

## PndPidFtofInfo.cxx

```
1/* This class can be used for Pid by Stof+Ftof which was intially for Ftof
2 The Fts3 last plane is defined in PndPidCorrpar.h and PndPidCorrpar.cxx
   as
3 Float_t GetFts3LastPlane() { return fFts3LastPlane; }; and also used
   the value of Fts3LastPlane as 467 cm
4 and Ftof cut as 36 cm^2 in all.par file*/
5#include "PndDetectorList.h"
6#include "PndPidCorrelator.h"
7#include "PndPidCandidate.h"
8#include "PndMCTrack.h"
9#include "PndTrack.h"
10#include "PndTrackID.h"
11
12#include "PndFtofHit.h"
13#include "PndFtofPoint.h"
14
15#include "FairTrackParH.h"
16#include "FairMCApplication.h"
17#include "FairRunAna.h"
18#include "FairRootManager.h"
19#include "FairRuntimeDb.h"
20
21Bool_t PndPidCorrelator::GetFtofInfo(FairTrackParH* helix,
   PndPidCandidate* pidCand)
22{
23  if(!helix)
24  {
25    std::cerr << "<Error> PndPidCorrelator::GetFtofInfo: FairTrackParH
   NULL pointer parameter."<<std::endl;
26    return kFALSE;
27  }
28  if(!pidCand)
29  {
30    std::cerr << "<Error> PndPidCorrelator::GetFtofInfo: pidCand NULL
   pointer parameter."<<std::endl;
31    return kFALSE;
32  }
33
34  if(helix->GetZ() < fCorrPar->GetFts3LastPlane())//changed for Stof
35  {
36    if (fVerbose>0) std::cout << "-W- PndPidCorrelator::GetFtofInfo:
   Skipping tracks not reaching the last plane of FTS3 layer" << std::endl;
37    return kFALSE;
38  }
39
40  if(helix->GetPz() <= 0.)
41  {
42    std::cout << "-W- PndPidCorrelator::GetFtofInfo: Skipping tracks
   going backward" << std::endl;
43    return kFALSE;
44  }
45
46  FairGeanePro *fProTof = new FairGeanePro();
```

PndPidFtofInfo.cxx

```

47  if (!fCorrErrorProp) fProTof->PropagateOnlyParameters();
48  FairGeanePro *fProVertex = new FairGeanePro();
49  fProVertex->PropagateOnlyParameters();
50
51  PndFtofHit *tofHit = NULL;
52  Int_t tofEntries = fFtofHit->GetEntriesFast();
53  Int_t tofIndex = -1;
54  Float_t tofTof = 0., tofLength = -1000, tofGLength = -1000,
    tofTrackLength = -1000;;
55  Float_t tofQuality = 1000000;
56
57  Float_t chi2 = 0;
58  TVector3 vertex(0., 0., -10000.);
59  TVector3 vertexrec(0., 0., -10000.);
60  TVector3 momrec(0., 0., -10000.);
61  TVector3 tofPos(0., 0., 0.);
62  TVector3 momentum(0., 0., 0.);
63  TVector3 hitPoint; //changed for stof
64
65  if (fGeanePro) // Overwrites vertex if Geane is used
66  {
67      // calculates track length from (0,0,0) to last point
68      fProVertex->SetPoint(TVector3(0,0,0));
69      fProVertex->PropagateToPCA(1, -1);
70      FairTrackParH *fRes= new FairTrackParH();
71      Bool_t rc = fProVertex->Propagate(helix, fRes, fPidHyp*pidCand-
    >GetCharge());
72      if (rc)
73      {
74          tofTrackLength = fProVertex->GetLengthAtPCA();
75          vertexrec.SetXYZ(fRes->GetX(), fRes->GetY(), fRes->GetZ());
76          momrec.SetXYZ(fRes->GetPx(), fRes->GetPy(), fRes->GetPz());
77      }
78  }
79
80  for (Int_t tt = 0; tt<tofEntries; tt++)
81  {
82      tofHit = (PndFtofHit*)fFtofHit->At(tt);
83      if ( fIdeal && ( ((PndFtofPoint*)fFtofPoint->At(tofHit->GetRefIndex
    ()))>GetTrackID() !=pidCand->GetMcIndex() ) continue;
84      tofHit->Position(tofPos);
85      hitPoint=tofHit->GetPosition(); //changed for stof
86      Float_t propX = helix->GetX() + (hitPoint.z()- helix->GetZ()) *
    helix->GetPx() / helix->GetPz(); //changed for Stof
87      Float_t propY = helix->GetY() + (hitPoint.z() - helix->GetZ()) *
    helix->GetPy() / helix->GetPz(); //changed for Stof
88      Float_t propZ = hitPoint.z(); //changed for Stof
89      vertex.SetXYZ(propX, propY, propZ);
90      tofGLength = (vertex-helix->GetPosition()).Mag();
91
92      if (fGeanePro) // Overwrites vertex if Geane is used
93      {
94          fProTof->SetPoint(tofPos);

```

PndPidFtofInfo.cxx

```

95     fProTof->PropagateToPCA(1, 1);
96     FairTrackParH *fRes= new FairTrackParH();
97     Bool_t rc = fProTof->Propagate(helix, fRes, fPidHyp*pidCand-
>GetCharge());
98     if (!rc) continue;
99     tofGLength = fProTof->GetLengthAtPCA();
100     vertex.SetXYZ(fRes->GetX(), fRes->GetY(), fRes->GetZ());
101 }
102
103     Float_t dist = (tofPos-vertex).Mag2();
104
105     if ( tofQuality > dist)
106     {
107         tofIndex = tt;
108         tofQuality = dist;
109         tofTof = tofHit->GetTime();
110         tofLength = tofTrackLength+tofGLength;
111     }
112
113     if (fDebugMode)
114     {
115         Float_t ntuple[] = {vertex.X(), vertex.Y(), vertex.Z(),
116                             vertexrec.X(), vertexrec.Y(), vertexrec.Z(),
117                             momrec.X(), momrec.Y(), momrec.Z(),
118                             helix->GetMomentum().Mag(), helix->GetQ(),
119                             helix->GetMomentum().Theta(), helix->GetZ(),
120                             tofPos.X(), tofPos.Y(), tofPos.Z(),
121                             dist, tofLength, tofGLength,tofTrackLength};
122         ftofCorr->Fill(ntuple);
123     }
124
125     if ( (tofQuality<fCorrPar->GetFTofCut()) || (fIdeal && tofIndex!=-1) )
126     {
127         pidCand->SetTofQuality(tofQuality);
128         pidCand->SetTofStopTime(tofTof);
129         pidCand->SetTofTrackLength(tofLength);
130         pidCand->SetTofIndex(tofIndex);
131         if (tofLength>0.)
132         {
133             // mass^2 = p^2 * ( 1/beta^2 - 1 )
134             Float_t mass2 = helix->GetMomentum().Mag()*helix->GetMomentum().Mag
135             ()*(30.*30.*tofTof*tofTof/tofLength/tofLength-1.);
136             pidCand->SetTofM2(mass2);
137         }
138     }
139     return kTRUE;
140 }
141

```