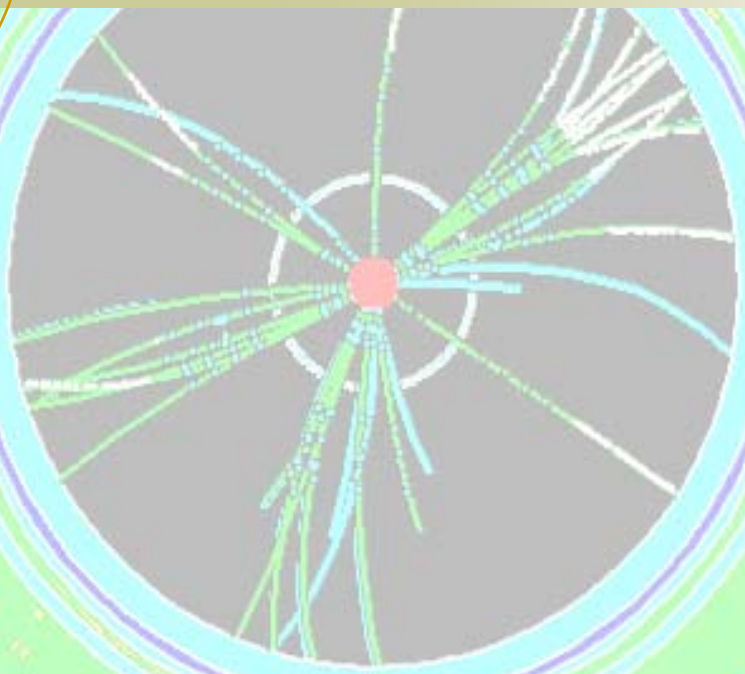
Track and vertex reconstruction

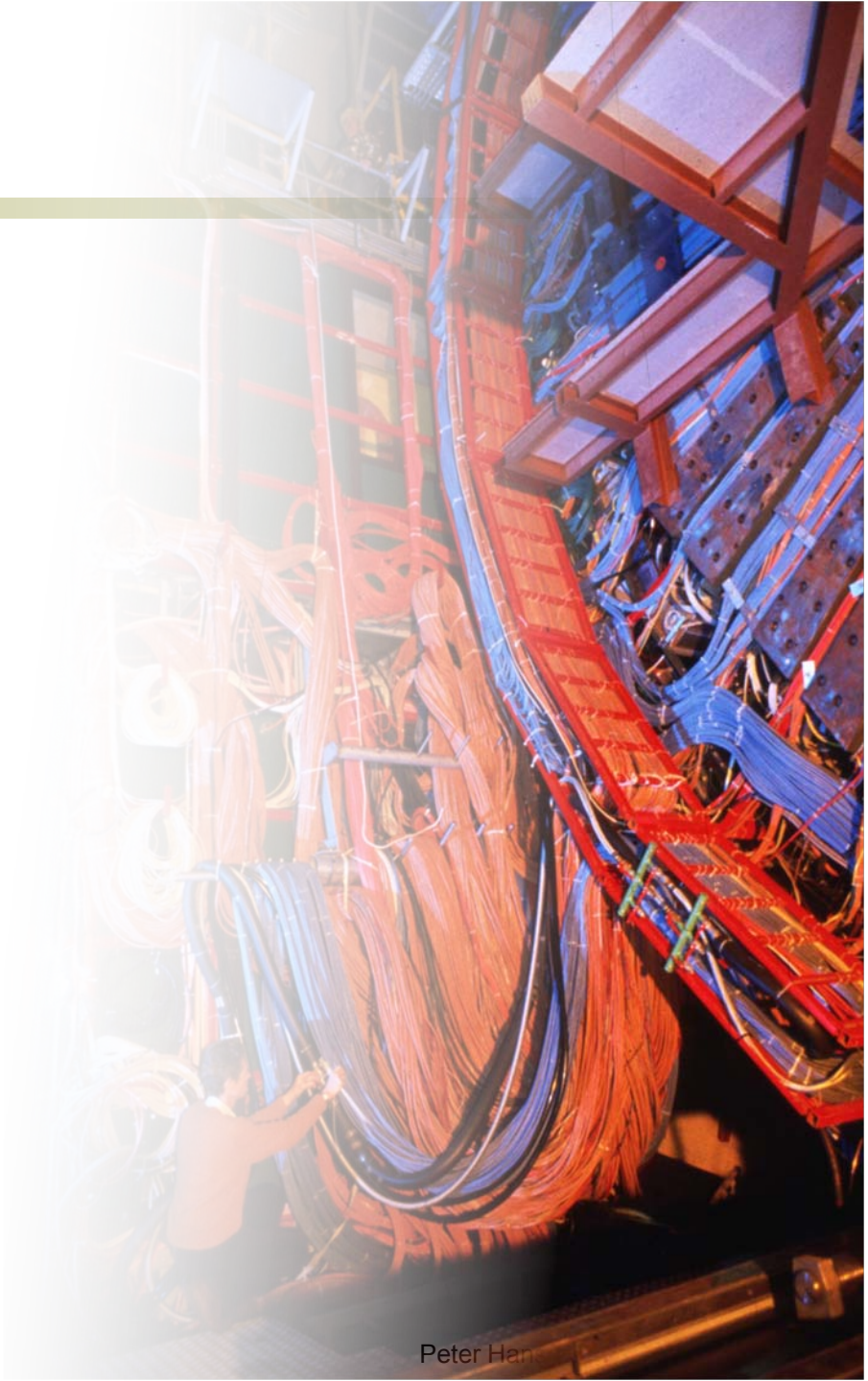


Peter Hansen, 12/9/08, NordForsk lectures

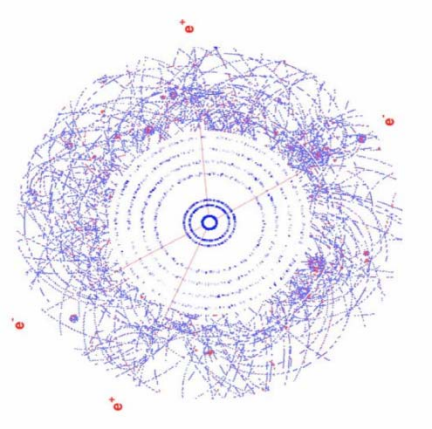
Overview

- Spacepoint formation
- Spacepoint calibration
- Pattern recognition
- Track fitting methods
- Vertexing and b-tagging
- Alignment

Many Thanks to colleagues from ATLAS



The tracking challenge



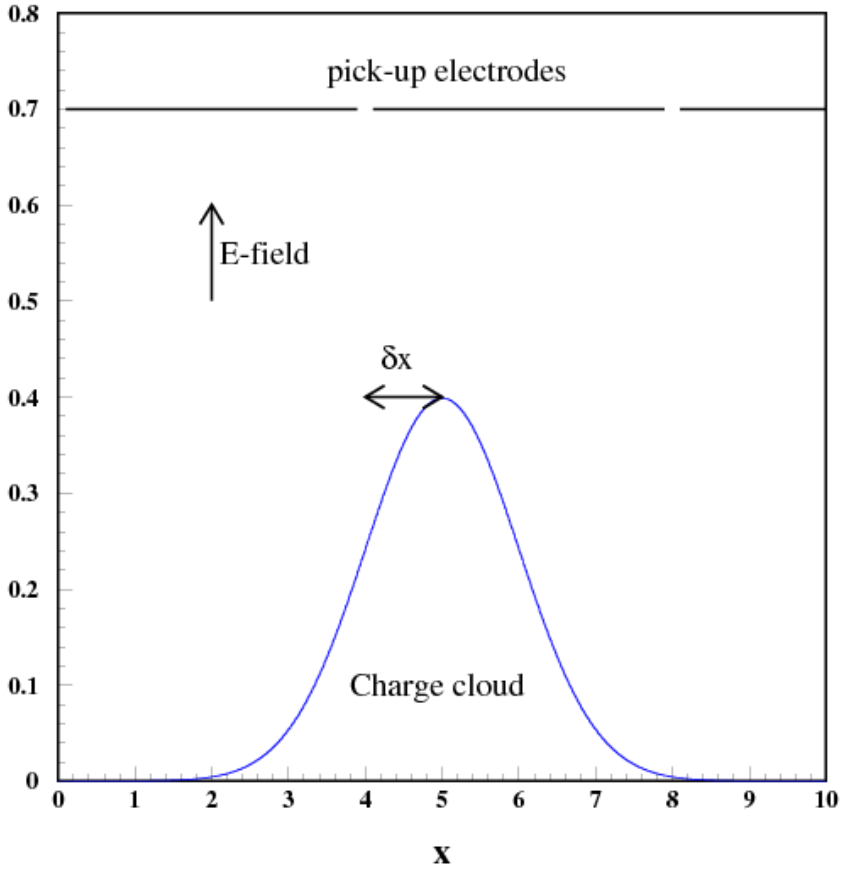
- Several 100 events, some with thousands of tracks, from several different collisions, are saved to mass-storage each second at the LHC.
- To cope with the high density and high momentum of these tracks, many, many channels are needed, some unavoidably noisy, as well as relatively large amounts of material (0.5-2 rl) in the tracking detectors.
- Still, the ambition is high precision tracking, in order to perform, for example, a 10 MeV W mass measurement.
- This requires highly sophisticated and error-tolerant track-finders and -fitters, as well as highly performant calibration and alignment of the tracker elements.

Spacepoint formation

- Most tracking detectors register “hits” from signals induced on pickup electrodes by an *electron cloud* made by a track.
- If all the signal is on one electrode, the precision is $\Delta / \sqrt{12}$ where Δ is the electrode size.
- Much better is it if the signal is distributed over two electrodes.

In that case $$\frac{\delta x}{\Delta} = \sigma^2 \log \frac{P_1}{P_2}$$

but you need to measure P and know σ….



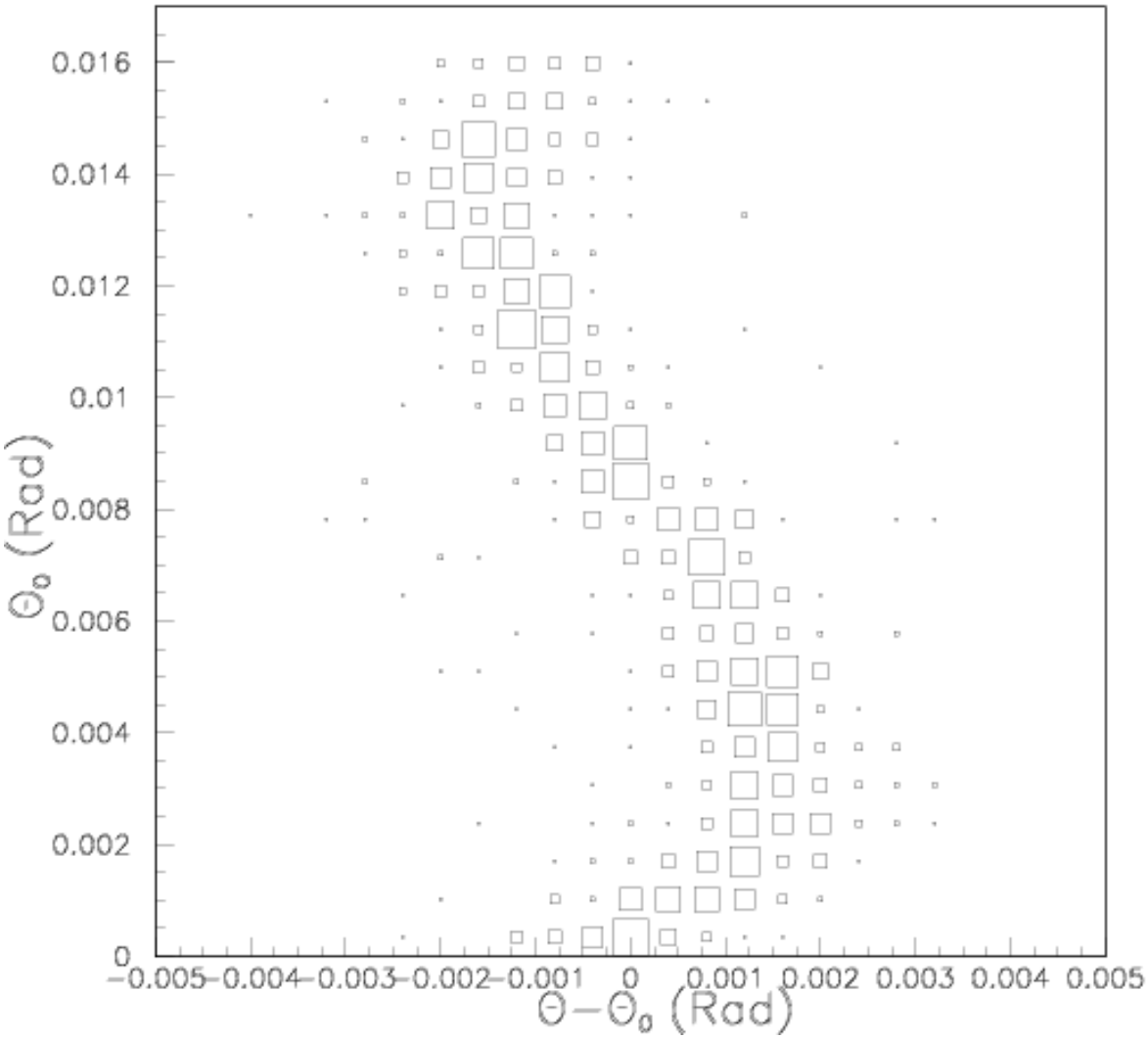
Spacepoint formation

- If 3 or more electrodes pick up signal for one track passing the detector layer, it is possible to fit the width of the electron cloud. In that case the barycenter (pulseheight weighted average of the electrode centers) is a popular estimator of the track position. The pulseheights must exceed a certain threshold and the electrodes must be spatially connected in order to identify clusters of electrodes belonging to different tracks:

$$x = \sum P_i x_i \,/\, \sum P_i$$

Spacepoint formation

- The barycenter is not perfect because of the finite size of the electrodes.
- This example is for 3x3cm electrodes in a lead-gas sampling calorimeter. You see that the estimate is only unbiassed at the border between two or at centre of one.



Stereo view

- If you do not have pixel detectors or a TPC, what about the second coordinate? (the third being the position of the detector *surface*.)
- In strip detectors double sided wafers are often used with strips on both sides having an angle between them.
- In e+e- machines, where a particular wafer is rarely hit by more than one track, 90 degrees is appropriate.
- In high track densities, 20-80 mrad is a better choice in order to avoid too many ghost hits.
- With surfaces ~perpendicular to particle velocity and strips ~parallel to B, this gives good resolution in the bending plane and some resolution in second coordinate.
- For drift tubes the second coordinate is mostly unmeasured.

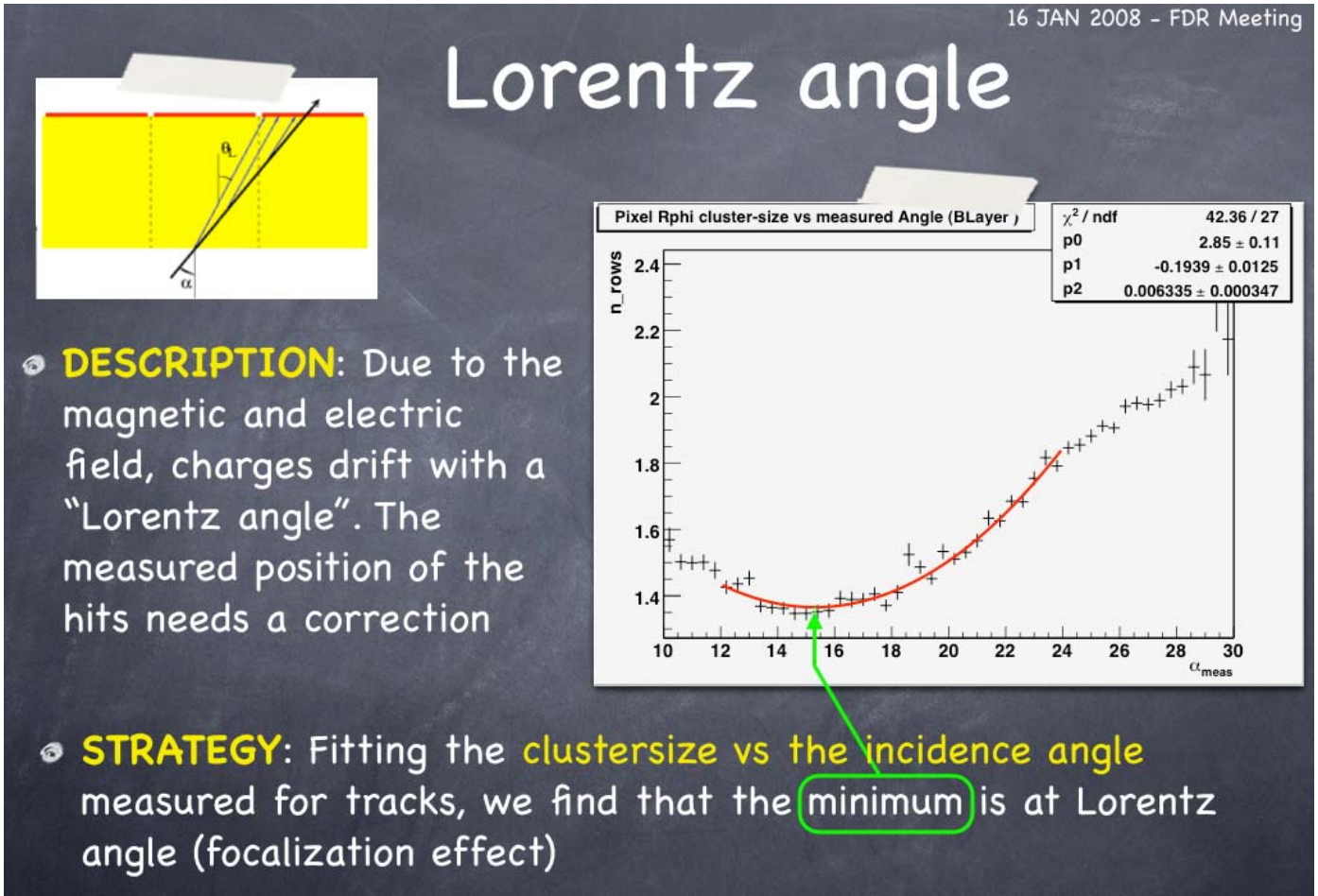
Spacepoint calibration

- In general we must know *the response function* , the probability distribution of induced pulseheights for a given track impact (actually, we would like the inverse: the pdf for the track impact, given the pulseheights. But unfortunately we do not measure this directly).
- The response function may vary from channel to channel and must be calibrated from data.
- The nightmare is when this function also varies in time ,for example due to radiation damage. In this case, massive calibrations are required at each run-period.

Spacepoint calibration

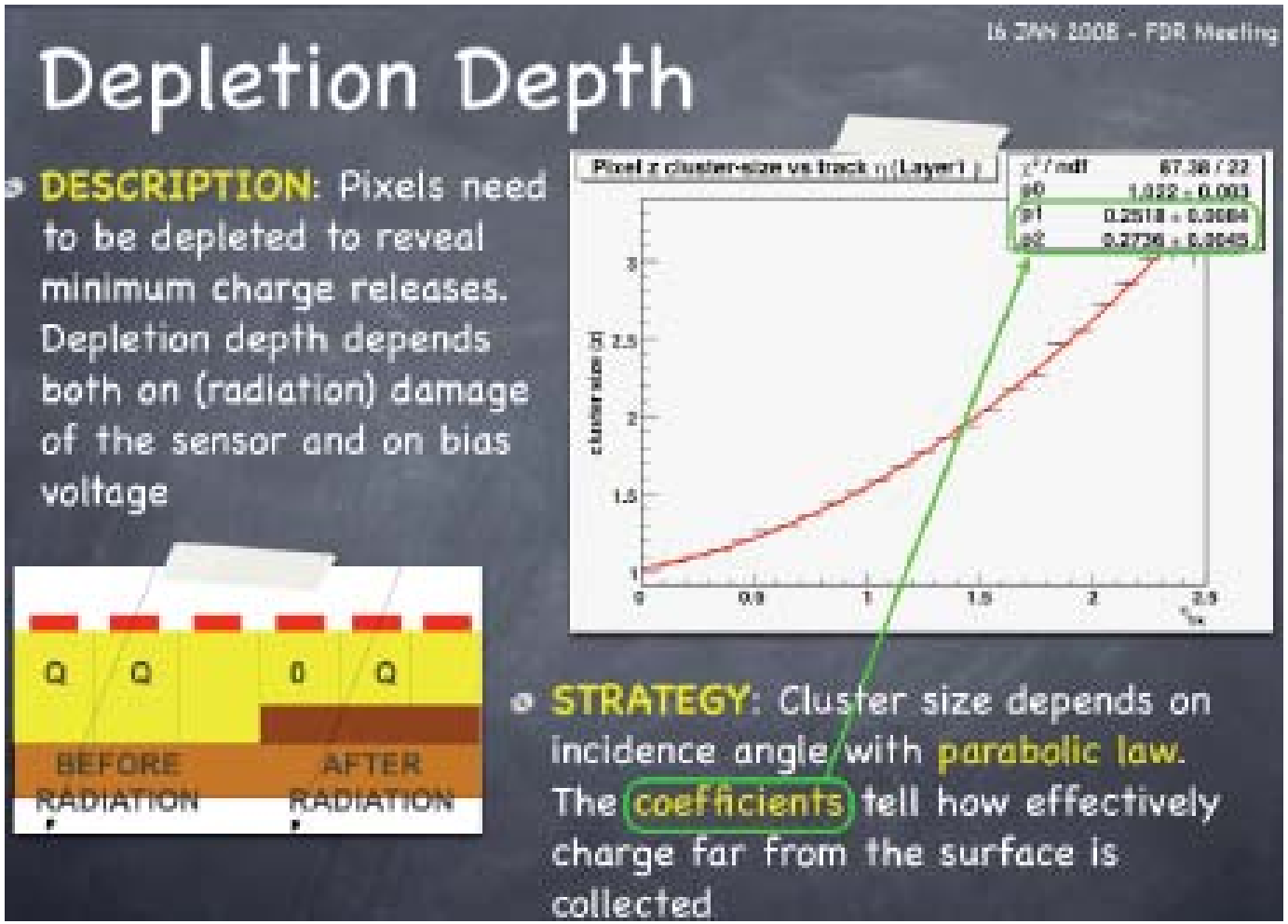
- Corrections are needed for crossing angle and change in electron drift direction caused by B.

  (pictures from Simone Montesano)



Spacepoint calibation

- The corrections can depend on hardware calibrations – different from module to module

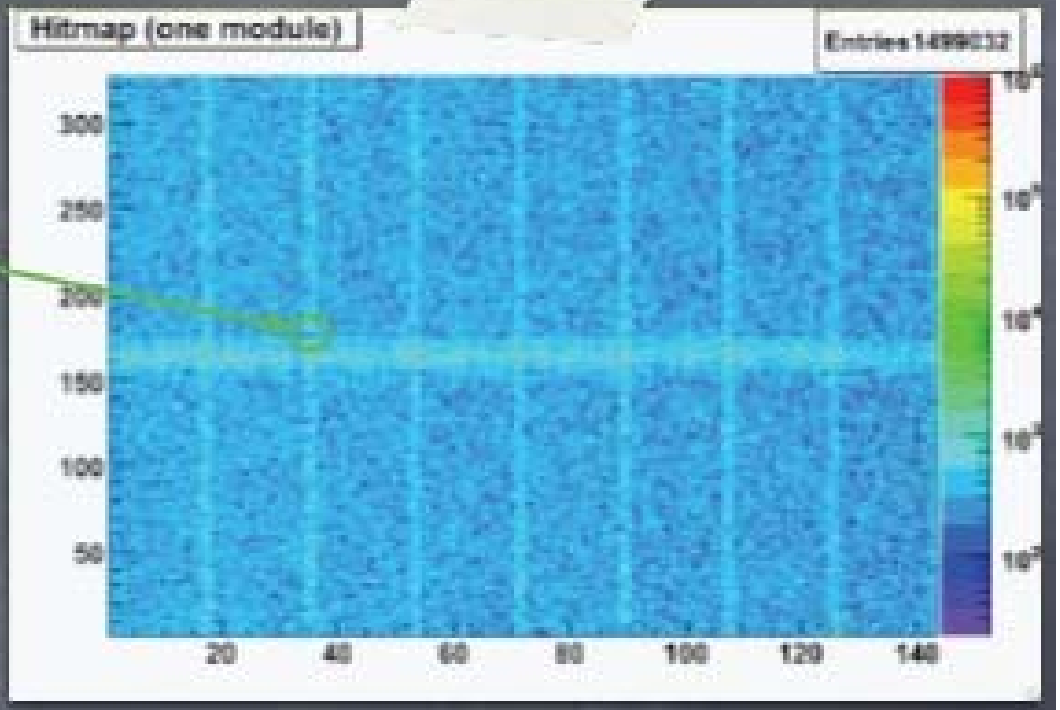


Dead and noisy channels

- Any clustering algorithm must handle dead or noisy channels to avoid false or split clusters.

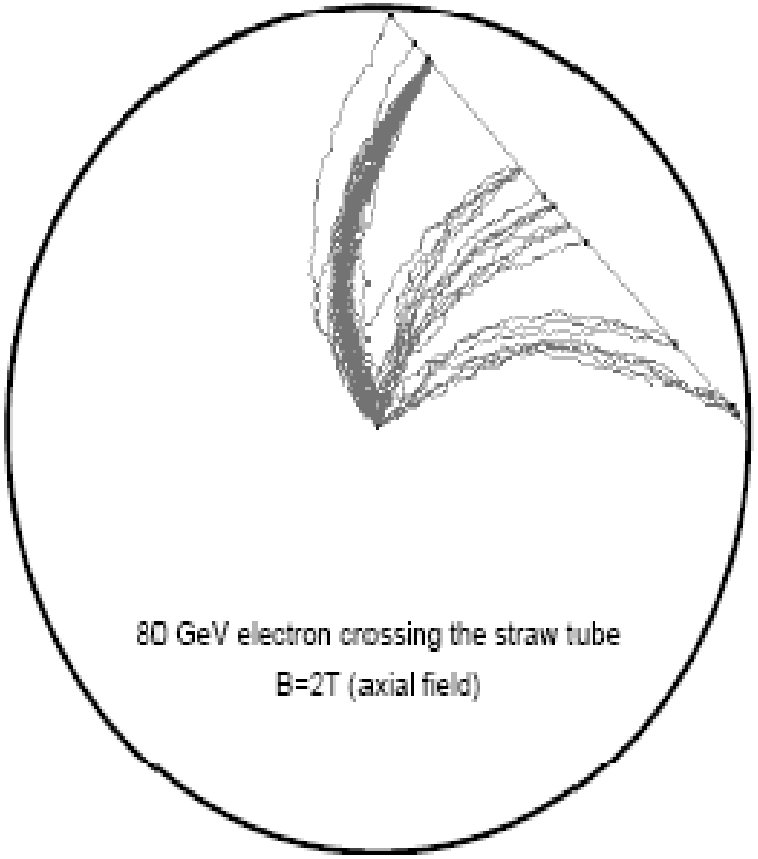
DESCRIPTION: some pixels have intrinsic high occupancy (noisy) other are not working (dead), during reconstruction we "mask" special pixels

STRATEGY: simply plot the occupancy and decide a threshold for dead and noisy. BTW: for dead pixels we need O(10^7) events!

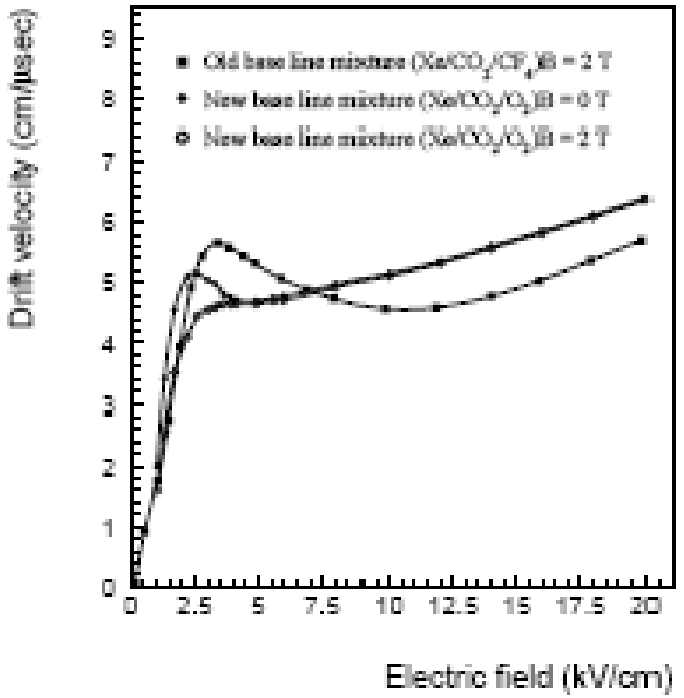


NB: this will be done with PixelMonitoring histograms!

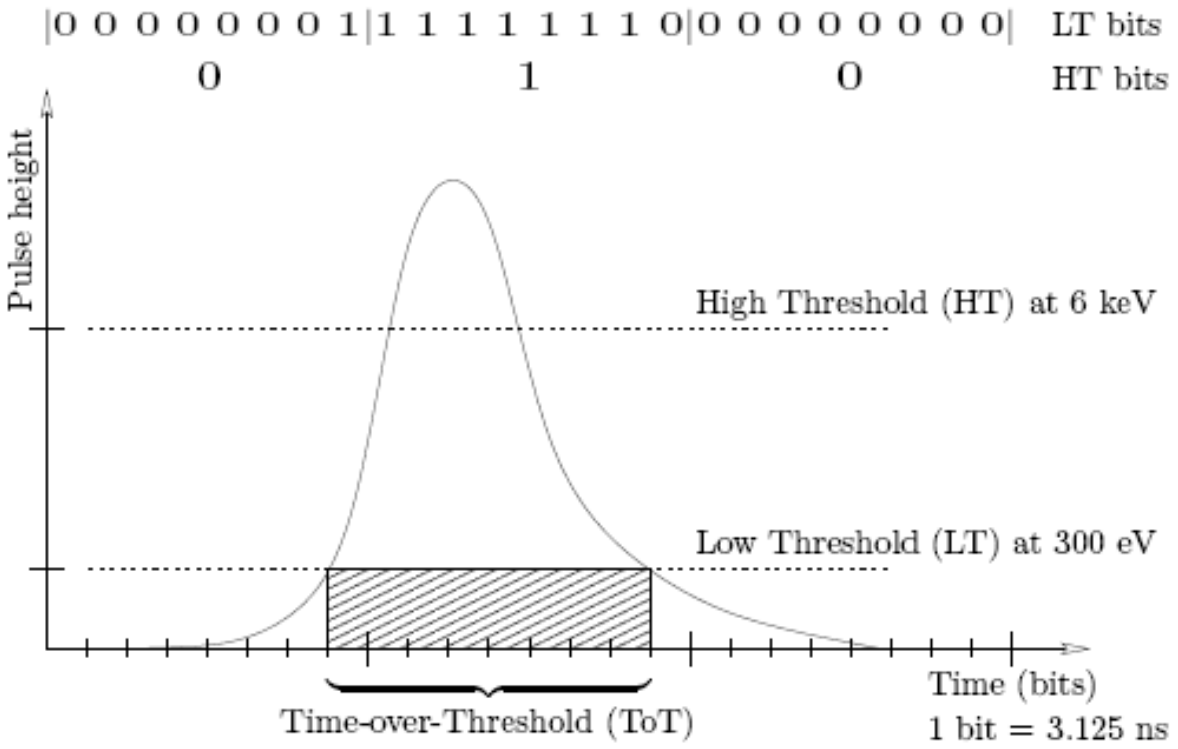
Spacepoints in drift-tubes



Drift velocity in different gasses.Measure time t at which signal exceeds some threshold. Must calibrate the distance R(t-t0) from the track to the wire.



Example: ATLAS TRT



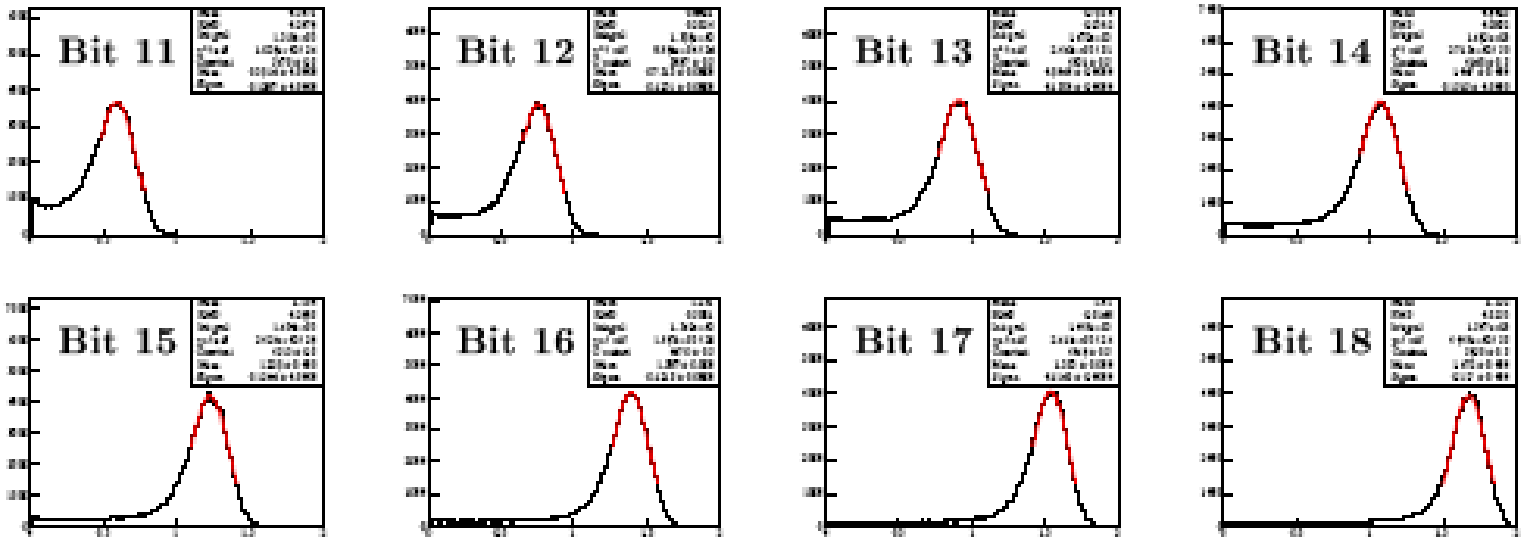
Picture from Thomas Kittelmanns thesis

Drift tube hit calibrations

- Both T0 and R(t-T0) may vary from channel to channel
- Final adjustments – not available at hit preparation time – may also be needed:
- Time-of-flight and signal propagation time (depending on the coordinate along the length of the tube)
- Pulseheight, if available (electrons will typically deposit more energy, so that the 300 eV threshold is reached earlier)

Drift tube calibration

- Either the minimum value of Rtrack(t-T0) yields T0, or a correction is obtained from a Gaussian fit to (Rtrack-Rhit)/v.
- The peak Rtrack in each 3ns bin of t-T0 yields Rhit(t-T0)
- (Note that the average position is not good because of tails at long arrival times for tracks passing close to the wire)



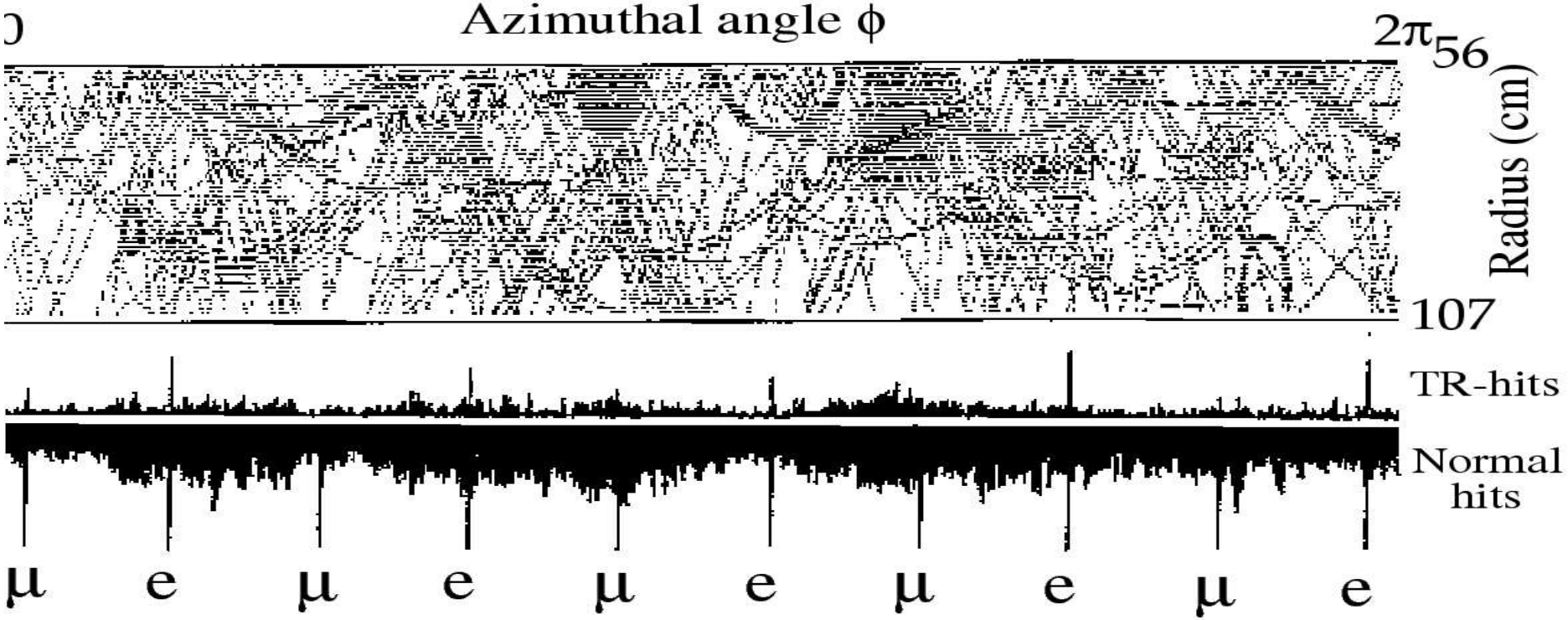
Pictures from Thomas Kittelmanns thesis

Pattern recognition

- The simplest method is to predefine a number of *templates*, ie patterns of fired cells that define an allowed track. The templates can be arranged in a hierarchical structure of increasing granularity to avoid too many templates.
- Such methods are often used in trigger algorithms. An example from the calorimeter world is the cell tower.
- Each hit has an associated surface in the feature space of compatible tracks. This is called a *Hough transform.* For straight tracks in two dimensions, each hit corresponds to a straight line in the slope-intercept plane. Locations in this plane where many lines intercept reveal the tracks, and this can be used in early track-finding.

Pattern recognition

- Histogramming methods (use-cases of Hough) may provide fast seeds for high momentum tracks



- by a ”conformal transformation” also low momentum seeds can be found by histogramming.

The Kalman-filter

- Determines the track state vector dynamically from measurements at each detector surface. These are either discarded or used to update the existing state vector.
- Needs only inversion of small matrices. Fast.
- Can account for noise, multiple scattering and energy loss at each surface. Efficient.
- Is equivalent to the least squares fit, but can provide also pattern recognition (you can skip a plane if the hit is too far from the prediction, count the number of skipped planes and eventually drop the track candidate).
- Hence its popularity.
- Ingredients in the following.

The state vector x

- At each detector *surface* the track has a *state vector* of *track parameters*.
- Example: a helix, where 90-λ is the track angle to the B field, R is the radius, s the path length and h is a sign

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x^0 + R \cdot (\cos(\alpha_0 + \frac{hs}{R} \cdot \cos\lambda) - \cos\alpha_0) \\ y^0 + R \cdot (\sin(\alpha_0 + \frac{hs}{R} \cdot \cos\lambda) - \sin\alpha_0) \\ z^0 + s \cdot \sin\lambda \\ \end{pmatrix}$$

Perigee parameters

- The perigee parameters

$$\overline{x} = (d_0, z_0, \phi_0, \theta, \frac{q}{p})$$

are often used to describe the track state at the closest point to the beam axis(z). We use q/p instead of qp, because q/p is generally measured with a gaussian uncertainty (see later slides). The impact parameter d0 has some sign convention, for example according to the angular momentum of the track wrt the beam axis, oriented along z.

The projection matrix H

- In order to compare with measurements, the track state needs to be projected onto the measurement frame.
- Consider, for example, a set of strips forming an angle α with the x axis. Let the track parameters be x and y at the surface. Then we must do

$$H\begin{pmatrix} x \\ y \end{pmatrix} = (\cos\alpha \quad -\sin\alpha)\begin{pmatrix} x \\ y \end{pmatrix}$$

in order to arrive at the predicted measurement in terms of strip counting.

The propagator F

- Let the track transport from layer k-1 to k be given by

$$\tilde{x}_k = f(x_{k-1})$$

- This is then the predicted state (denoted by a tilde). If f is not already linear in x, we can try to use its derivatives to get a linear equation:

$$\tilde{x}_k = F_k x_{k-1,} \quad C_k^{k-1} = F_k C_{k-1} F_k^T + Q_k$$

where Ck is the covariance matrix for the predicted state and Q contains the additional random perturbations occuring in the step such as, for example, multiple scattering.

The propagator F

- The propagator can be straight line or helix, but in regions with an inhomogenous B field, the preferred method is Runge-Kutta integration. Here the trajectory derivatives are sampled at a number of intermediate positions, weighted so that the error is 5^{th} power in h.

$$y' = f(t, y), y_0 = y_0(t_0)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

$$k_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

The residual r (covariance R)

- The difference between measurement m (can have several components) and the prediction of the track state x, is called the residual:

$$r_k^{k-1} = m_k - H_k \tilde{x}_k, \qquad R_k^{k-1} = V_k + H_k C_k^{k-1} H_k^T$$

where V is the diagonal covariance matrix of the measurements. H projects the track state onto measurement space.

(Note that the contribution from the track is here *added* to the measurement variance. If the hit contributes to the track, this contribution is instead *subtracted*, as we shall see).

The gain matrix K

- The matrix transforming the residual into a correction to the track state is:

$$K_k = C_k^{k-1} H^T (V_k + H_k C_k^{k-1} H_k^T)^{-1}$$

$$x_k = \tilde{x}_k + K_k r_k^{k-1}$$

$$C_k = (1 - K_k H_k) C_k^{k-1}$$

where x (no thilde) is now the updated (filtered) state.

(check with a one-dimensional update of weighted averages)

(from R.Mankel, Prep. Prog.Phys. &7 (2004) 553)

Alternatively:

- A completely equivalent update algorithm is to take a weighted average of the predicted track state and the state suggested by the new measurement:

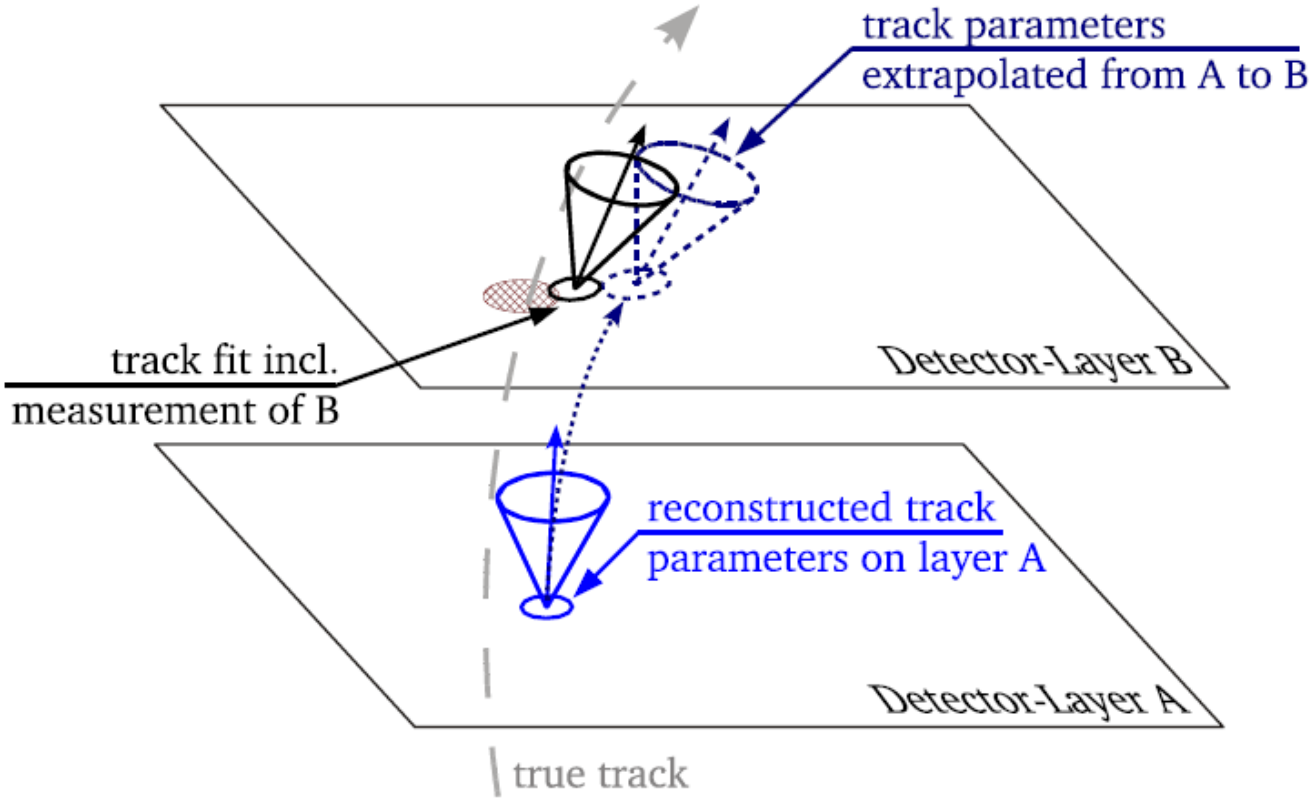
$$x_k = C_k((C_k^{k-1})^{-1} x_k^{k-1} + H^T V_k^{-1} m_k)$$

$$C_k^{-1} = (C_k^{k-1})^{-1} + H^T V_k^{-1} H$$

Filtered residuals

- The filtered residual, its covariance and chi2 are

$$r_k = (1 - H_k K_k) r_k^{k-1}, \quad R_k = (1 - H_k K_k) V_k, \quad \chi^2 = r^T R^{-1} r$$



Smoothing

- We have reached the end with n hits. Now the procedure is repeated, but backwards. This is used to update each measurement k with the information from all the others:

$$C_{k|n}^{-1} = C_{k|k}^{-1} + (C_{k|k+1}^{b})^{-1}$$

$$x_{k|n} = C_{k|n}(C_{k|k}^{-1} x_{k|k} + (C_{k|k+1}^{b})^{-1} x_{k|k+1})$$

  If the measurement at plane k is more than a few sigma from updated x , it will typically be ignored in the fit – as early as possible. Rejected measurements may be flagged.

- This policy turns out to yield better momentum resolution while retaining good track efficiency.
- Finally the innermost surface is extrapolated to the perigee.

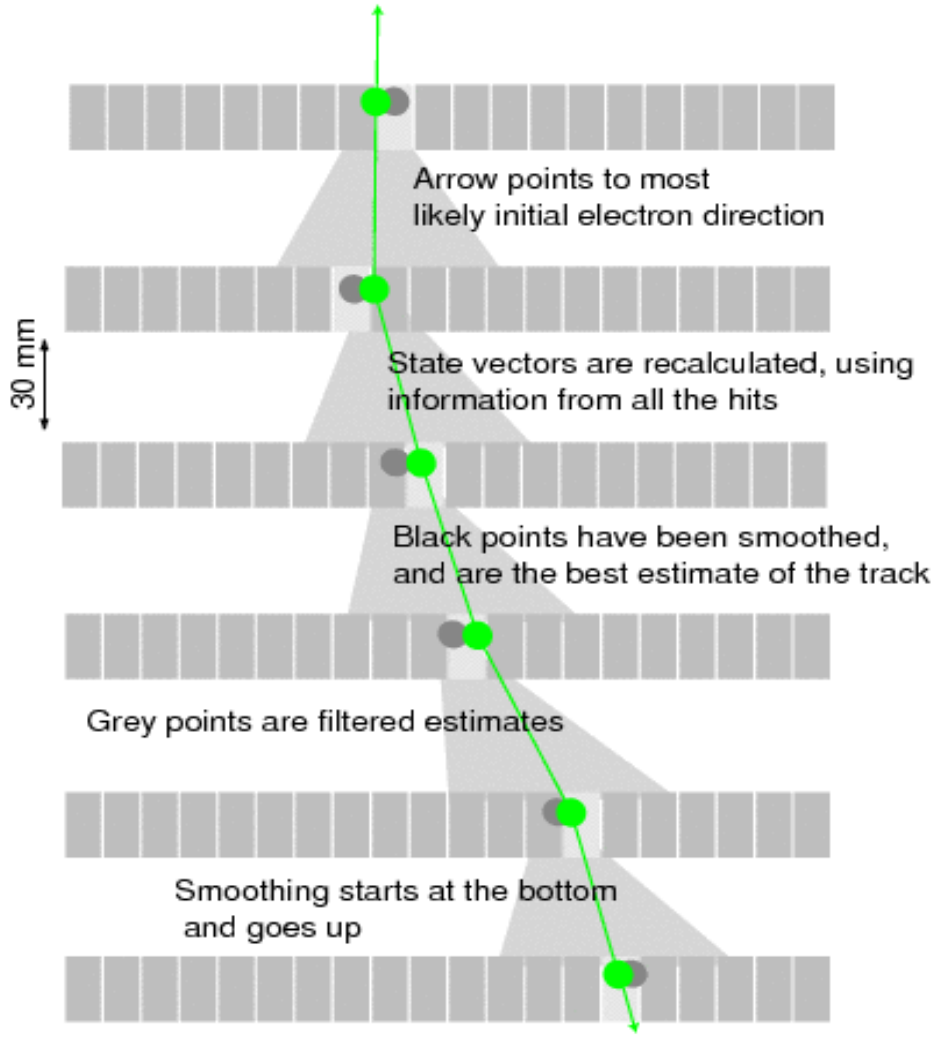


Fig. 2.— The Kalman smoothing process.

Newton-Raphson (global) fits

- ”Global” least-squares fitters minimize the distances between the fitted track and the assigned measurements, using a linear approximation for the dependence of the measured coordinates on the track parameters. Minimizing

$$\chi^2 = (\bar{m} - H\bar{x})V^{-1}(\bar{m} - H\bar{x})$$

where m and x now refers to *all the surfaces* yields

$$\bar{x} = (H^T V^{-1} H)^{-1} H^T V^{-1} \bar{m}$$

For normally distributed m, this is directly the maximum likelihood estimate of the parameters.

Newton-Raphson fit

- If the projection h(x) is not linear, we can still Taylor expand at an initial value x0 obtaining approximately:

$$\frac{d\ \chi^2}{dx}(x_0) = 2H^T V^{-1}(m - h(x_0))$$

$$\frac{d^2\chi^2}{dx^2}(x_0) = 2H^T V^{-1} H = Cov^{-1}(x_0)$$

yielding

$$x_1 = x_0 - (\frac{d^2\chi^2}{dx^2})^{-1}\frac{d\chi^2}{dx}$$

Then x1 may not be exactly the minimizing value – but it is, after all, better than x0.

Global optimization

- In dense track environments or very noisy environments, the problem arises of competing assignments of hits to the different tracks or to noise.
- Consider M tracks and N detector layers, each with nk hits.
- Let Mkia be the squared distance from track a to hit i in layer k.
- Let Skia be the ”assignment strenght” for hit i to be associated with track a. The Elastic Arms Algorithm then defines the problem as that of the minimization of an energy function

$$E = \sum_{a}^{M} \sum_{k}^{N} \left[S_{ka} \left(\sum_{i}^{n} S_{kia} M_{kia} \right) + \lambda \left(S_{ka} - 1 \right)^2 \right]$$

- Where Ska indicates summation over i.

Annealing

- The problem with the total energy minimization is that the energy landscape in track parameter space is very spiky and the risk is high to land in a local minimum of wrongly assigned hits.
- This problem is solved by *”annealing”:*
- One starts the track finding at ”high temperature” where the assignment probabilities are relatively large at large distances,
- then calculate new assignment probabilities according to a formula inspired by statistical mechanics,
- then repeat the process at a lower temperature, until T=0.

Deterministic Annealing Filter

- In a very complicated tracking environment it is not realistic to use a *global method* like the Elastic Arms where the approximate number of tracks must be known beforehand.

  Therefore Frühwirth and Strandlie proposed to modify the (*local*) Kalman filter using an assignment probability p_{ik}.
- Since the measurement probability density function is still assumed to be Gaussian, the assignment propability for n_k competing measurements is assumed to be proportional to a multivariate Gaussian:

$$\phi_k^i = \phi(m_k^i; H_k x_k, TV_k^i + H_k C_k^* H_k^T)$$

  where x here is the smoothed track state without involving layer k in the fit and T a temperature parameter. The last term, the track contribution to σ, can normally be neglected.

Deterministic Annealing Filter

- Allowing also for the hypothesis that no hit is assigned to the track in layer k, we normalise the assignment probs as

$$p_i^k = \frac{\phi_k^i}{\sum_j^{nk} (\Lambda_k^j + \phi_k^j)}$$

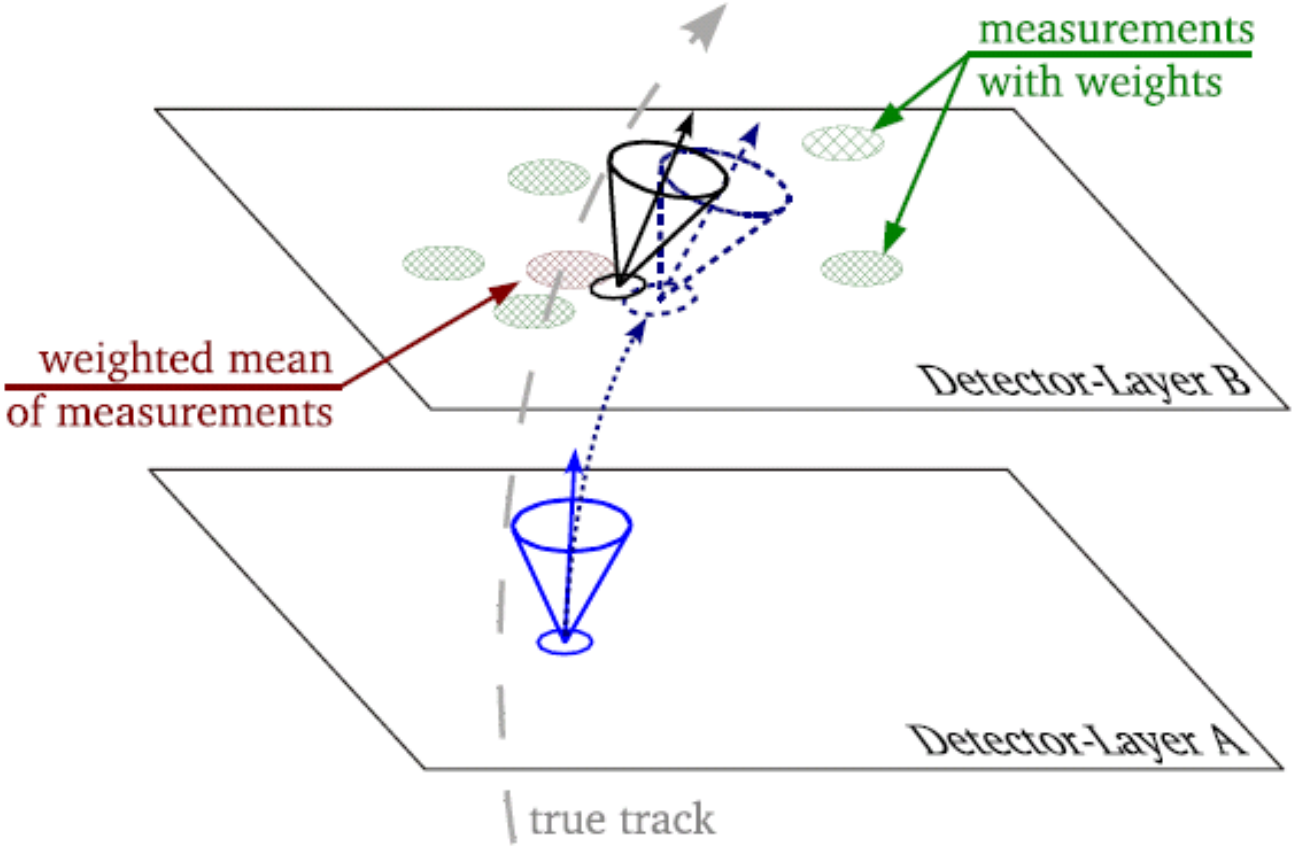
- The cut term is parametrised

$$\Lambda_k^j = \frac{1}{(2\pi)^{\dim(m)} \sqrt{T \det V_k^i}} \exp(-\frac{\lambda}{2T})$$

where λ acts as a χ^2 cut-off at low temperature.

Deterministic Annealing Filter

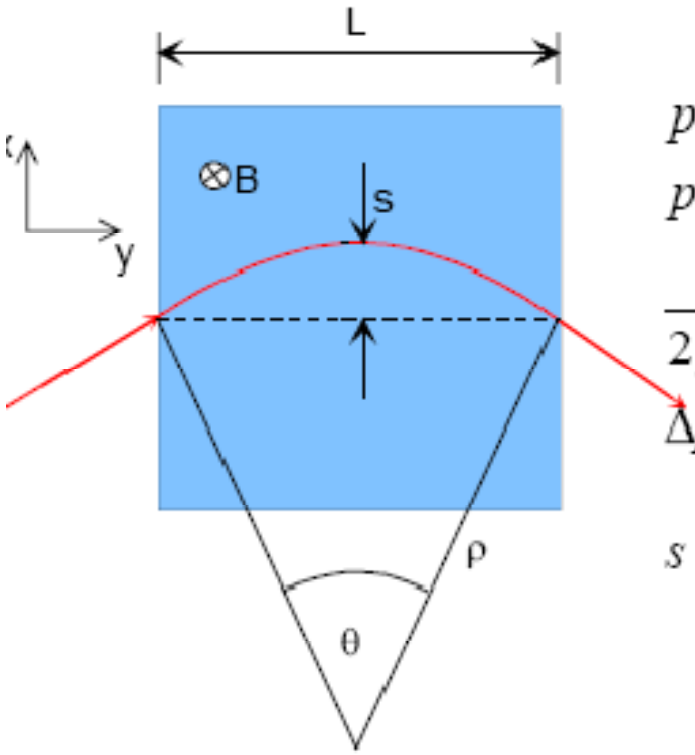
- Uses the same principles as the Kalman filter, but several measurements per detector layer are taken into account by using their weighted mean.



Deterministic Annealing Filter

- The Deterministic Annealing Filter has turned out especially effective in finding the best left-right choices in drift tubes.
- It is used by ATLAS as an ”afterburner” to the Kalman Filter and significantly improves momentum resolution.
- A promising possibility is to extend it to a multitrack fitter with its own pattern recognition. In this case the normalisation of assignment probabilities needs to be changed so that the sum runs over all tracks competing for the measurements.
- See Sebastian Fleischmanns thesis (ATLAS) for more detail.

Momentum measurement



$$p_T = qB\rho$$

$$p_T\ (\mathrm{GeV/c}) = 0.3B\rho \quad (\mathrm{T\cdot m})$$

$$\frac{L}{2\rho} = \sin\theta/2 \approx \theta/2 \quad \rightarrow \quad \theta \approx \frac{0.3L\cdot B}{p_T}$$

$$\Delta p_T = p_T \sin\theta \approx 0.3L\cdot B$$

$$s = \rho(1-\cos\theta/2) \approx \rho\frac{\theta^2}{8} \approx \frac{0.3}{8}\frac{L^2 B}{p_T}$$

From Christian Jorams summer student lectures

Momentum accuracy

the sagitta s is determined by 3 measurements with error $\sigma(x)$:

$$s = x_2 - \frac{x_1 + x_3}{2}$$

$$\left.\frac{\sigma(p_T)}{p_T}\right|^{meas.} = \frac{\sigma(s)}{s} = \frac{\sqrt{\frac{3}{2}}\sigma(x)}{s} = \frac{\sqrt{\frac{3}{2}}\sigma(x)\cdot 8 p_T}{0.3\cdot BL^2}$$

for N equidistant measurements, one obtains

(R.L. Gluckstern, NIM 24 (1963) 381)

$$\left.\frac{\sigma(p_T)}{p_T}\right|^{meas} = \frac{\sigma(x)\cdot p_T}{0.3\cdot BL^2}\sqrt{720/(N+4)} \qquad \text{(for N} \geq \approx 10\text{)}$$

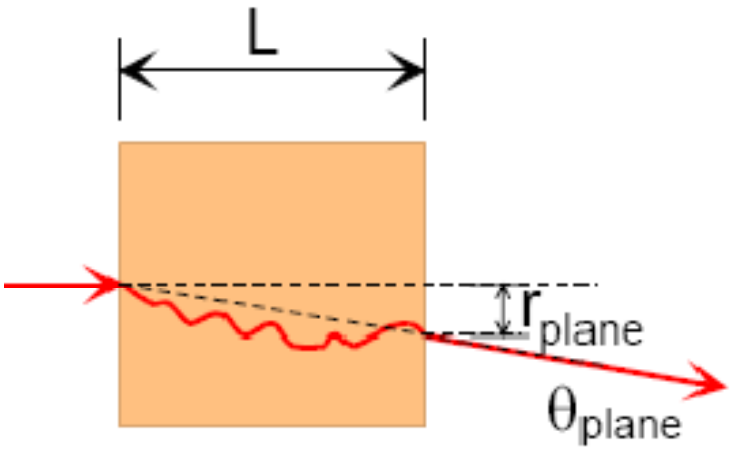
ex: p_T=1 GeV/c, L=1m, B=1T, σ(x)=200μm, N=10

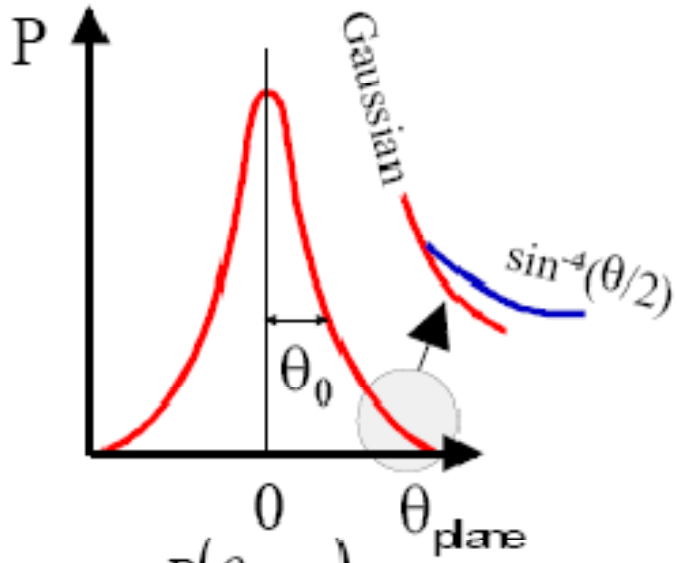
$$\left.\frac{\sigma(p_T)}{p_T}\right|^{meas.} \approx 0.5\% \qquad (s \approx 3.75 \text{ cm})$$

Multiple scattering

Sufficiently thick material layer

→ the particle will undergo multiple scattering.





$$\theta_0 = \theta_{plane}^{RMS} = \sqrt{\langle \theta_{plane}^{\ 2} \rangle} = \frac{1}{\sqrt{2}} \theta_{space}^{RMS}$$

$$P(\theta_{plane}) = \frac{1}{\sqrt{2\pi}\theta_0} \exp\left\{ -\frac{\theta_{plane}^{\ 2}}{2\theta_0^2} \right\}$$

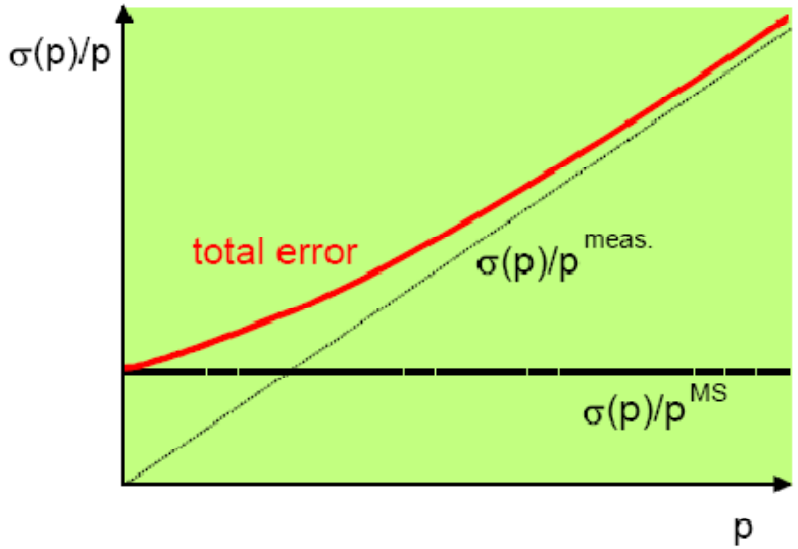
Total momentum error

contribution from multiple scattering

$$\Delta p^{MS} = p \sin\theta_0 \approx p \cdot 0.0136 \frac{1}{p}\sqrt{\frac{L}{X_0}}$$

$$\left.\frac{\sigma(p)}{p_T}\right|^{MS} = \frac{\Delta p^{MS}}{\Delta p_T} = \frac{0.0136\sqrt{\frac{L}{X_0}}}{0.3BL} = 0.045\frac{1}{B\sqrt{LX_0}}$$

independent of p !



Dealing with multiple scattering

- In the global least squares track fit, a finite number of scattering planes in the detector is defined and each MS angle is treated as an extra track parameter with an independent contribution to chisquared of $\left(\theta / \theta_0\right)^2$

- An alternative way is to incorporate MS in the covariance matrix of the measurements

$$V \rightarrow V + S\Theta S^T$$

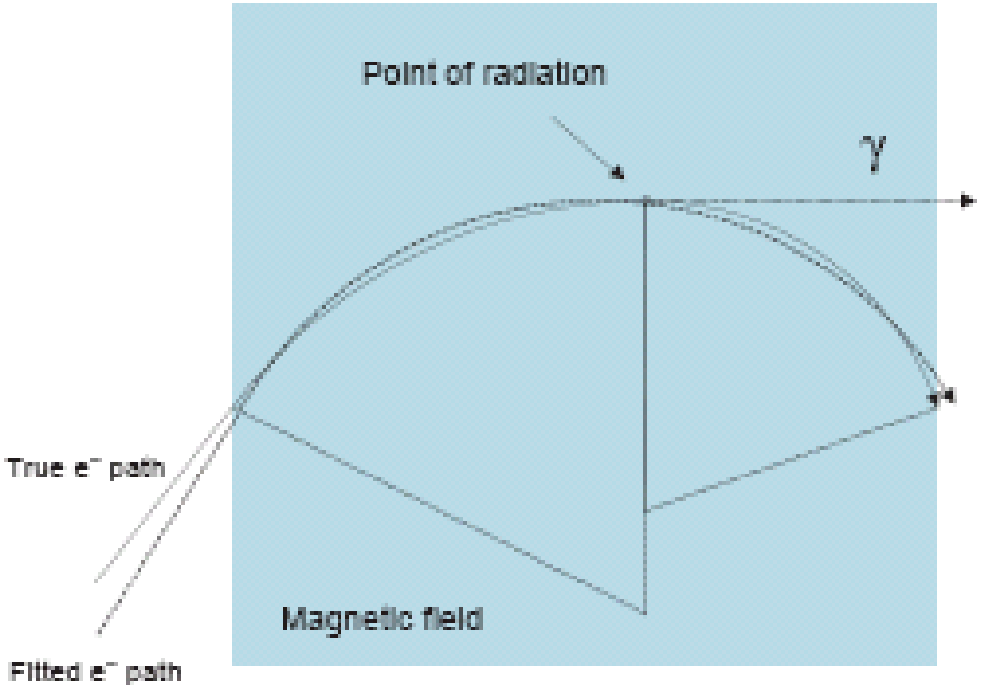
with $S = \partial r / d\phi$ $\quad \Theta_{jj} = \theta_{0j}^2$ for each scattering plane j.

This is more or less what happens in the Kalman filter.

Non Gaussian contributions

- All the methods presented up to now are complicated by non-gaussian influences, such as hard photon radiation where the probability density for the electron to retain a fraction z of its energy follows the Bethe-Heitler law

$$f(z) = (-\ln z)^{c-1}\Gamma(c) \qquad c = X_{rl} / \ln 2$$



Gaussian Sum Filter

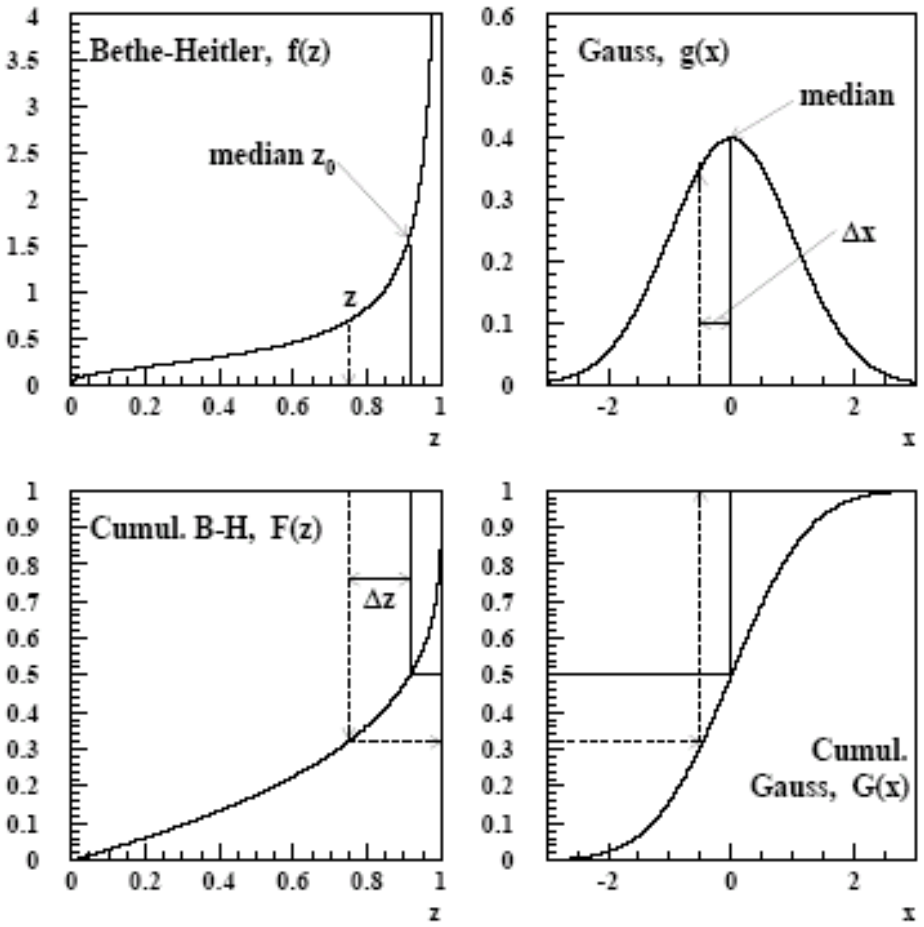
- One way to cope with non-Gaussian influences while retaining the techniques of the least squares fit is to branch the Kalman filter at each surface into parallel paths using a finite number of different Gaussian errors.
- The different paths carry weights according to the physics of multiple scattering, hard photons etc. The path with highest net probability is retained, preventing the occurrence of a measurement in the non-Gaussian tail to pull the track state as far away.
- This approach is very efficient in recovering from hard bremsstrahlung, but is also very CPU consuming due the multiplication of branches.

Dynamic Noise Adjustment

- To avoid the heavyness of GSF, another method can with success be applied to the bremsstrahlung of electrons.
- First, the z retained after a particular surface is estimated using the hits in the following planes.
- Then an adjustable noise level $\sigma(z)$ is calculated so that the Bethe-Heitler probability of a deviation from median z out to the found one equals that for $z=z(\text{median})+x\sigma(z)$, where x is drawn from a unit Gaussian.
- This way the Bethe-Heitler is mapped onto a Gaussian (google Kartvelishvili for more details)

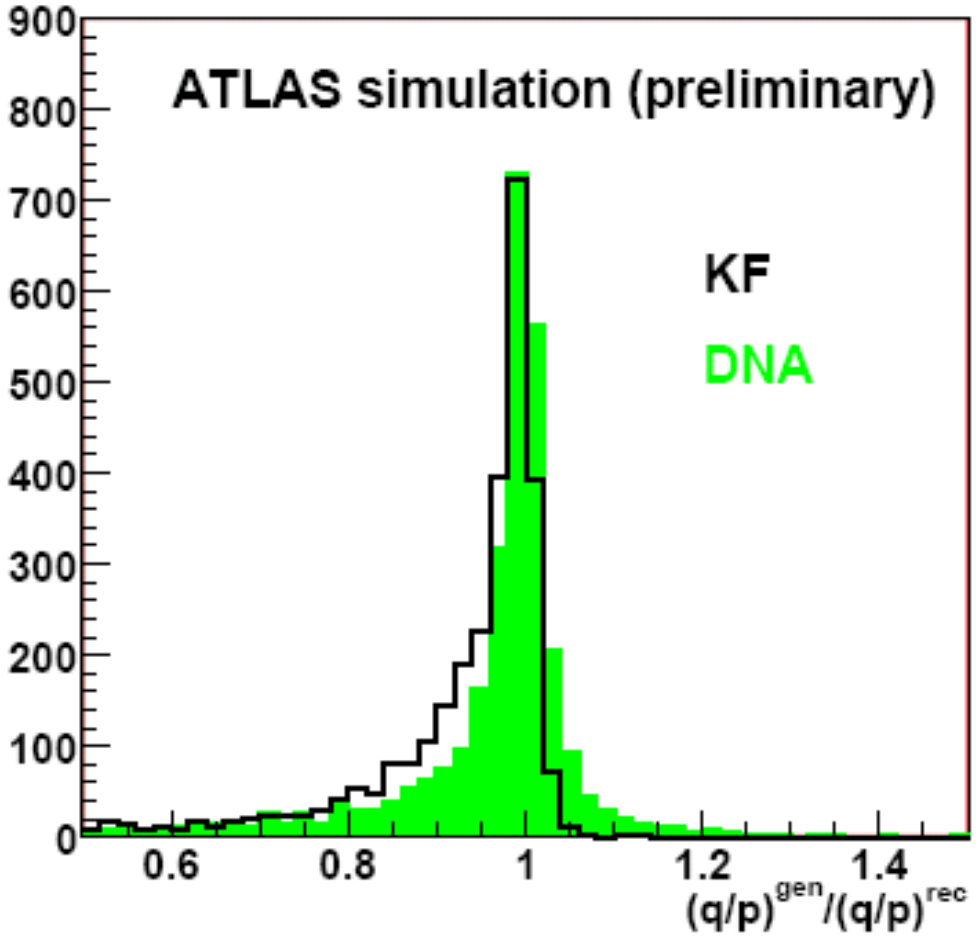
Dynamic Noise Adjustment

- The dynamically adjusted $\sigma(z)=\Delta z/\Delta x$ noise term is fed back to the Kalman Filter covariance matrix just like for multiple scattering.



DNA filter

- Using the DNA filter instead of just a fixed noise of width $< (z - z_{median})^2 >$ really helps for electrons:



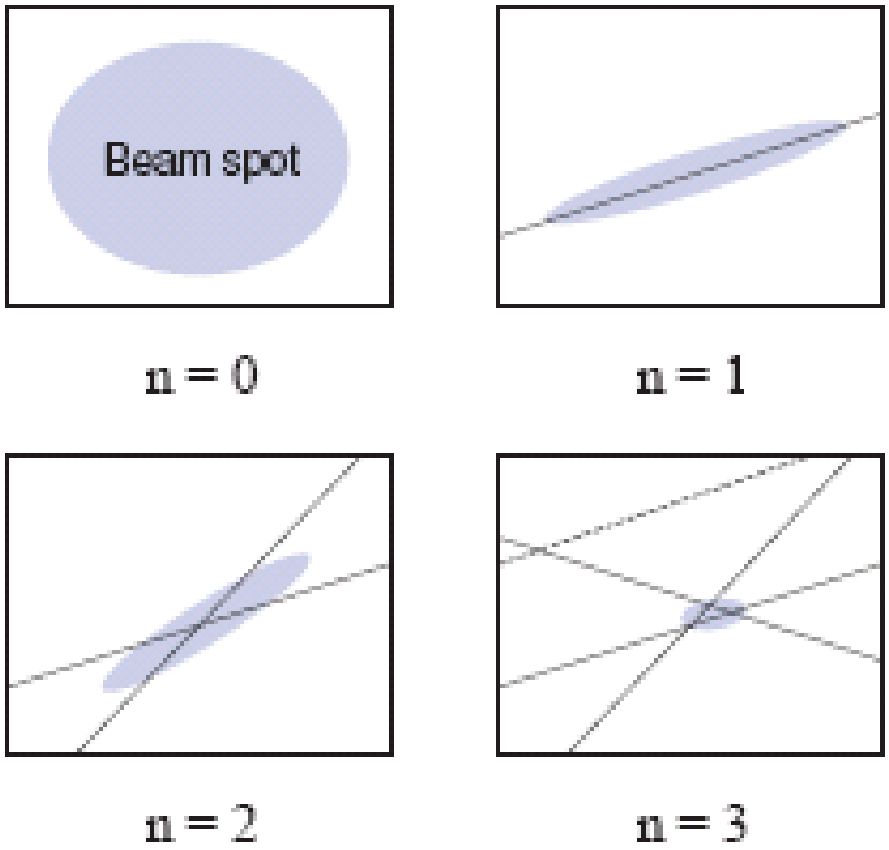
C++ implementation

- At the micro-level the CLHEP library is quite efficient for directly coded matrix equations. In the exercises we try out the ROOT TMatrix package.
- In ATLAS abstract interfaces are used for all components, reducing the compile-time dependencies and making the code general. The fitters do not care about which detectors made the measurement, which propagators are used etc..
- Any component, like a particular track-fitter, or a particular propagator, can be interchanged at run-time.
- Data objects are clearly distinguished from algorithms. These objects can be stored once, and accessed by all modules, but their basic properties can not be modified.

  This is for safety and for modularity of the code.

Finding the primary vertex

- Normally a limited ”beam-spot” is given by the machine-parameters , possible pick-up electrodes or preprocessing.
- Hereafter, just two tracks suffices to provide an accurate seed for the vertex



Finding the first vertex seed

- The danger is of course that the initial seed is wrong, so great care must be taken in this first step.
- In ZEUS, for example, all candidate track pairs were checked for compatibility with a common vertex on the proton beam-line. They were then ranked according to how many other pairs they agreed with. The best pair then started the chi-squared fit.
- In CMS, a similar procedure is used were the coordinates with the highest density of track crossings is found. In this evaluation, each track pair is weighted by a decreasing function of the distance between the two perigees.

The primary vertex fitter

- After the first seed is found, a *Kalman filter* least squares algorithm is normally used to add tracks one by one, continuosly updating the vertex state.
- In CMS, a weight is furthermore multiplied to each track, effectively blocking outliers from contributing. The CMS algorithm proceeds at progressively lower ”temperatures” just like the *Deterministic Annealing Filter*, letting the weights more and more steeply cut away outliers, so as to avoid local minima.
- Finally a *smoother* step refits each of the contributing tracks using the vertex constraint and provides a final update to the vertex.
- See eg J. Phys. G: Nucl. Part. Phys. **34** (2007) N343 ”Adaptive Vertex Fitting”.

Carrying on

- In the further reconstruction of the event *a priori* knowledge can be used with advantage. An example is the beam energy constraint used in the reconstruction of tracks from the decay of $\Upsilon(4S)$ in b-physics experiments, or the reconstruction of B meson cascade decay where the known masses of the D mesons are used as a constraint.
- Another example is fitting electron-positron tracks from photon conversions. The knowledge of zero photon mass can be imposed by using Lagrange Multipliers.
- (for more explanation by see eg www.phy.ufl.edu/~avery/fitting/kinfit_talk1.doc)

Lagrange Multipliers

- Let again x be the track parameters of the two tracks that form a conversion candidate. The constraints must be expressed as some functions of x being equal to zero.
- We expand around an approximate solution x_A:

$$\frac{\partial \bar{H}}{\partial \bar{x}}(\bar{x}-\bar{x}_A)+\bar{H}(\bar{x}_A)=\bar{D}\Delta\bar{x}+\bar{d}=\bar{0}$$

- The condition that the two tracks should emerge parallel from a common point is then expressed by something like

$$\bar{d}=\bar{H}(\bar{p}_1,\bar{p}_2,\bar{r}_1,\bar{r}_2)_A=(\frac{\bar{p}_1}{E_1}-\frac{\bar{p}_2}{E_2},\bar{r}_1-\bar{r}_2)_A$$

- Where the p's and r's refer to the start points of the tracks.

Lagrange Multipliers

- The function to be minimized is now:

$$\chi^2 = (x - x_0)^T V_0^{-1} (x - x_0) + 2\lambda^T (D(x - x_A) + d)$$

both wrt the track parameters x=(p,r) and the real constants λ (dropping the vector bars). The ”0” refer to the unconstrained solution and the “A” to the previous iteration. The solution have to be iterated since the constraint equation was linearized.

Solution to the constrained fit

- (check by differentiate chi2 and insert expressions below):

$$x = x_0 - V_0 D^T \lambda$$

$$\lambda = V_D (D(x_0 - x_A) + d)$$

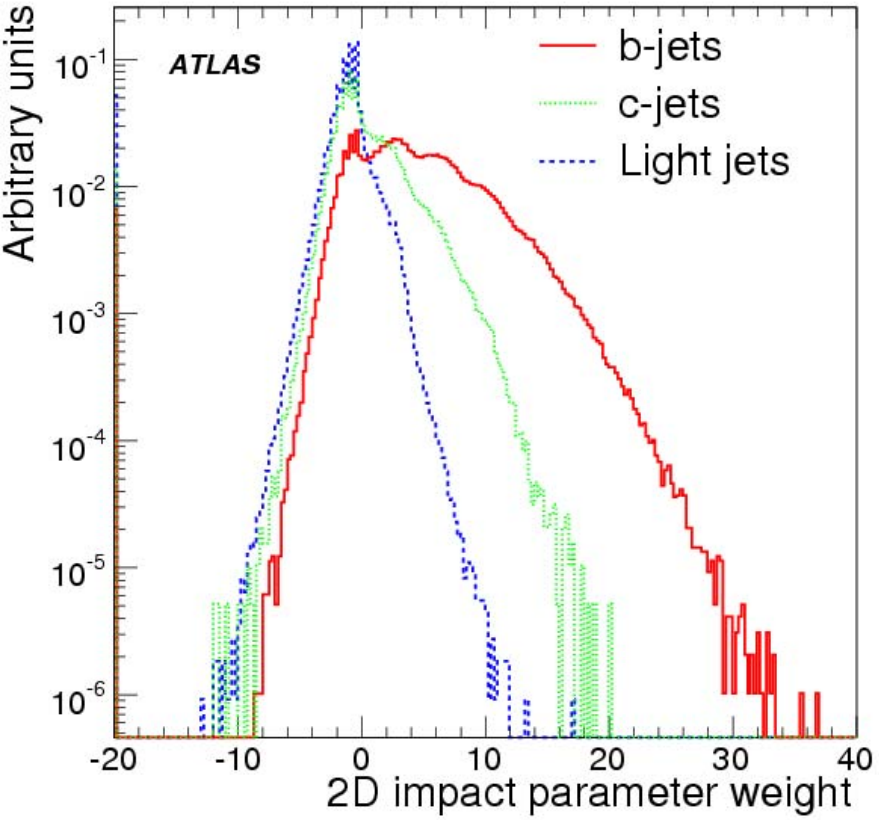
$$V_D = (D V_0 D^T)^{-1}$$

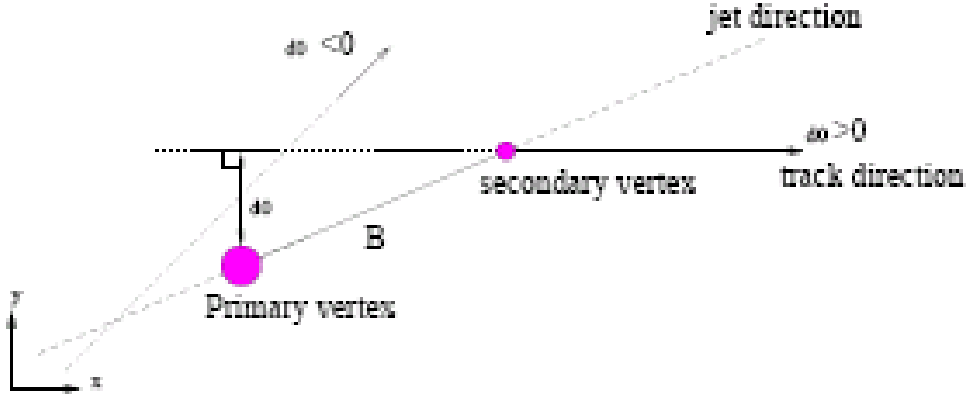
$$V_x = V_0 - V_0 D^T V_D V_0$$

$$\chi^2 = \lambda^T V_D^{-1} \lambda$$

B-tagging examples

- Impact parameter significance (S=d0/σ) based likelihood ratio (b-quark vs u-quark hypothesis):





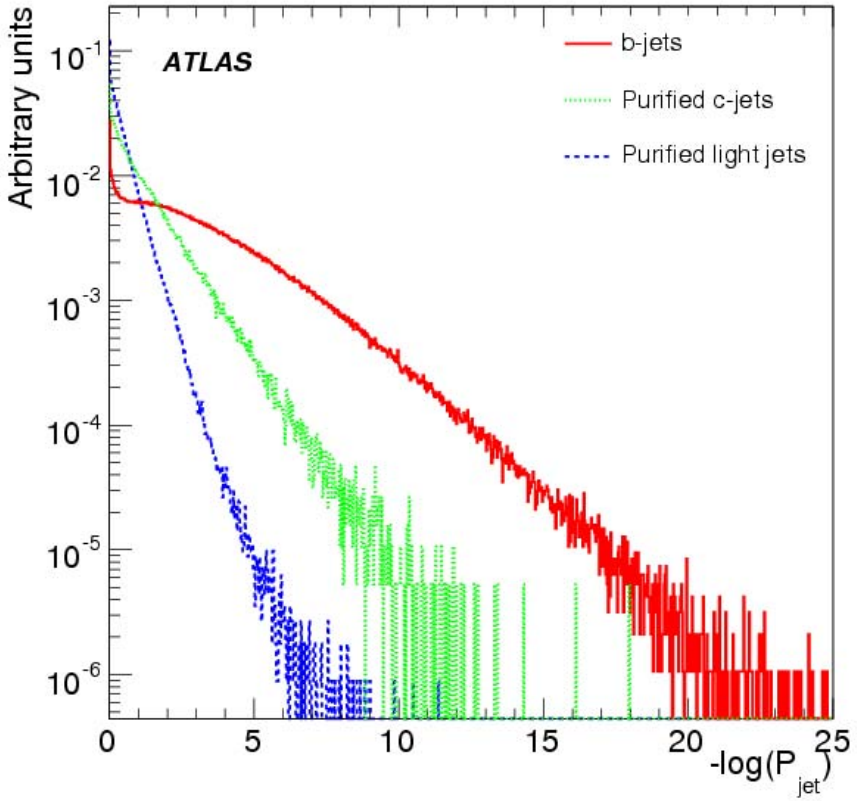
$$W_{jet} = \sum_{i=1}^{N_{tr}} \ln \frac{b(S_i)}{u(S_i)}$$

B-tagging examples

- If the resolution function R is known, we can test each tracks compatibility with the primary vertex. Let here d0 be the impact parameter projected on the jet axis (x). The probability for ”primary vertex origin” is:

$$\mathscr{P}_i = \int_{-\infty}^{-|d_0^i/\sigma_{d_0}^i|} \mathscr{R}(x)dx$$

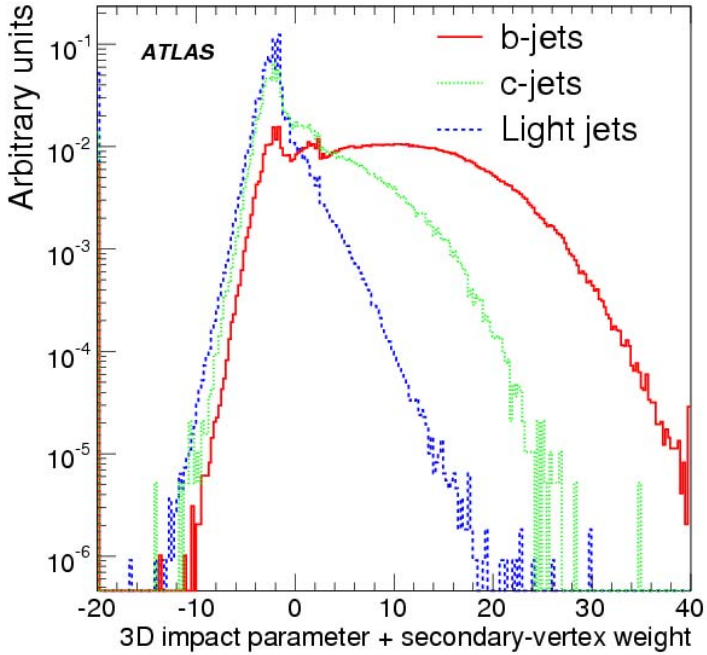
We plot –log (probability that all the tracks in the jet come from the primary vertex)



B-tagging examples

- Reconstructed secondary vertices can add extra orthogonal information to the impact parameter based discriminant:
  - Invariant mass of the attached tracks
  - Fraction of jet momentum carried by the attached tracks
  - Number of secondary vertices

more efficient b-tagging from combining all information



B-tagging examples

- Finally the electron and muon identification capabilities of the detector can be used to identify (semi)-prompt leptons from b-hadron decay.
- In general it pays off to consider separate probabilities for different track classes – such as the tracks with shared hits.
- All the various information can be combined in many ways. One of the ways that have been tried out is the Boosted Decision Tree (J. Bastos, ATL-COM-PHYS-2007-016).
- ROOT integrated implementaions of all such multivariate analysis algorithms are collected in the TMVA package
- http://tmva.sourceforge.net/docu/TMVAUsersGuide.pdf

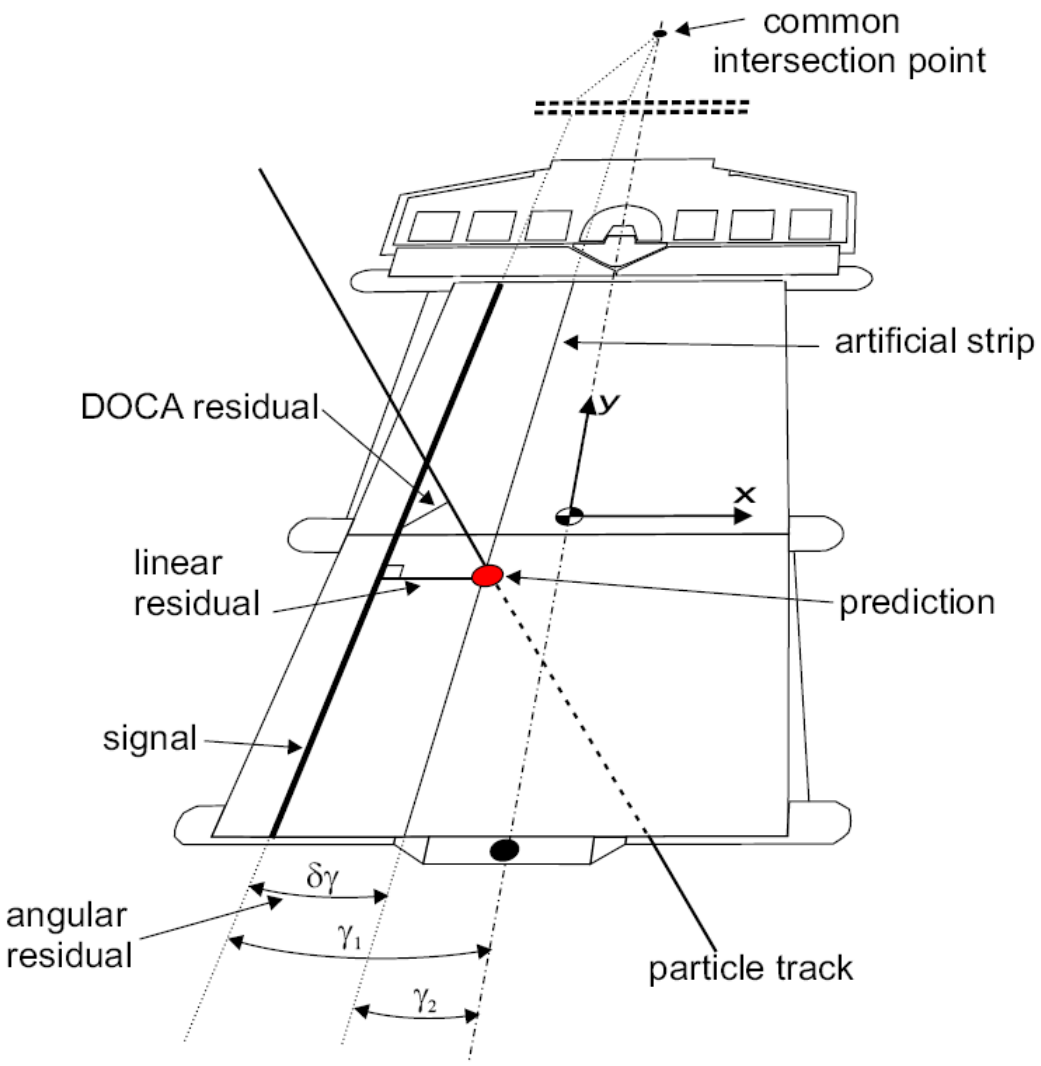
Alignment

- In order to have high resolution unbiased tracking, the detector elements must be correctly aligned. For the silicon elements we typically speak of a few microns or less.
- Part of this is achieved by elaborate laser alignment systems, such as the Frequency Scanning Interferometer and the muon alignment system in ATLAS, or the TPC laser calibration system in ALICE.
- While these devices are good at tracking short-term changes, the ultimate alignment precision is best achieved by using the fitted tracks themselves.

Alignment with tracks

- Consider a tracker with n planes along the x-axis. From the reconstructed tracks, we want to determine the alignment corrections to the position and orientation of each plane.
- Consider a single measured coordinate y_i and a track model $y=h(\mathbf{x},\boldsymbol{\alpha})$, where **x** is a vector of the positions of the planes and **α** are the alignment corrections.
- A straight forward estimate of α_i is simply the average residual $\langle r_i = y_i - h(x,\alpha) \rangle$, averaged over all fitted tracks.
- If the considered plane does not take part in the fitted track, it is called an unbiased residual.
- This *"local"* approach requires in general many iterations because the correlations between planes induced by the fitted tracks are ignored with this method.

Alignment with tracks



Alignment with tracks

- In the *"global"* approach we define a total *chi2* of a large track sample:

$$\chi^2 = \sum_{tracks} r^T V^{-1} r$$

$$r(x, \alpha, m) = h(x) - m$$

where r are the residuals (note the unusual sign), α the alignment parameters of the detector elements common for all the tracks and x the individual track parameters. What we want is to simultaneously minimise *chi2* both with respect to all the millions of x's and the many α's. Sounds impossible, but it isn't - as seen from the arguments in Bocci and Hulsbergen: ATL-INDET-PUB-2007-009 presented in the following.

Alignment with tracks

- At the minimum wrt α, we do not want any changes in the *chi2* derivatives wrt x when we change α by a small amount (here, the residuals are assumed linear in the alignment parameters). This can be expressed as

$$\frac{d}{d\alpha}\frac{\partial \chi^2}{\partial x} = \left(\frac{\partial}{\partial \alpha} + \frac{\partial x}{\partial \alpha}\frac{\partial}{\partial x}\right)\frac{\partial \chi^2}{\partial x} = 0$$

Inserting transposed expressions from slide 30 and 31 and using A for the partial derivative of r wrt α, we get for any total derivative of α:

$$\frac{d}{d\alpha} = \frac{\partial}{\partial \alpha} - A^T V^{-1} H C \frac{\partial}{\partial x}$$

Alignment with tracks

- Further ignoring any second derivative of the residuals we obtain for the derivatives of a single track:

$$\frac{d\chi^2}{d\alpha} = 2A^T V^{-1}(V - HCH^T)V^{-1}r$$

$$\frac{d^2\chi^2}{d\alpha^2} = 2A^T V^{-1}(V - HCH^T)V^{-1}A$$

where the matrix $R = V - HCH^T$ is the covariance matrix of the (biased) residuals in the track fit. (For the unbiased residuals we would get a plus sign).

Alignment with tracks

- The preceeding equations hold even if the x's are not yet fitted and even if all kinds of extra contributions are used in the *chi2 (eg other detectors, vertex, mult scatt etc)*
- But if also the initial track parameters are minimising the individual *chi2's*, the first derivative reduces to (slide31):

$$\frac{d\chi^2}{d\alpha} = 2A^T V^{-1} r$$

- Thus the first derivative is *local:* the derivative wrt some α receives only contributions from the local detector element for which the partial derivative A=δr/δα is non-zero.
- Requiring the sum of the derivatives over all tracks be zero results in M coupled equations – in general non-linear – where M is the number of alignment parameters.

Alignment with tracks

- If *chi2* is not already at minimum, we linearize the problem and get the following linear equations for the corrections:

$$\frac{d^2\chi^2}{d\alpha^2}\Delta\alpha = -\frac{d\chi^2}{d\alpha}$$

- Several algorithms exists for solving them. MILLIPEDE is a well-known example (google Blobel).
- Another example is MINRES, minimising the distance between the two sides of the equation (used by CMS).
- Yet others calculate eigenvectors and eigenvalues of the second derivative while exploiting the sparseness of this matrix (used by ATLAS).

Alignment with tracks

- The explicit solution to the alignment problem is thus

$$\Delta\alpha = -\left(\frac{d^2\chi^2}{d\alpha^2}\right)^{-1}\frac{d\chi^2}{d\alpha}$$

or equivalently

$$\Delta\alpha_i = -\left[\sum_{tracks}\frac{\partial\overline{r}}{\partial\alpha_i}R^{-1}\frac{\partial\overline{r}^T}{\partial\alpha_j}\right]^{-1}\sum_{tracks}\left[\frac{\partial\overline{r}}{\partial\alpha_j}R^{-1}r^T\right]$$

where $R = V - HCH^T$ is the covariance matrix of the residual vector of a track.

Alignment with tracks

- After diagonalising the second derivative, the solution reads:

$$\Delta\bar{\alpha} = \sum_{j=1}^{M} \frac{\bar{u}_j \bullet \bar{b}}{d_j} \bar{u}_j \qquad C(\Delta\bar{\alpha}) = \sum_{j}^{M} \frac{1}{d_j} \bar{u}_j \otimes \bar{u}_j$$

where u_j are the eigenvectors, d_j the eigenvalues. and b is minus the first derivative. Clearly there is a problem if $d_j=0$. Small eigenvalues corresponds to weak modes, ie alignment corrections that are poorly constrained by the data. An obvious example is the movement of the entire setup to another location in the world.

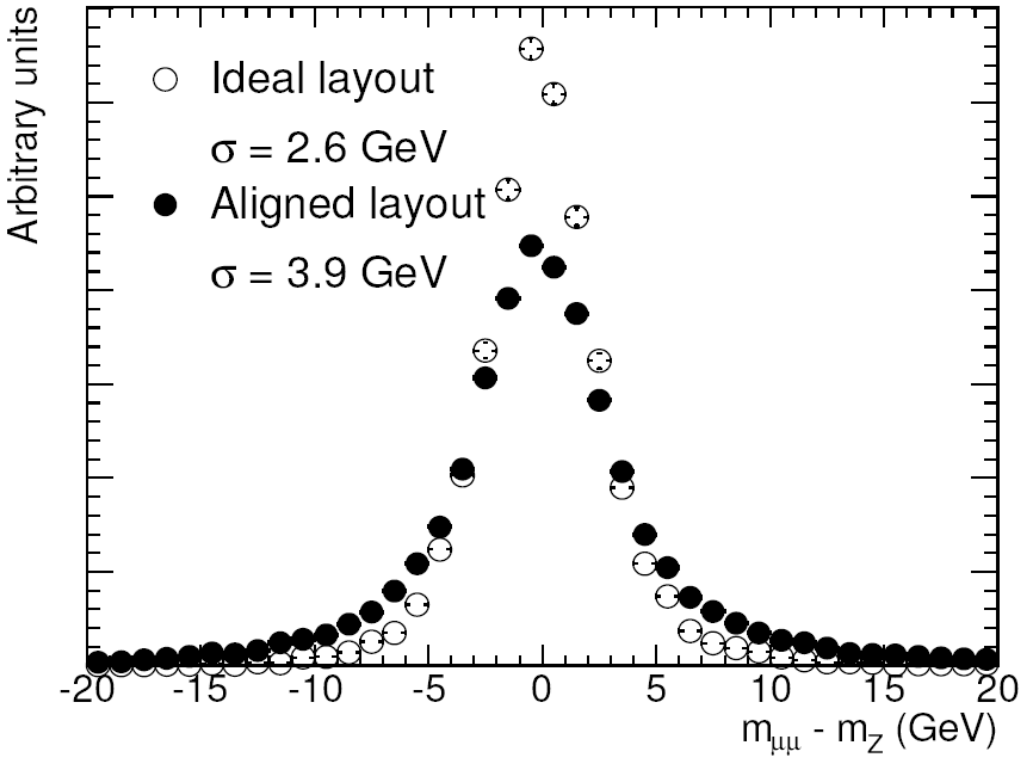
- Using weak modes will blow up detector resolution, so only those corrections should be applied whose uncertainty C is far below the detector resolution.

Weak modes

- The deformations corresponding to different eigenvectors give independent contributions to the chisquared.
- Small eigenvalues correspond to weak modes. Those are movements of the detector parts that are poorly constrained by track residuals.
- An obvious example is the translation of the entire detector to somewhere else.
- Another is deformations that somehow preserve the track model (eg helix->helix).
- Using the weak modes will blow up detector resolution. Therefore one should only apply corrections for modes whose uncertainty C is much smaller than the detector resolution.
- To obtain the corrections for the weak modes other alternative methods must be used (laser alignment, optical survey, cosmic tracks).

Effects of misalignment

- The LHC experiments have ~successfully exercised the method on misaligned Monte Carlo. Still room to improve on the infamous “weak modes”



Conclusion and outlook

- Profiting from increasing computing ressources, new sophisticated track finders and fitters have been applied to high energy physics experiments.
- These algorithms will face their crucial tests in the reconstruction of real collisions during the coming year.
- In the future we might see truly global methods (optimizing simultaneously all the tracks in regions of an event) and also methods benefitting at an early stage from particle identification.
- We will also see more detectors and constraints included in the global alignment procedures– at the cost of sharply increased computing power requirements – in order to avoid weak modes.

Exercises

- Consider a simple silicon telescope in a test beam:
- A Kalman filter example (with unrealistically simplified pattern recognition) using the ROOT TMatrix is in
- **http:/www.nbi.dk/~phansen/nordforsk/kalman.C**
- Try different outlier cuts and see what happens to the track efficiency and precision of the track parameters
- Compare with the equivalent Global Chi2 example:
- **http://www.nbi.dk/~phansen/nordforsk/globalchi2.C**

  (Something is fishy. Note that MS is ignored in the reco.)
- Try out the alignment algoritms :
  http://www.nbi.dk/~phansen/nordforsk/align.C

  (Note the elimination of the most important weak mode)