

Hello Stefano,

I wanted to start fixing the problem during today's lunch break, but I noticed that we might not have the described problem after all. (Sorry, I was quite tired when I looked through the code last night.) I looked at the following code excerpt from  
void PndFtsTrackerIdeal::Exec(Option\_t \* option):

(Note that iDet == 0 corresponds to the FTS)

```
// Detector loop
for(Int_t iDet=0;iDet<4;iDet++){
  if (kFALSE == fBranchActive[iDet]) continue; //skip manually switched off detector
  if(fVerbose>4) Info("Exec","Use detector %i",iDet);
  // Hit loop
  for (Int_t ih = 0; ih < fHits[iDet]->GetEntriesFast(); ih++) {
    ghit = (FairHit*) fHits[iDet]->At(ih);
    if(!ghit) {
      if(fVerbose>3) Error("Exec","Have no ghit %i, array size:
%i",ih,fHits[iDet]->GetEntriesFast());
      continue;
    }
    Int_t mchitid=ghit->GetRefIndex();
    if(mchitid<0) {
      if(fVerbose>3) Error("Exec","Have a negative mcHit %i",mchitid);
      continue;
    }
    myPoint = (FairMCPoint*)(fMCPoints[iDet]->At(mchitid));
    if(!myPoint) continue;
    Int_t trackID = myPoint->GetTrackID();
    if(trackID<0) continue;

    if(fVerbose>5) Info("Exec","Have a Hit %i at Track index %i",ih,trackID);

    // Continue Construction of a track candidate (start with FTS hits)
    // Track candidates and corresponding MC track index are saved in a map
    PndTrackCand* cand=candlist[trackID];
    if(NULL==cand){
      if(0!=iDet){
        if(fVerbose>5) Info("Exec","Skip Hit %i, it's not connected to a Track in FTS",ih);
        continue; // skip Tracks in MVD/GEM not going to FTS
      }
      if(fVerbose>5) Info("Exec","Create new PndTrack object %i",trackID);
      cand=new PndTrackCand();
      cand->setMcTrackId(trackID);
      if(fVerbose>5) Info("Exec","Creating new PndTrack object finished %i",trackID);
    }
    if(fVerbose>5) Info("Exec","add the hit %i to trackcand %i",ih,trackID);
```

```

cand->AddHit(fBranchIDs[iDet],ih,myPoint->GetTime());
// Figure out if the current hit is the earliest in the event
if(!firstHit[trackID] || firstPoint[trackID]->GetTime() > myPoint->GetTime()) {
    firstHit[trackID]=ghit;
    firstPoint[trackID]=myPoint;
}
// or the latest one
if(!lastHit[trackID] || lastPoint[trackID]->GetTime() < myPoint->GetTime()) {
    lastHit[trackID]=ghit;
    lastPoint[trackID]=myPoint;
}

candlist[trackID] = cand; // set
} // end loop over hits
} // end loop over detectors
// now we have track candidates

```

The most important part is actually this one:

```

PndTrackCand* cand=candlist[trackID];
if(NULL==cand){
    if(0!=iDet){
        if(fVerbose>5) Info("Exec","Skip Hit %i, it's not connected to a Track in FTS",ih);
        continue; // skip Tracks in MVD/GEM not going to FTS
    }
    if(fVerbose>5) Info("Exec","Create new PndTrack object %i",trackID);
    cand=new PndTrackCand();
    cand->setMcTrackId(trackID);
    if(fVerbose>5) Info("Exec","Creating new PndTrack object finished %i",trackID);
}

```

So we first look at all FTS Hits individually and later all hits from other detectors in the event. For each hit we figure out which MC track caused them. ONLY if there is at least one FTS (iDet == 0) Hit from a MC track, we create a PndTrackCand. If there was no FTS hit for a MC track (iDet != 0), then the algorithm fails to load the corresponding PndTrackCand and it continues (meaning it skips the hit).

So, I believe that the PndFtsTrackerIdeal does exactly what you suggested (write out PndTrackCands which have at least 1 FTS Hit). I would nevertheless like to change that behavior so that it requires at least 5 FTS hits in order to be more realistic and if no one objects I will change the algorithm accordingly in the next few lunch breaks / nights.