SPD Collaboration meeting
3-6 October 2022

Status of track reconstruction for SPD experiment

V. Andreev (LPI, Moscow)

Introduction

- Track reconstruction is usually divided into separate sub-tasks:

  - track finding
  - track fitting (in general on the base of Kalman filter method)

- Track finding (or pattern recognition):

  - division set of measurements in the tracking detectors into subsets
  - each subset contains measurements believed to originate from the same particle

- Track fitting:

  - starts with the measurements inside one subset as provided by the track finder

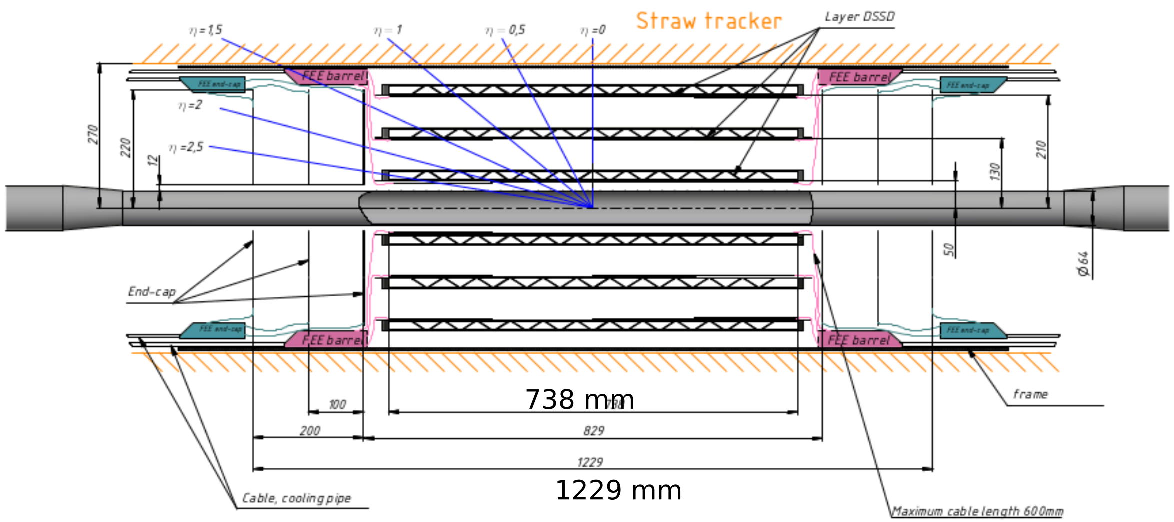
Different techniques can be used for track finding procedure:

- global method => simultaneous clustering detector hits into track candidates

- 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 used for SPD track reconstruction procedure

General conditions



1. 1-st version of vertex detector => 4 MAPS layers or 5 DSSD layers (CDR)

2. Last version from TDR => 4 MAPS layers or 3 DSSD layers (?)

3. Also vertex detector with 3 layers of MicroMegas detector is considered in TDR

4. Straw tube trackers (TDR) => 30 double layers in Barrel and 8 double layers in Endcap

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

6. Zero beam-crossing angle is foreseen in SPD experiment with Gaussian distribution of primary vertex ($\sigma = 30.0$ cm).

7. Solenoidal magnetic field till 1.2 Tesla is also foreseen in SPD with uniformity better than $\pm 2\%$ inside -900 mm $\leq z \leq 900$ 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 track seeds as input for constructing track candidate;
   - 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. do Kalman fit of remaining track candidates

6. write reconstructed track with the set of parameters to output root-file

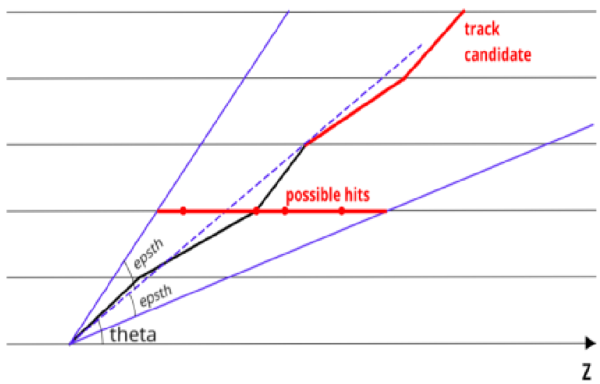
Present version of track reconstruction program is optimized for MAPS (4 layers) or DSSD (5 layers) option and only for tracks formed in primary vertex

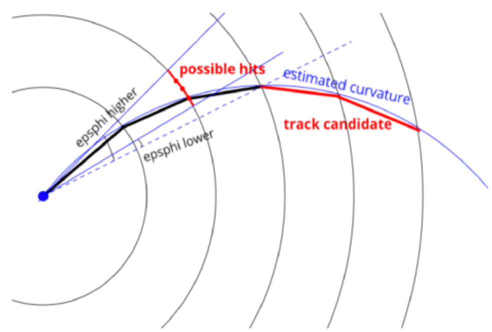
Constructing track seeds in vertex detector

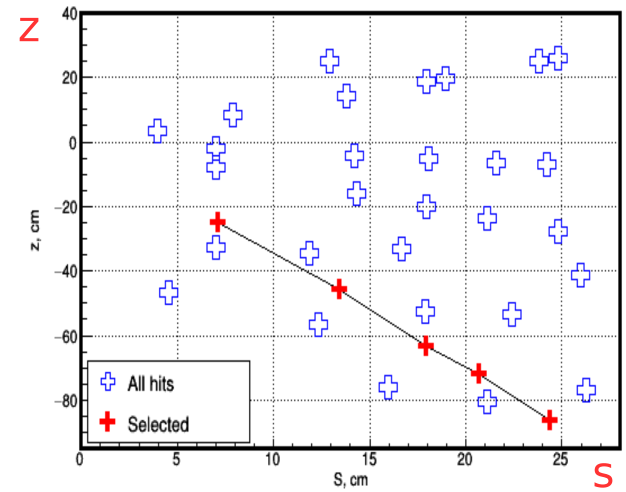
1. Charge particle trajectory in constant magnetic field is helix which can be described:
   a) in XY plane as circle with radius **$R = P_T / 0.3 \cdot B$** and
   b) z-coordinate is the function of arc length (s), **$z(s) = z_0 + s \cdot \tan\lambda$**, where $s = (\Phi - \Phi_0) \cdot R \cdot q$,
   Φ – azimuthal angle, λ – dip angle, z_0 and Φ_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 – coordinate

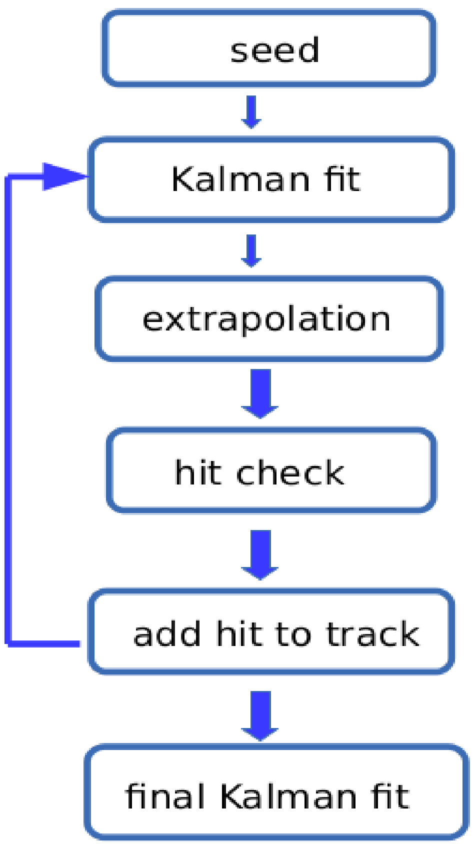
5. Do estimation of primary vertex z-position

6. Seed finding procedure provides the next track seed parameters:
   a) charge;
   b) curvature or radius;
   c) theta and phi angle;

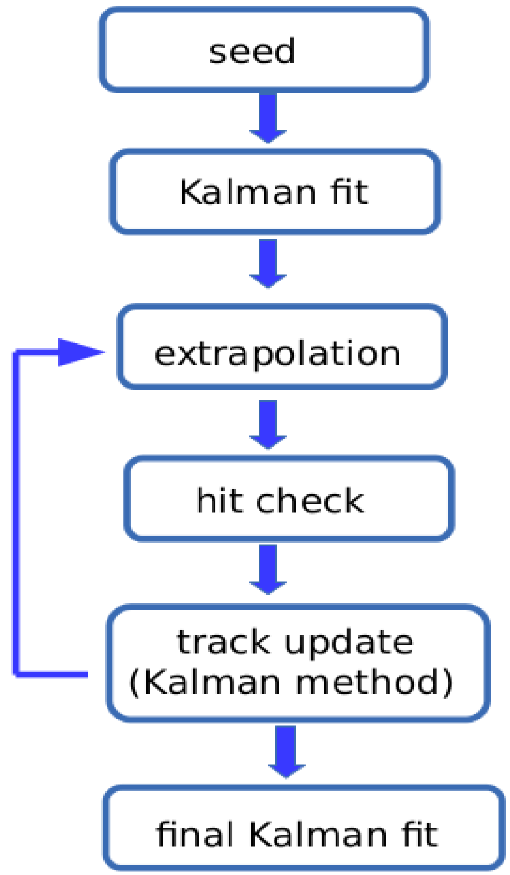
7. **Only tracks which contain 2 or more vertex hits can be reconstructed with this algorithm**

Scheme for adding straw hits

Full fit



Fast fit



Track seeds constructed from vertex detector hits are used as input for adding straw hits to track candidate. Two different procedures are implemented at the present time:

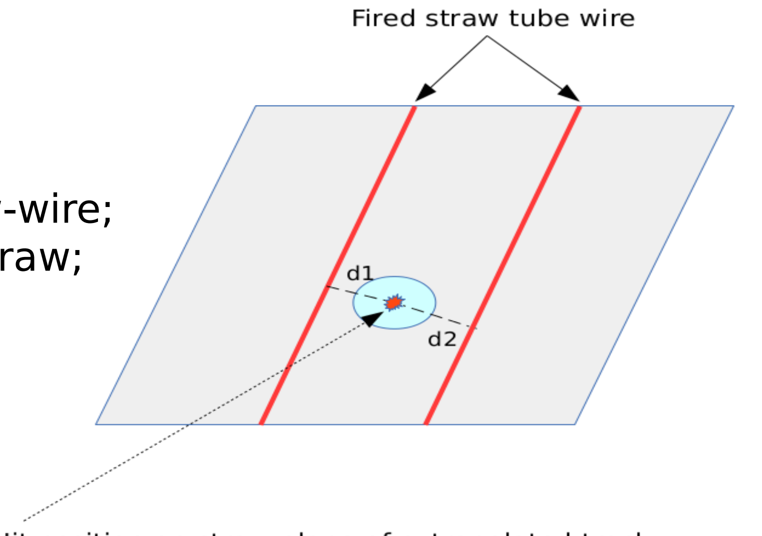
a) so called “full” fit – left scheme;
b) so called “fast” fit – right scheme (now used as default option).

Procedure of adding straw hits

1. Extrapolate track to some 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:

   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.



Hit position on straw plane of extrapolated track with error band;
d1, d2 – distance from hit position to fired wires

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) 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 do Kalman fit of remaining track candidates

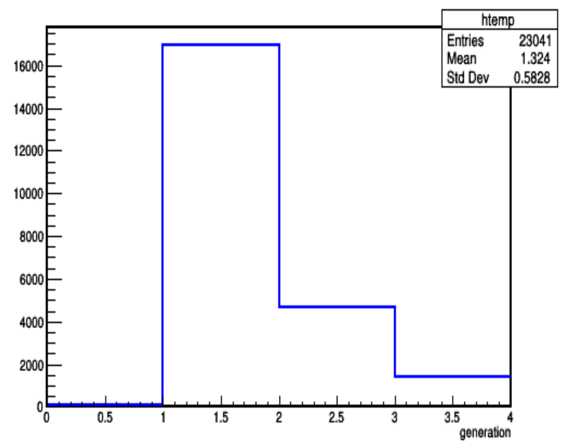
6. Write reconstructed track parameters in SpdTrackRC class with the same set of track parameters as for the usual SpdTrackMC

Track reconstruction (general points)

1. Minimum Bias events are simulated with Pythia8 at √s = 27 GeV (MAPS and DSSD options) with Gaussian distribution of primary vertex z-position (σ = 30.0 cm).

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

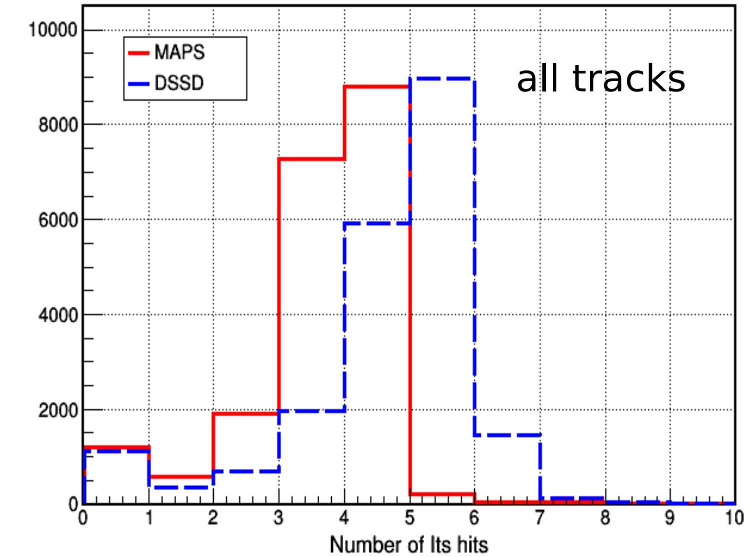
3. "Ideal" track reconstruction can be applied to the different particle generation level:
   a) 1-st generation - only primary particles from interaction vertex are considered;
   b) 2-nd generation – primary and secondary interactions and particles are considered;
   c) 3-d generation - primary, secondary and further interactions and particles are considered.

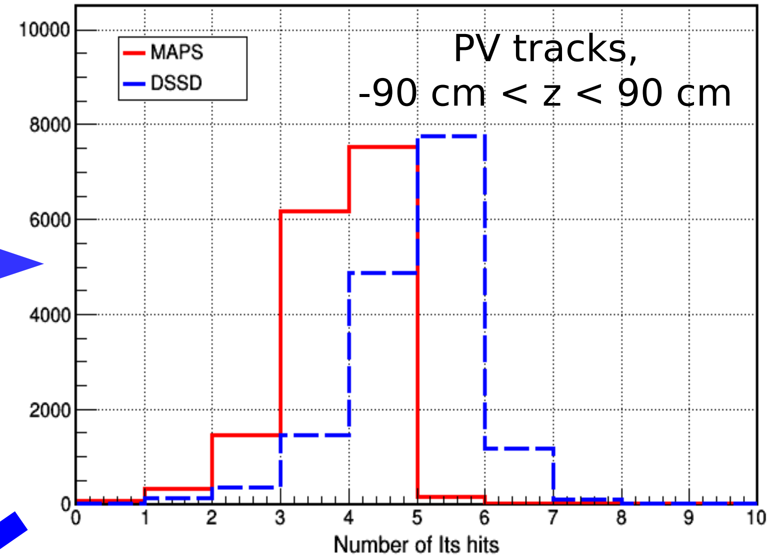


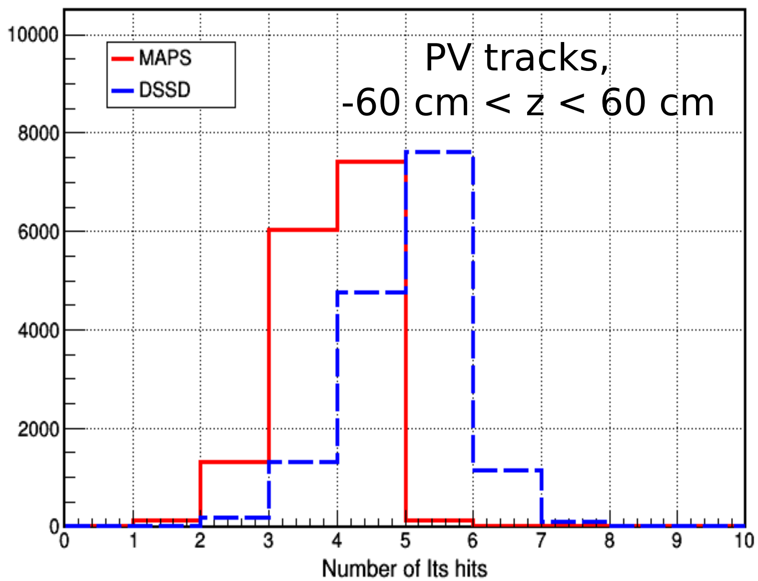
Number of "ideal" reconstructed track vs generation

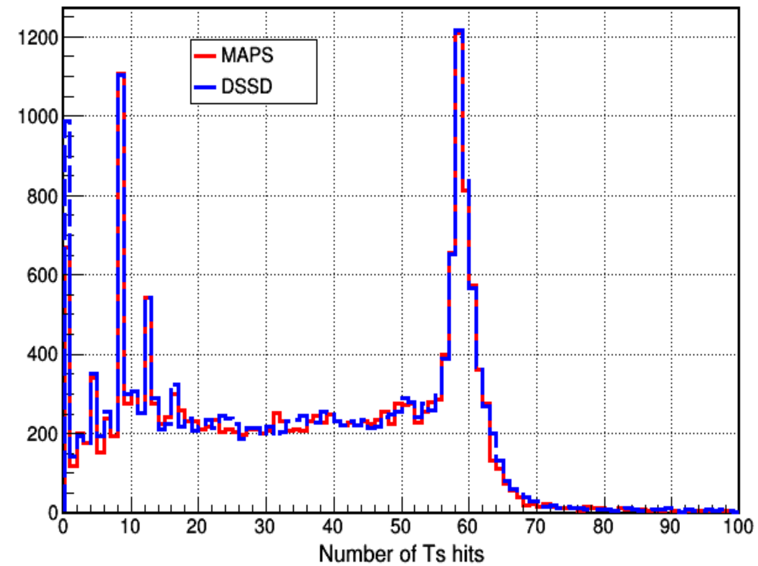
4. Time for reconstruction of Minimum Bias events (2.8 GHz notebook) for MAPS options:
   a) "ideal" fit (known MC hits) => ~6.0 sec/event (3 gen., Its hits >=0, Its+Ts hits >= 3)
   b) reco "fast" fit => ~4.2 sec/event (Its hits>=2, Its+Ts hits > 3)
   c) reco "full" fit => ~15.0 sec/event (Its hits>=2, Its+Ts hits > 3)
   d) reco Kalman tree => ~33.0 sec/event (Its hits>=2, Its+Ts hits > 3)
   e) **reconstruction time is strongly depended on applied conditions** =>
      (number of Its hits, generation level, minimum Pt and momentum of reconstructed track)

Number of vertex (Its) and straw (Ts) hits for simulated tracks









Printout example of reconstructed tracks

-I- <SpdMCEventHelper::PrintTracksRC> "Reco" tracks

ID	M(ID)	M(PDG)	NAME	V(ID)	FIT	NH	N(Its)	N(TsB)	N(TsEC)	NP
0	0	-211	pi-	0	yes	49	4	45	0	49
1	1	211	pi+	0	yes	40	4	36	0	40
2	2	211	pi+	0	yes	51	4	0	47	51
3	3	-211	pi-	0	yes	5	3	0	2	5
4	4	211	pi+	0	yes	50	3	0	47	50
5	5	211	pi+	0	yes	28	2	0	26	28
						223	20	81	122	223

-I- <SpdMCEventHelper::PrintTracks> "Ideal" reco tracks

ID	M(ID)	M(PDG)	NAME	V(ID)	FIT	NH	N(Its)	N(TsB)	N(TsEC)	NP
0	17	211	pi+	0	yes	51	4	0	47	51
1	24	211	pi+	0	yes	52	3	0	49	52
2	25	-321	K-	0	yes	3	3	0	0	3
3	28	211	pi+	0	yes	28	2	0	26	28
4	29	211	pi+	0	yes	42	4	38	0	42
5	30	-211	pi-	0	yes	52	4	48	0	52
6	55	13	mu-	7	yes	51	0	51	0	51
7	63	211	pi+	10	yes	51	3	0	48	51
8	64	-211	pi-	10	yes	50	3	0	47	50
						380	26	137	217	380

Secondary tracks

Track reconstruction (performance)

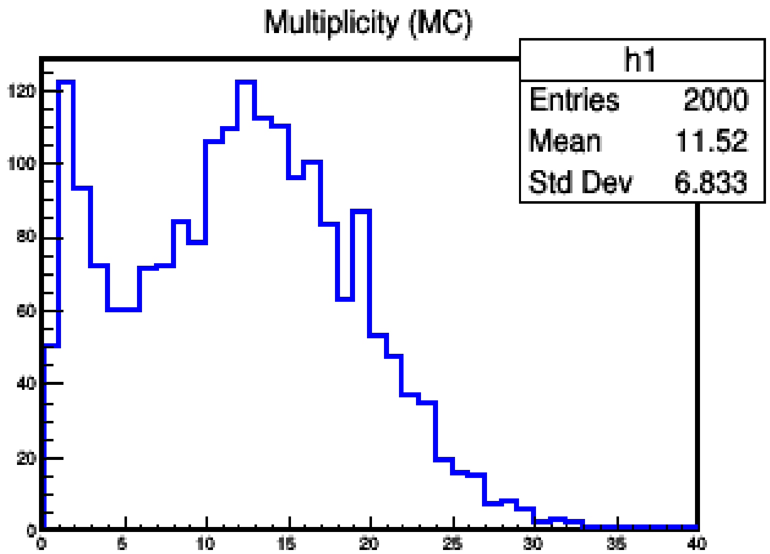
1. Procedure for estimation of track **reconstruction efficiency**:
   a) consider “ideal” reconstructed track;
   b) try to find among reconstructed tracks such track which can be considered as identical to “ideal” reconstructed track => that means this track has the some number common vertex hits (>=2) and also has > 50 % same “straw” hits.

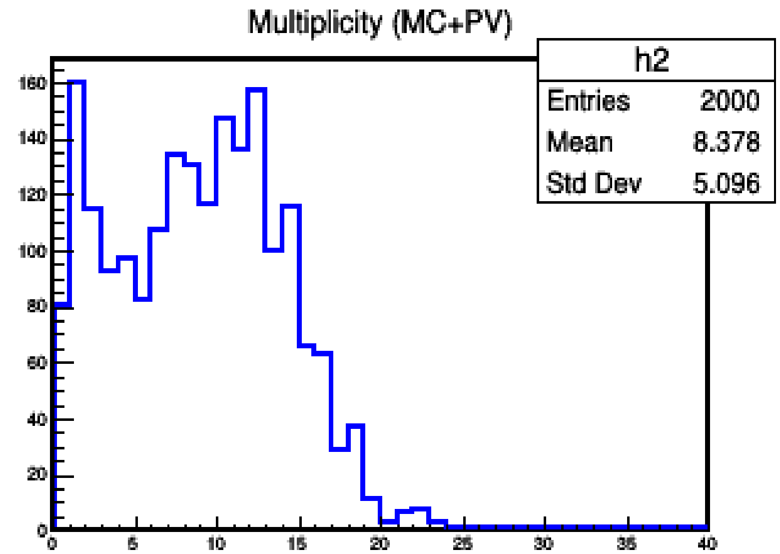
2. Estimation number of **clone track** at reconstruction:
   a) consider “ideal” reconstructed track;
   b) try to find among reconstructed tracks such track which can be considered as identical to “ideal” reconstructed track;
   c) if it was found two or more such tracks – this tracks are considered as clone tracks.

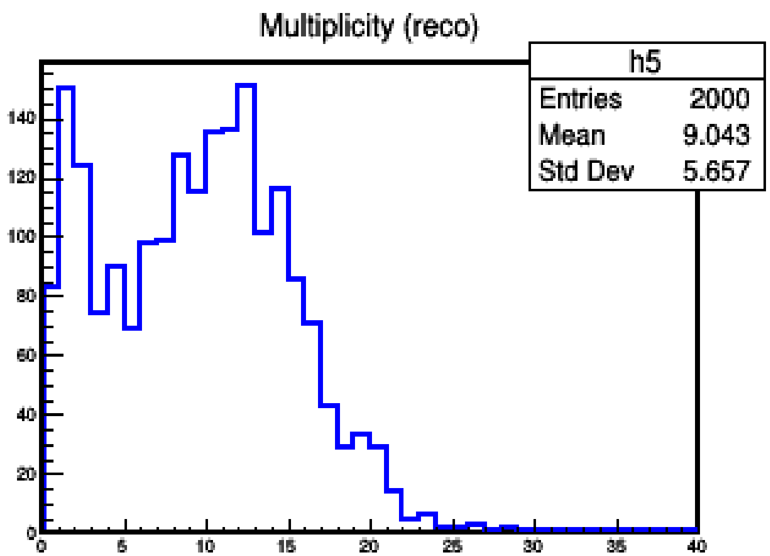
3. Procedure for estimation of **fake** reconstruction rate:
   a) consider reconstructed track;
   b) try to find among “ideal” reconstructed tracks such track which can be considered as identical to reconstructed track => that means this track has the some number of common vertex hits (>= 2) and also has > 50% same “straw” hits;
   c) if such “ideal” reconstructed track is not found => this reconstructed track is considered as fake track.

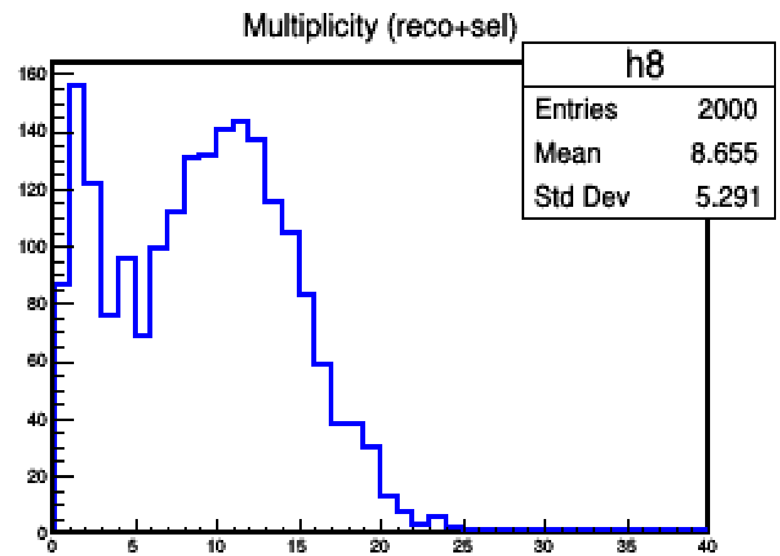
4. Next conditions are used for checking of the track reconstruction program:
   a) minimum **Pt > 25 MeV/c** and minimum **P > 75 MeV/c**;
   b) “ideal” reconstruction => **3 generation, Its hit >= 0, Its+Ts hits >= 3**;
   c) “real” reconstruction => **Its hits >=2, Its+Ts hits >= 3.**

Reconstructed track multiplicity (MAPS)

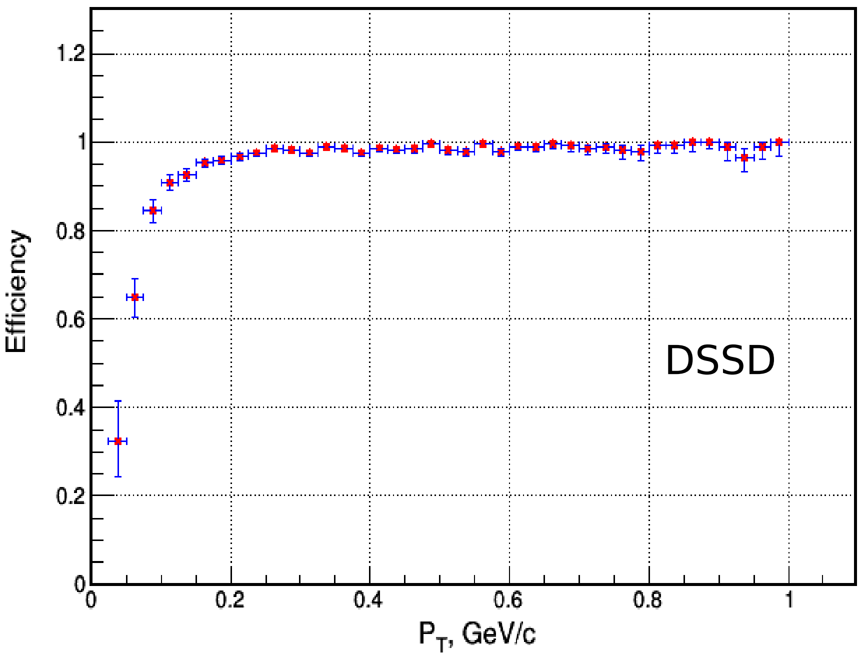


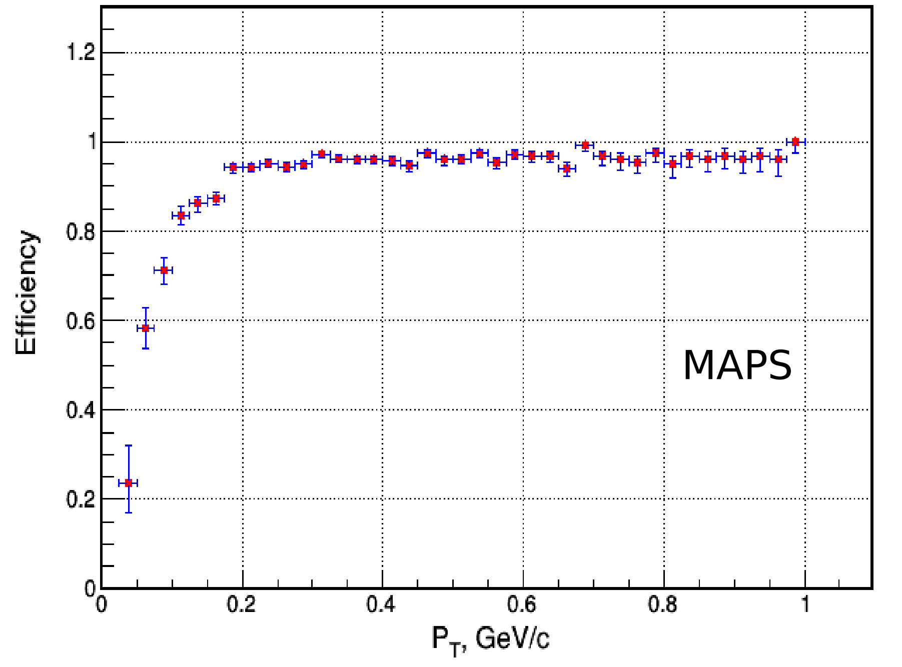






Track reconstruction efficiency (MB)

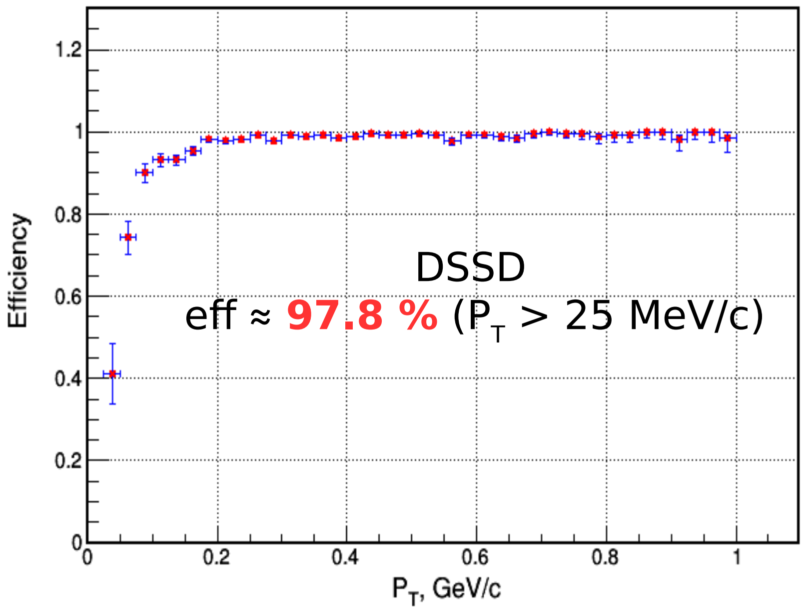


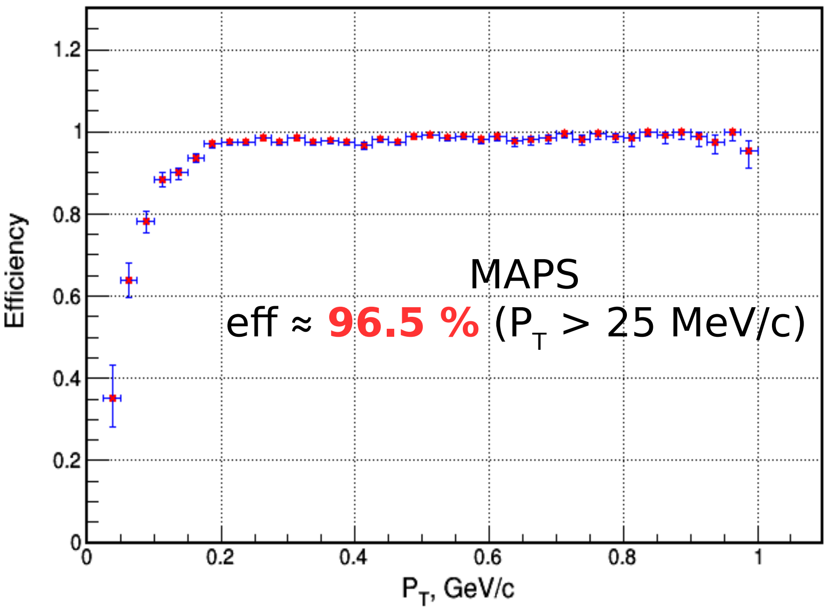


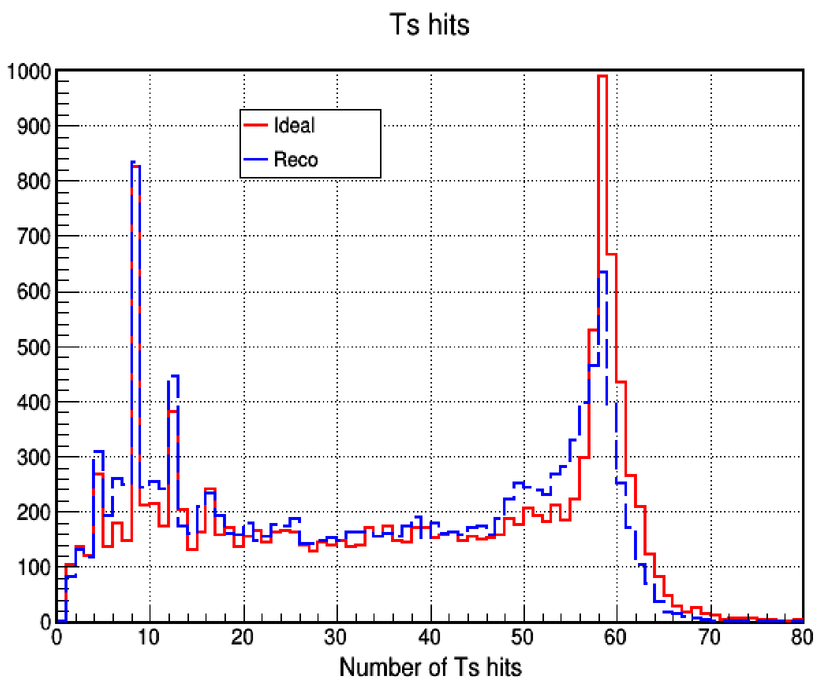
Integrated track reconstruction efficiency [%] for PV -60 cm < z < 60 cm

Pt (MeV)	> 25	> 50	> 100	> 150	> 200
MAPS	93.9	94.1	94.9	95.5	96.0
DSSD	97.0	97.2	97.7	98.0	98.3

Track reconstruction efficiency (MB) (z = 0.0 cm for PV)

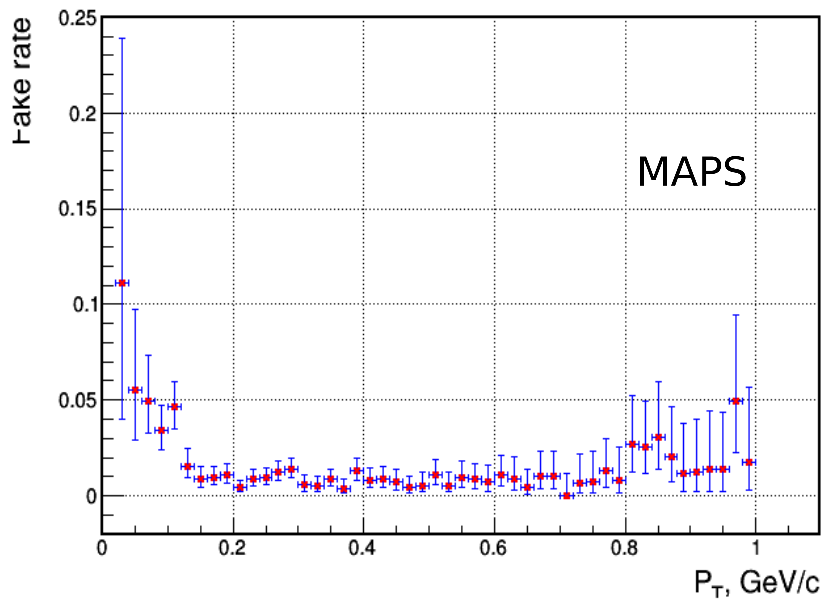






Track reconstruction efficiency is strongly depended on selection criteria !

Fake rate at track reconstruction (MB)



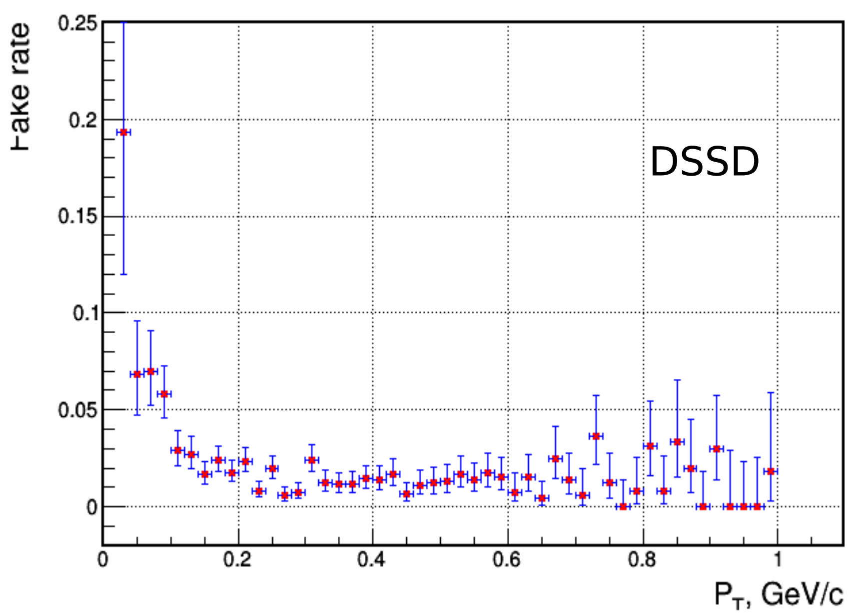
Total fake rate for P_T > 25 MeV/c

fake ≈ 1.3 %

Number of clone tracks:

a) MAPS ≈ **0.3 %**

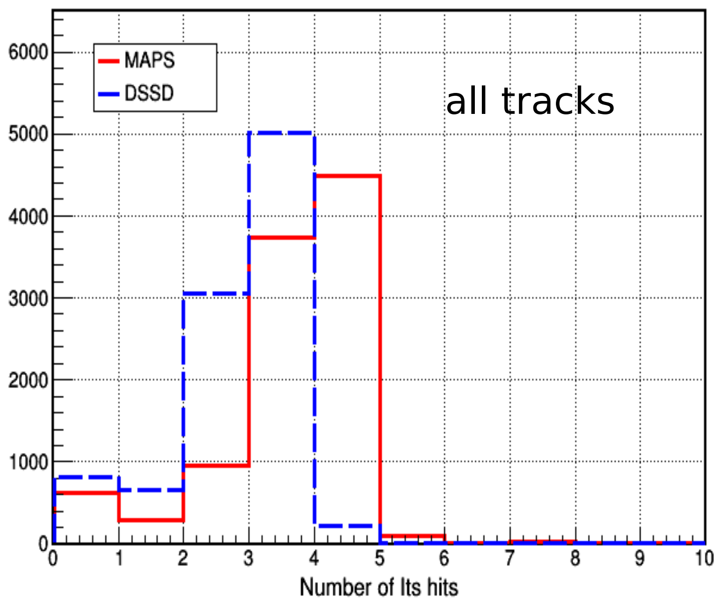
b) DSSD ≈ **0.6 %**

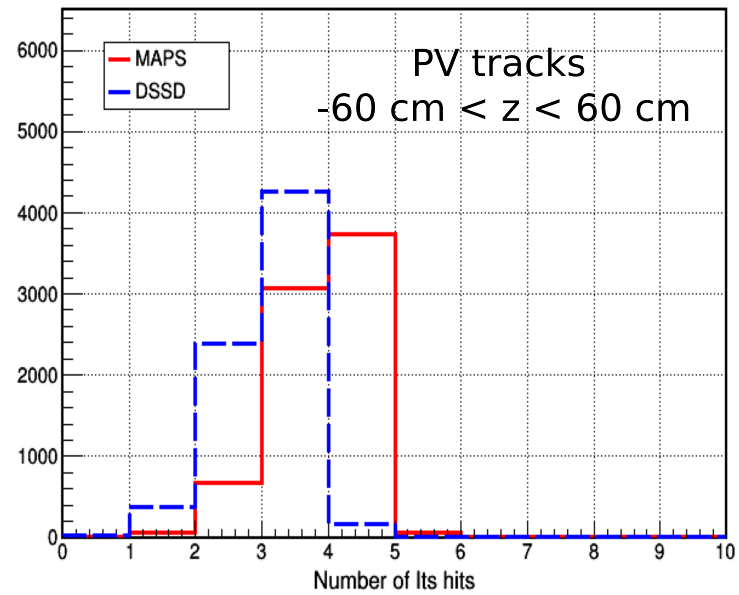


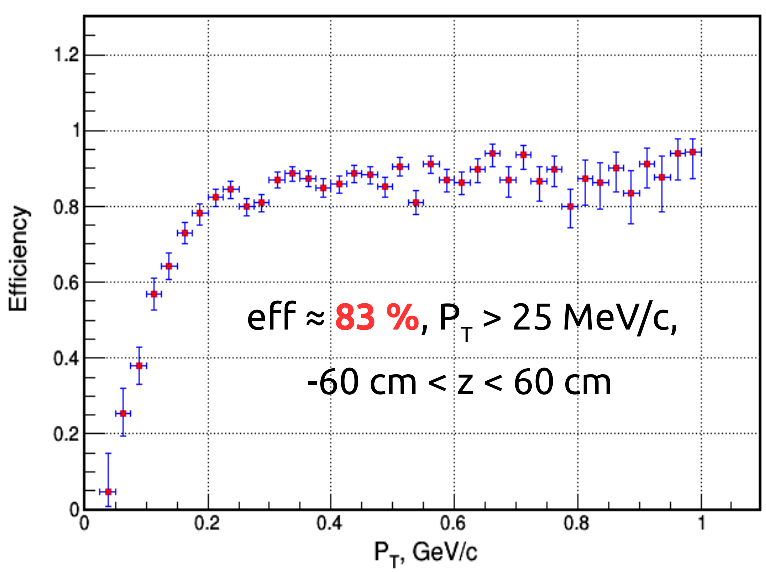
Total fake rate for P_T > 25 MeV/c

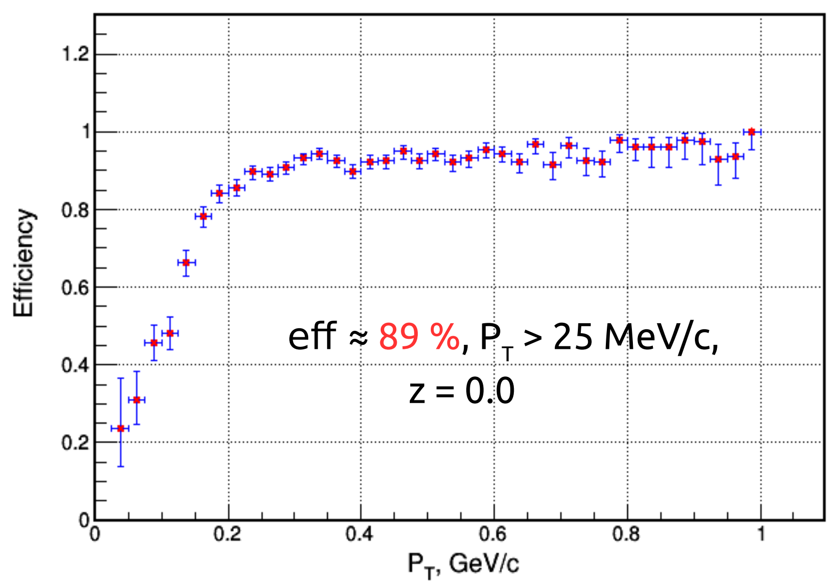
fake ≈ 2.0 %

Track reconstruction efficiency (MB) (DSSD, 3 layers









Summary and plans

1. 2-nd version of track reconstruction is ready for implementation in SPDroot

2. add new “fast” fit option for reconstruction algorithm

3. cleaning and simplification of code structure are done

4. **reconstruction algorithm shows good track reconstruction efficiency and low fake rate for MAPS and DSSD vertex detector options**

Future plans:

1. develop track reconstruction algorithm for primary vertex tracks with one or without vertex detector hits (important for 3 layers detectors)

2. develop the finding procedure for secondary tracks

3. add noise to vertex and straw detectors