SPD S&C meeting,
12 October 2021

Status of track reconstruction for SPD experiment (update)

V. Andreev

Track reconstruction (general scheme)

All changes are done inside SPDroot software

1. 1-step – event generation

2. 2-st step (simulation) => produce sim-hits using SPDroot with vertex and tracker

3. 3-d step => produce digi-hits (this options is now in SPDroot, contains hit coordinates with smearing for vertex and straw detectors)

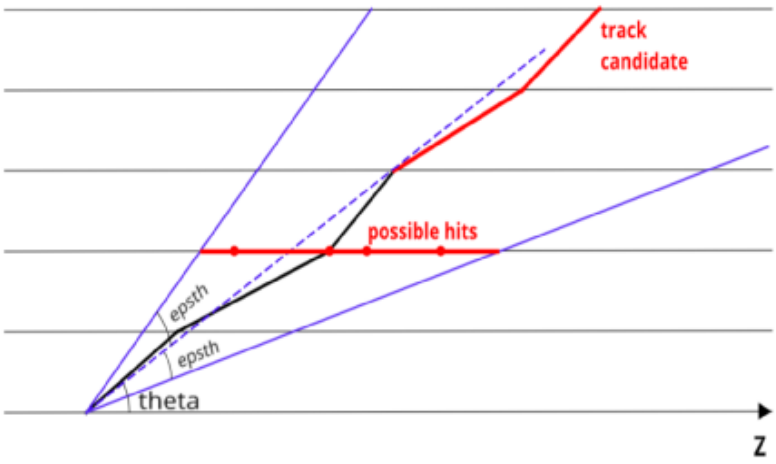
4. 4-th -step (pattern recognition) => produce track candidates with the set of vertex and straw hits

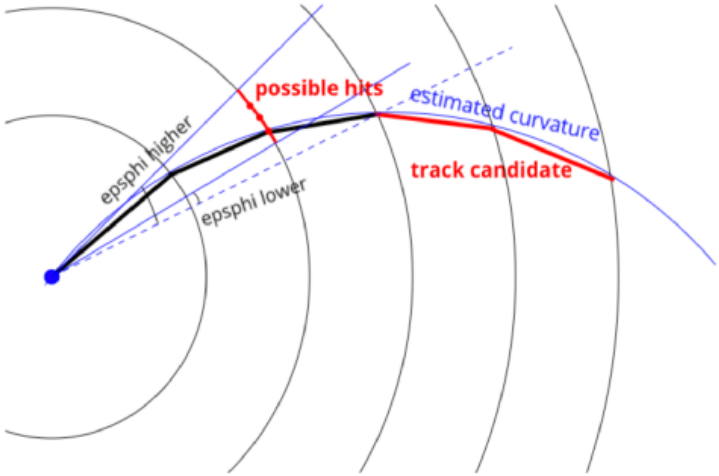
Pattern recognition means:
- construct track seed using hits in vertex detector;
- add consistently straw detector hits to track candidate;
- and finally create track candidate which contains as vertex and straw tracker hits.

5. final Kalman fit of track candidates

6. write reco-track to output root-file

Track seed in vertex detector





1. produce 2-points seed using combination of 2-hits on the different layers

2. then produce 3-points, 4-points and 5-points seeds adding additional point:

   a) use 2 points for primary estimation of theta direction and then check 3-d point (4-th or 5-th points) – all points should be inside some delta theta range

   b) use 3-points for curvature estimation (for 2-points seed use also zero point(0,0))

   c) next 3-d point (4-th or 5-th points) should be also inside some delta phi range

3. next merging procedure is applied:

   a) if 3-points seed contains all points of 2-points seed => 2-points seed is removed
   b) if 4-points seed contains all points of 3-points seed => 3-points seed is removed
   c) if 5-points seed contains all points of 4-points seed => 4-points seed is removed

Track seed in vertex detector (2)

4. z-coordinate of primary vertex is very important parameter for track reconstruction => clustering algorithm is applied for z-coordinate estimation which can produce several “primary” vertexes

5. check all seeds on compatibility with the set of “primary” vertexes

6. finally produce => 2-points, 3-points, 4-points and 5-points seeds

7. this seed finding procedure is done with separate Spd-task which produces the new SpdTrackSeed and SpdTrackSeedpar classes as output with the next information:

   a) estimated charge
   b) estimated radius (=> or Pt)
   c) estimated theta
   d) estimated phi (phi in 1-st seed point)
   e) estimated z-coordinate of primary vertex

Pattern recognition
(add straw detector hits)

New Spd task SpdRCTrackFinder is created for track reconstruction which uses vertex seeds as input data

1. do Kalman fit of vertex seed and find track candidate parameters

2. extrapolate track candidate to virtual cylinder with radius 48.0 cm (minimum radius of tracker barrel module) and find module in barrel or endcap in which track candidate hits

3. extrapolate track to 1-st straw layer of this tracker module and find hit points on the plane

4. check distance and chi2 between hit point and fired straw wire on this plane

5. add “good” straw hit to track candidate points, update track parameters (do new Kalman fit) and then extrapolate track to the next straw layer

6. if 2 or more straw hits on one layer are consisted with the track candidate => the new track candidate is created, all hits are copied to the new track candidate and then extrapolation and fitting procedure are applied for all new candidates =>

so called Kalman tree method

Pattern recognition (2)
(add straw detector hits)

7. finally, as the result “big” number of track candidates are produced which contains vertex and straw detector hits

8. then merging procedure is applied for track candidates which were produced from the same vertex seed:

   a) check common number of straw hits
   b) if common number of hits is more than some value (50%) -> remove track candidate with less number of straw hits, or
   c) if number of straw hits is equal → remove track candidate with greater chi2

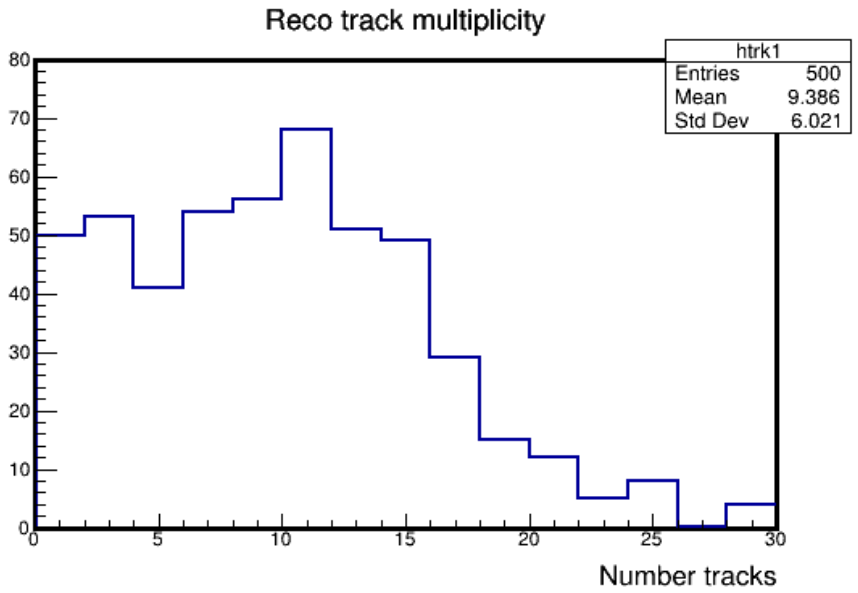
9. do final Kalman fit for remaining track candidates

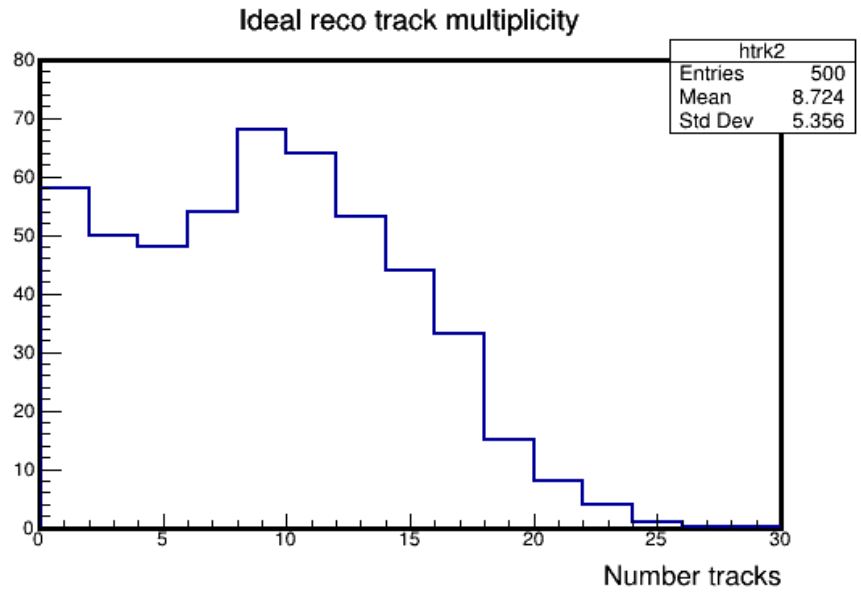
10. finally write reconstruction track parameters in SpdTrackRC

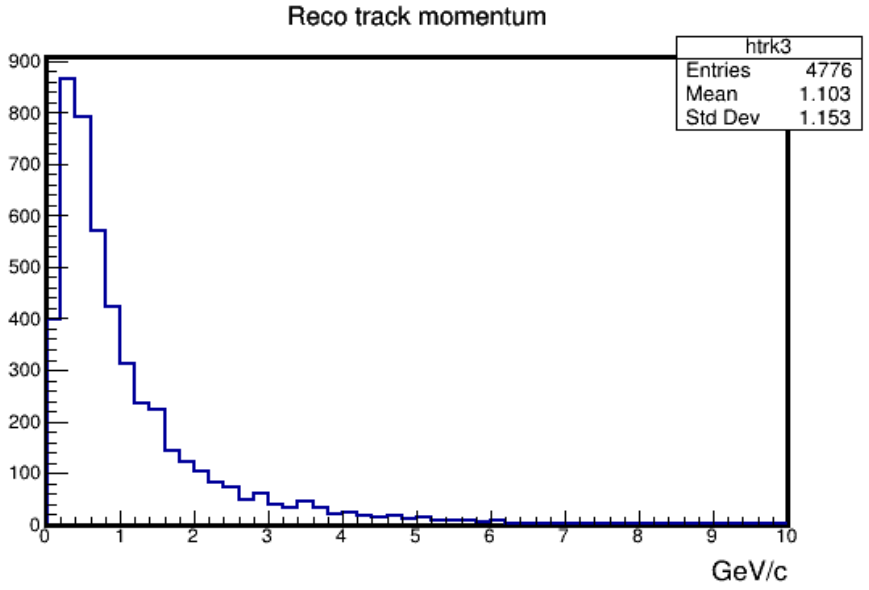
Result:

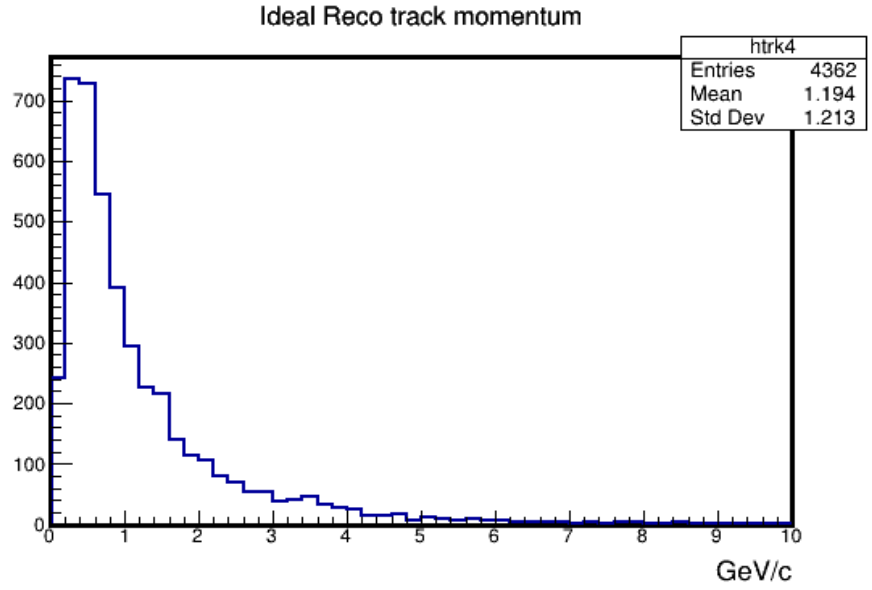
a) simulate 500 Minimum Bias events
b) do ideal pattern recognition (use MC information) => run standard fitting procedure for tracks and use this result as reference
c) run new track reconstruction task and compare result with reference ideal data
d) for each ideal track try to find the corresponding reconstructed track and compare track parameters

Results (1)

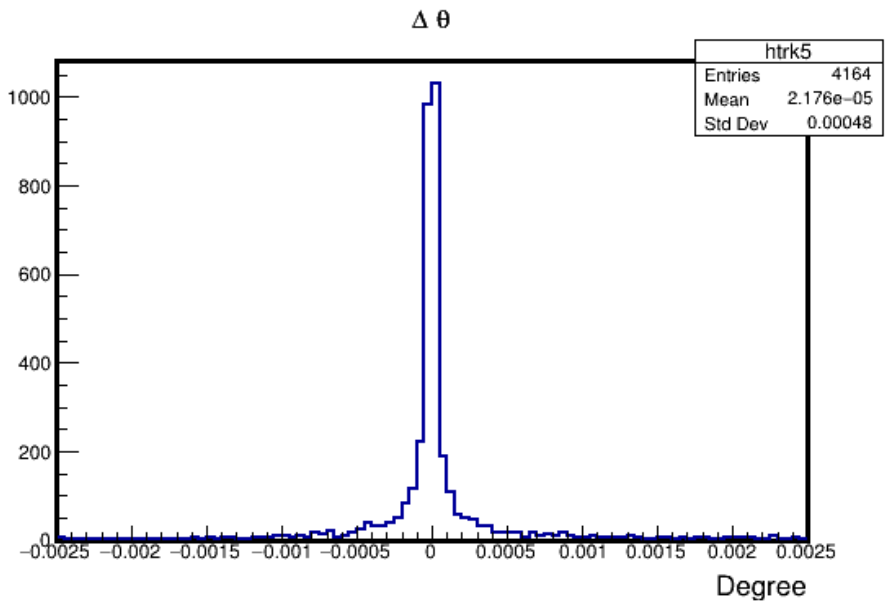

Reco track multiplicity
htrk1
Entries 500
Mean 9.386
Std Dev 6.021
Number tracks

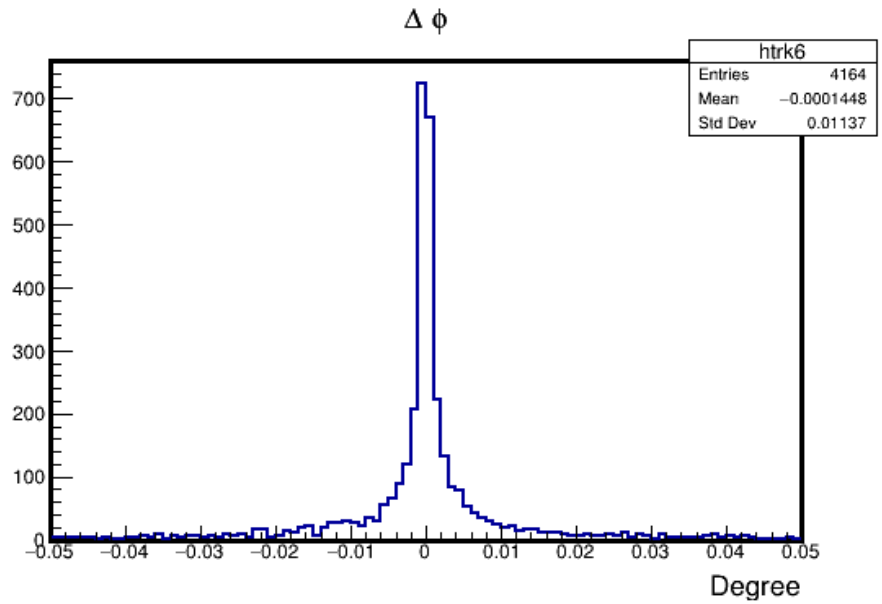

Ideal reco track multiplicity
htrk2
Entries 500
Mean 8.724
Std Dev 5.356
Number tracks

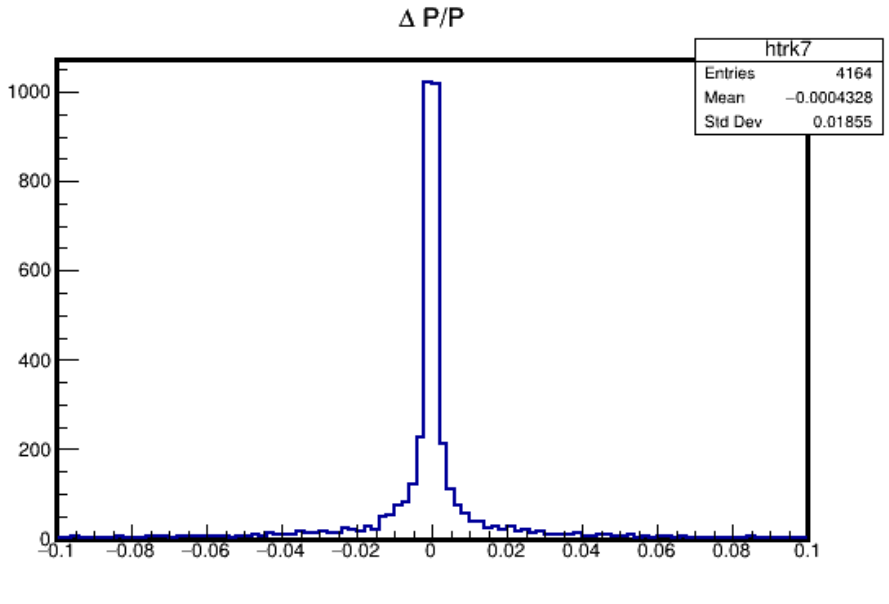

Reco track momentum
htrk3
Entries 4776
Mean 1.103
Std Dev 1.153
GeV/c

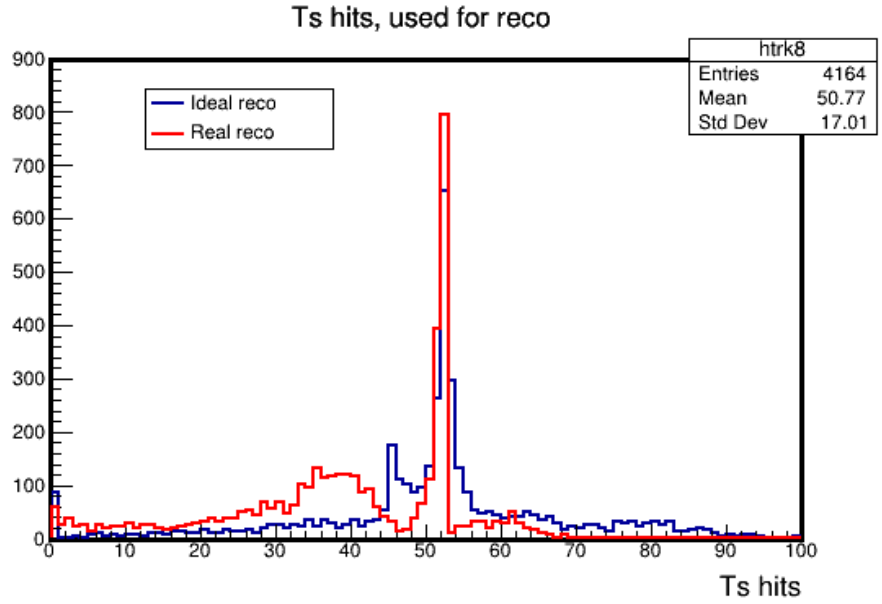

Ideal Reco track momentum
htrk4
Entries 4362
Mean 1.194
Std Dev 1.213
GeV/c

Results (2)



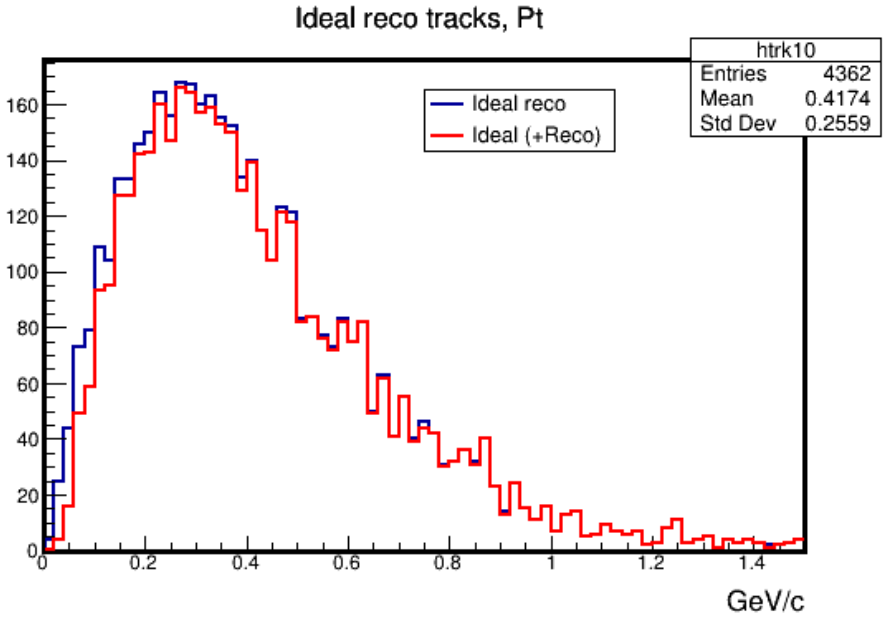


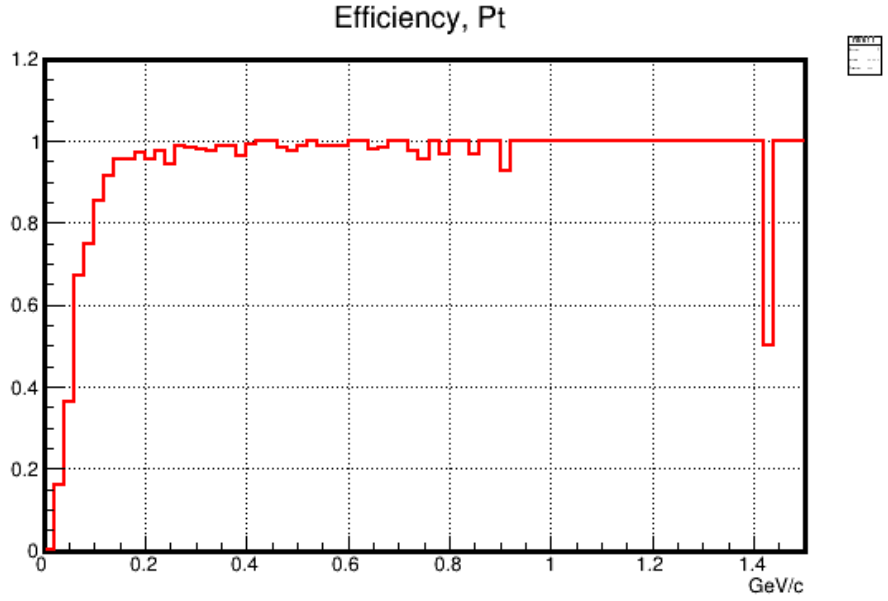


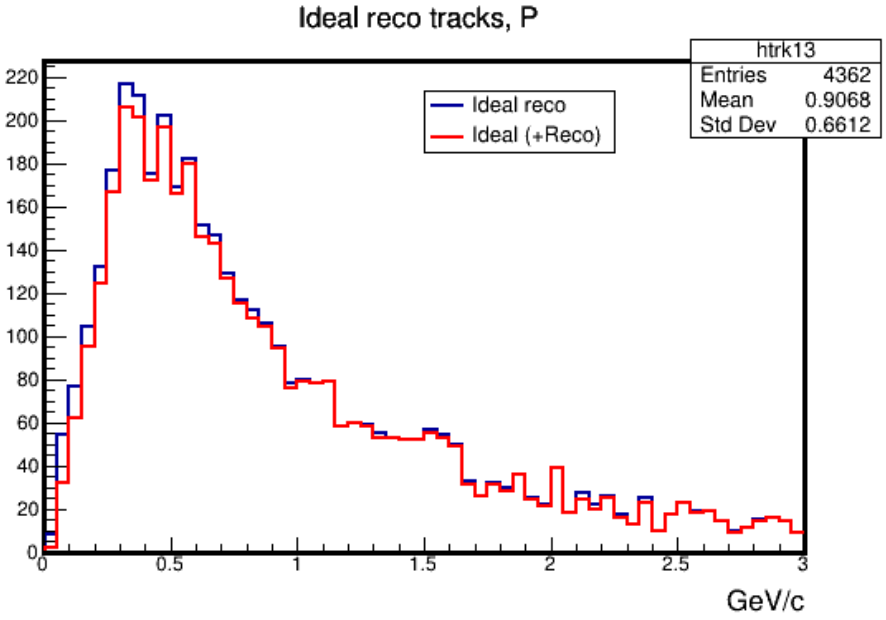


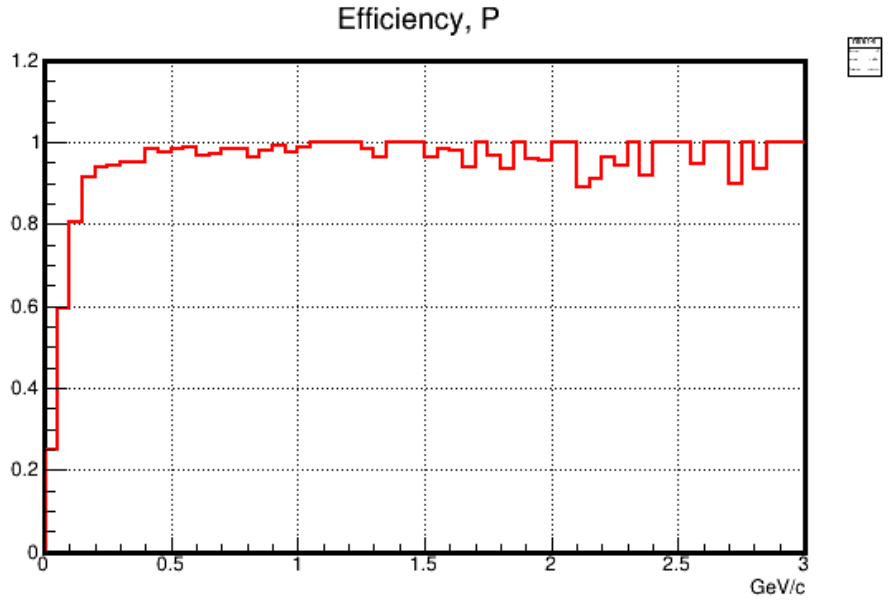
1. difference between ideal and reconstructed track parameters

Results (3)



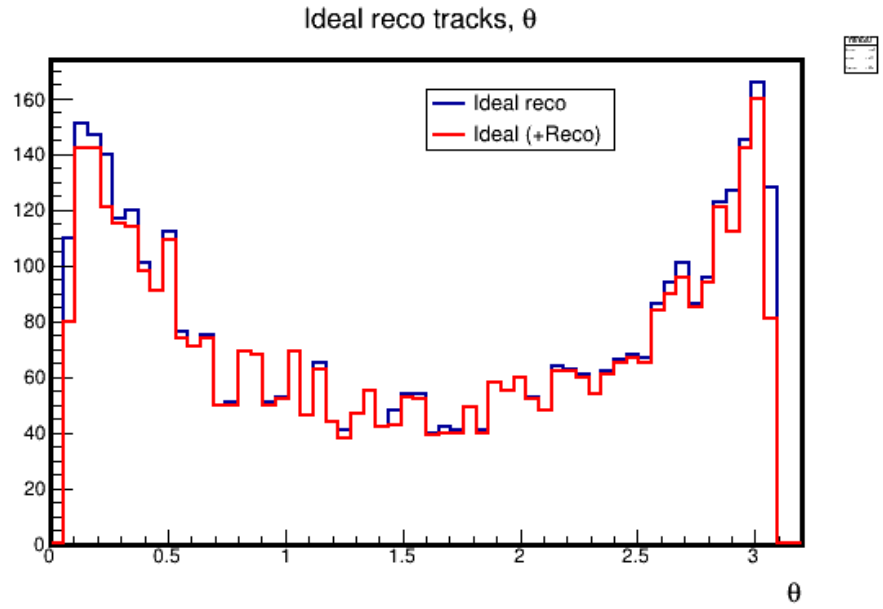


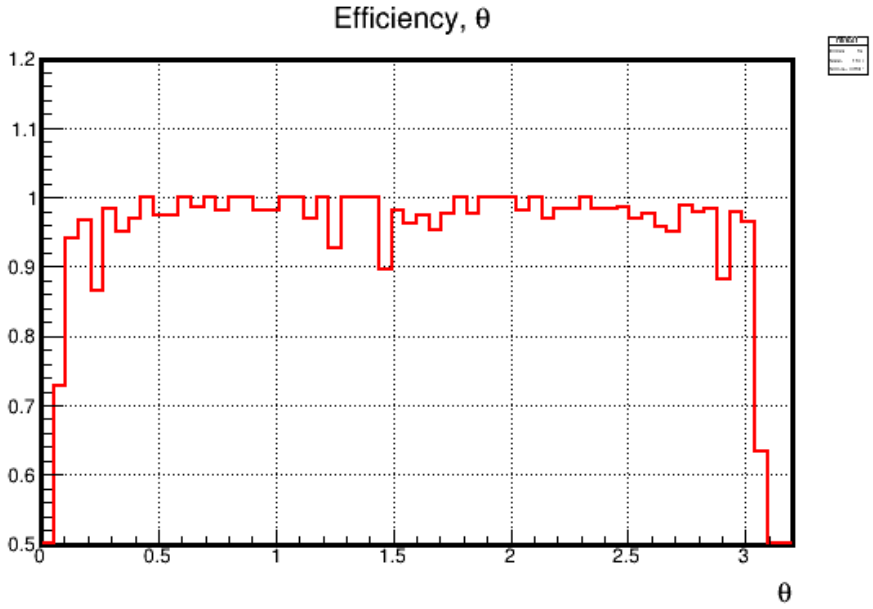


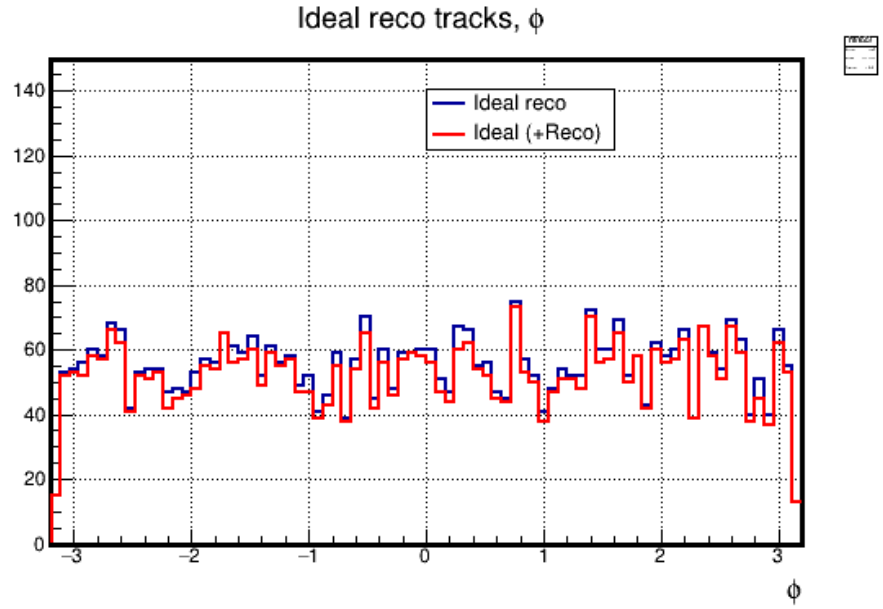


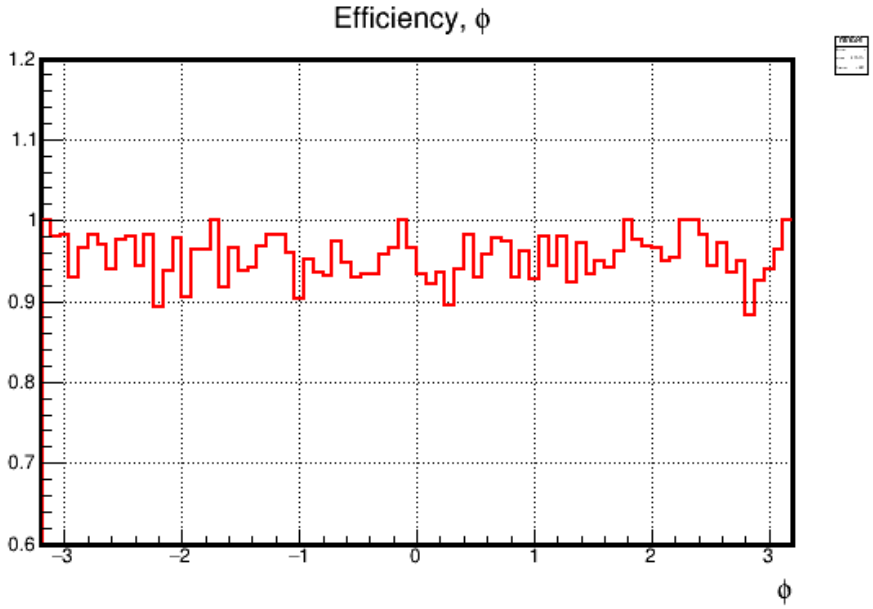
1. use ideal track fit as reference => Ideal reco (blue line)
2. ideal track which is matched with reconstructed track => Ideal (+Reco) (red line)

Results (3)


Ideal reco tracks, θ
Ideal reco
Ideal (+Reco)
θ


Efficiency, θ
θ


Ideal reco tracks, ϕ
Ideal reco
Ideal (+Reco)
ϕ


Efficiency, ϕ
ϕ

Summary

1. version of track reconstruction is ready and works (full chain)

2. time for track reconstruction is ~12 times greater than for ideal track fit
   (~25 sec/event with comparison ~2 sec/event on my notebook)

3. debugging and tuning procedure is underway