Tracking issues

suggestions on methods and implementations

Franz Klein, CUA

Tracking in general

- **charged particle tracking in B field:**
  5 d.o.f. -> typical: $\mathbf{k=(q/p,\lambda,\varphi,d_{\perp},d_{\parallel})}$

- **track based on associated hits in detectors, detector geometry, calibration, B-fields**

- **procedure:**

1. **exploration** **- pattern recognition (segments)**
2. **selection** **- segment combining (linking)**
3. **fit** **- global fit (spline, global track model)**
4. **final selection** **- single, multi-track fitting**

- **requirements: precision, efficiency (,speed, memory)**

Pattern recognition

- **collect hits and combine to segments and track candidates:**
- **template matching -> associated hit clusters**
- **circle or helix fits of any 3 hits $(x,y,z,dx/dz,dy/dz)_i$**

  + track following (successive combination)

  note: large exploration tree = combinatorial problem
- **projection onto 2D circles or lines (+helix fit or lin.regression)**
- **conformal mapping (Riemann fit: exact linear fit in 3D)**
- **Hopfield net (neural network)**
- **principal components analysis (interpolation via track road method)**

Hopfield net

- **neurons=track segments**
- **weighting based on track samples (MC data, real data)**
- **energy minimization** (M_{ia}=distance of hit *i* to search arm *a*)

  (s_{ia}=decision unit: s_{ia}=1: hit *i* associated to *a*)

$$E(s_{ia}\mathbf{k}) = \sum_{n,m} s_{ia} M_{ia} + \lambda \sum_{n} \left(\sum_{m} s_{ia} - 1 \right)^2$$

⇒ hypotheses screened on method of minimum description length:
$L_{meas}(m)=min$ ⇔ $F(m)=m^T Cm=max$ (with $c_{ij}=K_1 n_{ij}-K_2 \xi_{ij}-K_3 N_i N_j/(N_i+N_j)$)

Principal Components Analysis

- **transformation of *(N)* measurements to subset of components u_1 to u_n such that $u_{n+1} \approx \ldots \approx u_N \approx 0$, i.e. u=A(m-z)**
- **unknown hit candidate is then predicted by**

$$0 \approx u_N = \sum_{i=1}^{N} A_{Ni}(m_i - z_i) \Rightarrow \quad m_j = \frac{-1}{A_{Nj}} \sum_{\substack{i=1 \\ i \neq j}}^{N} A_{Ni} m_i + \frac{1}{A_{Nj}} \sum_{i=1}^{N} A_{Ni} z_i = \sum_{\substack{i=1 \\ i \neq j}}^{N} a_{ij} m_i + b_j$$

- **coefficients a_{ij} and b_j determined via training (MC data)**

Tracking in CLAS: global track model

- **global predictor-corrector:**
  (note: **k** defined at track "vertex"=intersection with beamline plane)
- **PR: linking using road maps (hit combinations with estimates for k)**
- **adjustment via swimming (helix steps) and fit to DC hits (twice):**
- HIT BASED (+extrapolation to CC,SC,EC)
- TIME BASED (using β, doca, LR fit)
  d_t^i via intersection with layer (cylinder)

$$\chi^2 = \min \sum_{i=1}^{n} \left(\frac{d_m^i - d_t^i(g, B)}{\sigma_{d_m^i}} \right)^2$$

- **variations: 5 superlayer tracking** (>possibly 6th SL hits neglected<)
  k on **1st DC layer** (+extrapol. into target field)
  vertex ($v_\parallel$, $v_\perp$) + SCpos. constraints

helix and quintic spline fit

- **concept: local PR + spline fit**
- **helices of any 3 hits** $(x,y,z,dy/dx,dz/dx)_i$
- **fit via quintic splines** (incl.energy loss)

$$py''(x) = q\sqrt{1+y'^2+z'^2}\left[B_x z' + B_y y' z' - B_z (1+y'^2)\right]_z, \quad y(x) = a_1 + a_2 x + \frac{q}{p} Y(x)$$

$$pz''(x) = q\sqrt{1+y'^2+z'^2}\left[-B_x y' - B_z y' z' - B_y (1+z'^2)\right]_z, \quad z(x) = b_1 + b_2 x + \frac{q}{p} Z(x)$$

$$\mathbf{Bk} = \mathbf{m}, \quad \mathbf{k} = -(\mathbf{B}^{\mathrm{T}}\mathbf{WB})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{Wm}, \quad \mathbf{k} = (a_1, a_2, \frac{q}{p}, b_1, b_2)^T, \quad \mathbf{m} = (y_1, z_1, \ldots, y_n, z_n)^T$$

W=error matrix, cov(**u**)=$(\mathbf{B}^{\mathrm{T}}\mathbf{WB})^{-1}$

- **covariance matrix** (incl. contrib. of mult.scattering)

Kalman filter

- **concept: track finder+fitter**
- **estimation procedure (k_0 from helix fit, covariance C_0)**
  (**m**=measurements, **ε**=uncorrelated errors with covariance V)
- **geometry uncertainty taken into account (G=covariance)**
- **improved filter with annealing factor** $\alpha^{(k)}$

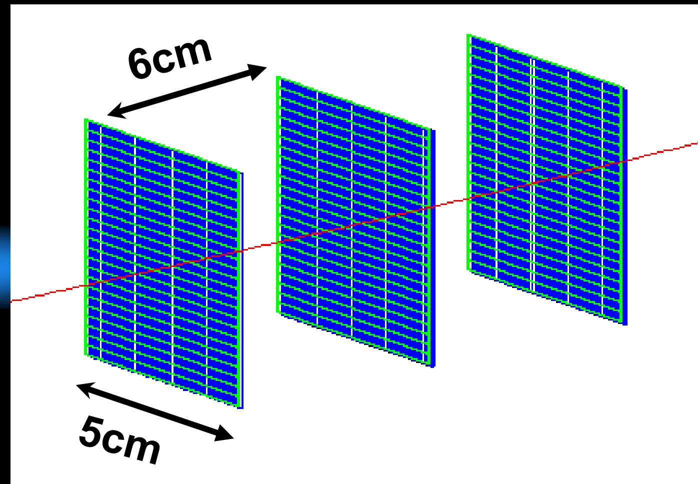
$$\mathbf{m} = \mathbf{f}(\mathbf{k},\mathbf{G}) + \boldsymbol{\varepsilon} = \mathbf{f}(\mathbf{k}_0) + (\mathbf{H}+\mathbf{G})(\mathbf{k}-\mathbf{k}_0) + \boldsymbol{\varepsilon}, \quad \mathbf{H} = \frac{\partial \mathbf{f}(\mathbf{k}_0)}{\partial \mathbf{k}}$$

$$\mathbf{k}_1 = \mathbf{k}_0 + \mathbf{W}_0\mathbf{H}^T\mathbf{W}\left[\mathbf{m} - \mathbf{f}(\mathbf{k}_0)\right], \quad \mathbf{C}_1 = \mathbf{C}_0 - \mathbf{C}_0\mathbf{H}^T\mathbf{W}\mathbf{H}\mathbf{C}_0$$

$$\mathbf{W} = \left[\alpha^{(k)}\mathbf{V} + \mathbf{H}\mathbf{C}_0\mathbf{H}^T + \mathbf{H}\mathbf{G}\mathbf{H}^T\right]^{-1}$$

- **problem: good error estimates for 1st iteration**

toy model

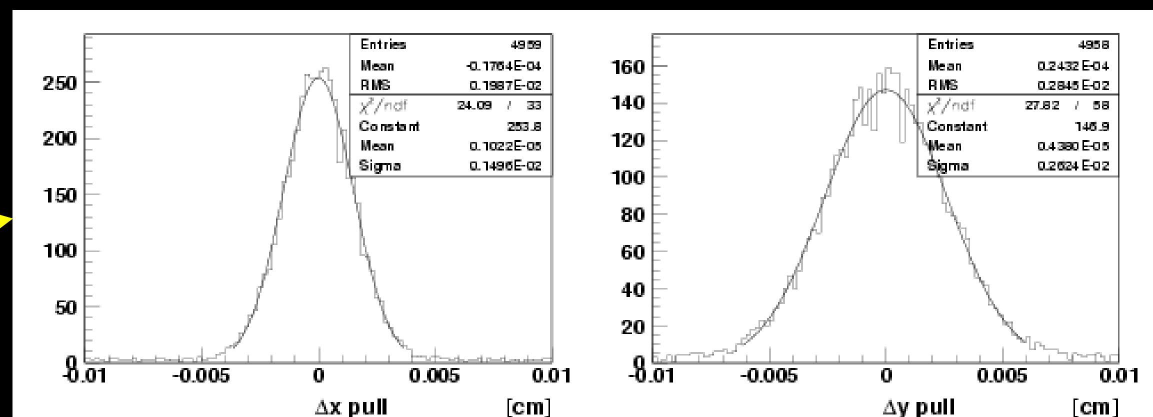


- 3 double-layer MSD pitch: dx=50μ,dy=100μ
- position known to 25μ
- uniform B=2.0±0.01T
- 5000 evts p=1.0±0.1GeV

reconstruction:

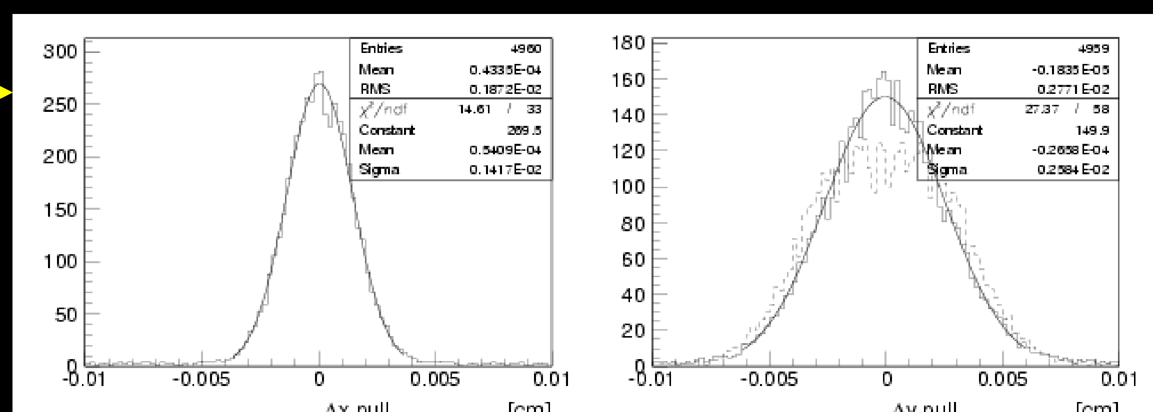
helix-spline:

both directions ok



Kalman fitter:

great! can even account correctly for alignment errors



Outlook

- **many algorithms tested over the years for HEP** (Zeus,H1,CDF,Belle,Barbar,CMS,ATLAS)
- **Several algorithms should be tested!**
- **probably more efficient than global track model**
  (e.g. low mom. tracks and large background)

- **at least modification of global track model:**
- **improved PR and road maps**
- **≥1 hit per layer (double hits)**
- **energy loss (RK extrapolation)**
- **kinematically constrained fits for poorly reconstructed tracks**