Subject: [FIXED] Ideal Tracking Bugs(?) when using FairLinks
Posted by André Zambanini on Tue, 12 May 2015 14:38:54 GMT
View Forum Message <> Reply to Message

Hello everyone,

for my analysis I had a closer look at some events in the eventDisplay and found some strange behavior of the ideal track finding. Basically, I found two problems and my guess is, they are independent of each other, but I'm not sure.

General Information
First, some information beforehand.
FairSoft: mar15
FairRoot: master (fb738d60 from 26.03.2015)
PandaRoot: r27581

I'm simulating events with EvtGen (momentum 4.07 GeV/c) using the decay chain: pbar p -> Xi+ Xi(1690)- -> Lambdabar pi+ Lambda K- and the lambdas decaying to pi p. The main message from the decay for you is that I have a lot of displaced vertices with distances of several centimeters to the IP.

For the reconstruction I'm using ideal track finding, both with the old track finder (PndSttMvdGemTrackingIdeal) and the new one (PndMCIdealTrackFinderNewLinks). The simulation chain uses FairLinks (fRun->SetUseFairLinks(kTRUE).

Wrongly Assigned GEM Hits
The first thing Tobias and me noticed were wrongly assigned GEM hits. From what I understood from Tobias and Stefano, this is a known issue. To illustrate this a bit, see these screenshots here:

Both, the old and the new track finder seem to assign GEM hits which don't belong to the track. The white dots indicate the MC points associated with the selected track - which works fine for MVD (blue squares) and STT (purple) hits, but seems to be off for GEM (red) hits.

Along with this comes wrong track information, as you can see with the red and blue circles, which indicate the first and last track parameters, respectively.

Tobias and me had a closer look at the new track finder and found out, that the spurious hit assignment happens when more than one MC point belong to a hit. The quick fix introduced by us is to simply ignore those hits. This came to the PndMCMatchNewLinks class with r27667 in the trunk.

My conclusion here is, that for now this is okay but maybe someone should have a detailed look at this.

Track Reconstruction with Kalman Task
After resolving the spurious GEM hits, the reconstructed track parameters still looked quite odd. I did some comparison and found out, that in a few cases the Kalman task messes things

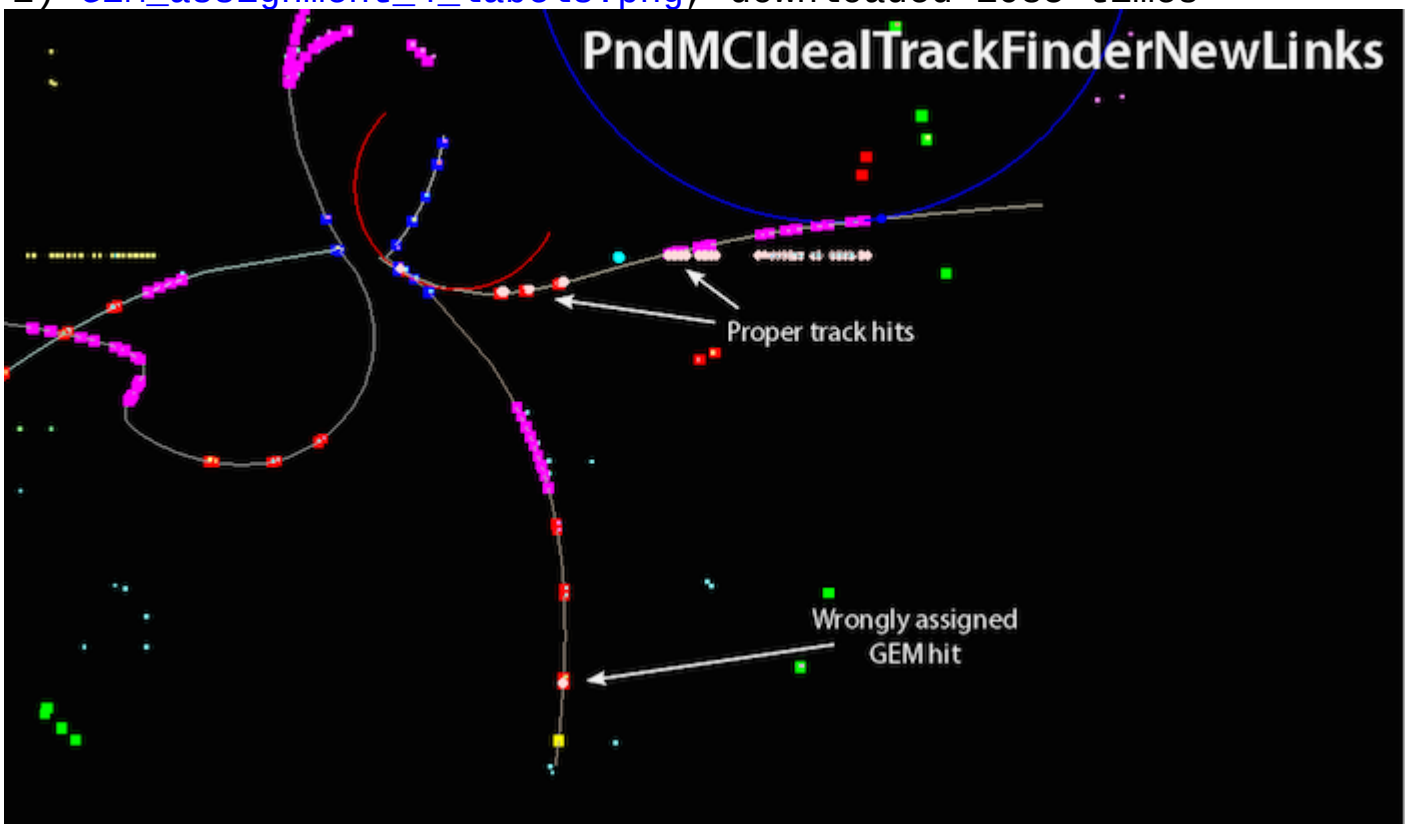
up, both for genfit 1 and 2. The following screenshots all show the result of the new ideal track finder (PndMCMatchNewLinks):

For this event, the Kalman filter produces strange tracks, independent of the genfit version. Most other events I had a look at seemed ok with genfit 2, while genfit 1 sometimes causes charge flips for instance (as visible in the screenshot above).
Ideally, I would leave the Kalman filter out, especially because for ideal tracking it seems a bit odd to use it. But this doesn't work for the PID, which seems to require the covariant matrices or something else filled by the Kalman.

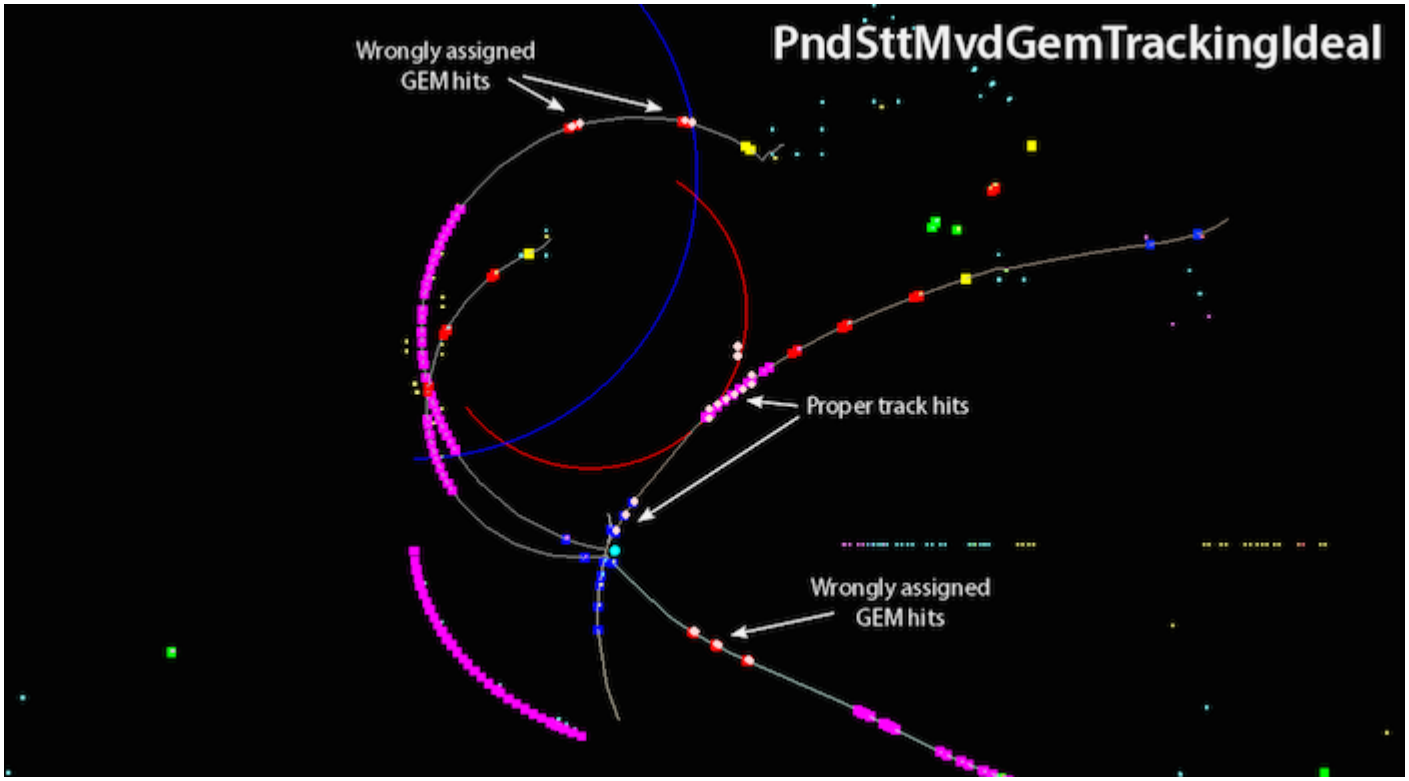
So concluding, I have two questions: Any ideas, what could cause the Kalman to produce these results? And secondly, why is it required to have it in the first place, shouldn't ideal PID be based on MC information?

File Attachments

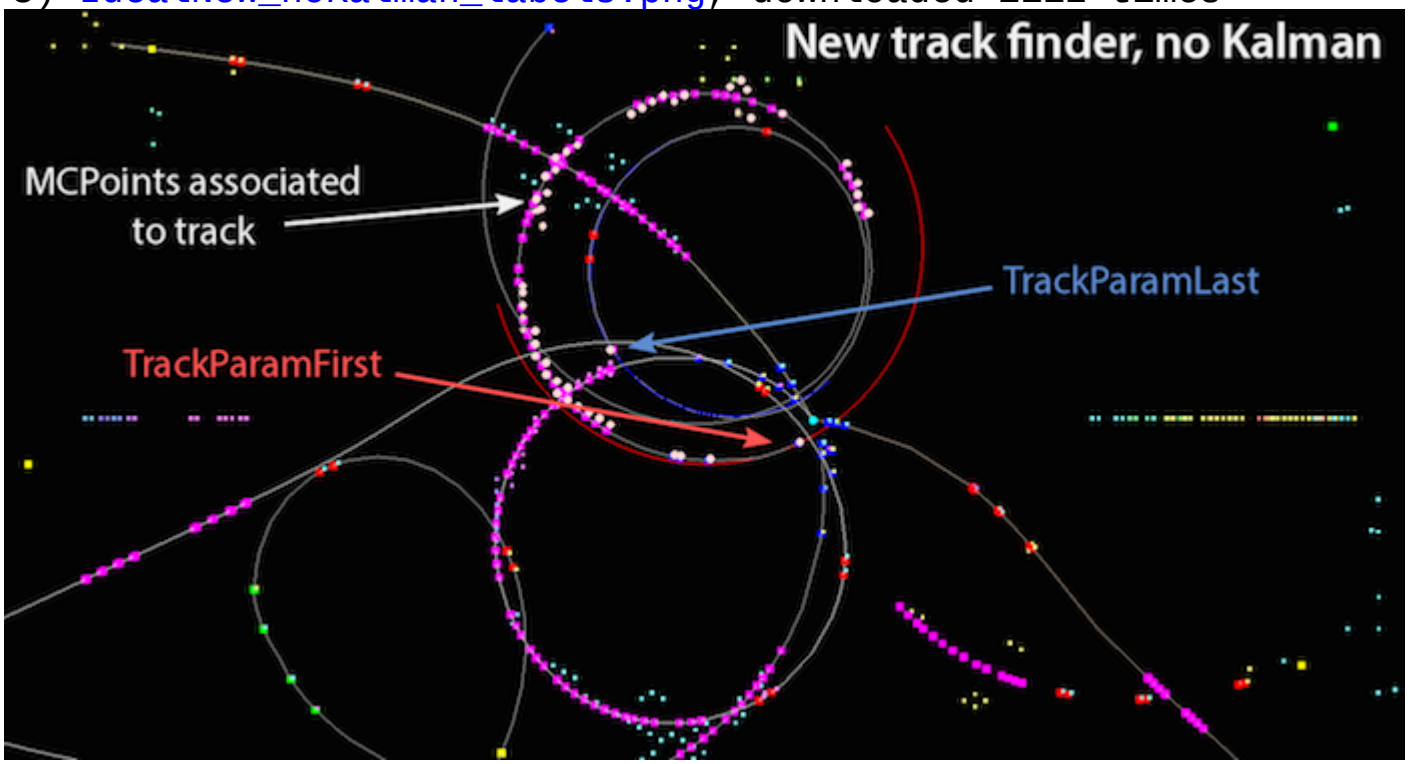
1) GEM_assignment_4_labels.png, downloaded 1085 times



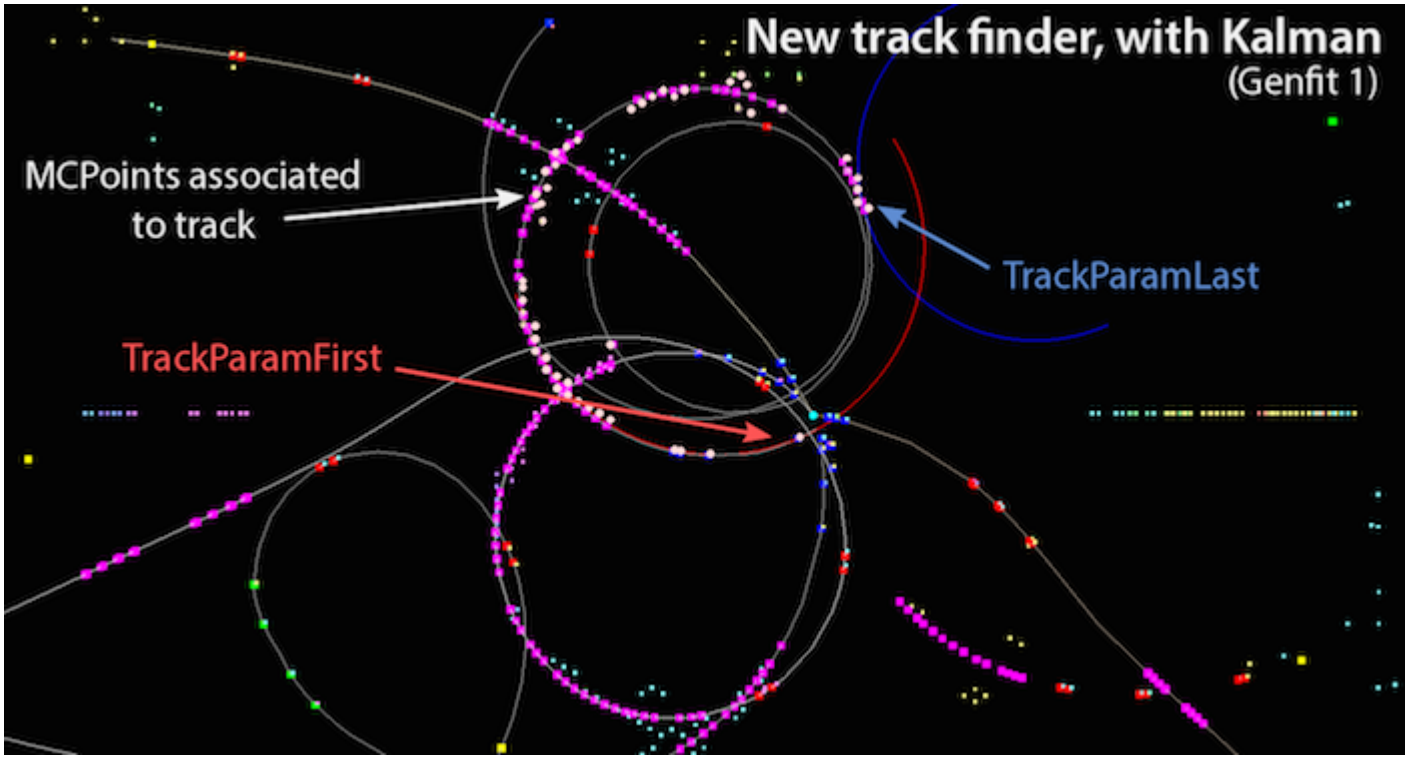
2) GEM_assignment_Ideal_Trackfinder_labels.png, downloaded 1074 times



3) IdealNew_noKalman_labels.png, downloaded 1112 times



4) IdealNew_withKalman_labels.png, downloaded 1064 times



5) IdealNew_withKalman2_labels.png, downloaded 1006 times