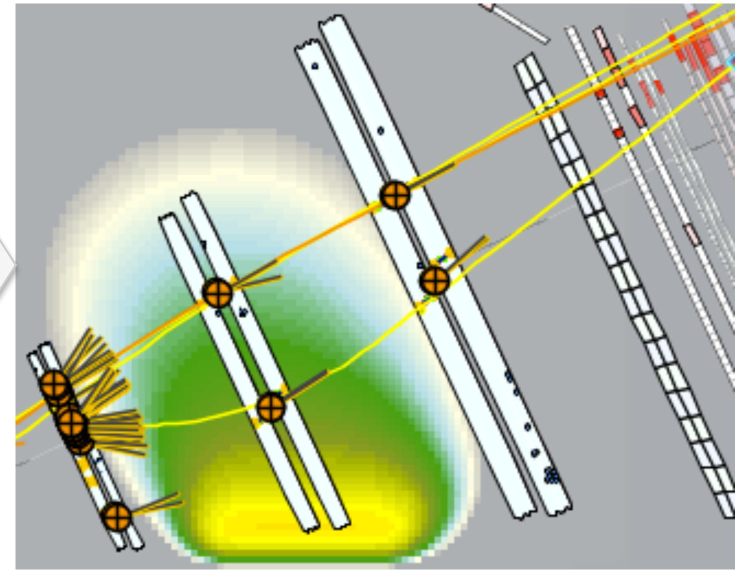
Out-of-time-hits rejection

Out-of-timers signature

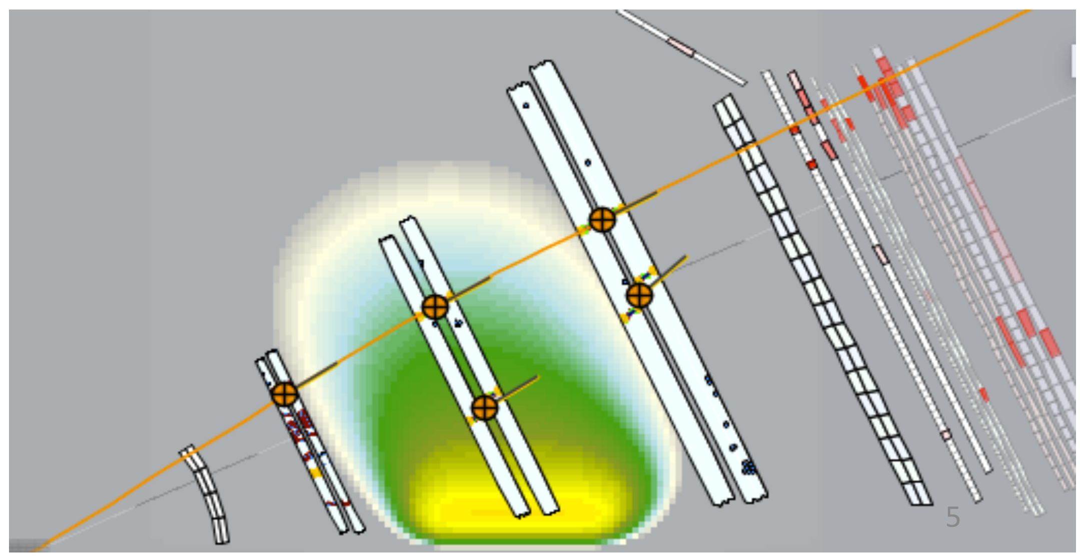
$\Sigma_{i(i:\ \text{hit in segment})}$ doca ~ Σ_i cell-size



produce lots of combinatorials



After O.T. hit rejection: Fewer Time-Based track candidates to fit → speed improvement

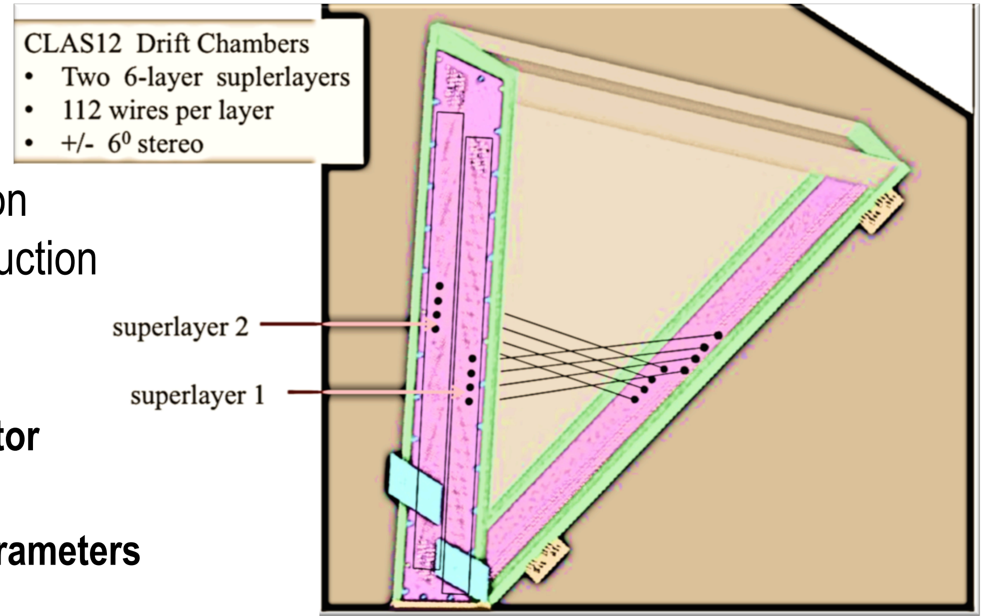


❑DC Tracking

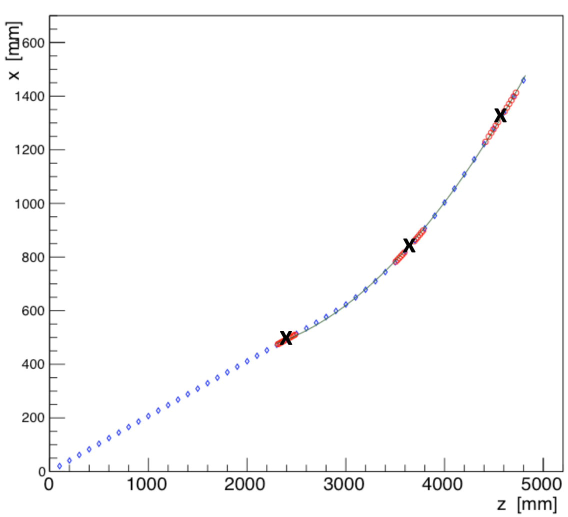
- New Kalman Filter
  - Overview

DC Reconstruction Algorithms (reminder)

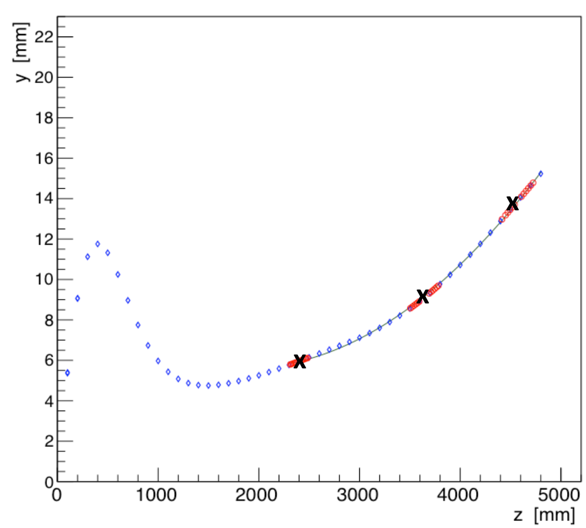
- Obtain a trajectory from hit-based track & subsequently time-based (using timing information to refine the hit position) track segment reconstruction
  - Fits to the wires → extended to a plane → point & direction
- **Gives a "cross" object a position and direction vector**
- **Fit to the crosses to obtain a trajectory → Initial parameters to Kalman Filter**



In x – z plane



In y – z plane

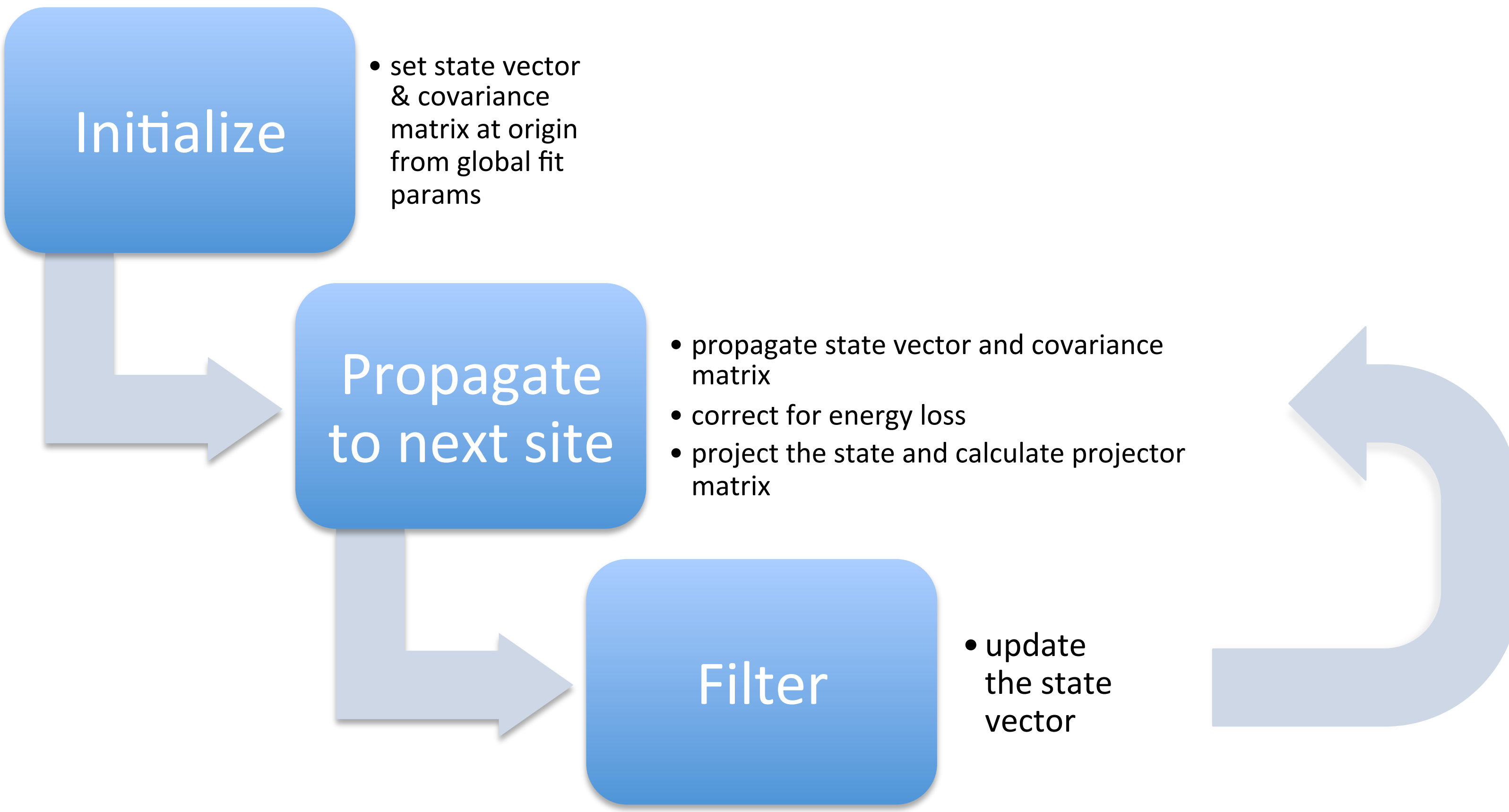


$$\frac{q}{p} = \frac{theta3 - theta1}{0.3\int B\,dl}$$

$theta_i \sim R_i$ segment fit slope ($i = 1,3$)

← Quadratic fit

How the Kalman Filter works in a Nutshell

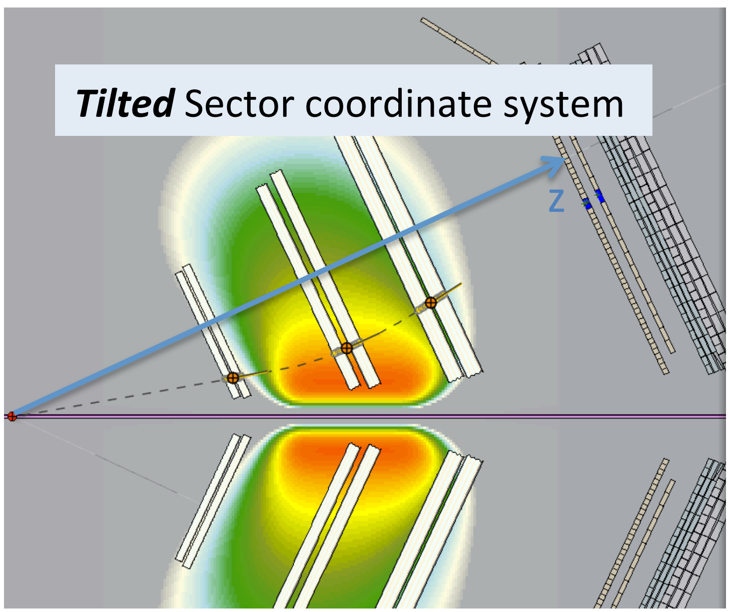


- ◆ Start process by propagating the state vector from first measurement plane using the pattern recognition estimates (swim back from point at first cross to first plane)

StateVecs class

1) State Vector descriptor

p and $\int B\,dl$ estimated from pattern recognition prior to fitting; global fitting method problematic due to inhomogeneity in the field.



Event display of reconstructed tracks in CLAS12 DC

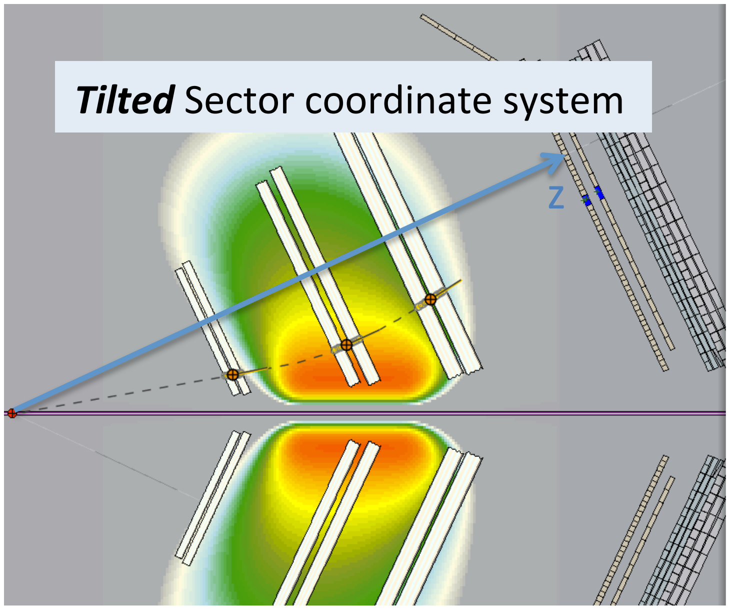
- **site**: DC layer plane where a fired (k = 1…36);
  - in tilted coordinate system, planes are perpendicular to z, so measurement sites are equidistant

- **state**: 5-parameter track representation

$$\tilde{x}(z) = \begin{pmatrix} x \\ y \\ t_x \\ t_y \\ q \end{pmatrix}, \quad \begin{aligned} t_x &= p_x/p_z \\ t_y &= p_y/p_z \\ q &= Q_e/\left|\vec{p}\right| \end{aligned}$$

StateVecs class

2) Propagator methods



- **Propagate from initial state estimated at layer 1 to outermost layer in DC sector**
- **<u>state propagator</u>**
  - Solve equation of motion directly assuming B is constant over a small enough step size δz

$$
\begin{aligned}
x(z) &= x_0 + t_{x0} \cdot s + \tfrac{1}{2} \cdot q_0 \cdot v \cdot A_x \cdot s^2 \quad , \\
y(z) &= y_0 + t_{y0} \cdot s + \tfrac{1}{2} \cdot q_0 \cdot v \cdot A_y \cdot s^2 \\
t_x(z) &= t_{x0} + q_0 \cdot v \cdot A_x \cdot s \quad , \\
t_y(z) &= t_{y0} + q_0 \cdot v \cdot A_y \cdot s \quad .
\end{aligned}
$$

- Adaptive step size
- Placeholder for using Runge-Kutta 4 (or 5) method (work with D. Heddle (CNU) → speed optimization

$$
\begin{aligned}
A_x &= (1 + t_x^2 + t_y^2)^{\frac{1}{2}} \cdot [t_y \cdot (t_x B_x + B_z) - (1 + t_x^2) B_y] \quad , \\
A_y &= (1 + t_x^2 + t_y^2)^{\frac{1}{2}} \cdot \left[-t_x \cdot (t_y B_y + B_z) + (1 + t_y^2) B_x\right]
\end{aligned}
$$

- **<u>state covariance matrix propagator</u>**: Jacobian of state projector

$$
\boldsymbol{F}_{k-1} \equiv \left(\frac{\partial \boldsymbol{a}'}{\partial \boldsymbol{a}}\right) = \begin{pmatrix} \frac{\partial d'_\rho}{\partial \boldsymbol{a}} \\ \frac{\partial \phi'_0}{\partial \boldsymbol{a}} \\ \frac{\partial \kappa'}{\partial \boldsymbol{a}} \\ \frac{\partial d'_z}{\partial \boldsymbol{a}} \\ \frac{\partial \tan \lambda'}{\partial \boldsymbol{a}} \end{pmatrix}
$$

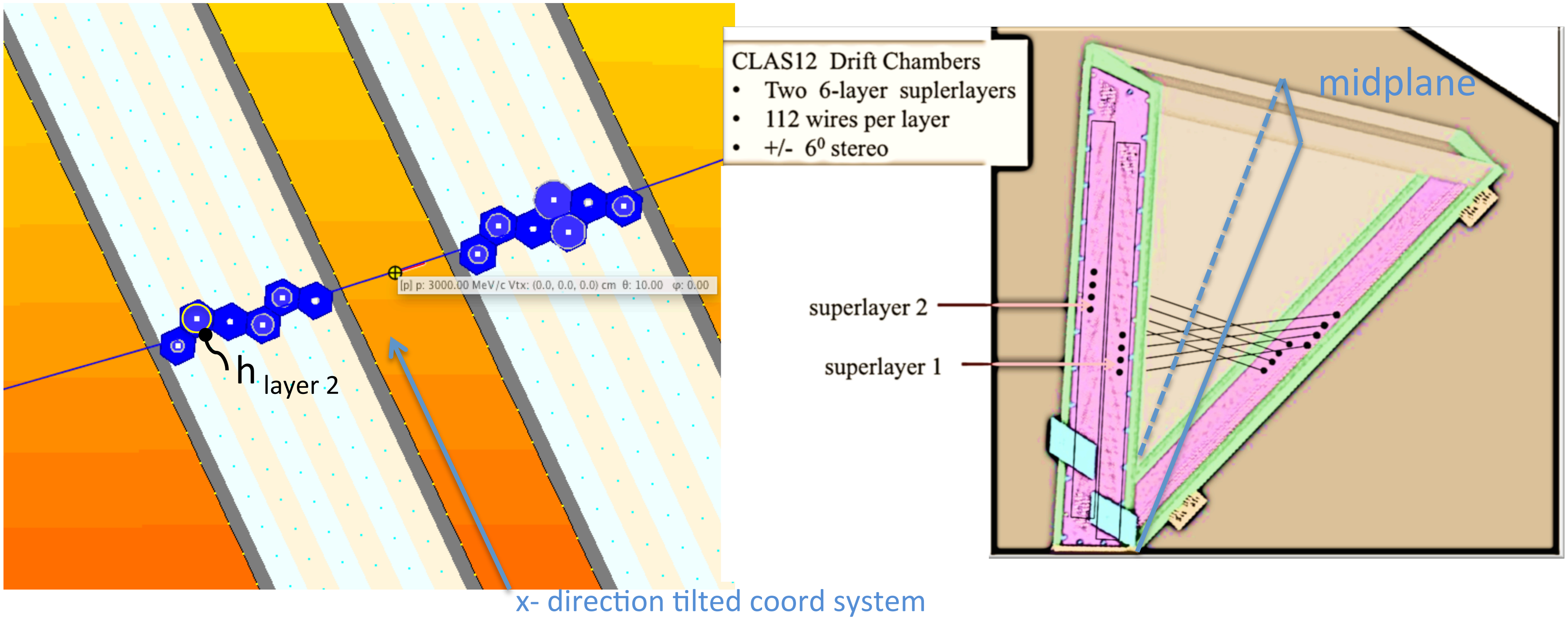
<u>Ref</u>:

Optimized Integration of the Equations of Motion of a Particle in the HERA-B Magnet

Alexander Spiridonov
DESY Zeuthen / ITEP Moscow

MeasVecs class

3) Measurement descriptor and projector



Projection of state vector to the doca position @ midplane:

$$h_k(a_k^{k-1}) = (x)_k - \tan((sprlyr - 1)\cdot stereoAngle)(y)_k$$

$sprlyr = \text{superlayer},\ \ stereoAngle = \pm 6 \text{ degrees.}$

Projector matrix:

$$\Rightarrow H = (1, -\tan((sprlyr - 1)\cdot stereoAngle))$$

maps state onto an observable

- Easy to drop sites (relevant for tracking with missing superlayers (next topic)

KFitter class

4) State Vector and Covariance Matrix Filter

Filtering of the state vector

$$\boldsymbol{a}_k = \boldsymbol{a}_k^{k-1} + \boldsymbol{K}_k\left(\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{a}_k^{k-1})\right), \quad C_k = \left[\left(C_k^{k-1}\right)^{-1} + \boldsymbol{H}_k^T G_k \boldsymbol{H}_k\right]^{-1}$$

Kalman ***Gain*** matrix

$$\boldsymbol{K}_k = \left[\left(C_k^{k-1}\right)^{-1} + \boldsymbol{H}_k^T G_k \boldsymbol{H}_k\right]^{-1} \boldsymbol{H}_k^T G_k.$$

- Contains a method for smoothing (revisiting past sites) → in development

❑DC Tracking

- Five-Out-Of-Six Superlayer Tracking

  Algorithm developed

  - using MC sample with single e- tracks [p from 2 to 6 GeV/c, theta from 10 to 35 deg., all azimuth, 1000 events].
  - turn off one superlayer at a time, at hit-reading stage.

Sequence of Steps for Tracking with missing Superlayers

Note: a track segment in a superlayer requires a minimum of 4 layers out of 6. So with the FOOSST algorithm, a total of 18 layers can be missing in a track candidate.

Determine where missing segment would be using existing segments

- odd superlayers
- even superlayers

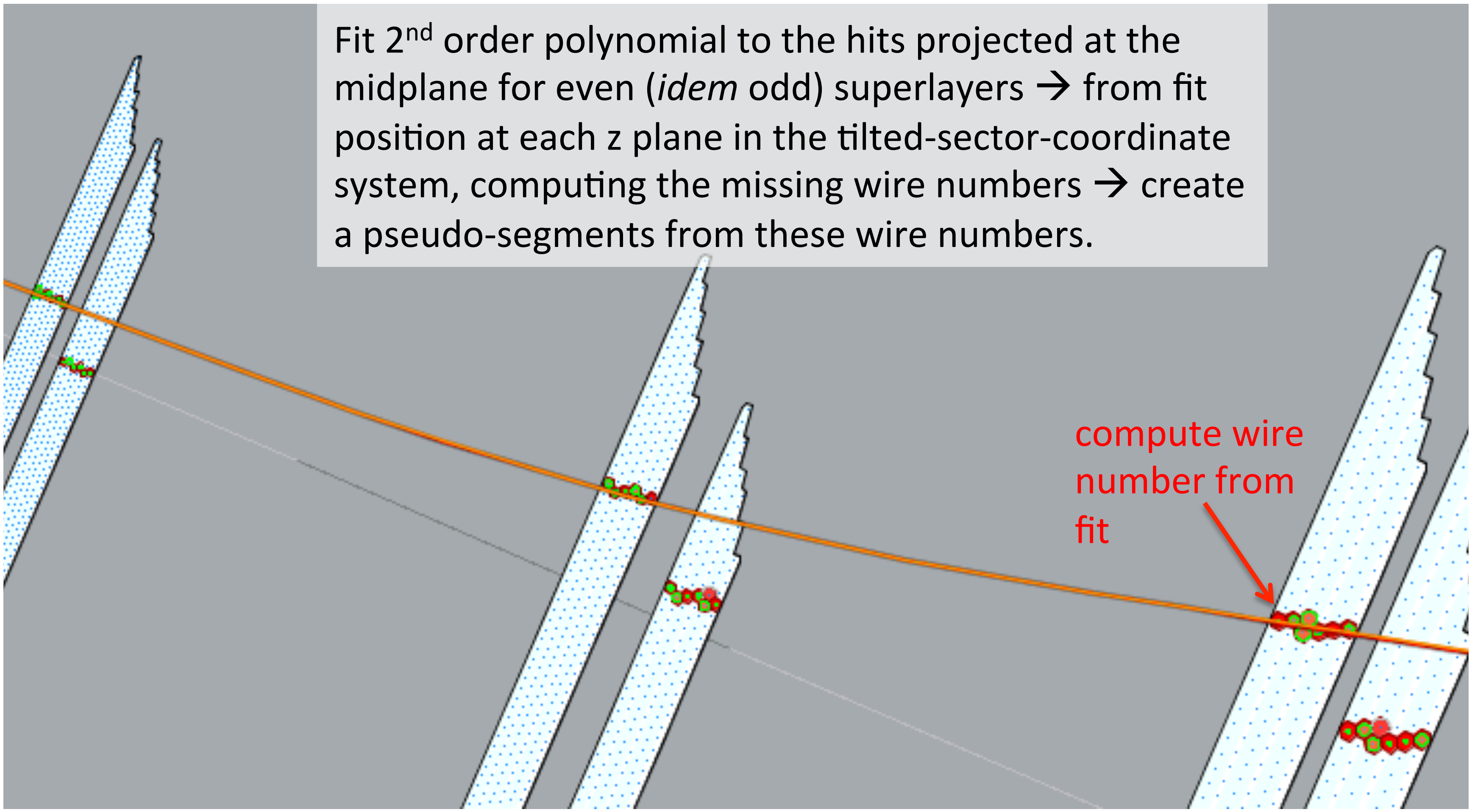
Fit missing segment to get a pseudo-cross

- use pseudo-cross as part of pattern recognition as before

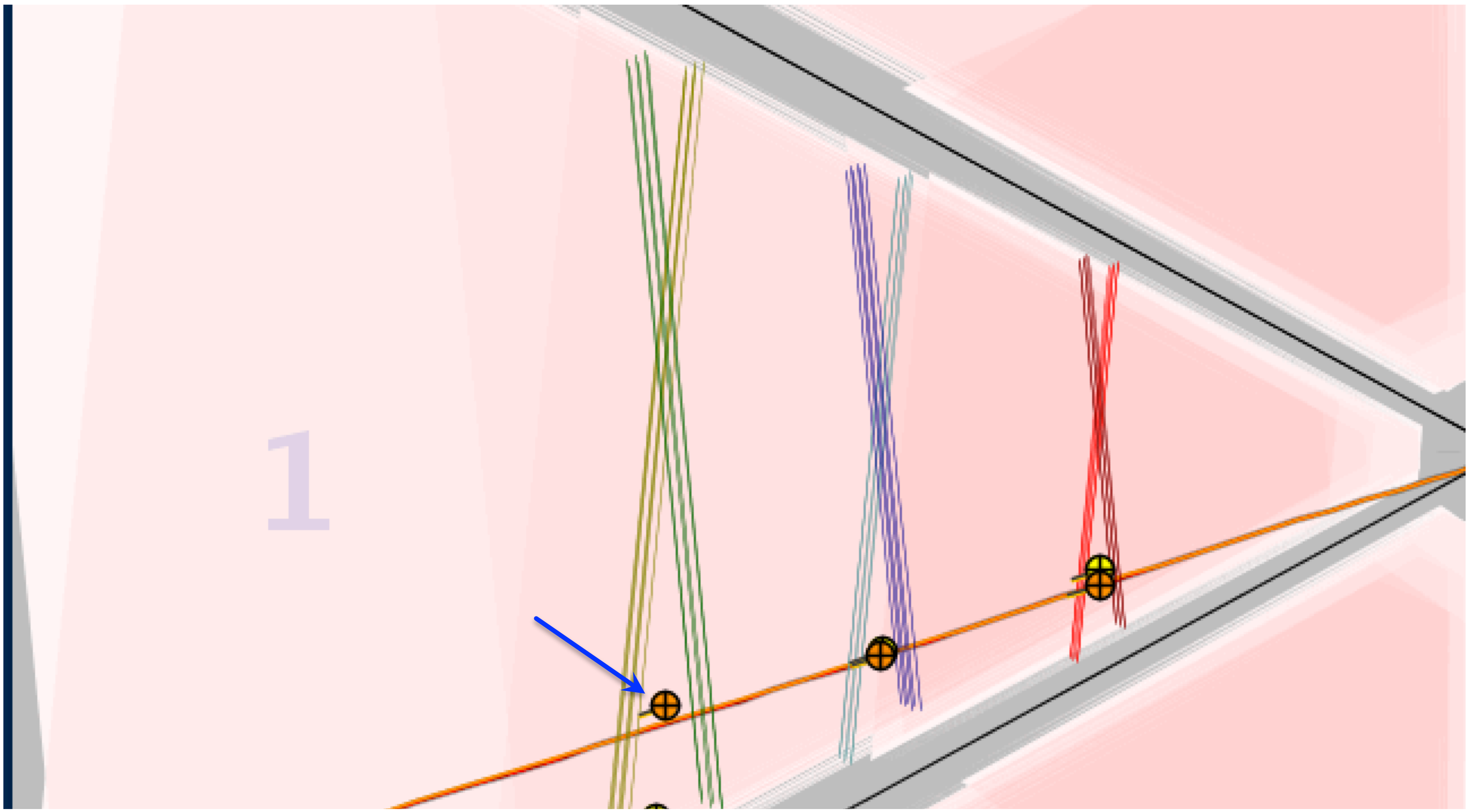
Once initial parameters are obtained from pattern recognition run the Kalman Filter to fit the track

- use only non-pseudo-hits

1) Estimate of missing segment



2) Compute the Pseudo-Cross

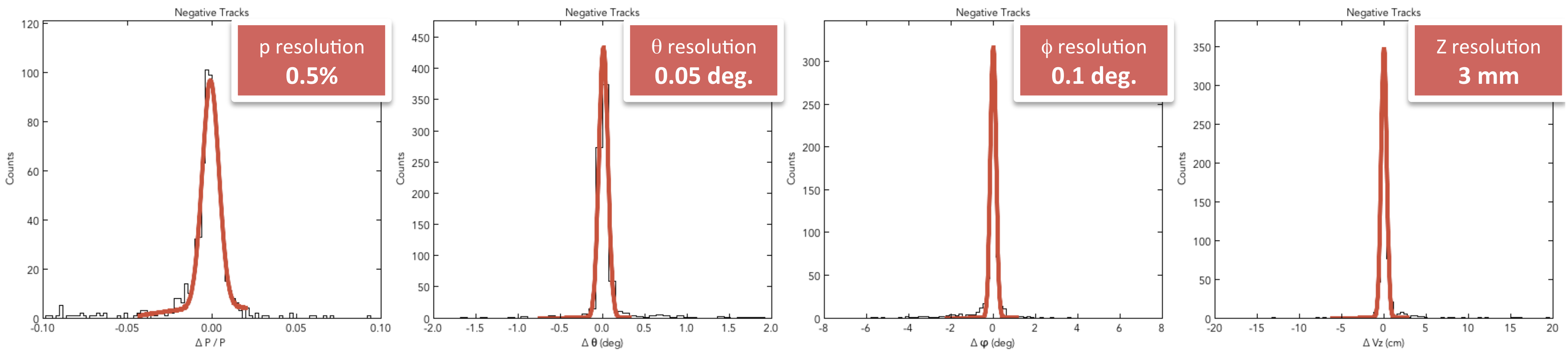


Preliminary MC Studies

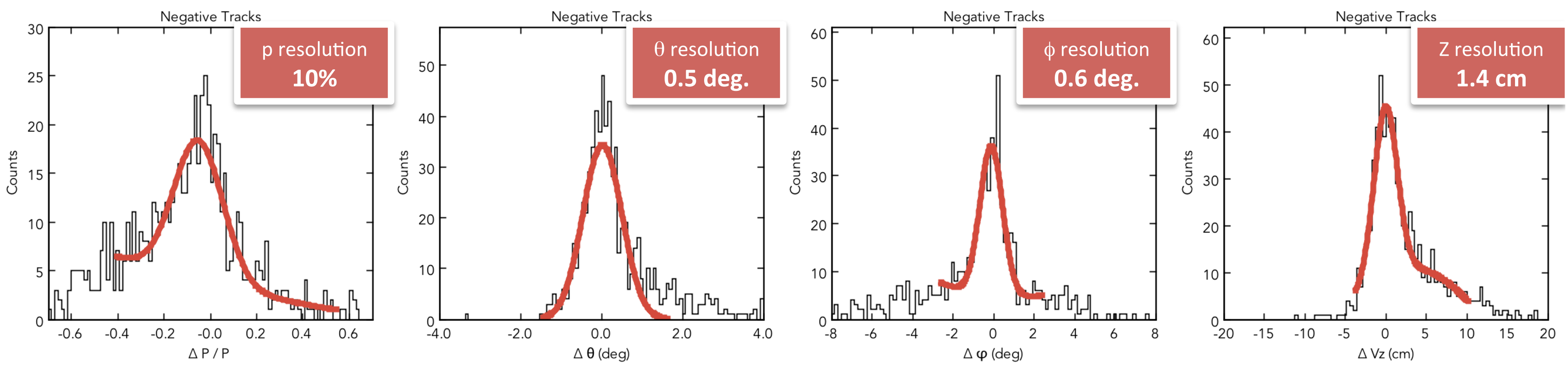
- Loosing a superlayer has a minimal effect on tracking resolutions
- Inefficiencies due to missing SL: 5% for SL1, 10% for SL2, less than 3% for all other SLs

Can we use this for 2-regions tracking ?

- only 1 superlayer missing



- 2-regions tracking (superlayers 5 & 6 missing): ~79% efficient



- Needs testing using low momentum tracks

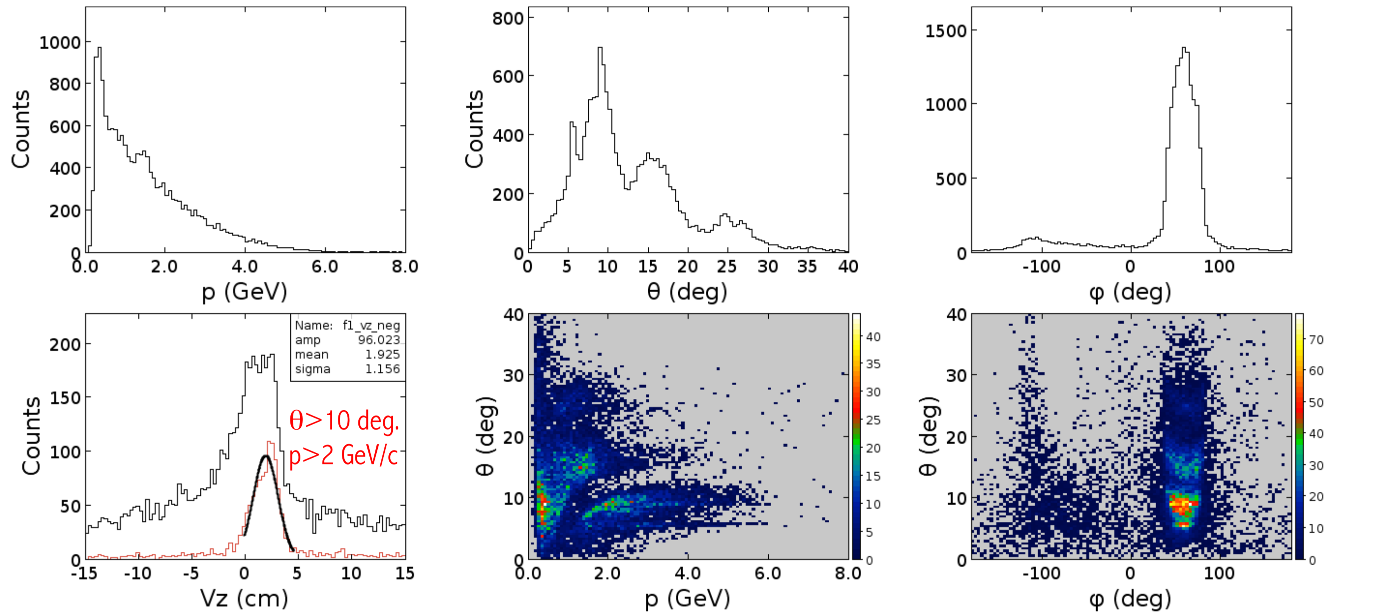
New Tracking Results

Torus Current *-1900A*
***outbending* negative tracks**

Previous Results

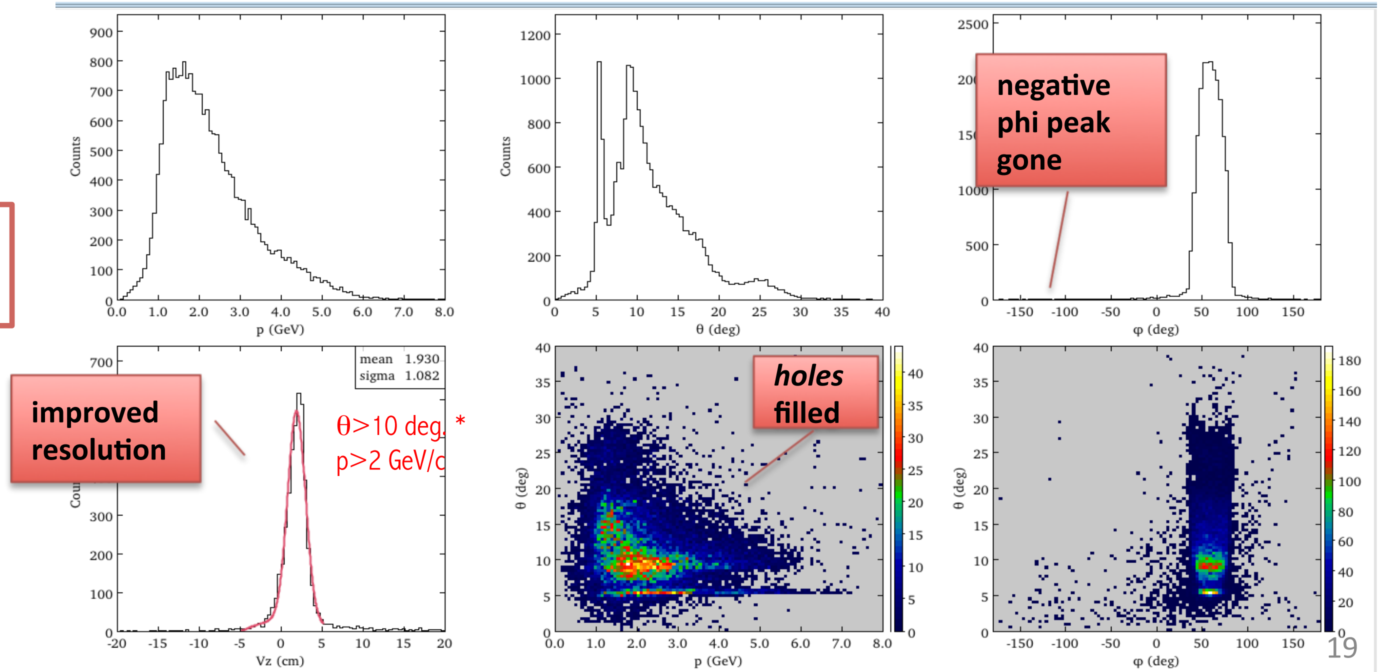
- old Kalman Filter
- 6 out of 6 superlayer tracking
- T0 values fixed at 135 ns

Negative Time-Based Tracks

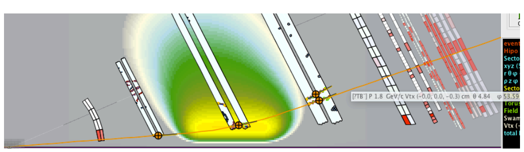


New Results

- new Kalman Filter
- 5 out of 6 superlayer tracking
- T0 values from calibration (Krishna)



* very low θ tracks produce tail in z-vtx distribution



New Tracking Results

Chef Raffaella

Torus Current *1900A*
***inbending* negative tracks**

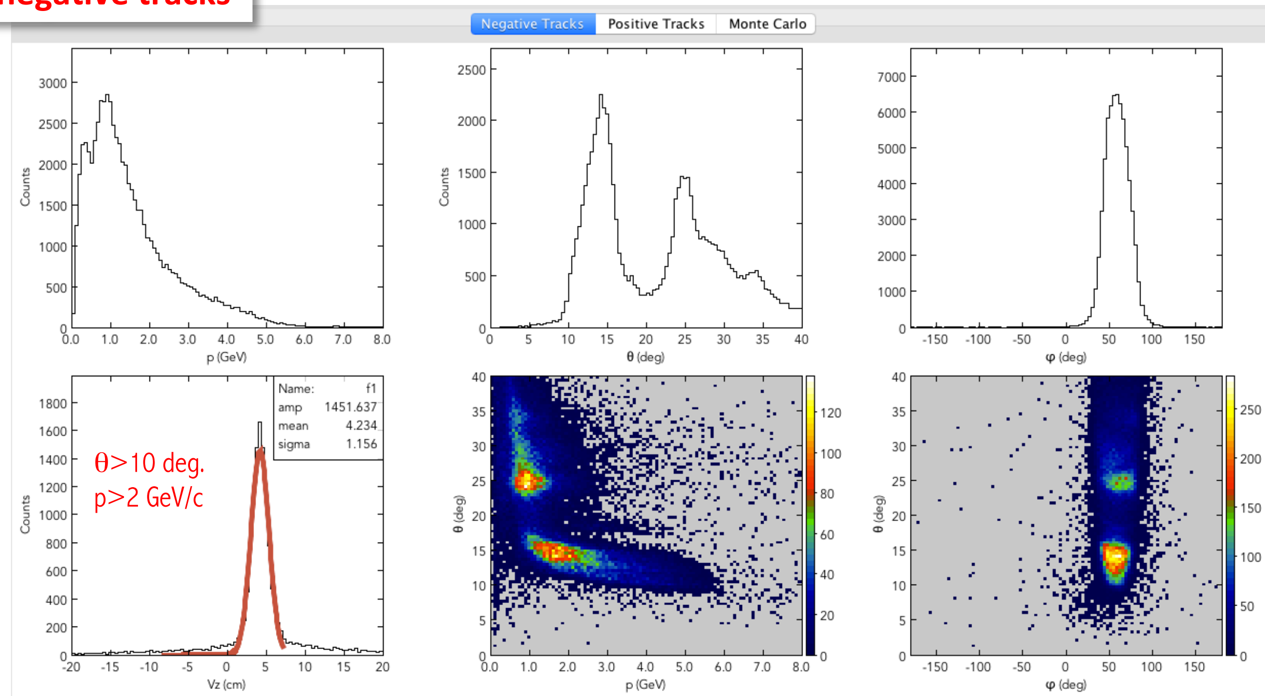
Previous Results
Runs 805--808

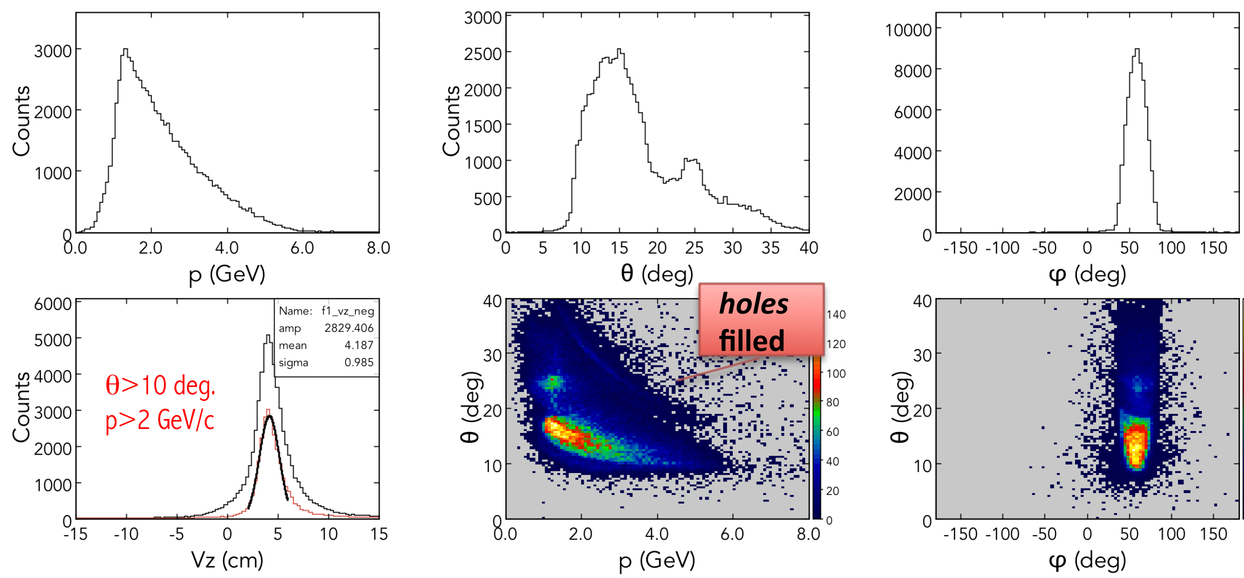
- old Kalman Filter
- 6 out of 6 superlayer tracking
- T0 values fixed at 135 ns

New Results
Runs 804--808

- new Kalman Filter
- 5 out of 6 superlayer tracking
- T0 values from calibration (Krishna)

negative tracks





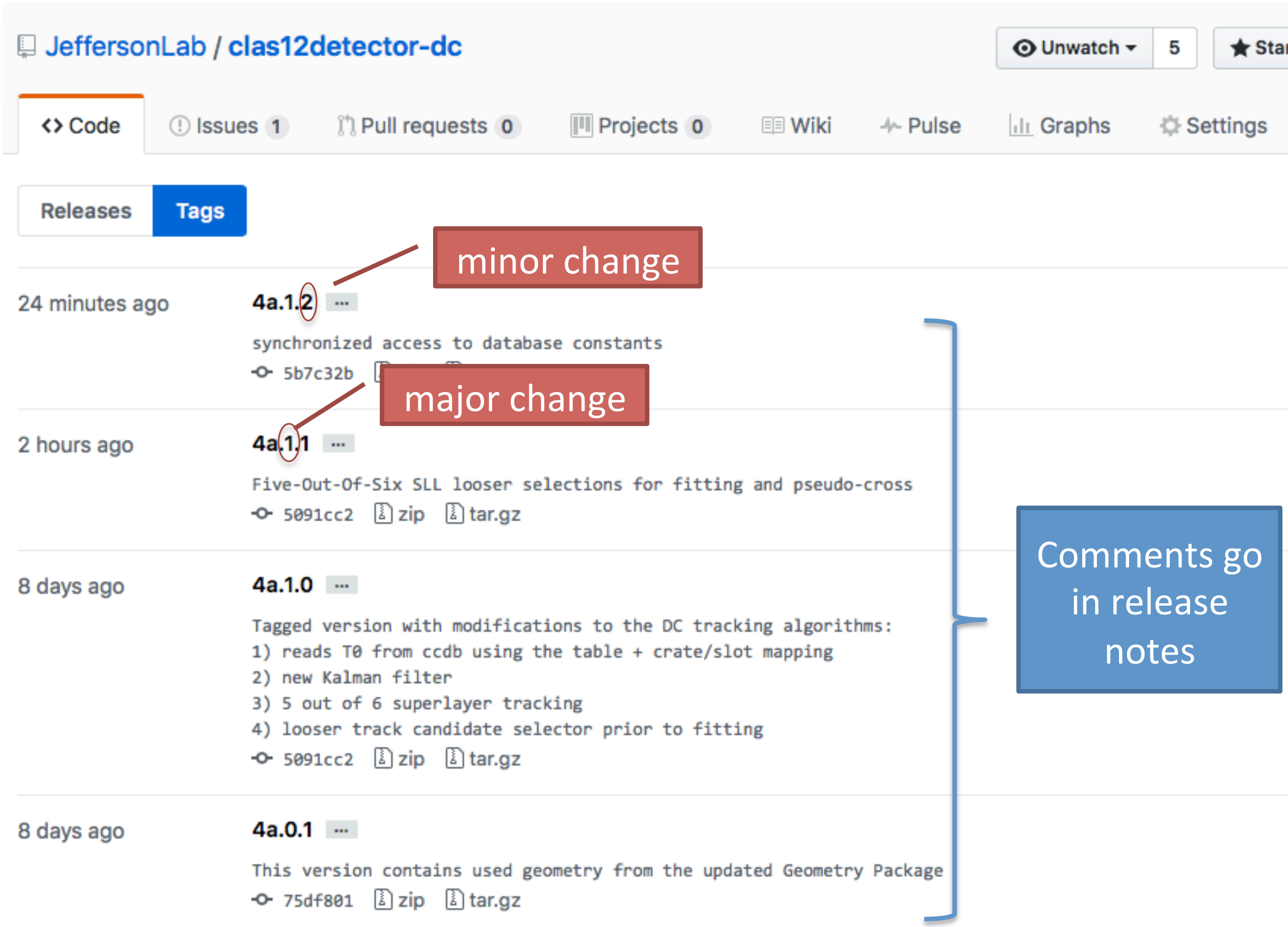
❑DC Tracking

- Code release tags

Code Organization

- Reconstruction tagging scheme utilized…

Code in Git



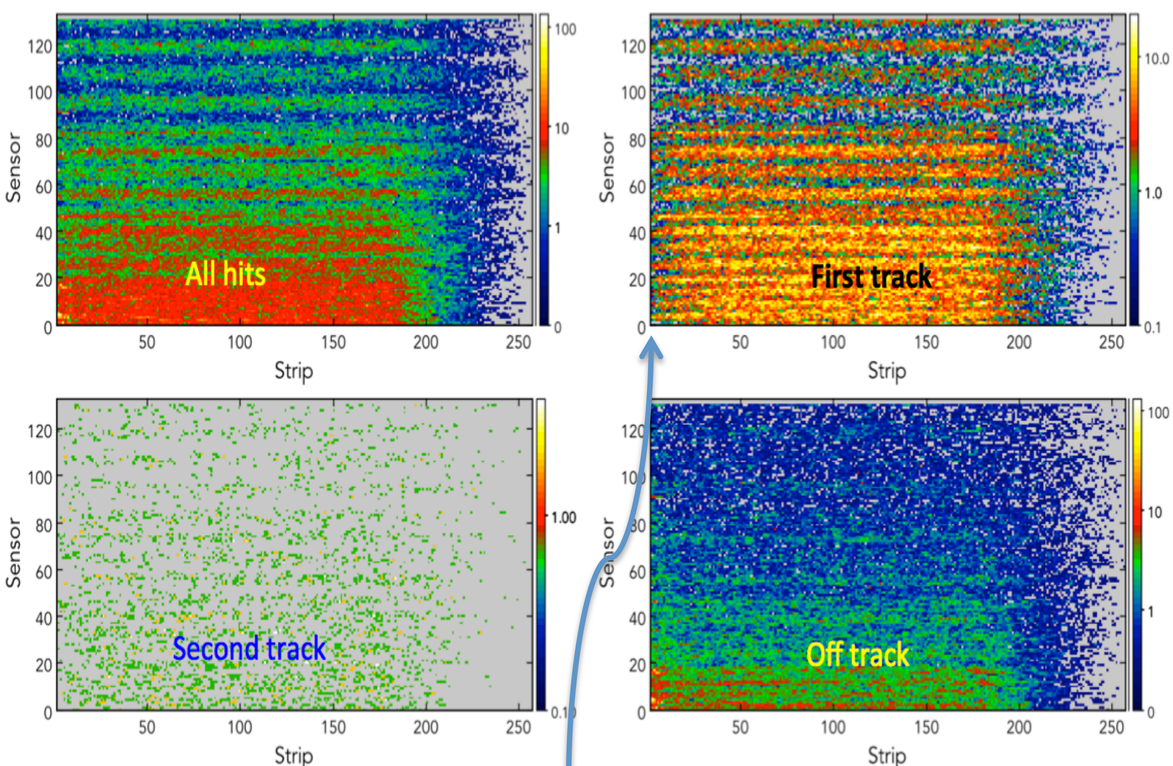
- Github reorganization in progress (Nathan)

❑SVT Tracking

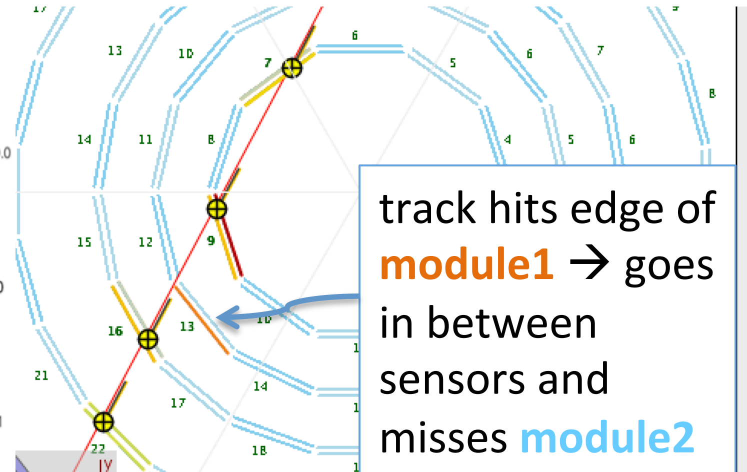
- Monitoring Suite Development (Yuri Gotra)
  - ongoing implementation of SVT Validation suite into MON12
  - tracking algorithms validation

Tracking developments (CVT)

SVT Hit Occupancy (KPP run 799, SVT standalone trigger)



depletion



track hits edge of **module1** → goes in between sensors and misses **module2**
➔ no cross, **hence hits were not used to fit the track**

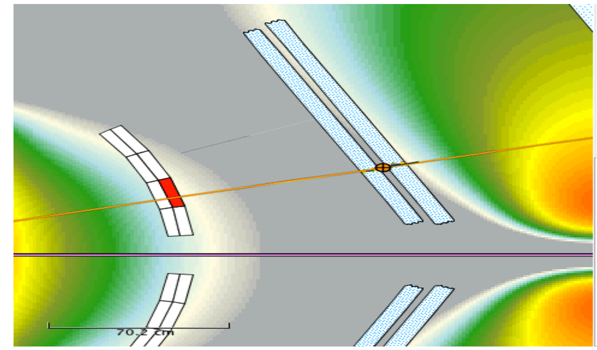
After fitting search for unassociated hits on track trajectory (|calcCentroid – clusterCentroid| <3)

❑Event Builder

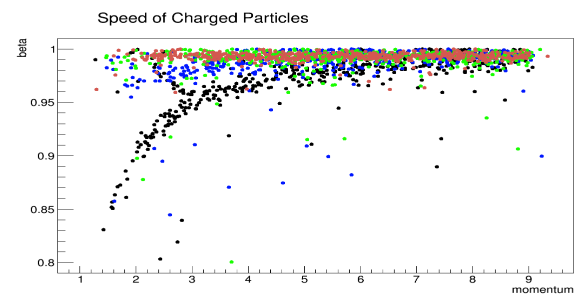
- EB Code Development (<u>Joseph Newton</u>, Stepan Stepanyan, Gagik Gavalian)
  - detector hits-to-track matching
  - PID

New Event Builder (J. Newton [ODU])

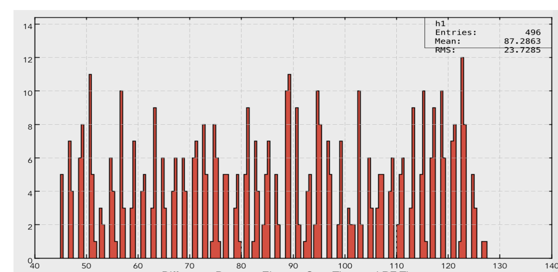
- Geometrical matching between HTCC hits and DC tracks
- Particle Identification
- CCDB parameters access
- New Output Banks
  - REC::Cherenkov = All Cherenkov Hits and their positions and number of photoelectrons
  - REC::Tracks = All Tracks Found at Hit-Based and Time-Based levels
  - REC::Event = Contains event-by-event information such as the event start time



HTCC Hit Matching based off reconstructed angles



Particle identification based off speed of tracks, which is reliable at low momentum



Difference between the electron vertex time and the RF beam bunch time

Conclusions

- DC Tracking
    - Calibration constants included in tracking
        - T0-subtraction & Time-to-distance parameters working
    - New Kalman Filter
        - Optimization ongoing
        - Saves fit χ^2 → subsequent use for track selection
    - 5 out of 6 superlayer tracking
        - Optimization ongoing
        - Efficiency studies ongoing
    - Further Studies to tune the tracking for inefficiencies
    - Code ready for next release

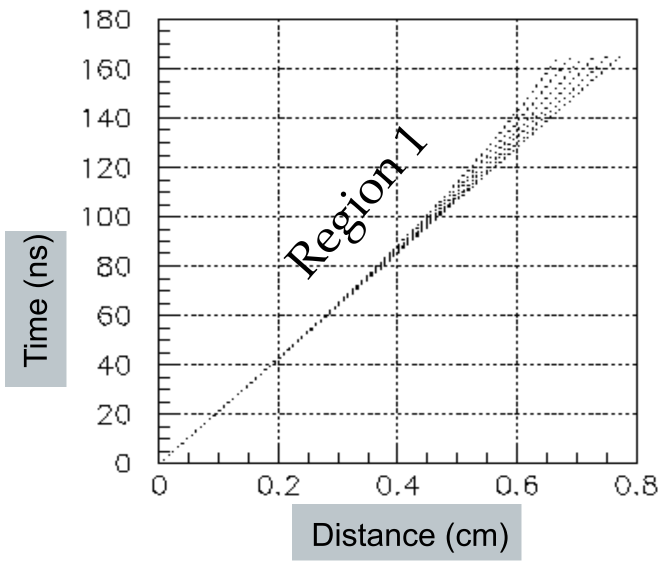
- SVT Tracking
    - Monitoring suite to be integrated in Mon12 (Yuri Gotra)
    - Ongoing: alignment, CTOF-SVT matching → needs CTOF Z coordinate reco., BCO time info to reject the noise hits (in KPP, large trigger window ➔ over half of out of time noise hits can be removed offline prior to track finding)

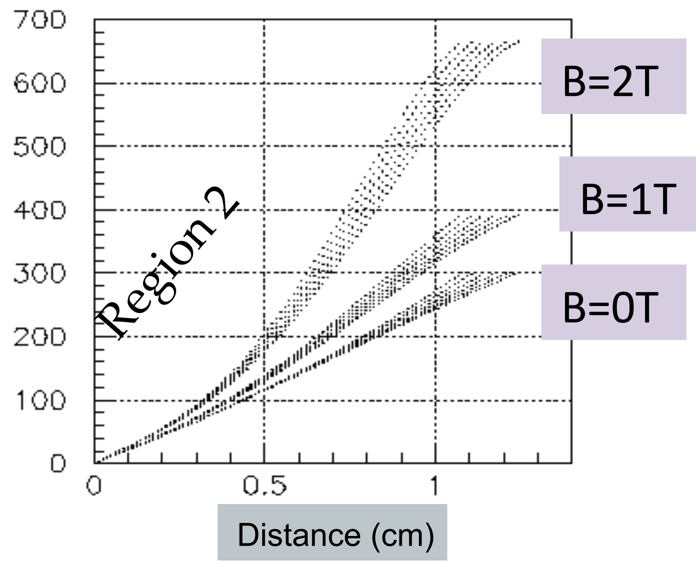
- Event Builder
    - FTOF & HTCC matching in place
    - New output banks in use for PID information extraction

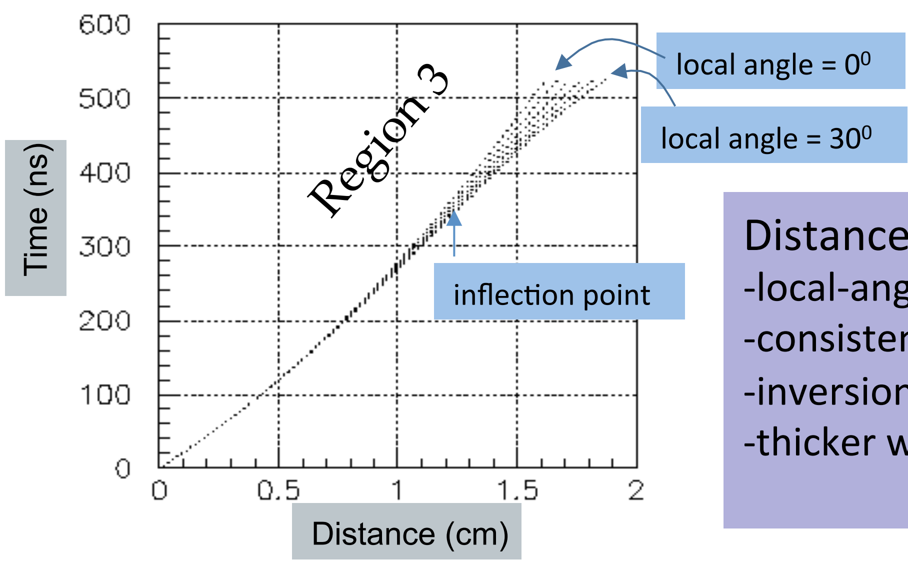
BACK-UP SLIDES

Time-to-distance parameterization

M. Mestayer

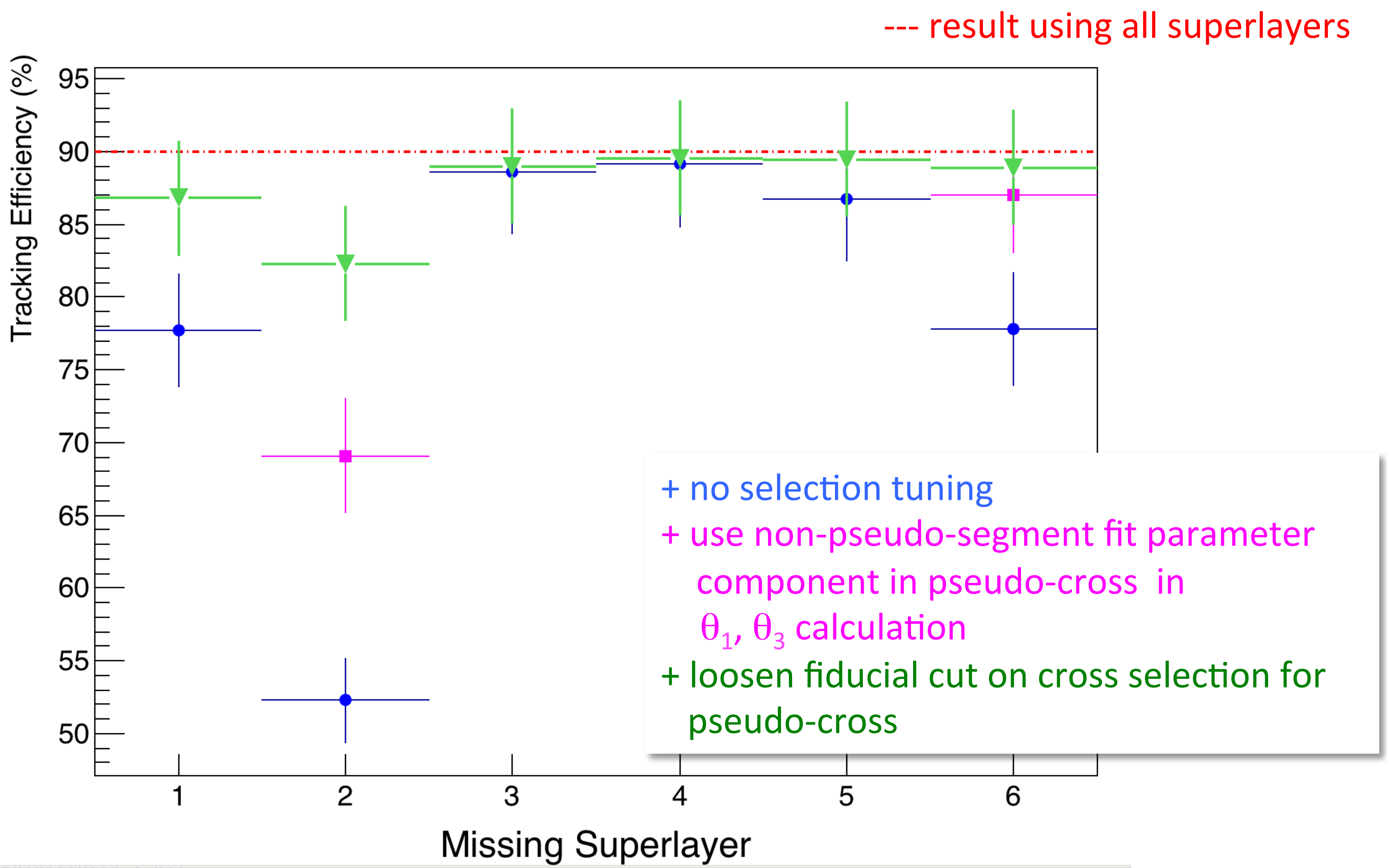






Distance → Time
-local-angle and B-field dependence
-consistent with GARFIELD
-inversion done numerically
-thicker wire → more linear
→ easier to calibrate

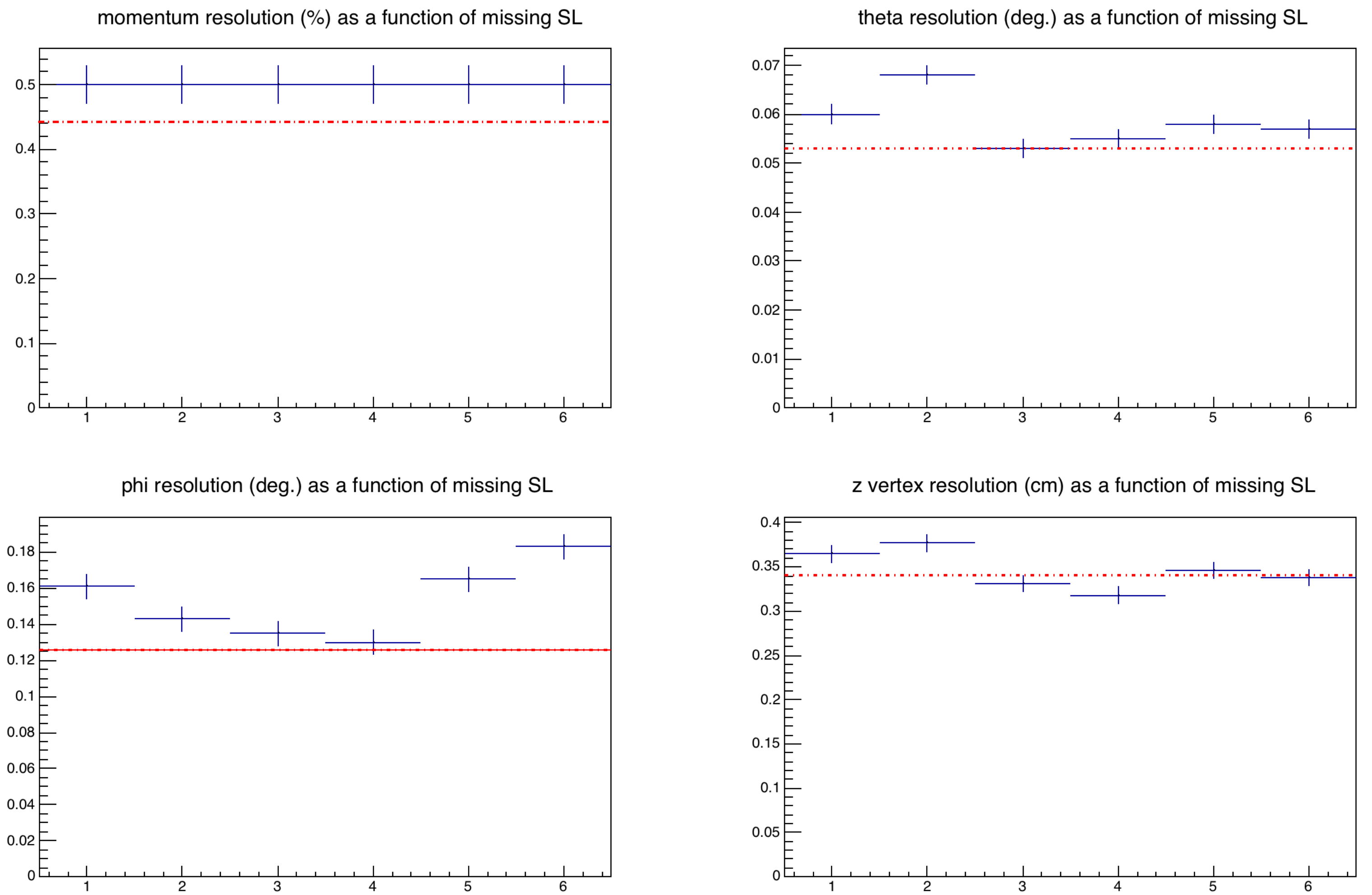
Tracking Efficiency Studies



- Further optimization ongoing

Resolution Studies

Loosing a superlayer has a minimal effect on tracking resolutions



- integrated over all p, θ, φ range

--- results using all superlayers

Vertex theta dependence (KPP data)