Track reconstruction efficiency and momentum resolution

Malgorzata Gumberidze

IPN Orsay, France

The reference track set

Assessment of track finding efficiency requires, a definition of a reference set of tracks that an ideally performing algorithm should find.

- Particles arising from secondary interactions in the material that normally are not within the physics scope but merely an obstacle and should be excluded.
- Particles travelling outside of the geometrical acceptance.
- Particles straddling the border of a detector and, e.g. traversing only a small number of tracking layers, regarded as outside of the design tracking volume. A typical convention may be to regard particles which traverse O(**80%**) of the nominal tracking layers as constituents of the reference set.

The definition of the reference set can then be regarded as a definition of effective geometrical acceptance

$$\varepsilon_{geo} = \frac{N_{ref}}{N_{total}}$$

with N - the number of particles of interest in the reference set and in total

Rep. Prog. Phys. 67 (2004) 553–622

Track efficiency

The definition of the track finding efficiency requires a criterion which specifies whether a certain particle has been found by the algorithm or not.

There are two rather different concepts:

1. ***Hit matching.*** Analyzing the simulated origin of each hit in the reconstructed track using the MC truth information. If the qualified majority of hits, (e.g. at least 70%), originates from the same true particle, the track is said to reconstruct this particle.

   This method is stable in the limit of very high track densities, but it requires the MC truth information to be mapped meticulously through the whole simulation.

2. ***Parameter matching.*** The reconstructed parameters of a track are compared with those of all true particles. If the parameter sets agree within certain limits the corresponding track is said to reconstruct this particle.

   This method requires less functionality from the simulation chain, but there is a danger of accepting random coincidences between true particles and artifacts from the pattern recognition algorithm.

Track efficiency

The definition of the track finding efficiency requires a criterion which specifies whether a certain particle has been found by the algorithm or not.

2. Parameter matching. The reconstructed parameters of a track are compared with those of all true particles. If the parameter sets agree within certain limits the corresponding track is said to reconstruct this particle.

This method requires less functionality from the simulation chain, but there is a danger of accepting random coincidences between true particles and artifacts from the pattern recognition algorithm.

$$\varepsilon = \frac{N_{ref}^{reco}}{N_{ref}}$$

Criteria for the particle tracks

- ✓ It need to be primary particle. -> condition for the reference particle
- ✓ Reconstructed particle need to have MC index.
- ✓ If we have more than 1 reconstructed particle per MC track.
  We take particle which has reconstructed momentum closest to the truth MC one.

One should also control the abundance of non-reference tracks which are reconstructed ($N^{reco}_{non-ref}$), normally by the relation:

$$\frac{N^{reco}_{non-ref}}{N_{total} - N_{ref}} << \varepsilon_{reco}$$

should hold, otherwise the reference criteria might be too strict.

How to do it technically …

How to get inputs for our studies:

cd $VMCWORKDIR/macro/pid

1. **root –l -q 'run_sim_sttcombi_pgun.C(100, 13, 0.2, 3)'**
2. **root –l –q run_digi_sttcombi.C**
3. **root –l –q run_reco_sttcombi.C**
4. **root –l –q run_pid_stt.C**

- **100** events
- **1μ** per event
- momentum **0.2** ÷ **3** GeV/c

what we get is

points_sttcombi.root
params_sttcombi.root
} **1.**

digi_sttcombi.root → **2.**

reco_sttcombi.root → **3.**

pid_sttcombi.root → **4.**

+ Fair… .root

Analysis code (1)

```
{
   // setting input names of different levels from the simulation
 TString inSimFile    = "points_sttcombi.root"
 TString inPidFile    = "pid_sttcombi.root";

 // opening input file and getting corresponding tree
 TFile *inFile = TFile::Open(inPidFile,"READ");
 TTree *tree=(TTree *) inFile->Get("cbmsim") ;
 tree->AddFriend("cbmsim",inSimFile);

 // Monte Carlo TCloneArray
 TClonesArray* mc_array=new TClonesArray("PndMCTrack");
 tree->SetBranchAddress("MCTrack", &mc_array);

 // PidCandidate TCloneArray
 TClonesArray* cand_array=new TClonesArray("PndPidCandidate");
 tree->SetBranchAddress("PidChargedCand", &cand_array);

 // output files definition
 TFile *out = TFile::Open("nt_efficiencyStudy.root","RECREATE");

 // definition of the cotrol ntuple
 TNtuple *nt_eff = new TNtuple
                        ("nt_eff","nt_eff","momMC:thetaMC:phiMC:momReco:thetaReco:phiReco");
```

Analysis code (2)

```
Int_t nevents =  tree->GetEntriesFast();

for (Int_t j= 0; j< nevents; j++)        { // looping over entries
      tree->GetEntry(j);
       Float_t mc_mom  = 0,  mc_theta = 0, mc_phi  = 0;    // variables declaration
       Float_t rec_mom = 0, rec_theta = 0, rec_phi = 0;
      for (Int_t mc = 0; mc < mc_array->GetEntriesFast(); mc++)
       {  // loop over monte carlo tracks
         PndMCTrack *mctrack = (PndMCTrack*)mc_array->At(mc);
         if (mctrack->GetMotherID()!=-1) continue;  // only primary particles
          mc_mom   = mctrack->GetMomentum().Mag();
          mc_theta = mctrack->GetMomentum().Theta()*TMath::RadToDeg();
          mc_phi   = mctrack->GetMomentum().Phi()*TMath::RadToDeg();

            Int_t cand_mult = 0; // counter for the candidate per event
            for (Int_t pp=0; pp<cand_array->GetEntriesFast(); pp++)
                  { // loop over PID candidate (Charge canditate)
                   PndPidCandidate *pidCand = (PndPidCandidate*)cand_array->At(pp);
                    if (pidCand->GetMcIndex()!=mc) continue;    // only with an index to MC
                     if ( (cand_mult==0) || ((cand_mult>0)
                                 && (fabs(rec_mom-mc_mom)>
                               fabs(pidCand->GetMomentum().Mag() - mc_mom))) ) {
                                  // only one reconstructing track corresponding to one MC track
```

Analysis code (3)

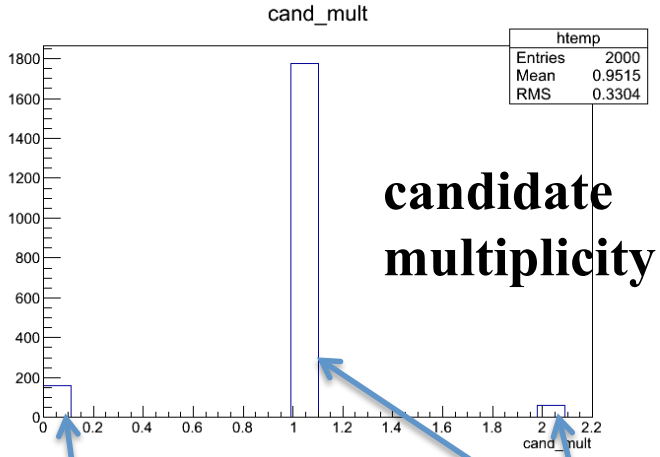
```
                rec_mom = pidCand->GetMomentum().Mag();
                rec_theta  = pidCand->GetMomentum().Theta()*TMath::RadToDeg();
                rec_phi    = pidCand->GetMomentum().Phi()*TMath::RadToDeg();
            } // end of if on cand_mult
            cand_mult++;    // increasing of counter to  candidate multiplicity per event
        } // end of loop on PidCand
    } // end of loop on MC

    Float_t var_eff[] = { mc_mom,mc_theta,mc_phi, rec_mom, rec_theta,rec_phi };
    nt_eff->Fill(var_eff);
} // end of loop on events

out->cd();
nt_eff->Write();
out->Save();
} // end of program
```

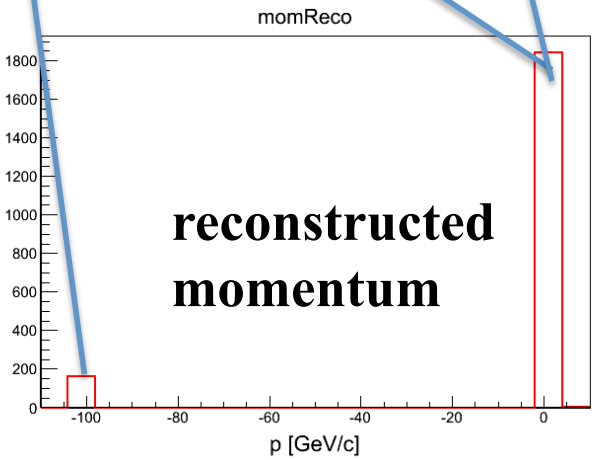
Ntuple … distributions …

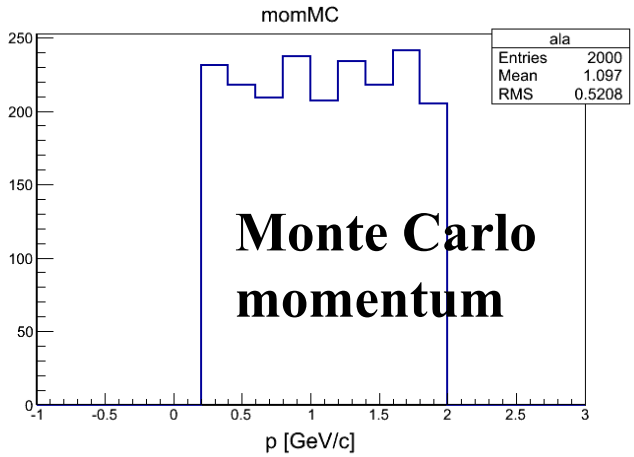
Terminal — root.exe — 84×35

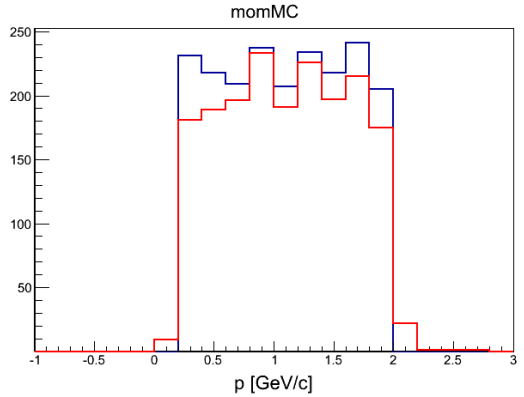
root.exe | bash | bash | bash | efte

```
root [2] nt_eff->Print()
******************************************************************************
*Tree    :nt_eff    : nt_eff                                                  *
*Entries :     2000 : Total =           60258 bytes  File  Size =      44510 *
*        :          : Tree compression factor =   1.29                       *
******************************************************************************
*Br    0 :momMC     : Float_t                                                 *
*Entries :     2000 : Total  Size=       8546 bytes  File Size  =       7121 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.13     *
*.............................................................................*
*Br    1 :thetaMC   : Float_t                                                 *
*Entries :     2000 : Total  Size=       8556 bytes  File Size  =       7277 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.11     *
*.............................................................................*
*Br    2 :phiMC     : Float_t                                                 *
*Entries :     2000 : Total  Size=       8546 bytes  File Size  =       7523 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.07     *
*.............................................................................*
*Br    3 :momReco   : Float_t                                                 *
*Entries :     2000 : Total  Size=       8556 bytes  File Size  =       6880 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.17     *
*.............................................................................*
*Br    4 :thetaReco : Float_t                                                 *
*Entries :     2000 : Total  Size=       8566 bytes  File Size  =       6970 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.16     *
*.............................................................................*
*Br    5 :phiReco   : Float_t                                                 *
*Entries :     2000 : Total  Size=       8556 bytes  File Size  =       7236 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.12     *
*.............................................................................*
*Br    6 :cand_mult : Float_t                                                 *
*Entries :     2000 : Total  Size=       8566 bytes  File Size  =        730 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=  11.07     *
*.............................................................................*
root [3]
```









Efficiency ….

2000 – Monte Carlo tracks

1820 – reconstructed track for each MC

$$\varepsilon = \frac{1840}{2000} = 0.92 \Rightarrow 92\%$$

global number
average over all θ, p

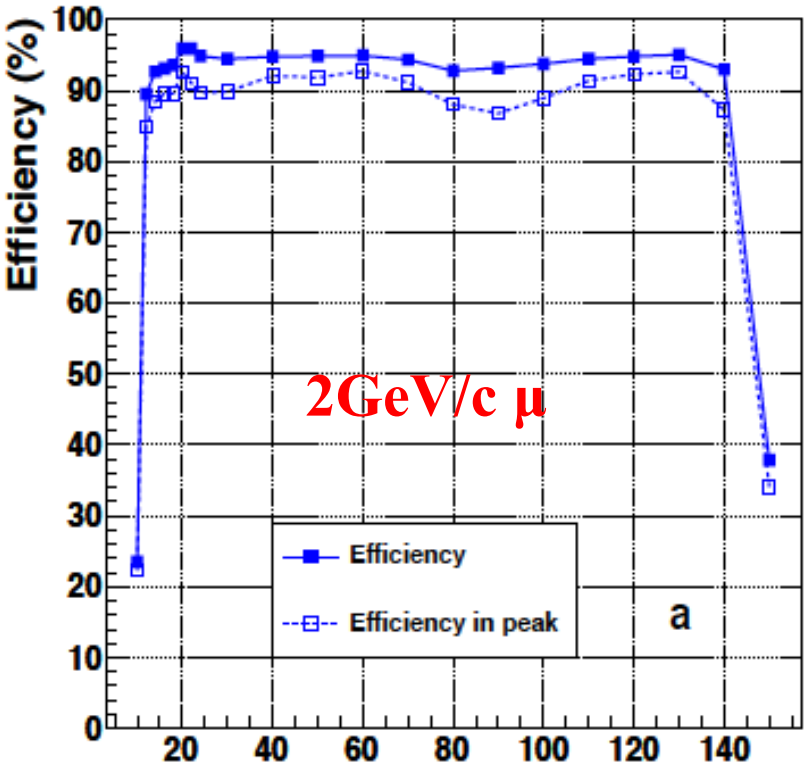
Efficiency ….

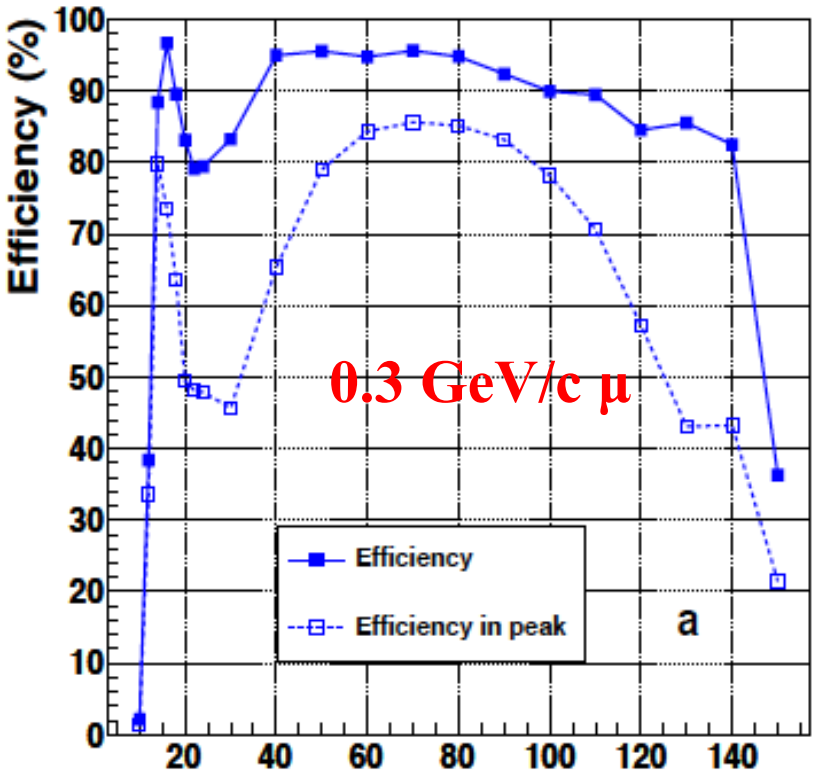
2000 – Monte Carlo tracks

1820 – reconstructed track for each MC

$$\varepsilon = \frac{1840}{2000} = 0.92 \Rightarrow 92\%$$

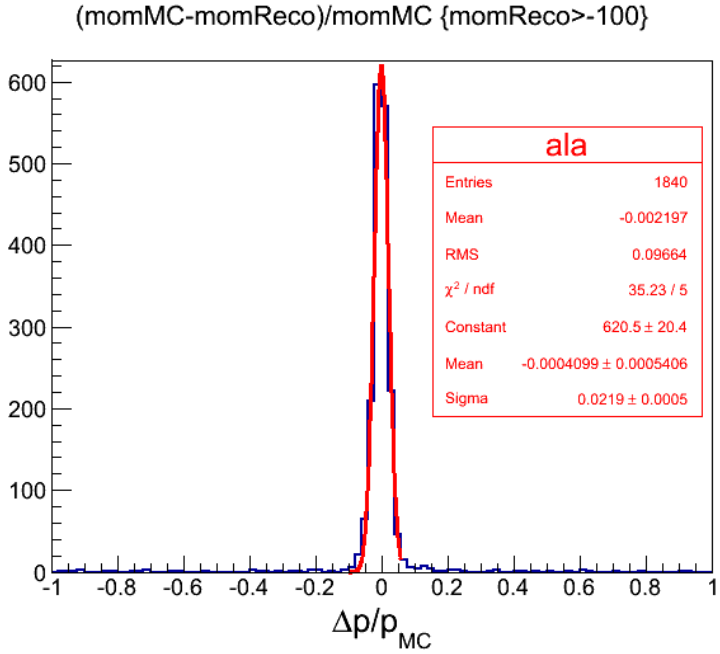
global number
average over all θ, p





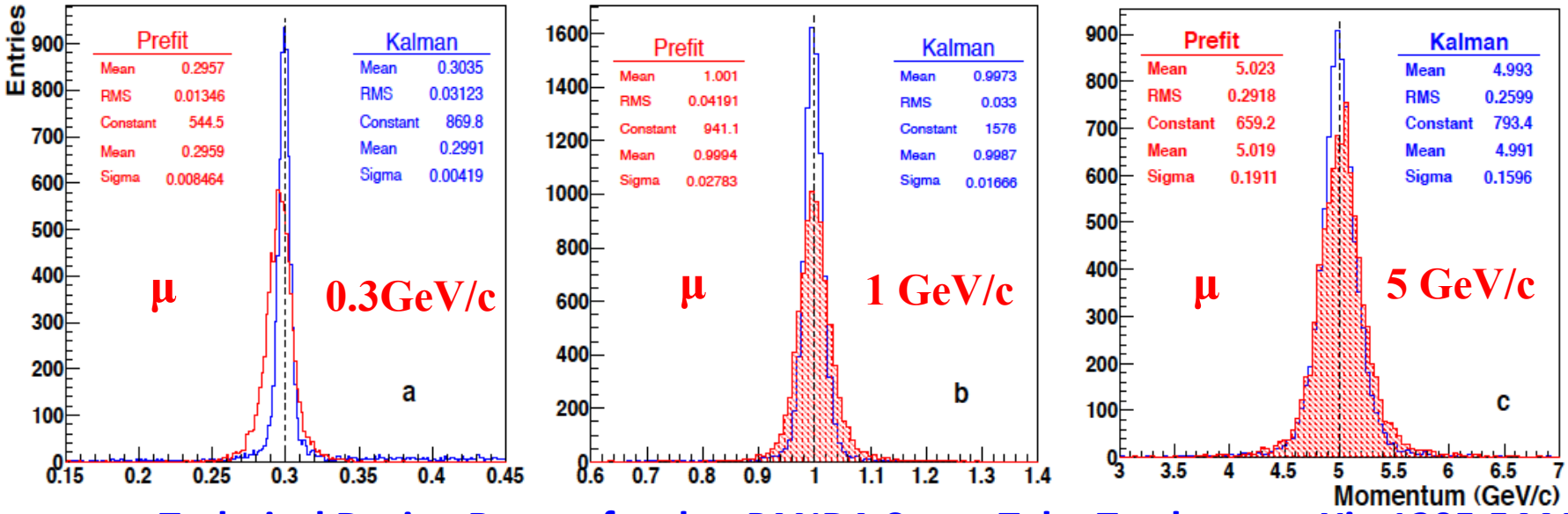
Technical Design Report for the: PANDA Straw Tube Tracker. arXiv:1205.5441

Momentum resolution



$\Delta p = p_{MC} - p_{Reco}$

global number
average over all θ, p



Technical Design Report for the: PANDA Straw Tube Tracker. arXiv:1205.5441

More detector information needed for PID in ntuple …

```
//EMC detector ....
clusterZ20    = pidCand->GetEmcClusterZ20();
clusterZ53    = pidCand->GetEmcClusterZ53();
emc_lat       = pidCand->GetEmcClusterLat();
emc_Er        = pidCand->GetEmcRawEnergy();
emc_qa        = pidCand->GetEmcQuality();
emc_index     = pidCand->GetEmcIndex();
emc_crystal = pidCand->GetEmcNumberOfCrystals();

// STT informations
stt_dedx    = pidCand->GetSttMeanDEDX();
stt_hits      = pidCand->GetSttHits();

// muon detector informations
muo_index     = pidCand->GetMuoIndex();
muo_nbLayer  = pidCand->GetMuoNumberOfLayers();
muo_module  = pidCand->GetMuoModule();
muo_iron       = pidCand->GetMuoIron();
muo_qa          = pidCand->GetMuoQuality();
muo_momIn   = pidCand->GetMuoMomentumIn();
```

try to add it into the macro …