

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:  $spu = \text{sign}[p \cdot (u \times v)]$  (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 immediately updated in `trackrep`, since before doing this we have to make the (kalman) filter step calculations (i.e. the weighted 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 1490 times

