Subject: Problem with secondary tracks
Posted by Felix Boehmer on Thu, 28 Jul 2011 16:10:26 GMT
View Forum Message <> Reply to Message

Dear pandarooters,

in the course of figuring out what is going wrong with the MC PID correlation, we came across some strange behavior concerning trackIDs of secondary particles.

What happens is that the trackID we extract in the processHits() method of the TPC Detector class from the MCStack is no longer a valid position in the MCTrack array after the simulation. This only happens for secondary tracks and explains why in some generator channels we get invalid PDG codes.

We have been trying to find out if/where these IDs are altered in the framework, but failed so far. Does anybody know what is happening?
Since time is short right now we would be very grateful for any help on that matter. We will be running simple example macros to find out what's going on.

Cheers

Felix

Subject: Re: Problem with secondary tracks
Posted by Mohammad Al-Turany on Thu, 28 Jul 2011 16:29:05 GMT
View Forum Message <> Reply to Message

Hi,

There is a filter on secondary particles, by default, this filter will skip secondary particles that:
1. do not hit any detector
and
2. do not have any daughter that hit any detector

such particles can never be detected and we simply delete the and reindex the whole mother-doughter relations in the TClonesArrays and the MCTrack
 (all what goes to the Tree is consistence)

Now if you have classes that do not inherits from the MCpoint classes in framework or you do your own things (you save the particle Id your self in the simulation in some privet classes), then I can imagine that you get some discrepancies!

Anyway, this filter can be easily switched off in the g4/3Config.C macro:

```
/// create the Specific stack
   PndStack *stack = new PndStack(1000);
   stack->StoreSecondaries(kTRUE);
   stack->SetMinPoints(1);
```

if you set the min points to zero, i.e:

/// create the Specific stack
   PndStack *stack = new PndStack(1000);
   stack->StoreSecondaries(kTRUE);
   stack->SetMinPoints(0);

Then all particles will be saved to the output without any filtering. But this could increase the file sizes dramatically.

I hope this will help you.

regards from the sunny Jordan with 40 degrees in shadow.

Mohammad

Subject: Re: Problem with secondary tracks
Posted by Felix Boehmer on Thu, 28 Jul 2011 19:12:22 GMT
View Forum Message <> Reply to Message

Hi Mohammad,

thanks for your answer.
We certainly don't want to use any information of tracks outside the detector... However, we want to store the sec. track ID in the TPC points during processHits of the detector class... How can we assure to have valid trackIDs also for secondaries after the framework did its cleanup?

Cheers

Felix

PS: 40 degrees is more or less 4 times what we have here...

Subject: Re: Problem with secondary tracks
Posted by Mohammad Al-Turany on Thu, 28 Jul 2011 21:24:59 GMT
View Forum Message <> Reply to Message

Hi,

If I understand your code in "PndTpcDetector::ProcessHits" you are miss-using the trackID variable, this member variable comes from the base class FairMCPoint and the whole filtering is based on this inheritance (The FairMCStack has no idea about detector or experiments code).

To explain this: The trackID is used normally as unique identifier of a track, i.e: it is the index of this track in the array, and when you get the mother ID of a track you get also the index of that mother in the array. In your code you are setting the trackID to the mother ID:

```
 Bool_t  PndTpcDetector::ProcessHits( FairVolume *v)
  ........
    TParticle* mother=gMC->GetStack()->GetCurrentTrack();
    if(mother->IsPrimary()) secID = 0;
    else while(!mother->IsPrimary()){
    trackID=mother->GetFirstMother();
    mother=dynamic_cast<PndStack*>(gMC->GetStack())->GetParticle(trackID);
    //std::cout<<"Fetching mother id="<<trackID<<std::endl;
  }

 // trackID is now ID of primary mother
 // if the mother is already primary, secID is 0
 // else secID is an (arbitrary) number !=0
.............

 //gotta love TClonesArray syntax!
  PndTpcPoint* p=AddHit(trackID, secID, volumeID, pos.Vect(), mom.Vect(),
   time, length, eLoss);
```

As you see, the PndTpcPoint has a variable trackID which does not correspond to the track index anymore, and this will miss up the whole filtering!   So I would suggest you to define a new variable in your TpcPoint which hold the info you need without miss-using the trackID.

Hope this will help you, now I will enjoy the rest of my vacation!

Cheers,

Mohammad

Posted by Johannes Rauch on Fri, 29 Jul 2011 14:48:44 GMT
View Forum Message <> Reply to Message

Dear Mohammad,

when I just assign the trackID (and not trackIDs of mothers) everything works fine!

Thanks for your help and have a nice holiday!

Johannes