Subject: Re: PndRecoMultiKalmanTask
Posted by Anonymous Poster on Wed, 31 Mar 2010 15:31:09 GMT
View Forum Message <> Reply to Message

Hi Stefano and others,

I spent a few hours getting up to speed on what is going on here. First of all: I really appreciate your effort in making a common fitting task for everybody!

Concerning your results:

- The fact that the chi2/nfd distributions peak at 0.5 is due to the fact that we do not estimate the resolution on the TPC (and possibly others) reco hits correctly. The smearing of TPC clusters depends on the full realistic digitization chain. The exact results for the point resolutions are not known at the moment. However, I can verify that scaling the point resolution by a constant factor of 0.8 makes the distribution peak at ~1. At this point this can not be considered as a major problem, but it will be subject to optimizations by all detector code developers, responsible for putting the right measurement errors into the corresponding reco hits. I can say that in the TPC it looks like we are already within ~20%, which is not so bad.

- The tracks which have NDF==0: I could reproduce this effect. I found out that it is due to the fact that all (or almost all) hist in these track failed int he Kalman filter, meaning in particular that the GEANE extrapolation had failed for all hits. You can see that if you get rid off the GFException::quiet() switch. The way to catch this, is to evaluate the number of failed hits. You can access it by

GFTrack::getFailedHits(int repId)

If you do not pass a repId, it defaults to the cardinal track representation. You might want to pass this info into PndTrack and to the adapter function.

I didnt check, why all the hits fail for these tracks. But I guess it could be due to the fact that the starting values are really wrong or that the starting point maybe is already much behind the first hit or so. If you have further problems in finding out what the problem is, I can try to help.

- about separating different particle hypothesis from chi2/ndf: This would be very interesting for PID purposes. I am not surprised that the chi2/ndf distributions look similar. The difference in chi2 must be very small for each fit. But it would be very interesting to know:
Is the chi2/ndf for the right hypothesis (if only slightly) better that for the others? Maybe you could plot (chi2/ndf_kaonHyp - chi2/ndf_pionHyp) for you fits. I am eager to see that!
Maybe it might be better to actually use the chi2 probability for cuts or evaluation instead of chi2/ndf. You can get it in ROOT via TMath:Prob(chi2,ndf). It should be better especially if ndf differs a lot between fits.

Cheers, Christian