Exercises in Tracking Algorithms (Detector school Oct 2012)

1)Consider our standard spectrometer with 4 or 6 planes:
http://www.nbi.dk/~phansen/nordforsk/spectrometer.C

This is a non-OO ROOT script[1] that simulates a compact spectrometer with very thin CMOS pixel layers (the special 50 micron MIMOSA chip - normal chips are at least 300 microns thick) and a weak magnet in the center. The goal is to measure positron production in the range 50MeV-1GeV from a diamond target just upstream of the first plane. The script simulates single positrons emitted along the x-axis. They are traced through the spectrometer, including multiple scattering, and then digitized including noise, resolution tails and inefficiencies.

A single track is reconstructed, first using a Kalman Filter for pattern recognition and then a global linear chisquared fit, requiring hits in all planes.

1a) Insert everywhere following an //INSERT comment in the functions *kalmanFilter* and *globalChi2* the required matrix code as in the slides.
Warning: If you transpose or invert a TMatrixD (like HT=H.T() or Cinv=C.Invert()) then H or C is *also* left in the transformed state, so you need to do H.T() or C.Invert() again to restore H or C – or some other trick.

1b) Is the spectrometer configuration optimal for its purpose? The length (try e.g. 60cm)? Should we add two extra planes (numberOfPlanes 2->3)? The adjustable noise occupancy (try 10^-4 with eff=0.98)? Use the 1/p residuals, the chisquared and the reconstruction efficiency as quality monitors. Try both 0.05 and 1 GeV in each case.

1c) How would you measure efficiency with data? Which improved flexibility in the code would be required in that case?
In fact you can (and should) do it easily for the last plane.

1d) The bremsstrahlung on slide 47 is not taken into account. How serious is the trouble you expect from this?

2 Now let us consider the case of our spectrometer being traversed by two oppositely charged pions from a K0 decays which has happened somewhere upstream. The measured state vector is the fitted one (z0, z', y0, y', q/p) with its covariance matrix. In this notation z'=dz/dx.

2a) Can you write down an initial estimate of the decay vertex v0 and the derivative matrices D and E from slide 66 and 67? Take, for example, the state vector x as (z,z',y,y',1/p) at the first plane.

---

[1] There is a much more sophisticated framework, using GEANT4 and supporting many track models, in the GENFIT package (genfit.sourceforge.net/)
It is still ROOT-based and reasonably lightweight.

2b) Assume then that we know the beam of K0's is propagating along the x-axis with a Gaussian beam profile of sigma b (cm). Let us also assume that the decaying particles are really K0. How could you take advantages of these "constraints" in the Billoir vertex fit?

3) The same script (6-plane option) with added alignment code is found in
http://www.nbi.dk/~phansen/nordforsk/align.C
Here we use a 100 GeV pion beam with no magnet and no target.

We imagine each plane is shifted randomly in y and z with a sigma of 100 microns.

Both the local and global method (see slide 82 and onwards) are implemented (the first and second derivatives of chisquared, as well as the average residuals, are accumulated in the function *accumulateResiduals)*.

3a) What accuracy seems to be obtained? How does that compare with the expected statistical errors on the alignment corrections? Are there any sign of "weak modes" giving trouble? Look at the plot of the alignment error versus plane number and the printout. Does it help to make iterations? Try 3. Is global better than local?

3b) We have only considered shifts parallel to the nominal planes. How would you implement in the alignment, for example, the possibility of some plane being rotated around its local z-axis?

Send answers to phansen@nbi.dk.
Peter Hansen