

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.