



SPD Physics Weekly meeting  
29 November 2022

Proposal for changes in  
primary vertex reconstruction  
structure

V. Andreev (LPI, Moscow)

## Introduction

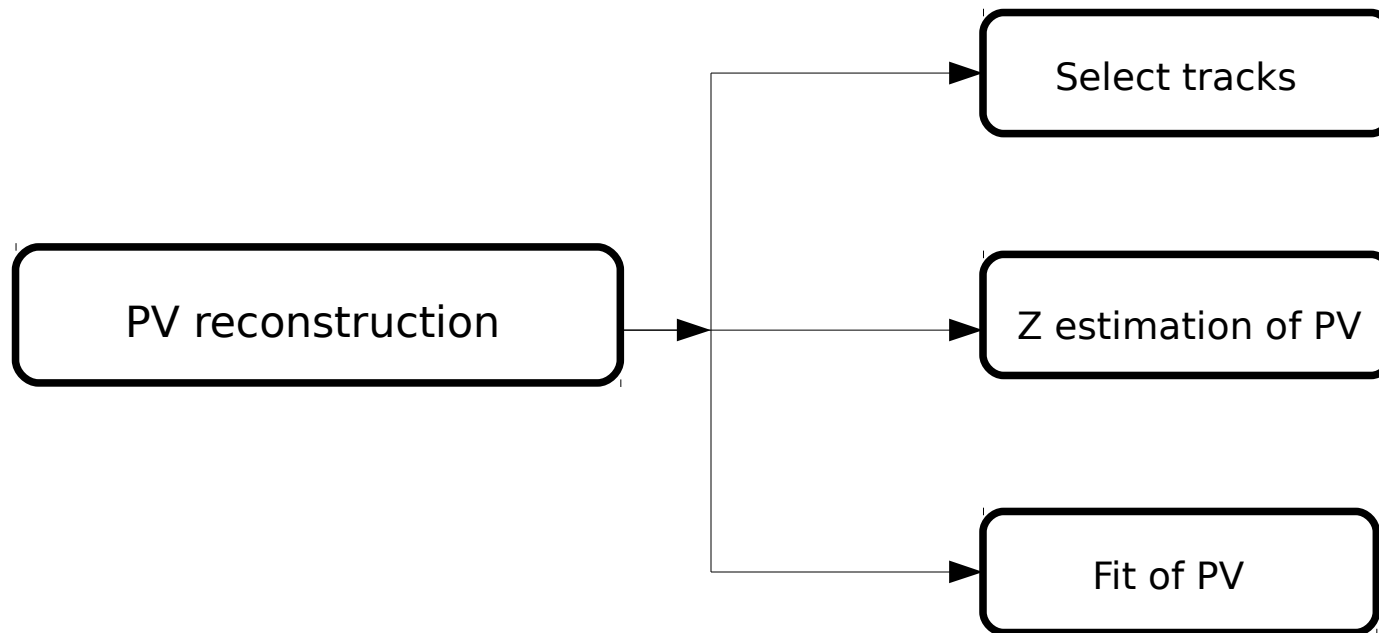
Main idea of these changes **to have the possibility to re-run** some reconstruction process during user physical analysis:

- now it is possible => to reconstruct the secondary decay particles with the different selection cuts (to do with KFParticle package and findDecayV0.C example);
- purpose => to reconstruct primary vertex with new set of selected tracks;
- need to change the structure of PV reconstruction program for this purpose.

### **Primary vertex reconstruction has 2 separate sub-tasks:**

- estimation position of z-coordinate for primary vertex (very important for SPD as we have broad distribution of z-coordinate );
- primary vertex fit (usually use the Kalman filter algorithm).

## Proposal for PV reconstruction



Present version of primary vertex reconstruction => one process with track selection, estimation of z-coordinate and fit of PV.

Proposal - inside this task to split this PV reconstruction process on 3 different sub-processes (or methods), in order to have the possibility to use each sub-process (**maybe except track selection**) outside of the main reconstruction program - in user analysis code as example.

No changes in algorithm itself is foreseen.

## PV reconstruction in user code

### In user analysis code (standard PV reconstruction):

a) select set of tracks => produce vector with tracks `std::vector<SpdTrackMC*>`  
and vector with particle code `std::vector<int>`;

b) `SpdRCVericesFinder fPVfinder` - initialize object;

```
fPVfinder.FitPrimaryVertex(std::vector<SpdTrackMC*>, std::vector<int>) - fit PV;  
fPVfinder.GetVertex();  
fPVfinder.GetCov();
```

### Now there is possibility to provide PV reconstruction using KFparticle package in user code:

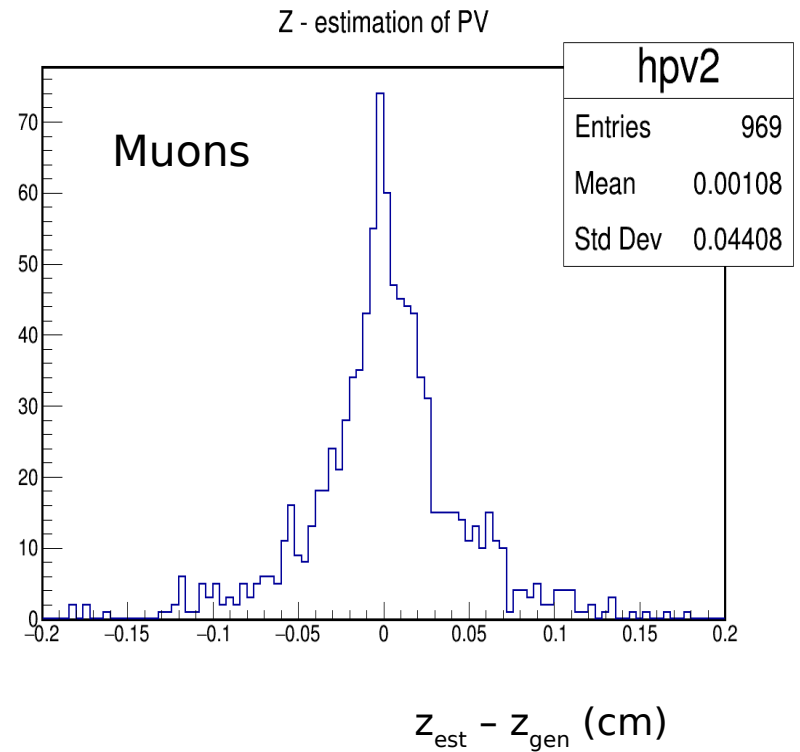
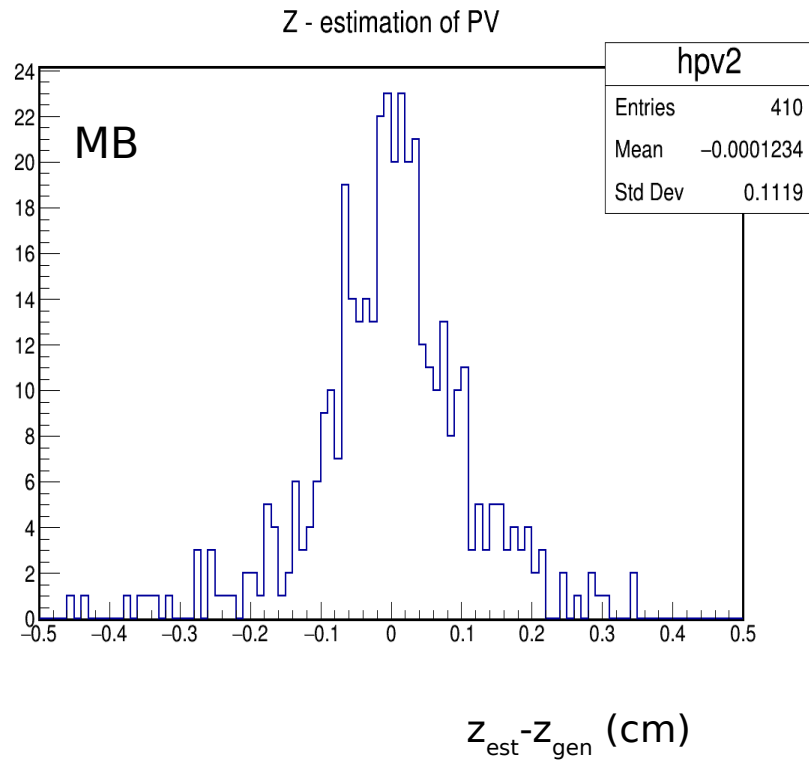
a) select set of tracks => produce `std::vector<KFParticle>` and vector with particle code `std::vector<int>`

b) `KFParticleTopoReconstructor fTopoReconstructor` - initialize object;

`z_pv = fPVfinder.FindPrimaryVertexCand(input_tracks, p_codes)` - z of PV estimation;

```
fTopoReconstructor.SetZofPrimaryvertex(z_pv);  
fTopoReconstructor.Init(KFParticles, p_codes);  
fTopoReconstructor.ReconstructPrimVertex();  
fTopoReconstructor.GetPrimKFVertex(0).GetX();  
fTopoReconstructor.GetPrimKPVertex(0).getChi2();
```

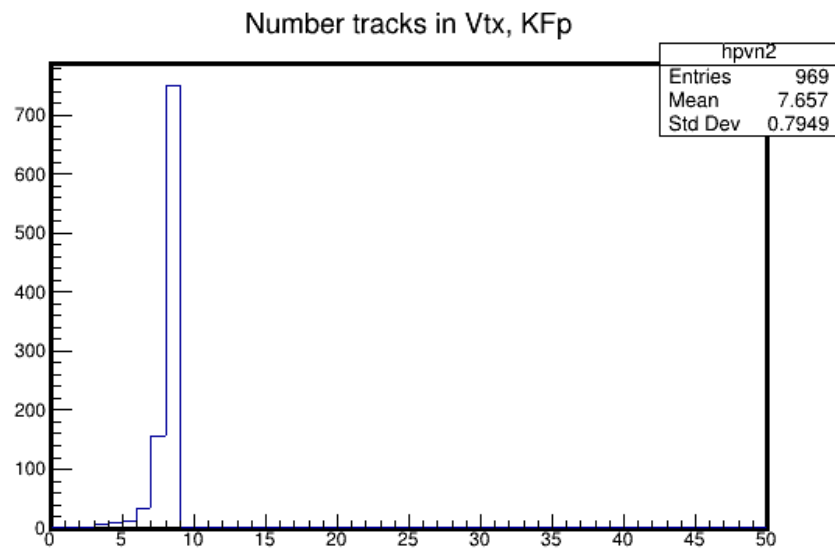
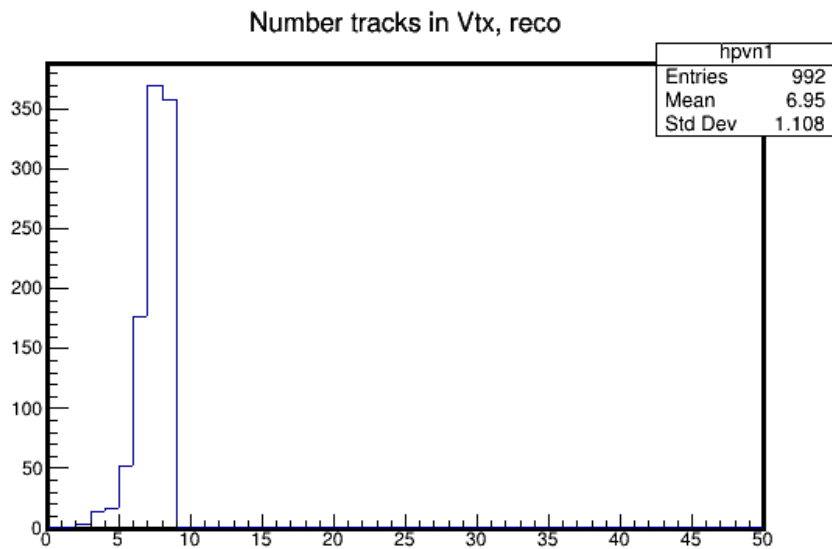
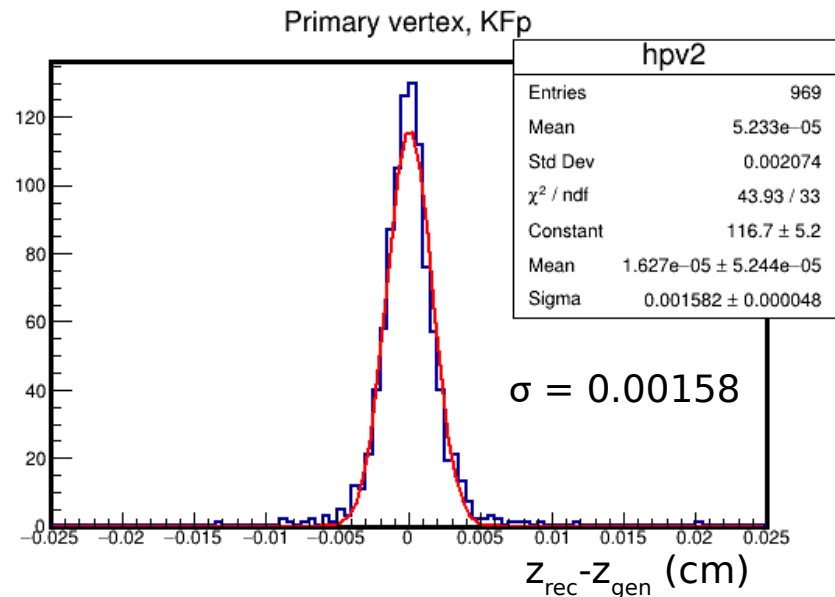
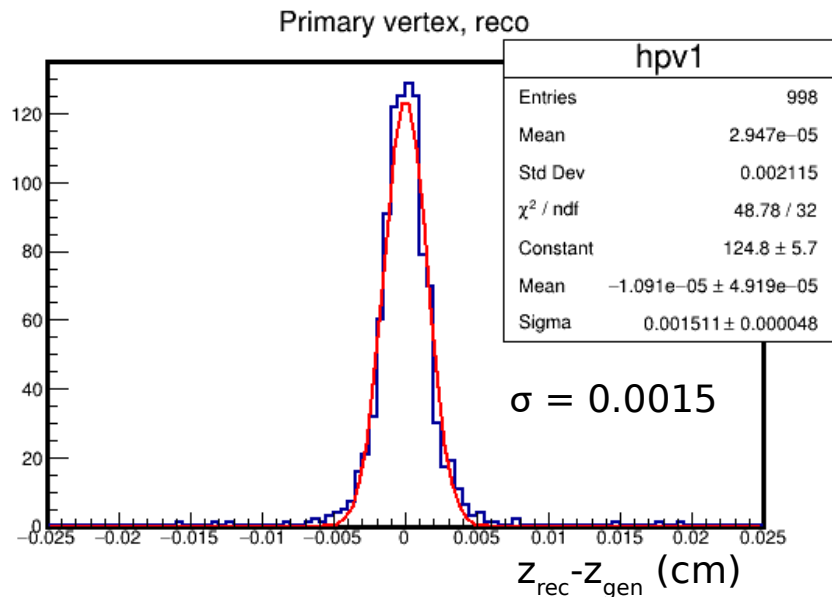
## Z estimation



It were generated, simulated and reconstructed 1000 events with 8 muons of 1 GeV/c uniform distributed in theta and phi angles and 500 events of Minimum Bias with Gaussian vertex position distribution  $\sigma_z = 30$  cm,  $\sigma_{x,y} = 0.1$  cm and for MAPS version of vertex detector

```
z_est = fPVfinder.FindPrimaryVertexCand(input_tracks, p_codes)
```

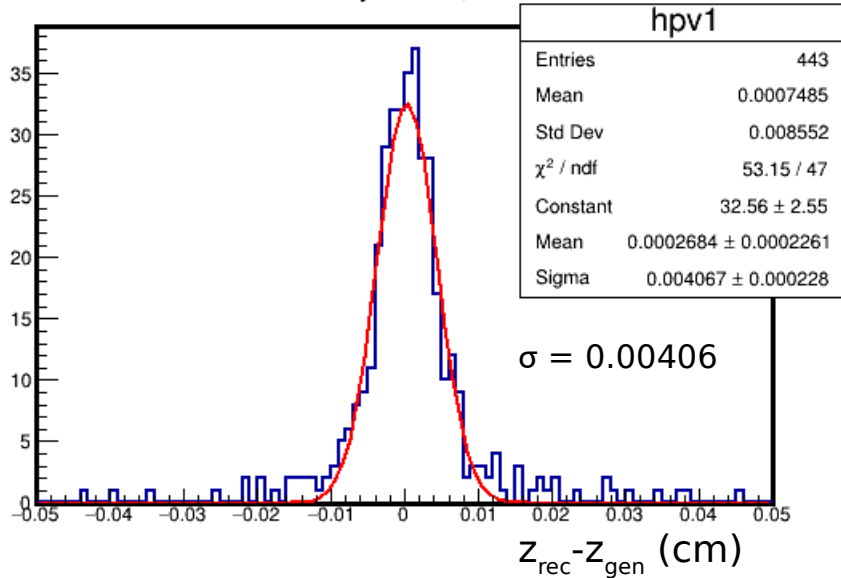
# Preliminary results (muon)



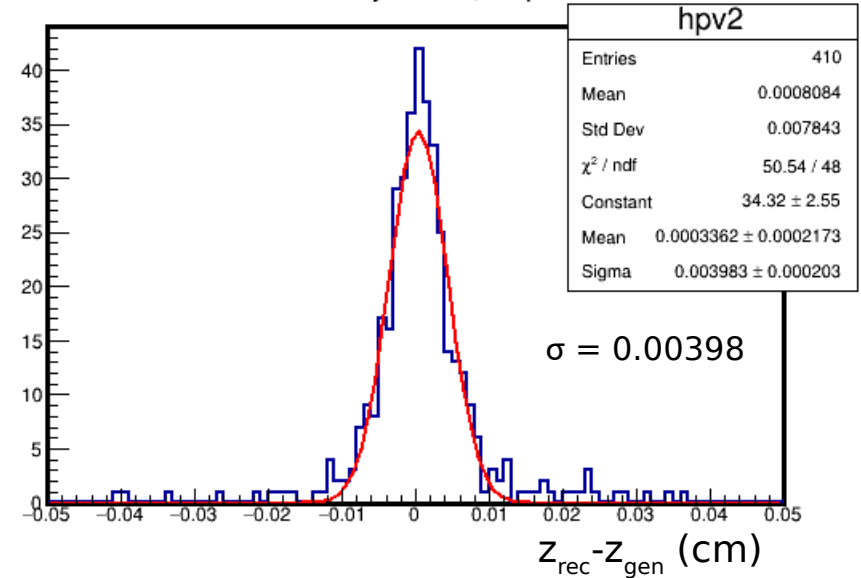
1000 events sample of 8 muons for MAPS version of vertex detector

# Preliminary results (MB)

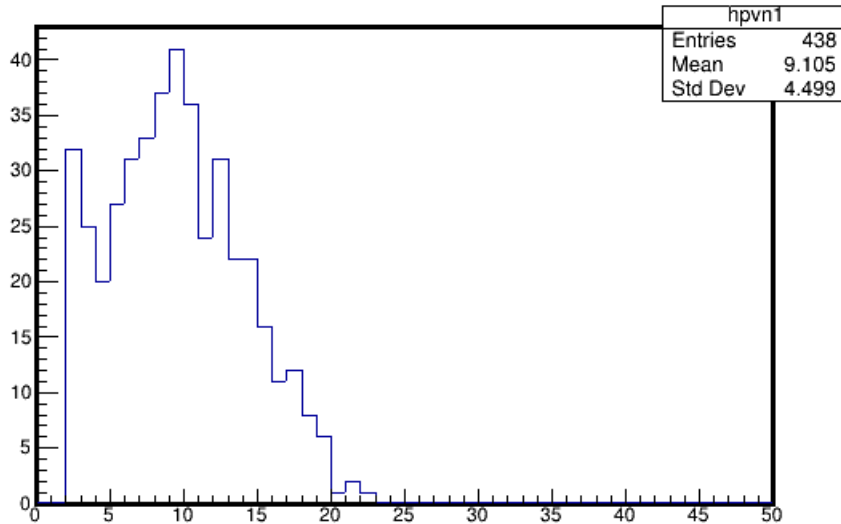
Primary vertex, reco



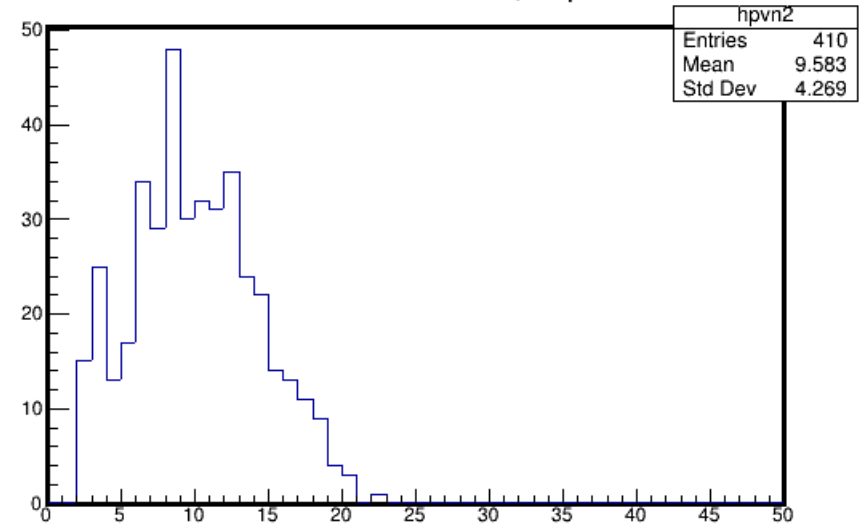
Primary vertex, KFp



Number tracks in Vtx, reco



Number tracks in Vtx, KFp



500 events sample of Minimum Bias events for MAPS version of vertex detector

## Summary and plans

1. new possibility to re-run primary vertex reconstruction is demonstrated
2. need some tuning and checking
3. foresee to add this option to SPDroot in ??