SPD Collaboration meeting
24 May 2024

Status of pattern recognition and vertex fitting in SpdRoot

V. Andreev (LPI, Moscow)

Introduction

This talk contains two parts: **track reconstruction and vertex reconstruction tasks.**

Track reconstruction is usually divided on two separate sub-tasks:

a) track finding (or pattern recognition);
b) track fitting (in general on the base of Kalman filter method).

Track finding:

a) division set of measurements in the tracking detectors into subsets;
b) each subset contains measurements believed to originate from the same particle.

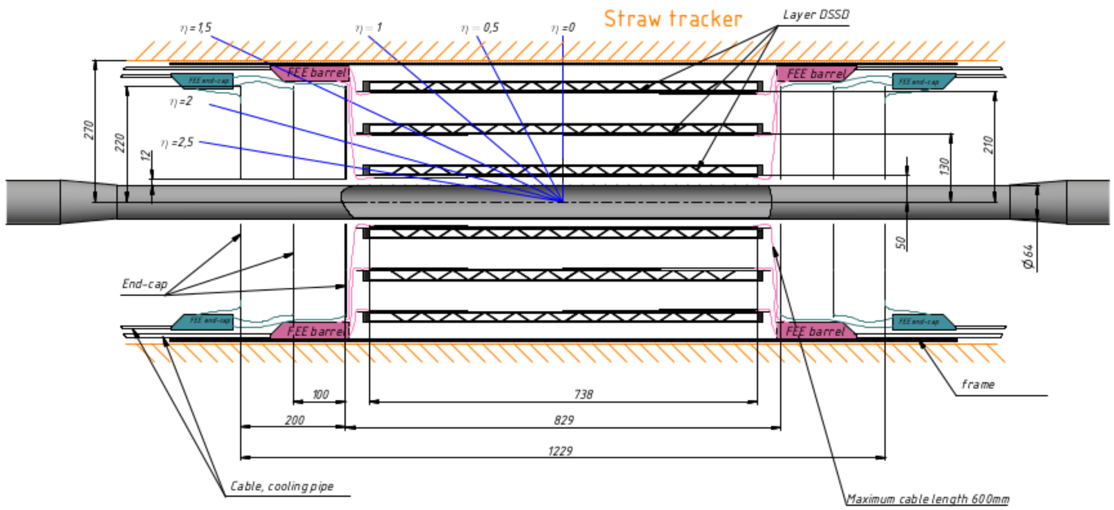
Track fitting - starts with the measurements inside one subset as was provided by the track finder.

Different techniques can be used for track finding procedure:

a) global method => simultaneous clustering detector hits into track candidates;
b) local method => generate seeds in one detector and then use these seeds to complete them by hits from another detector and form track candidates.

The local track finding method is now realized for SPD track reconstruction procedure.

General conditions



1. CDR => vertex detector: MAPS with 4 layers or DSSD with 5 layers.

2. TDR => vertex detector: 4MAPS with 4 layers or DSSD with 3 layers.

3. 1-st stage of SPD (TDR) => vertex detector with 1 or 3 layers of MicroMegas detector.

4. Straw tube trackers (TDR) => 31 double layers in Barrel and 8 double layers in Endcap.

5. SPD global coordinate system => z-axis is oriented along the nominal beam direction, y-axis is vertical and x-axis is perpendicular to z-y axis and is directed toward the center of the collider ring.

6. Zero beam-crossing angle is foreseen in SPD experiment and interaction range will have Gaussian shape with σ = 30.0 cm along z-axis and σ = 0.1 cm for x,y-axis.

7. Solenoidal magnetic field till 1.2 Tesla is also foreseen in SPD with uniformity better than ±2% inside −900 mm ≤ z ≤ 900 mm and radius 150 mm range.

Track reconstruction (general scheme)

1. Event generation (Pythia8 or another particle generators).

2. Simulation => produce sim-hits in vertex and straw-tracker detectors using SPDroot.

3. Produce “digi” - hits => apply smearing for vertex and straw hits.

4. Pattern recognition => find track candidates with the set of vertex and straw hits:

   - construct primary track seeds using hits in vertex detector;
   - use these seeds as input for constructing track candidate in straw tracker;
   - add consistently straw-hits from tracker detector to track candidate;
   - finally track candidate will contain vertex and straw tracker hits;
   - apply cleaning procedure (remove duplicates).

5. Apply Kalman fit for remaining track candidates.

6. Present track reconstruction program was optimized for MAPS (4 layers) and DSSD (5 layers) vertex detector, also was checked for DSSD (3 layers) option.

7. Only primary vertex tracks are considered for track reconstruction in present version of program.

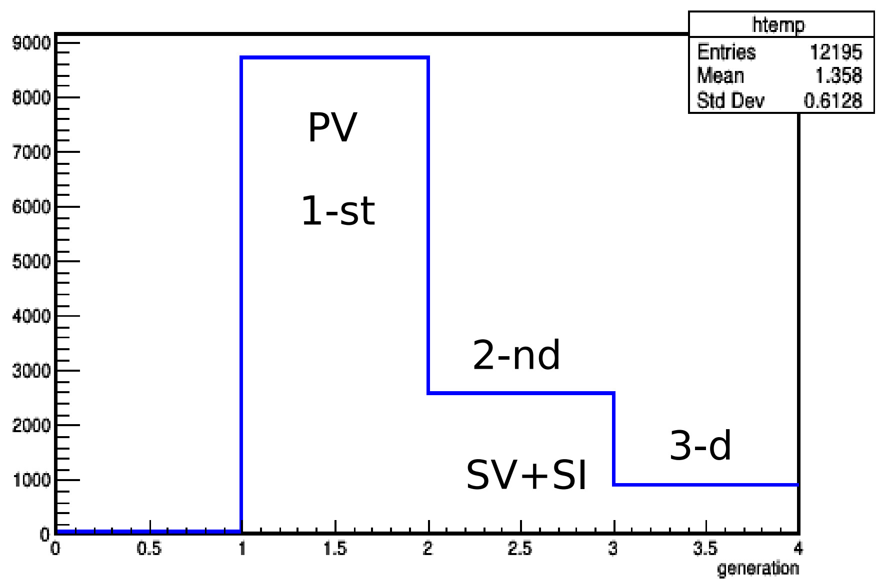
Track reconstruction (general points)

1. "Ideal" track reconstruction – Kalman fit uses exact MC information about hits which are belonged to the track and particle code.

2. Track reconstruction can be considered for the different particle generation level:

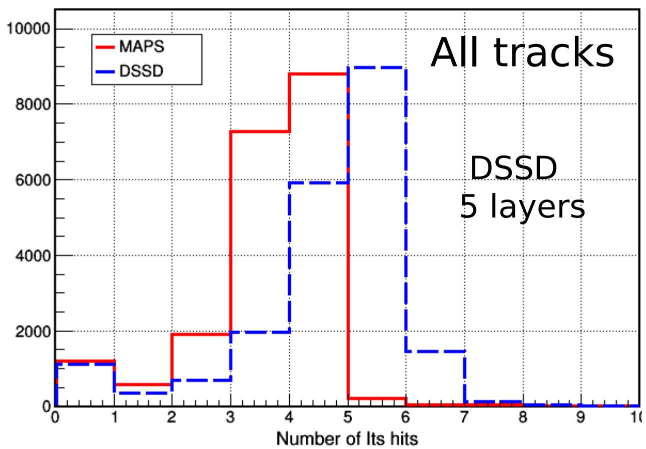
   a) 1-st generation - only primary particles from interaction vertex are considered;
   b) 2-nd generation – secondary interactions and short-lived particles are considered;
   c) 3-d generation - further interactions and particles are considered.

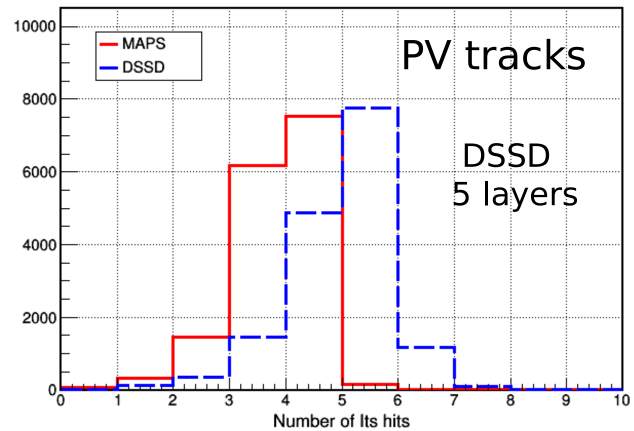
3. Minimum Bias events are simulated with Pythia8 at $\sqrt{s}$ = 27 GeV with Gaussian distribution of primary vertex z-position (σ = 30.0 cm).

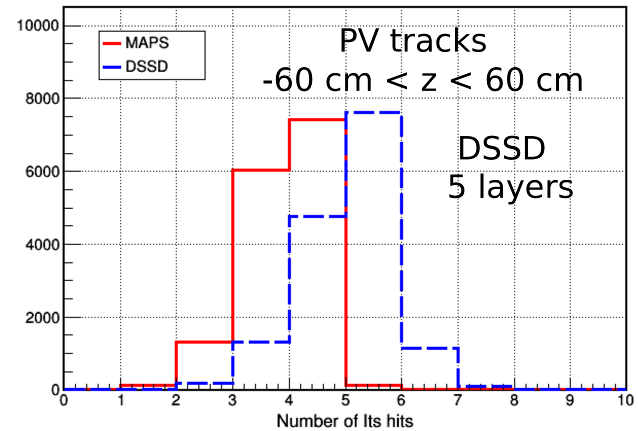


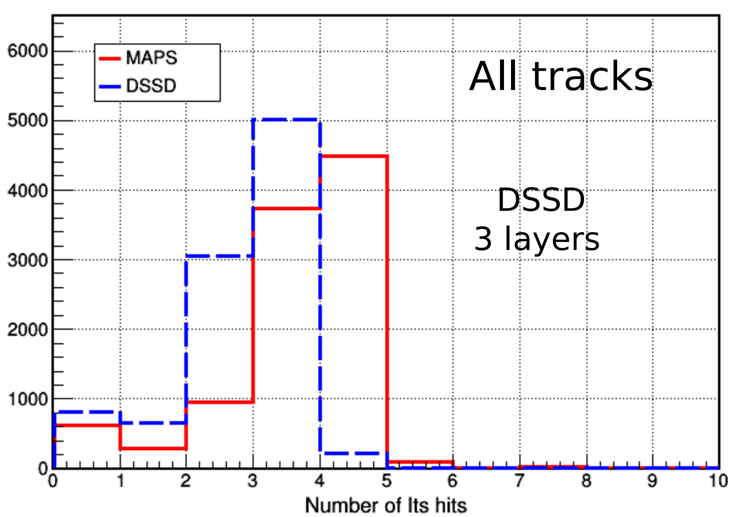
Number of tracks for the different generation level
(~70 % primary vertex tracks)

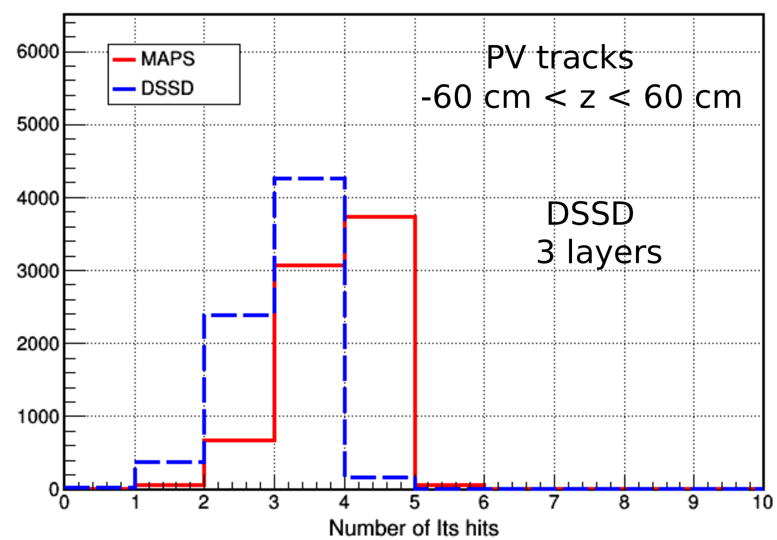
Number of hits in vertex detector









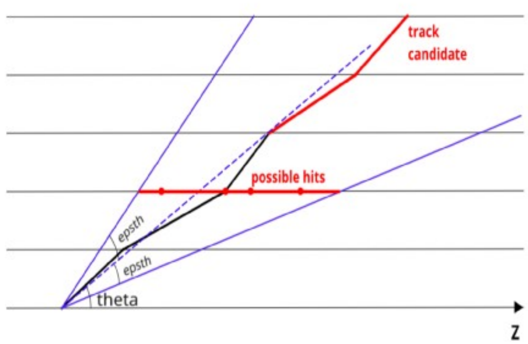


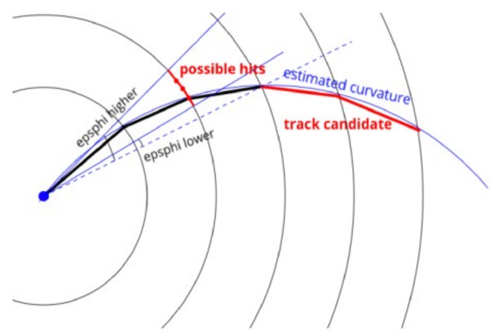
Track seeds in vertex detector

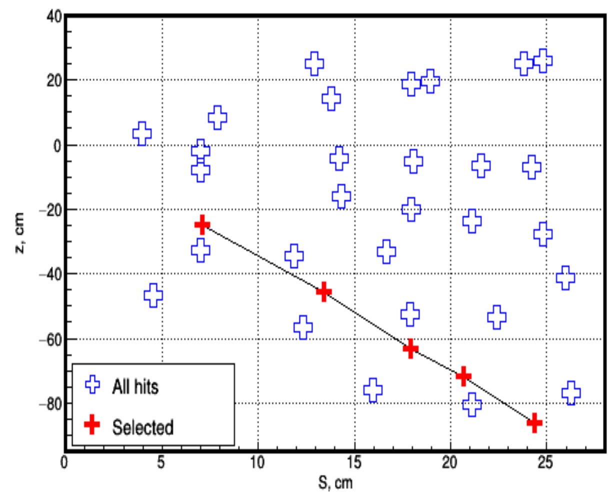
1. Charge particle trajectory in constant magnetic field is helix which can be described:
   a) in XY plane as circle with radius $R = PT / 0.3 \cdot B$ and
   b) z-coordinate is the function of arc length (s), $z(s) = z0 + s \cdot \tan\lambda$, where $s = (\Phi - \Phi 0) \cdot R \cdot q$,
      Φ – azimuthal angle, λ – dip angle, z0 and $\Phi 0$ - parameters in starting point or in primary vertex

2. Construct 2-points track seeds using combination of 2 hits from the different vertex layers and provide first estimation of track parameters (just adding x=y=0 point as 3-d point)

3. Then produce 3, 4 and 5-points seeds from 2-points seeds adding new hit from another layer with taking into account some $\Delta\theta$ and $\Delta\varphi$ – conditions and update track seed parameters





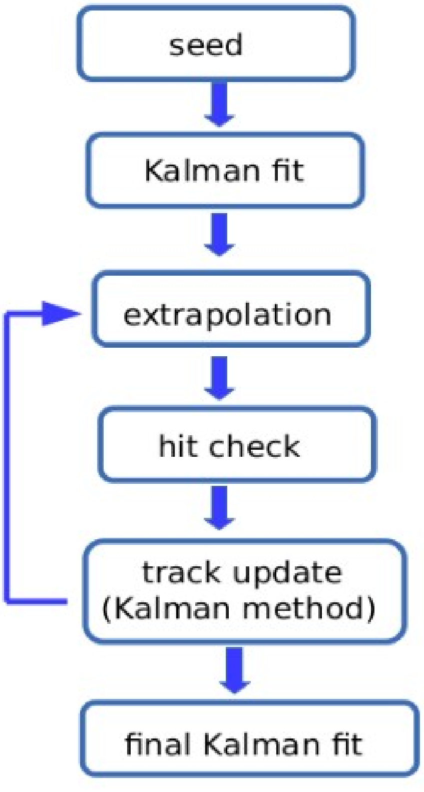


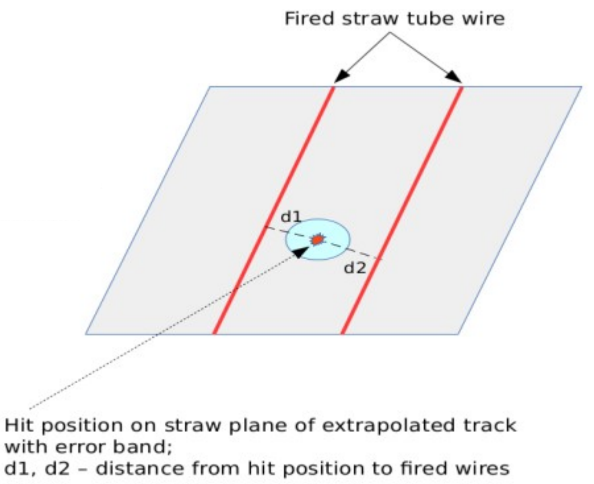
4. Finally track seed can be presented as the line in sz – coordinates

6. Seed finding procedure provides the next track seed parameters:
   a) charge;
   b) curvature or radius;
   c) theta and phi angles;
   d) estimation of primary vertex position.

7. Only track which contains 2 or more vertex hits can be reconstructed with this algorithm.

Add straw hits to track seed





1. Extrapolate track to straw plane (starting from 1-st layer).

2. Check the distance of track hit position on straw plane (red point on picture) to fired straw-wires (red lines on picture):
   a) if distance is greater than ~1.0 cm => skip this straw-wire;
   b) if distance is less than 0.5 cm => accept hit of this straw;
   c) if the distance is between 0.5 cm and 1.0 cm =>check this straw-wire with taking into account error (blue area) of track extrapolation;
   d) check chi2 increment with new straw hits at the track updating procedure.

3. If 2 fired straw-wires (red lines on picture) on plane are compatible with the track candidate, the next reconstruction options could be used:
   a) use both straw hits for updating track parameters;
   b) use the best hit for updating track parameters;
   b) if the option Kalman tree is used => then for the second hit the new track candidate is created, all previous hits are copied to the new track candidate and then extrapolation and fitting procedures are applied for all new candidates.

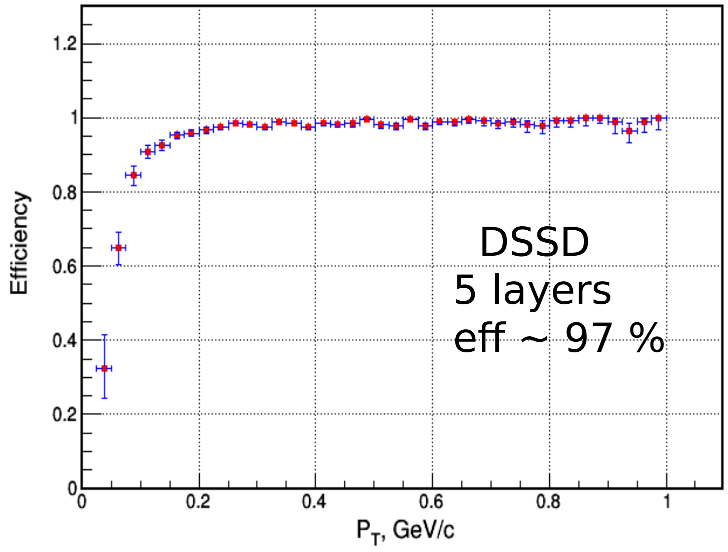
4. Finally, “big” number of track candidates are produced and each track candidate contains vertex and straw hits.

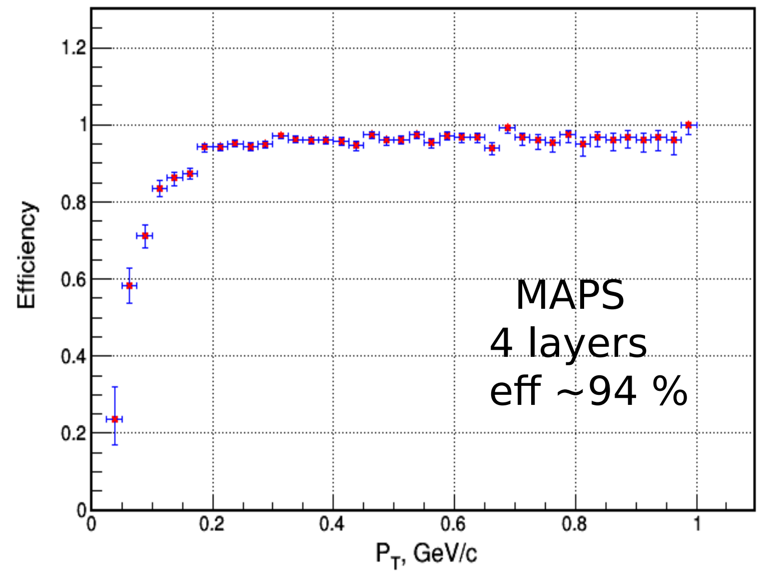
5. Remove duplicates and apply Kalman fit to remaining track candidates.

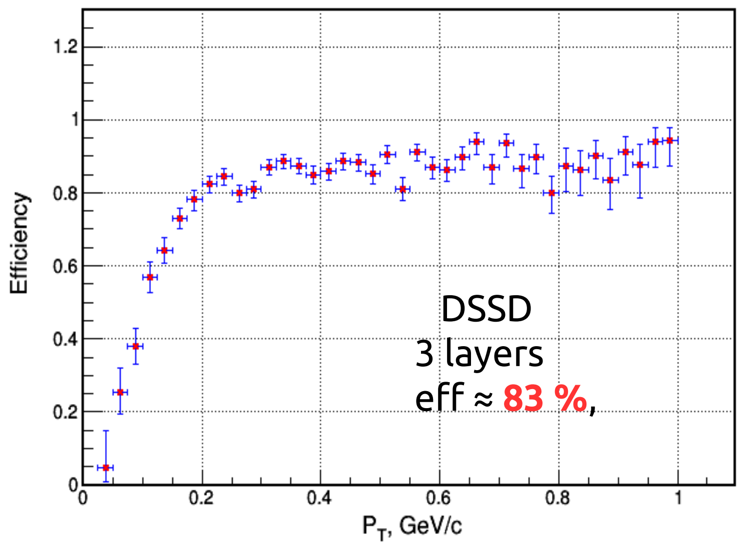
6. Write reconstructed track parameters in SpdTrackRC class with the same set of track parameters as for the usual SpdTrackMC for adding straw hits to track candidate.

Results of track reconstruction

Track reconstruction efficiency for charged tracks with Pt > 25 MeV/c and for primary vertex position inside z-range -60 cm < z < 60 cm shows good performance, low fake rate < 2 % and number clone tracks < 1% for all geometries.







Procedure for estimation of track reconstruction efficiency:

a) consider "ideal" reconstructed track;

b) try to find among reconstructed tracks such track which can be considered as identical to "ideal" reconstructed track => that means these tracks ("ideal" and reconstructed) have the some number of common vertex hits (>=2) and also have > 50 % same "straw" hits.

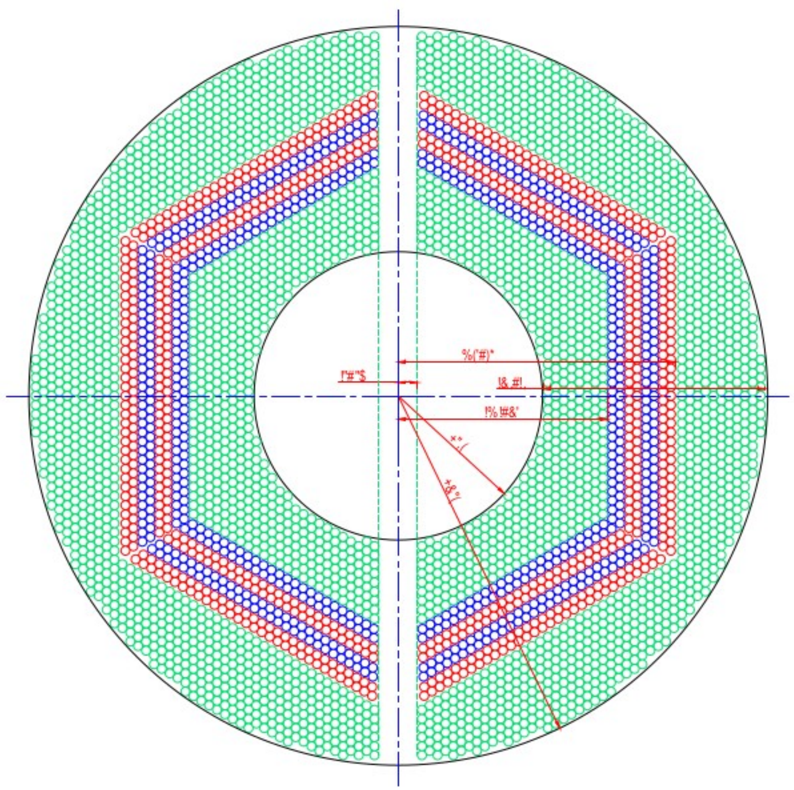
Need special reconstruction algorithm for track with 1 or less vertex hits for DSSD (3 layers) vertex options.

Pattern recognition in straw detector

1. The track reconstruction procedure in PANDA experiment will be presented as example:

   a) green tube – straw tubes which are parallel to z-axis;

   b) red and “blue” tubes – skewed straw tubes (tubes are tilted by $\pm 2.9^0$ with respect z-axis, 8 layers).



2. Pattern recognition is also consisted of two parts:

   a) pattern recognition in XY - plane.

   Charge particle trajectory in constant magnetic field is considered as helix which can be described as circle with radius $R = PT / 0.3 \bullet B$ and can be represented the next 5 parameters (as example):

   - xc horizontal coordinate of the helix center;
   - yc vertical coordinate of the helix center;
   - ω = Q/r signed curvature of the helix;
   - $\phi 0$ azimuthal angle at the reference z0;
   - tan λ dip angle of the helix with λ = arctan(pz /pt).

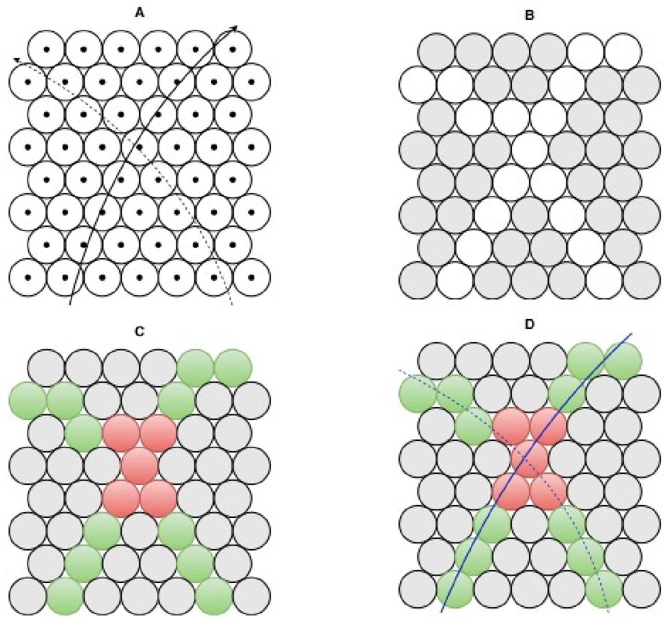
   b) longitudinal tracks reconstruction:

   z-coordinate is considered as the function of arc length(s),

   $z(s) = z0 + s \bullet \tan\lambda$,

   where $s = (\Phi - \Phi 0) \bullet R \bullet q$, Φ – azimuthal angle, λ – dip angle, z0 and $\Phi 0$ - parameters at starting point or in primary vertex

Celluar Automaton base on hit clusterization



A Tracks traverse detector

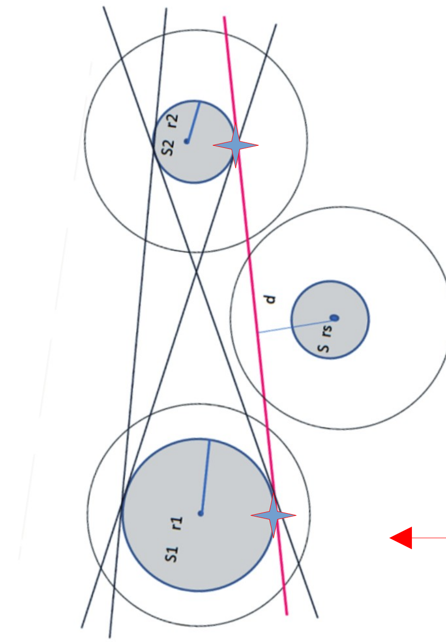
B Hits are marked as active cells

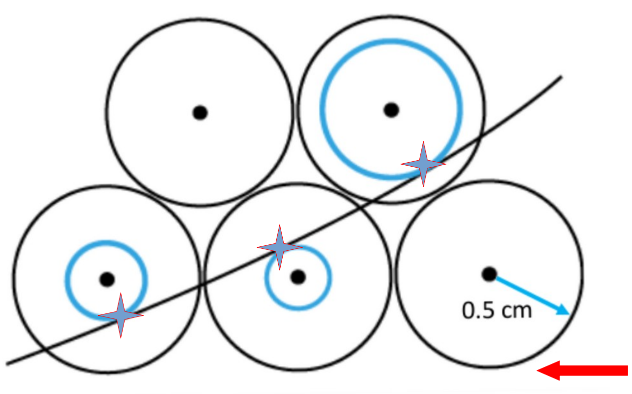
C Active cells are classified as

- unambiguous: ≤ 2 active neighbours
- ambiguous: > 2 active neighbours

D Ambiguities are resolved using track fits

- GPU version has been implemented





Blue circle is drift radius in each straw tube or the isochrone radius which describes a cylindrical surface around the wire and contains all possible positions that the particle might have traversed.

Start from unambiguous tubes and for each three neighbours tubes the “best” tangent line from 4 possible combinations can be chosen as shown on picture and also track hit position ( ) can be defined.

Estimate helix parameters (radius and center position) for unambiguous hits and after add ambiguous hits.

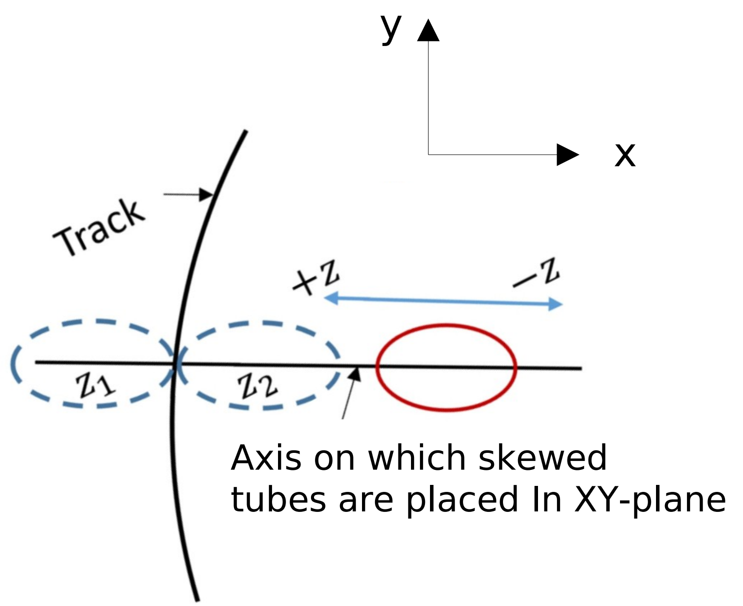
The resolution of the isochrone radius will be < 150 μm, whereas the resolution of z-position the isochrone resolution will be about 3 mm

Longitudinal track reconstruction

1. The z-position for each hit in a skewed straw tube is extracted through an alignment procedure as illustrated below. The track radius is determined before by the pattern recognition procedure in XY-plane.

   Since these tubes are skewed, the projection of the drift radius onto the XY-plane becomes an ellipse. The drift ellipse is aligned such way that its center position lies along x-axis of layer and is tangential to the particle trajectory.

   This alignment provides two solutions, introducing a left/right ambiguity with one solution on either side of the trajectory



In Panda software three different algorithms are considered:

a) Combinatorial approach.

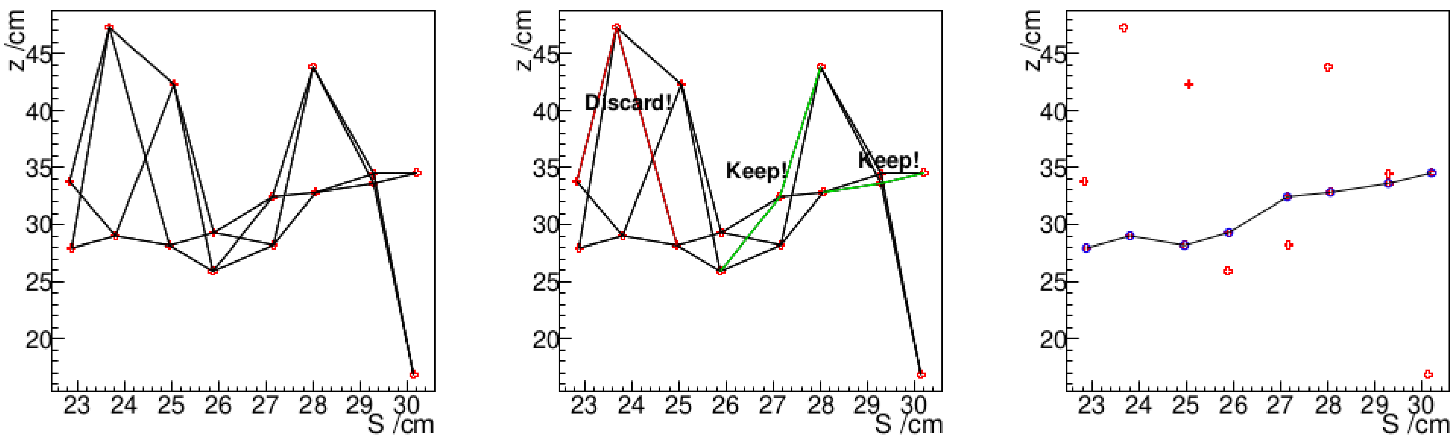
b) Hough transformation.

c) Recursive annealing fit.

Longitudinal track reconstruction

Combinatorial approach:
a) determine all possible connections between layers;
b) calculate angles between neighboring lines;
c) reject paths with $\theta < 90^0$;
d) select path by minimizing $$w = \sum_{j}^{N-1} (180° - \theta_j)^2,$$

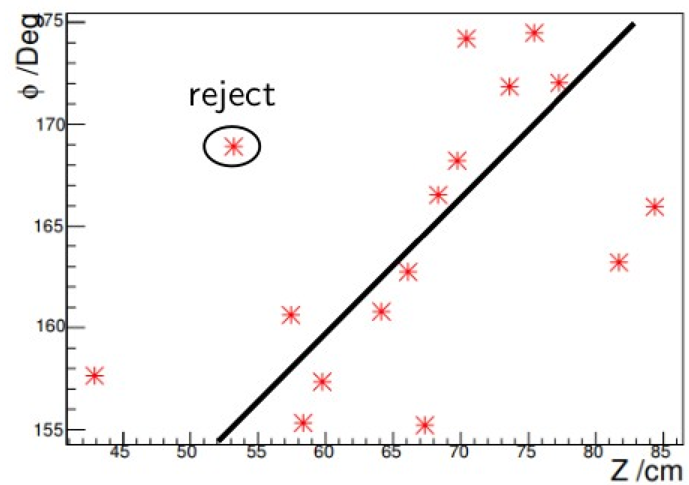
Hough transformation:
a) generate set of lines around point;
b) fill line parameters in accumulator;
c) repeat for all points;
d) select maximum in accumulator.



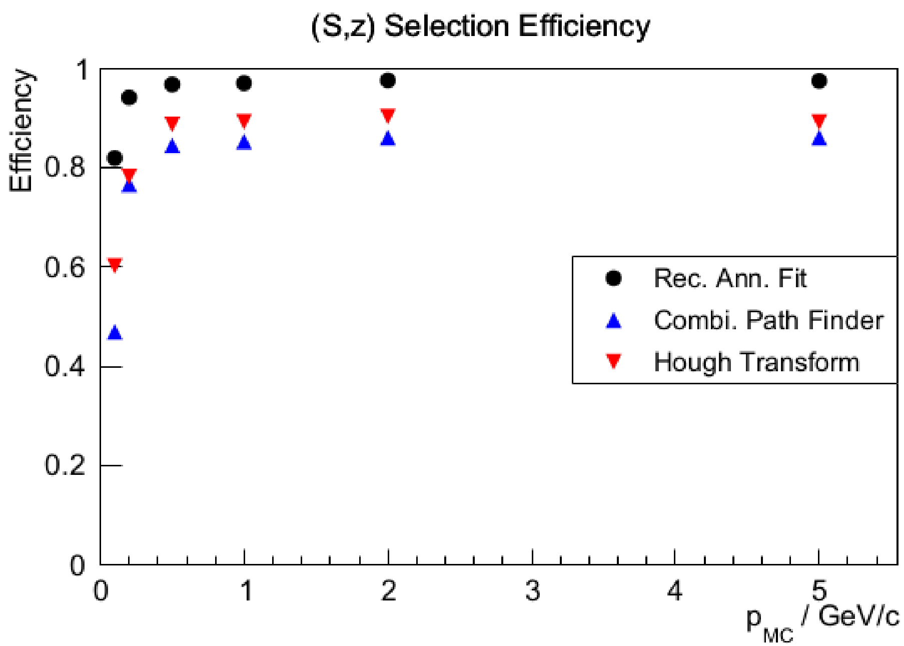
(a) (b)

Recursive annealing fit:
a) fit by line to all points;
b) remove point with largest residual;
c) calculate new line fit;
d) repeat until one point has been rejected for each straw tube;
e) do final line fit.

$$\chi^2 = \sum_{i}^{n} \frac{(z_i - kS_i - z_0)}{\sigma_i^2},$$



Longitudinal track reconstruction (results)



General results:

a) hits selection efficiency:
- Combinatorial Path Finder reaches up to ≈ 85%.
- Hough transformation achieves efficiencies of ≈ 90%
- Recursive Annealing Fit performs best with efficiency above 95%.

b) hit z-resolution ~3 mm;

c) longitudinal momentum P_l resolution ~2% at P = 1 GeV/c;

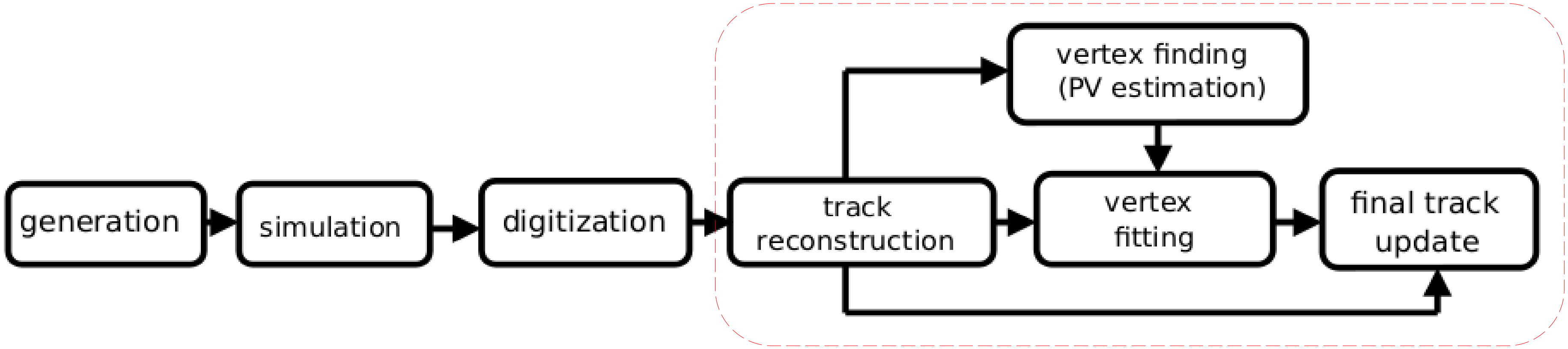
Some ideas of PANDA track reconstruction methods are now studied and under implementation for SPD track reconstruction procedure in straw detector.

General scheme of primary vertex reconstruction algorithm

Primary vertex reconstruction algorithm consists of two parts:

1. Initial approximation of primary vertex (vertex finding).

2. Fitting procedure for primary vertex (vertex fitting).



3. The current primary vertex reconstruction algorithm was introduced in SPDroot in 2019 and it’s performance was checked with the next procedure:

   a) comparison with MC vertex position;

   b) comparison with the others primary vertex reconstruction algorithms used in High Energy Physics.

4. Current reconstruction algorithm (on the base of CBM algorithm) shows the good performance.

5. The two different primary vertex reconstruction algorithms will be presented in this talk.

Primary vertex finding (1-D)

Primary vertex finding procedure is very important part of the vertex reconstruction algorithm.

The main specific feature of SPD experiment is the very broad primary vertex distribution in z-direction (σ_z = 30. cm). In SPD the vertex finding procedure is realized in assumption, that the beam distribution in transverse XY-plane is relatively small and has Gaussian shape with $\sigma_{x,y}$ = 0.1 cm.

The next algorithms are realized for initial approximation of the primary vertex:

A) 1-D clustering procedure:

- select good fitted tracks;

- extrapolate tracks to the beam axis (x=y=0) and assign to each track a weight which is proportional to inverse track error at the point of closest approach to the beam axis;

- remove tracks which are faraway from the beam axis;

- construct 1-st estimation of z-vertex position from the selected tracks with taking into account the track weight;

- find the “best” and “worst” tracks:
  a) “worst” track – track with maximum distance from z-vertex position;
  b) “best” track – track with maximum weight (or minimum error) inside some range around z-vertex position;

- produce 2-nd estimation of z-vertex position without “worst” track;

- do next iteration;

- final vertex estimation => construct cluster around “best” track using only tracks inside some range around z-position of the “best” track with taking into track weight.

Primary vertex finding algorithms (KFP)

B) 3-D clustering procedure on the base of KFParticle package.
But need to remember that KFParticle package are valid only for constant magnetic field.

Vertex construction procedure in KFParticle package from n-daughter particles:

- determine point of closest approach for two first daughter particles;
- extrapolate daughter particle to this point;
- update position of this point using standard Kalman filter equations;
- track parameters are remained unchanging during this procedure.

Procedure for primary vertex finding algorithm:

- select good fitted tracks;

- extrapolate tracks to the beam axis (using Runge-Kutta algorithm) and construct “new” KFParticle track parameters in the point of closest approach to the z-axis;

- remove tracks which are faraway from the beam axis;

- construct vertex from tracks using KFParticle procedure (put Bz value at x=y=z=0.0);

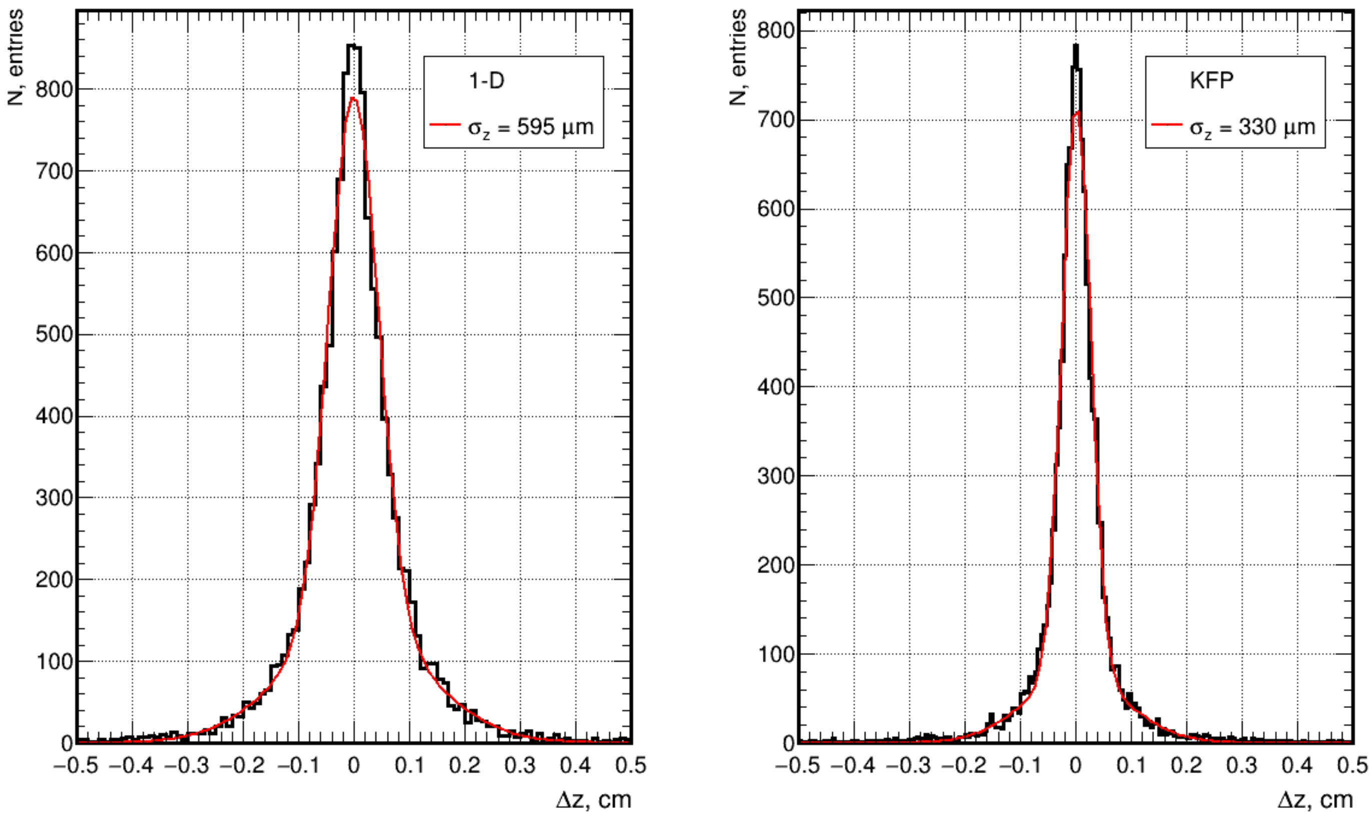
- find “ worst” track with maximum chi2 to the constructed vertex and larger than some threshold value;

- remove “ worst” track;

- do next iteration until all “worst” tracks will be removed or only 2 tracks are remained;

- construct the final primary vertex estimation using remaining KFParticle tracks.

Vertex finding (results)



Plots show the difference of z-coordinate between the generated primary vertex position and vertex position which is provided by the vertex finding methods (1-D and on the base of KFParticle package).

Data are for Minimum Bias events and for vertex detector with 1 layer of MicroMegas.

Fit is done by two Gaussian functions and sigma is determined as mean value of two sigmas with corresponding weights.

Algorithm on the base of KFParticle package shows the better vertex finding precision (~1.8 times) with comparison of 1-D method.

Primary vertex fitting algorithm (present)

The vertex fitting procedure uses the vertex position which is defined in vertex finding algorithm as the 1-st approximation.

All algorithms use the Kalman filter method.

A) Algorithm base on the CBM experiment method. A special feature of this algorithm is the next:

- track is extrapolated to the some virtual plane z_{ref} (z_{ref} is determined from vertex finding algorithm);
- then track parameters are calculated on this virtual plane;
- remove "bad" tracks which are faraway from vertex position;
- then use the second order curve for description of the track trajectory and after do linearization track parameters in the vicinity of this point;
- update vertex position using standard Kalman filter equations;
- repeat this procedure several times.

This approach gives possibility to fit the primary vertex without including the track parameters into the vertex state and so to simplify the calculations.

The current CBM primary vertex fitting algorithm has the some "weak" points:

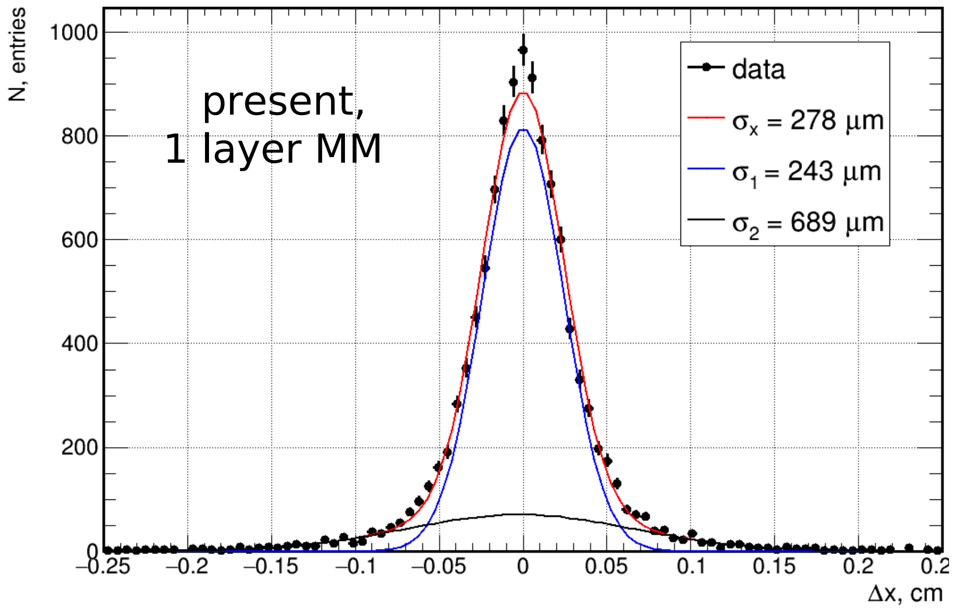
- does not include track with theta angle $\pm 3^0$ around the vertical plane in fitting procedure;
- use the second order curve for track trajectory description;
- track parameters are considered unchanged during fitting procedure.

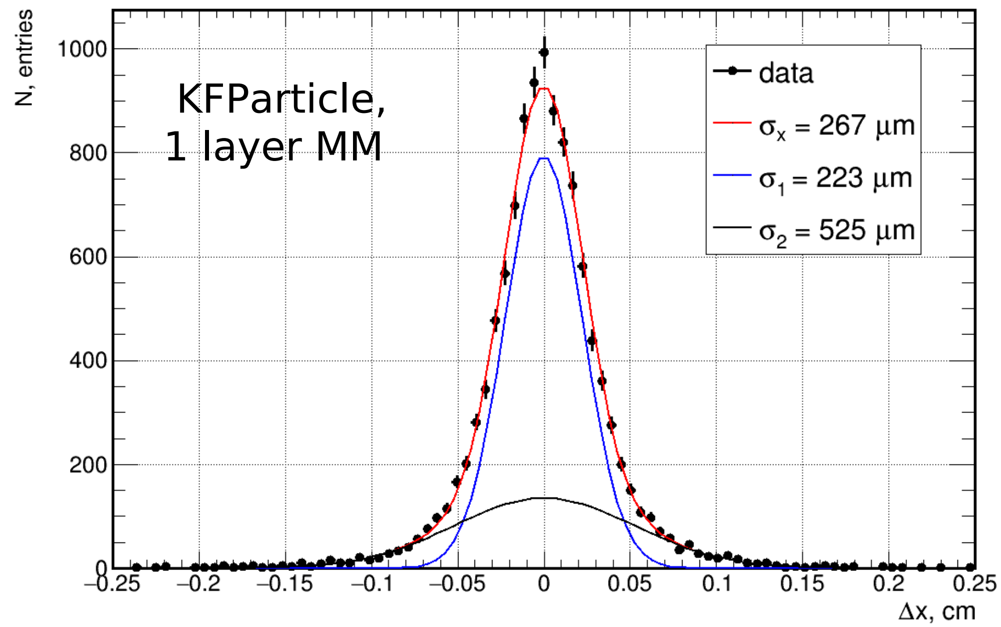
Primary vertex fitting algorithms (KFParticle)

B) Primary vertex fitting algorithm which uses the KFParticle package:

- extrapolate track to the 1-st estimation of primary vertex (from vertex finding) using Runge-Kutta method;
- construct KFParticle tracks with the new first “measured” point as the point of closest approach to the 1-st estimation of primary vertex;
- in the small vicinity of this vertex the magnetic field can be considered as constant and KFParticle package can be used;
- remove tracks which are faraway from the vertex;
- construct vertex using KFParticle procedure (put Bz value at xv, yv, zv);
- find “ worst” track with maximum chi2 from the constructed vertex and larger than some maximum value;
- remove “ worst” track;
- do iteration until all “worst” tracks will be removed or only 2 tracks are remained;
- construct final primary vertex using remaining KFParticle tracks;
- track parameters are considered unchanged during the fitting procedure;

Primary vertex reconstruction precision (x)





For checking of these primary vertex reconstruction algorithms (present and on the base of KFParticle package) 10000 Minimum Bias events was simulated with Pythia8.

Different vertex detector options were also considered: without vertex detector, with 1 and 3 layers of MicroMegas detector, DSSD and MAPS detectors.

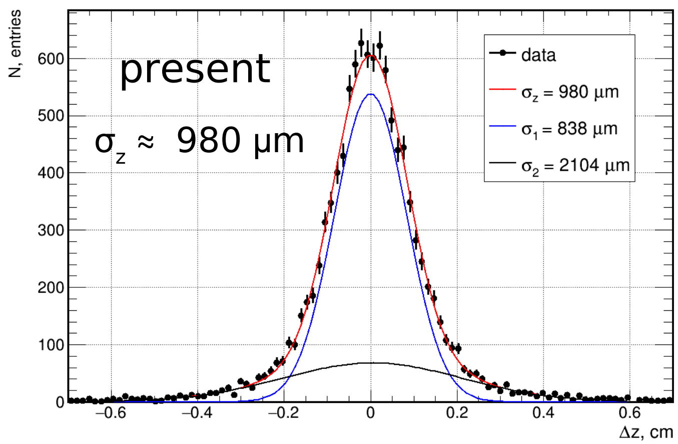
Primary vertex was smeared with Gaussian shape ($\sigma_{x,y}$ = 0.1 cm and σ_z = 30 cm).

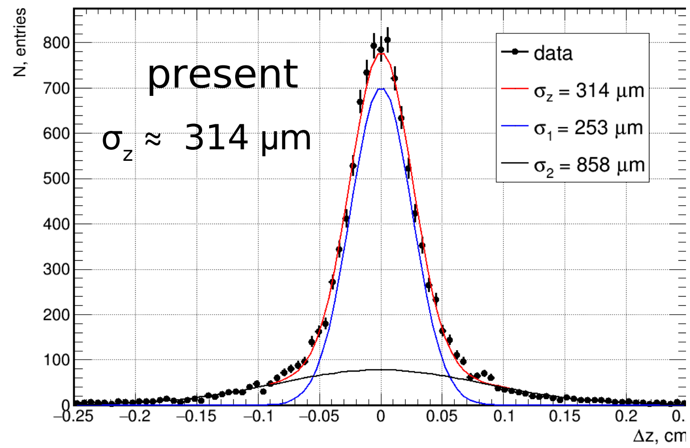
Select "good" reconstructed tracks: fitted track with chi2/ndf < 12 and with number hits > 8 (in vertex and straw detector). Presented results are obtained for three or more tracks used for the reconstruction procedure.

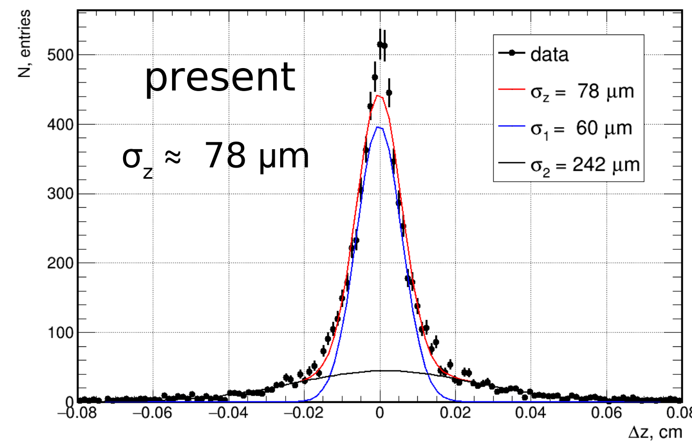
Reconstruction precision Is determined as the difference between reconstructed and generated value.

Fit is done by two Gaussian functions and sigma is determined as mean value of two sigmas with corresponding weights.

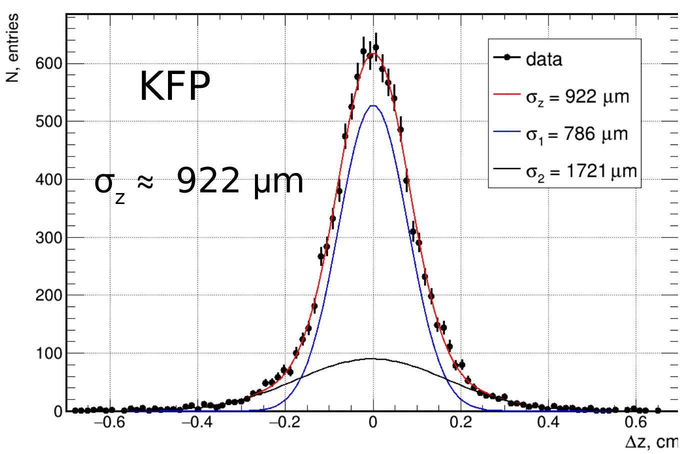
Primary vertex reconstruction precision (z)

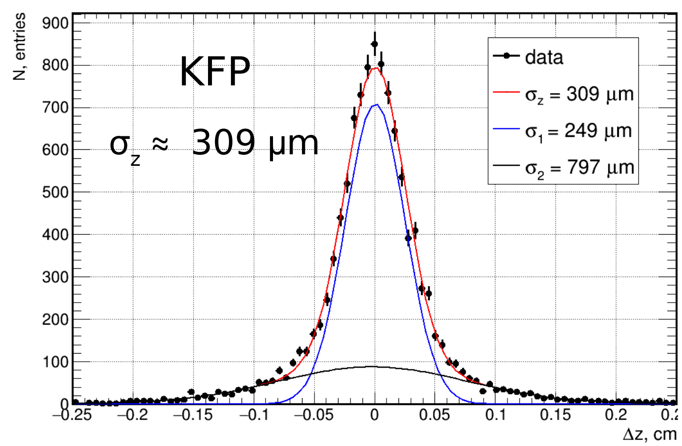


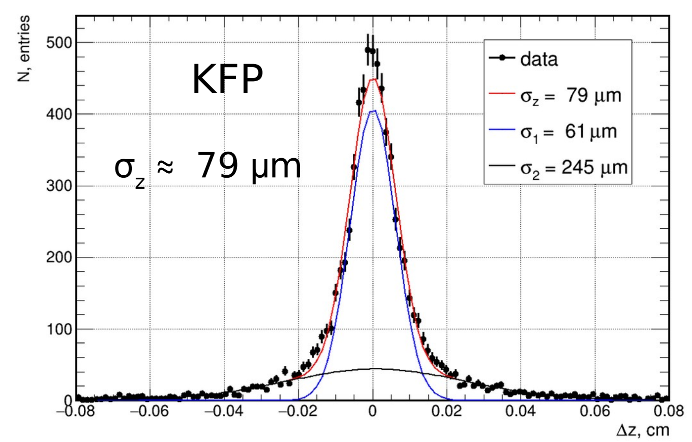




noVtx | MM, 1 layer | MAPS

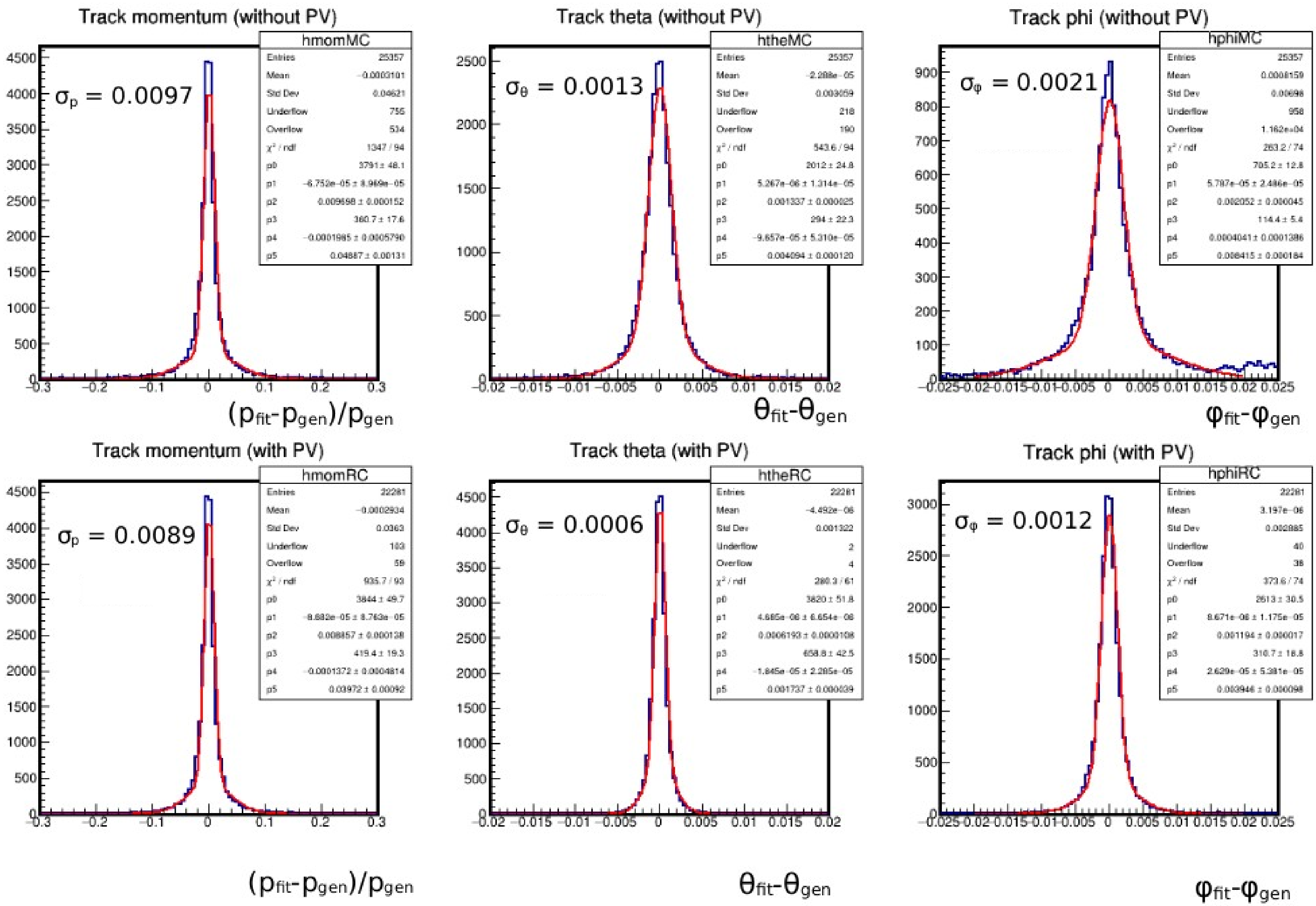






Both algorithms (present and KFParticle) show the similar primary vertex reconstruction precision.

Track parameters with PV



Adding of PV to the track fit procedure provides essential improvement of track parameters.

Summary

1. Two primary vertex reconstruction algorithm were presented.

2. Both algorithms show the similar results.

3. The update SPDroot which will include new vertex finder on base of KFParticle is foreseen in 2 weeks.

4. Track reconstruction program (track finder and fitting) is available for SPD geometry for the MAPS and DSSD vertex detector.

5. Need some update for this reconstruction algorithm for the TDR conditions.

6. Important task now is developing track reconstruction algorithm for straw tracker.