Subject: Question on GeaneTrackRep.
Posted by Lia Lavezzi on Fri, 06 Mar 2009 16:30:22 GMT
View Forum Message <> Reply to Message

Hi Christian,
I know you are busy but, since I see you are cleaning up the genfit classes, I take this chance to ask you one question, just to hear your opinion on this problem.

The problem concerns GeaneTrackRep: in this representation there is an additional variable (in addition to the state and the cov matrix) that is important and it is the spu (it is +1 or -1 whether the momentum is parallel or antiparallel to the detPlane z axis) which should be taken at each extrapolation and should be updated after the end of the Kalman step (exactly as state and cov are).

Since the spu variable is defined only in GeaneTrackRep the only way we found to set it was to do this within the:
GeaneTrackRep::extrapolate(const DetPlane& plane, TMatrixT<double>& statePred, TMatrixT<double>& covPred, TMatrixT<double>& jacobian) function, after the propagation is performed: _spu = result.GetSPU().
At the beginning this seemed quite ok, but actually this is not completely correct and may also cause mistakes: to set up things in a right way, spu should be updated together with state and cov. So in the extrapolation step only a spuPred should be filled: spuPred = result.GetSPU() and later the update should be done within the Kalman.cxx: rep->setSPU(spu)
This is not just a matter of principle, if we keep things as they are now, it may also cause problems: for example let's suppose I have my representation defined in point A and I want to know the momentum on two detector planes, one in point B and the other in point C. First I call the getMom function to have the momentum in B and "internally" the extrapolation from A to B is made and the momentum is returned. But here the spu may change (!) and if, later, I want to get the momentum in point C, the extrapolation from A to C is performed once again but the starting representation has now the WRONG spu value (the one which belongs to B point)... (I hope I have been able to explain things, I know it' s quite a mess!)

So, let me summarize:
1) the spu should NOT be set within the extrapolate function, but in the Kalman.cxx, where the representation is updated:
```
rep->setState(state);
rep->setCov(cov);
rep->setReferencePlane(pl);
rep->setSPU(spu);
```
2) To do this, however, also the extrapolate function should be changed to fill also a spuPred (in addition to statePred, covPred):
virtual double extrapolate(const DetPlane& plane, TMatrixT<double>& statePred, TMatrixT<double>& covPred, TMatrixT<double>& jacobian, double spuPred)=0 ;

I tried to find a solution, but I didn' t find an easy one.
The only thing I could think about is to change the AbsTrackRep directly (to add the spu and change the extrapolate/predict functions), but I know that this would affect also the LSLTrackRep (that does not need the spu, as far as I know) and so this is not the best solution at all... That' s why I' m asking your opinion, do you see a simple way to fill this variable together with state and cov?
What would you do if you had a representation which requires a variable which is not present

among the usual ones defined in AbsTrackRep?
I would be very grateful if you could give me some suggestion since I don' t find how to handle this!
I will keep thinking about this... looking for an inspiration

Thank you and ciao,
Lia.

Subject: Re: Question on GeaneTrackRep.
Posted by Anonymous Poster on Fri, 06 Mar 2009 16:38:03 GMT
View Forum Message <> Reply to Message

Hi Lia,

thanks for your interesting question. The problem of direction information is one of my two left items on my "cleaning up GENFIT" business. I do not have a quick answer for you right now. But next week I will sit down with Felix here in Munich to think about our open questions and to further clean up. I will keep you posted on what we come up with.

More next week

CU, Christian

Subject: Re: Question on GeaneTrackRep.
Posted by Anonymous Poster on Mon, 09 Mar 2009 20:28:28 GMT
View Forum Message <> Reply to Message

Hi Lia,

well your question is a really good one. I do not have an easy answer in the moment. It is a general problem with all tracks with all track representations. This is not specific to genfit as far as I can tell.
For example in COMPASS where we use x,y,dx/dz,dy/dz,q/p and z as the free parameter (which is some sense the DetPlane in genfit) we just know that tracks fly along the z-axis. I ALICE, whose track model I also looked at, you also dont have a parameter which gives the direction of the track. In most cases it is clear: it comes from the vertex. For secondary vertices with low momenta this doesnt work and we need a different idea.

Here are several points for discussion (Sebastian give some of your brains into this please, too):
- should we just have a sixth paramter? I dont really see a problem with that
- can we tackle the problem from hit sorting, saying that hits have to be sorted before the fitter so they always go with increasing track length?
- maybe more...

Here is my thought about the issue of direction: We need this info to decide whether to propagate a track fowards or backwards. We can leave this decision to the Kalman filter algorithm. Then this info should never appear in the trackRep and is just an argument to extrapolate(). Or we could leave it completely to the track representation then this info should not appear anywhere else. In this case we would maybe need the sixth parameter.

If i dont overlook any detail, the idea of an additional parameter looks good. In that case the trackRep is itself responsible for determining forward or backward tracking.

Any thoughts? What did I miss?

Christian

Subject: Re: Question on GeaneTrackRep.
Posted by Sebastian Neubert on Tue, 10 Mar 2009 09:26:43 GMT
View Forum Message <> Reply to Message

Dear colleagues,

I thought about this issue for some time at several occasions in the past and I always came to the conclusion, that there is no failsafe way to handle this ambiguity. From the measured hits alone one cannot tell in which direction the particle was going. The most general way is to fit both hypotheses.

One could make some tentative assumptions based on the general topology (most tracks come from the IP) but this is not general.

In my view it is a question that cannot be solved in the fitter! The direction of the track has to be decided upon during pattern recognition.

The hits have to be sorted along the track before the fitting! This has always been pointed out. It is also part of the pattern recognition (and not the easiest one).

Cheers!
Sebastian.

Subject: Re: Question on GeaneTrackRep.
Posted by Anonymous Poster on Tue, 10 Mar 2009 10:15:58 GMT
View Forum Message <> Reply to Message

Hi,

thanks for your thoughts. You are right that this is an ambiguity. However, it isnt a very bad one:

The only reason for having to know the direction parameter in the track rep is because sometimes sorting beforehand is not perfect. Sometimes it happens that the sorting is wrong for a few hits in a track. If we make the rule though that hits are sorted from their vertex on, and

we assume the sorting will hold true for the first three hits or so, the initialized direction parameter in the track rep would take care of the rest if it is initialized as "forward". This seems like a good solution to me, and we should work it out. I dont know when I am going to start doing that. And for sure I will need lots of help from Pavia

Cheers, Christian

Subject: Re: Question on GeaneTrackRep.
Posted by Lia Lavezzi on Tue, 10 Mar 2009 15:30:23 GMT
View Forum Message <> Reply to Message

Hi Christian,
actually the spu variable and the direction problem are two separate problems.

Concerning the direction, in my opinion the Kalman filter (and so the geane extrapolation) requires a starting point and you must decide yourself to go in the forward or backward direction, not to let geane decide itself.
The hits have to be ordered along the track somehow, then you know whether your starting point is at the beginning or at the end of the track: usually you choose a reconstructed vertex and so you know (or at least you can suppose) your track is going in the forward direction, but you can also choose to start from outside and go toward the center of the detector, but in this case you know that you are going backward.
In any case I think you have to tell geane to go forward or backward at the beginning of the track (it should be enough to set the direction at the very beginning, when the track rep is created, with setPropDir(...)) and then the Kalman procedure runs through all the track hits and arrives to the end; at this stage, in case of more than one iteration, it changes the propagation direction going in the opposite one.
I agree that a bigger problem in this case would be to be able to order hits.
I also think Sebastian' s hypothesis to fit both forward and backward track could be a solution if we don' t have a previously reconstructed vertex and so we don' t have any idea on where the track starts from.

Concerning the spu: the spu variable is not directly linked to the direction of the track (forward/backward), or at least not only; it is linked to the orientation of the detector plane frame with respect to the momentum direction: it is the sign of the momentum component perpendicular to the detector plane.
I try to explain this with a drawing (in attachment).
See figures A and B: here the track (green line) is going in the forward direction, but the spu is +1 for A and -1 for B (the same in cases C and D for the backward direction).
This happens in the case of STT, but I think it is a more general problem. In the case of STT the plane axes are chosen this way:
- u axis: from the wire through the PCA (= point of closest approach to the wire itself)
- v axis: the wire itself (in the direction of increasing z).
- w is obviously normal to the uv plane.
With this choice, the momentum is parallel or antiparallel to the w axis whether the PCA is (let' s say) at left side or right side of the tube (it is more clear from the drawings...).

Even if you add a variable which tells the direction forward or backward, the SPU variable is not univocally defined, because it depends on the plane frame orientation; it is given by: spu = sign[p ·(u x v)] (see in FairGeaneUtils for example). I can have a propagation along a track in

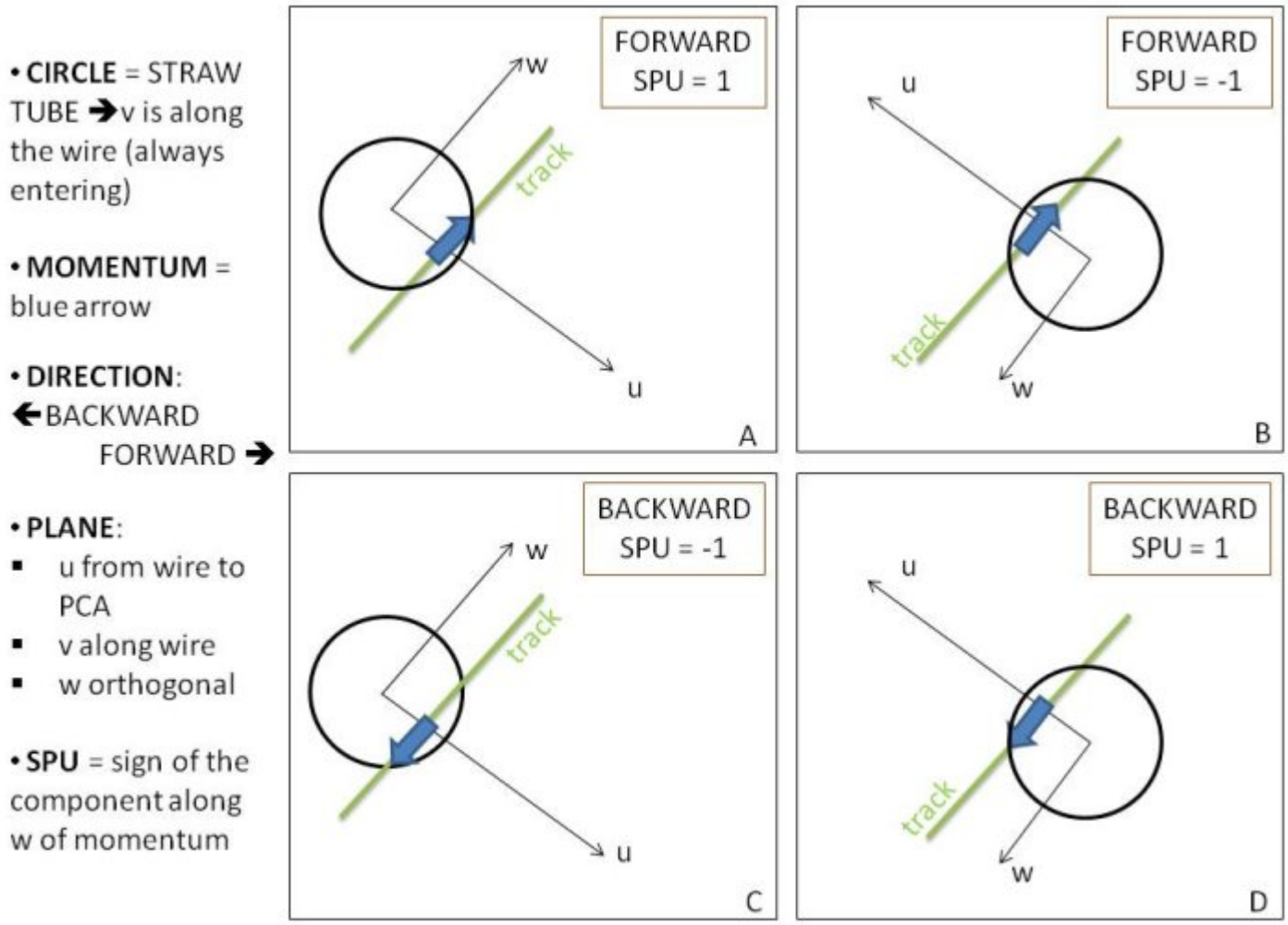
the forward direction alone and see the spu changing from one plane to another one.
So, let' s suppose we have all the hits ordered from the vertex and we are propagating forward, then at the beginning we give as input to kalman/geane the StartPos and StartMom in the master reference system, together with the starting detPlane, and so the starting spu is calculated. Then geane extrapolates to the first plane and on that plane it knows both the momentum direction and the frame orientation and so it can give the spu value (together with the state and cov on that plane).
Concerning the state and the cov they are not immediatly updated in trackrep, since before doing this we have to make the (kalman) filter step calculations (i.e. the wheighted mean) and only at the end of this step the updated state and cov are set to the trackrep in order to be used as starting point for the next extrapolation. So it should happen to the spu: the problem is that the spu is not updated here at the moment, but it is updated just after the extrapolation step and this causes problems.

I tried to find a way to update the spu (or to calculate it in such a way that it does not need to be updated at each step), but up to now I didn' t find one. One solution could be to add this variable (but without inserting it in the state vector). Anyway, I want also to try once again to calculate it inside GeaneTrackRep, to avoid the need of this new variable... I will let you know, but I already tried to do this without success ...Let' s see if I will have more luck!

Ciao,
Lia.

File Attachments

1) bckfwd.jpg, downloaded 1774 times



Subject: Re: Question on GeaneTrackRep.
Posted by Anonymous Poster on Tue, 10 Mar 2009 15:43:28 GMT
View Forum Message <> Reply to Message

Hi Lia,

I will look into your message in more detail tomorrow, but there is the main thought I want to reply even before:

I think the spu parameter, which I propose to use as 6th track parameter, actually defines the direction of the track for extrapolation. You have a DetPlane in which the track parameters are defined. So, if you have to extrapolate to another plane, you can determine whether you need to propagate the track forwards or backwards. With a plane with a defined orientation and the spu parameter the direction is fully defined. Am I missing something here?

Also, I didnt want to suggest to leave it to geane to decide whether to use "b" in the extrapolation, it is GeaneTrackRep.

Cheers, Christian

Subject: Re: Question on GeaneTrackRep.
Posted by Lia Lavezzi on Tue, 10 Mar 2009 16:14:45 GMT
View Forum Message <> Reply to Message

Quote:
You have a DetPlane in which the track parameters are defined. So, if you have to extrapolate to another plane, you can determine whether you need to propagate the track forwards or backwards. With a plane with a defined orientation and the spu parameter the direction is fully defined. Am I missing something here?

Ok, if you have a plane with a defined orientation and the spu parameter you have the momentum direction fully defined (I agree), but this does not define if you are propagating forward or backward (you decide it yourself).
This is what I mean: spu is connected to the momentum direction in the plane frame and not to the extrapolation direction (fwd or bwd) along the track... with the same spu you can go forward or backward.

I understand you would use the information on the momentum direction to decide whether to go forward or backward (is this correct?), but, as I said, I think this should be decided a priori and not "hit by hit"...

Ciao,
Lia.

Subject: Re: Question on GeaneTrackRep.
Posted by Anonymous Poster on Tue, 10 Mar 2009 17:03:52 GMT
View Forum Message <> Reply to Message

Hi Lia,

I know it is better to decide it before, and in most cases we will do so effectively. The point is that in the TPC the points are very close and sorting will go wrong for some hits. And I am saying that in the trackrep you will catch these cases and decide for the right hits to go in the other direction.
Is there a problem with my idea?

Thanks for this discussion. This is fun!

Christian

Subject: Re: Question on GeaneTrackRep.
Posted by Lia Lavezzi on Tue, 10 Mar 2009 18:25:51 GMT
View Forum Message <> Reply to Message

Hi Christian,

thank you for the explanation. No problem with this idea!
I think it might be feasible to "correct" things the way you suggest and if there is a small number of hits in the wrong place I guess they can be recovered by changing the track direction... or you could avoid to consider them, but in this case you lose some info. I guess you should try and see which is the best solution...

Ciao,
Lia.

Subject: Re: Question on GeaneTrackRep.
Posted by Anonymous Poster on Thu, 09 Apr 2009 16:11:52 GMT
View Forum Message <> Reply to Message

Hi,

just to sort of close this thread: As I wrote in my other message this afternoon, the direction problem is solved. The solution is actually pretty simple. If you should extrapolate from you point to a plane, you calculate the perpendicular vector to that plane. Then you decide by the scalar product with your momentum vector whether it is a forward or backward extrapolation. It is in the GeaneTrackRep.cxx file and you can take a look.

The spu parameter is just saved inside GeaneTrackRep. I think this solution is fine. It is not a good idea to make it a 6th track parameter since we can not calculate any covariances and most importantly because we dont need it.

In fact I checked for the value of spu and not surprisingly it is equal to 1 all the time. It makes a lot of sense since we usually orient our detector planes such that the normal vector points away from the interaction point. But this is not necessary and it should work just fine, if you'd do it differently.

Happy Easter!!

Christian
