Subject: Re: Smoothing method in class Kalman
Posted by Anonymous Poster on Mon, 16 Mar 2009 12:45:29 GMT
View Forum Message <> Reply to Message

Hi Ola,

ah I wasnt aware that your hit needs a custom residualVector implementation. That is not a problem though, I will make the residualVector method virtual again, immediately, such that your overwritten method is used. I am not sure about the functionality of C++ in this case, but I guess that if you provide your own implementation with the same prototype, but in the super class the thing was not virtual, if the method is called on a super-class pointer, the super-class implementation is called and not yours. We do want to avoid that! As I said, I will make it virtual again now.

What I will write next will be complicated, but it will fully answer your question. You will have much more documentation on Genfit soon, and then it will be easier to understand what I am talking about. But I will give you a full explanation here anyway. Afterall, the code is not a lot to look through...

For an explanation why I changed the prototype to contain the DetPlane and also why I made the AbsTrackRep pointer to point to a const AbsTrackRep:
In residual vector you need the DetPlane of the hit. In general this DetPlane is not fixed for the hit, but it depends on the track. This is implemented in the TPC through SpacePointHitPolicy. This means that if you would not give a DetPlane to the residualVector method, it would have to calculate it. This is the way it worked before. However, this calculation is expensive in computing time, and it has been done before, so we dont want to do it again. I made the AbsTrackRep pointer const, so extrapolate (which before was called through AbsTrackRep::getVirtualDetectorPlane over the SpacePointHitPolicy) is not possible anymore. There is another reason why I want to prohibit extrapolate calls inside any method of AbsRecoHit besides extrapolateToPoca (which is called by getVirtualDetectorPlane): I do multi-pass fitting in Kalman. When I reached the last point and I turn around, this first hit now after turning around does not require any extrapolation, since I am already there. These kind of null-extrapolations need to be avoided since Geane is instable for them, right now. So I dont do the extrapolation in these cases (look inside Kalman::ProcessHit()), but then I need to make sure that the residualVector method does also not do any extrapolations.

I commit the change for virtual residualVector in a minute.

Cheers, Christian