Tracking Algorithm Follow Up Studies

John Paul Chou (Rutgers), Keegan Humphrey (UofT)

August 3rd 2022

Follow Up From Previous Presentation

- The wrong Psuedo-Random number generator in ROOT was implemented (Trandom was being used instead of Trandom3)
  - And it was implemented incorrectly
- Found unaccounted for uncertainty in the propagation step of the kalman filter
  - Changed how propagation step is done and accounted for the uncertainty introduced
- Following the helpful comments and questions from the previous presentation, we conducted studies of
    - Pull distributions comparing reconstructed to generated track parameters
    - Efficiency of the tracker
    - Beta residuals and pulls before and after removing the cut imposed during reconstruction
- ***Conclusion: we find good cores but broad tails, we have an idea for why (to be discussed later)***
  - We plan to implement a corresponding upgrade over the next few months

Beta Sanity Check - With Beta Cut (Dropped 0.8 < beta < 1.2)

Study was conducted with a muon gun sample fired above the floor

	Beta [c]	Beta Pulls []
Fit Mean **	0.998	-0.0266
Fit STD **	0.0181	0.414
HWHM	0.0119	0.438
Plot STD	0.0778	2.41

** From Gaussian Fit to the core of the distribution

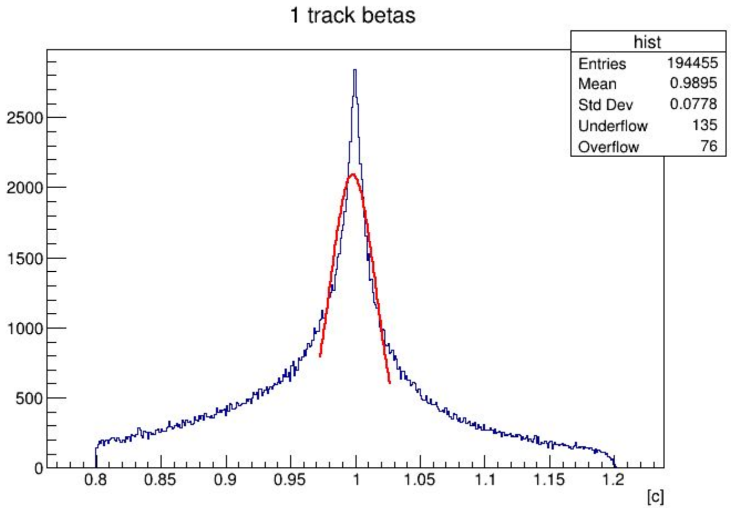
Beta Sanity Check - Without Beta Cut

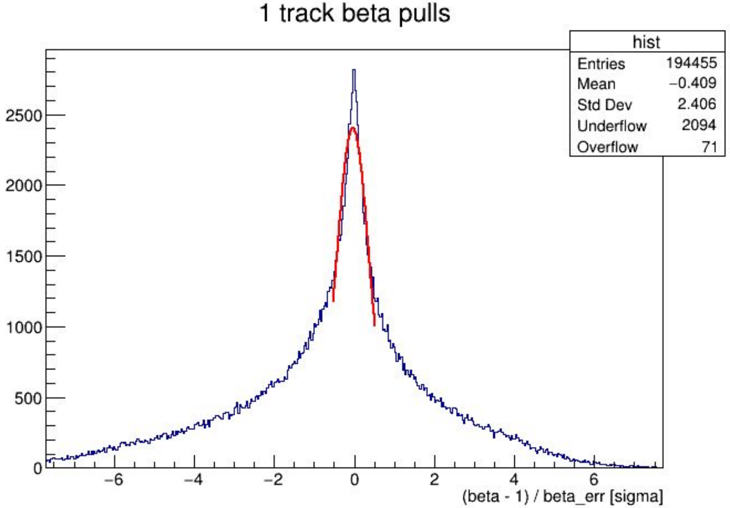
Study was conducted with a muon gun sample fired above the floor

	Beta [c]	Beta Pulls []
Fit Mean **	0.995	-0.103
Fit STD **	0.0351	1.16
HWHM	0.00938	0.506
Plot STD	0.149	5.54

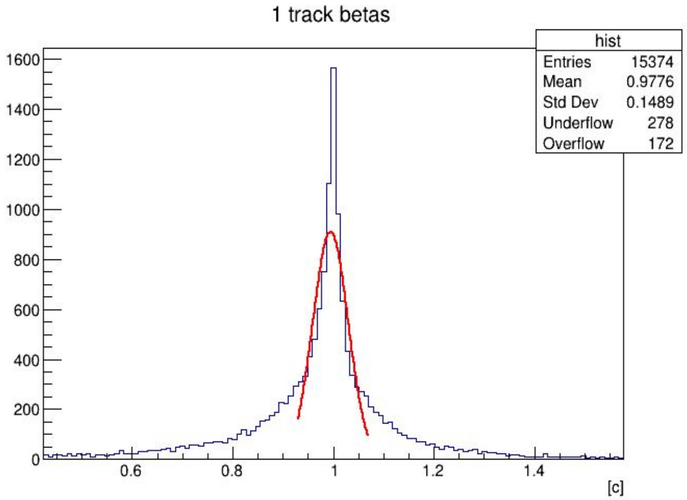
** From Gaussian Fit to the core of the distribution

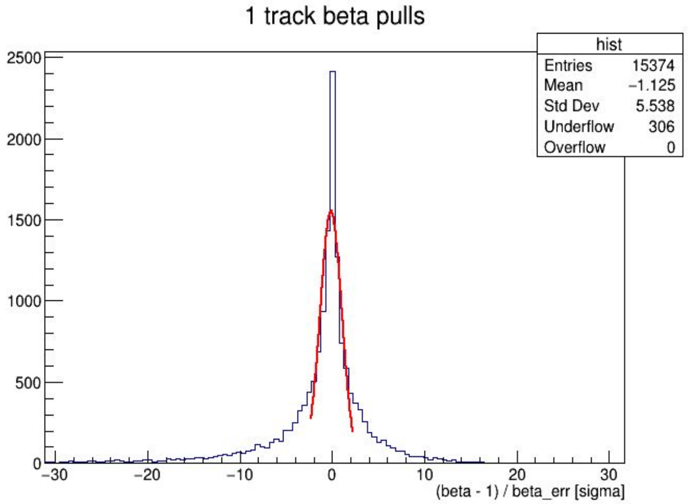
Beta Sanity Check - With Beta Cut





Beta Sanity Check - Without Beta Cut





The beta sanity check (among other things) tell us that constraining beta = 1 during reconstruction is a potentially fruitful way to improve its performance

Track Resolution - Pulls

- We use the muon gun sample generated **above the floor**
- We impose the condition of exactly **1 Track in the event** and a ***maximum*** of **1 digi hit per layer**, to reject soft delta rays
- We propagate the generated muon (given it's generated velocity) to the height of the **lowest hit in the track** and account for multiple scattering

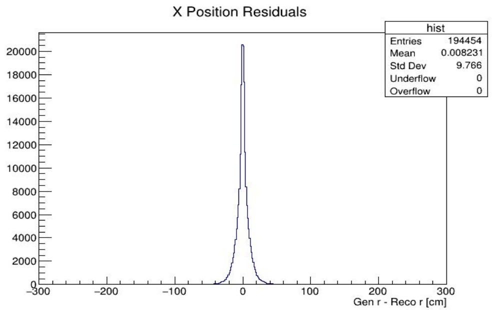
Y	Vertical
Z	Beamline
X	Other

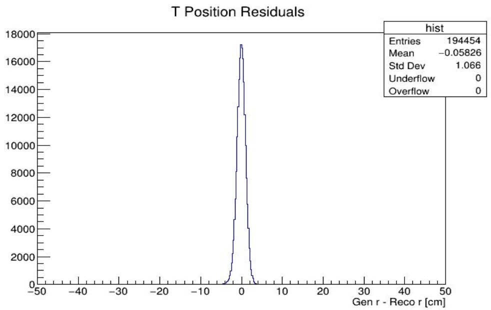
Track Parameter Resolution

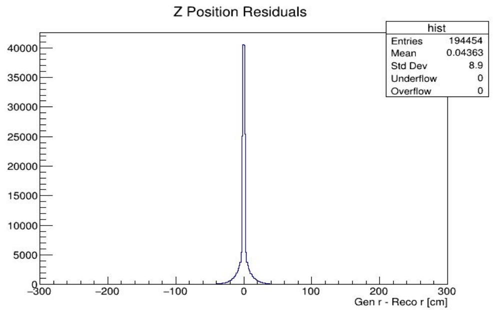
	x [cm]	t [ns]	z [cm]	vx [cm / ns]	vy [cm / ns]	vz [cm / ns]	angle [rad]
Mean	0.00823	-0.0583	0.0436	-0.00285	0.373	-7.58e-4	0.0282
STD	9.77	1.07	8.90	0.972	2.81	0.6354	0.0292
HWHM	3.41	1.02	2.05	0.219	0.524	0.229	0.00916
65% Boundary	(-6.60, 6.61)	(-1.02, 0.914)	(-2.38, 2.41)	(-0.539, 0.533)	(-1.52, 2.47)	(-0.377, 0.377)	0.0304
95% Boundary	(-20.3, 20.2)	(-2.22, 1.99)	(-19.5, 19.9)	(-2.03, 2.02)	(-4.84, 5.29)	(-1.27, 1.26)	0.0882

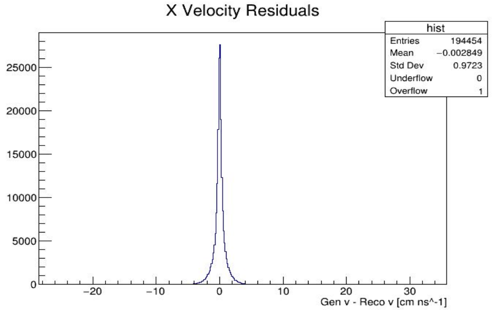
Track Parameter Resolution Plots

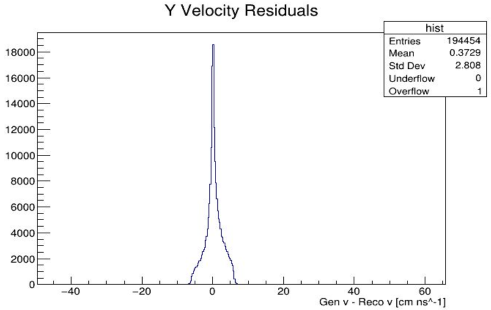
Y	Vertical
Z	Beamline
X	Other

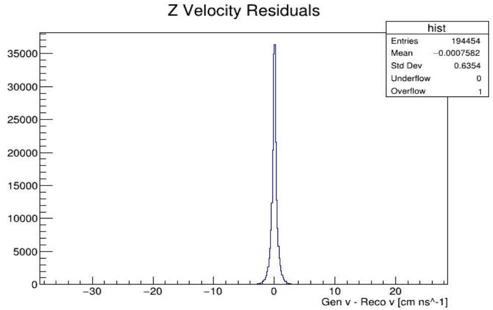


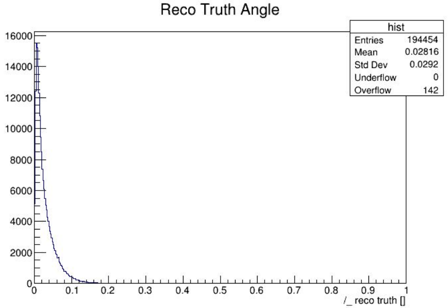












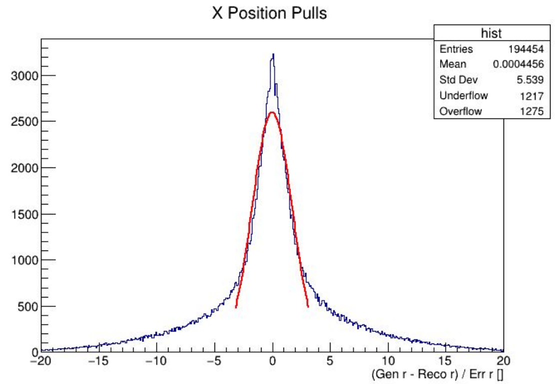
Pull Parameters

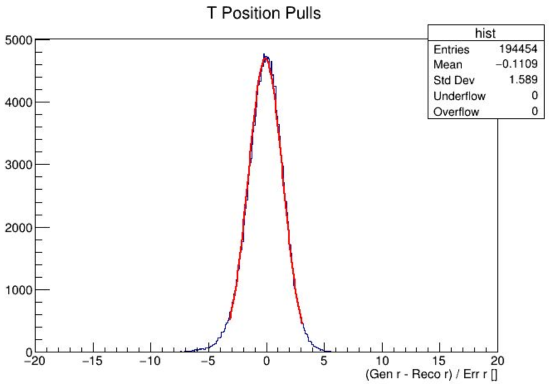
Y	Vertical
Z	Beamline
X	Other

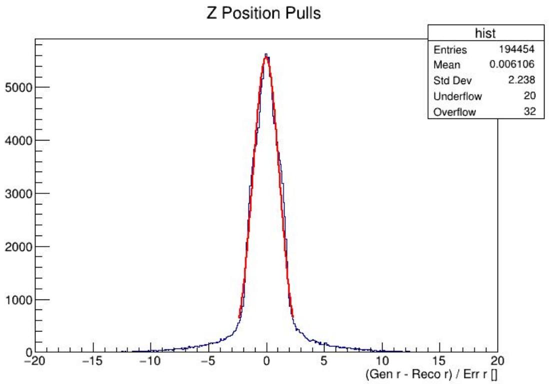
	x	t	z	vx	vy	vz
Fit Mean	0.00836	-0.0484	0.00319	-0.00149	0.0858	5.91e-4
Fit STD	1.72	1.47	1.13	1.13	1.02	0.861
HWHM	1.41	1.68	1.45	0.772	0.455	0.864
Plot STD	5.54	1.59	2.24	2.56	2.57	1.49

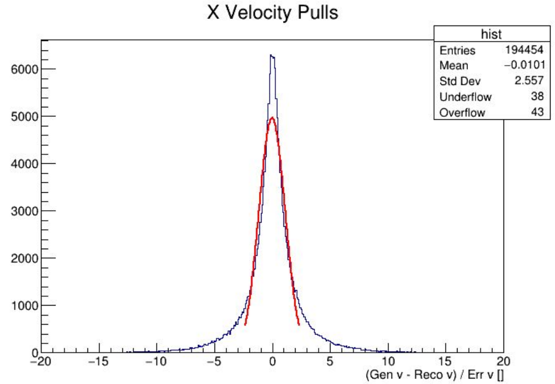
Track Parameter Pulls

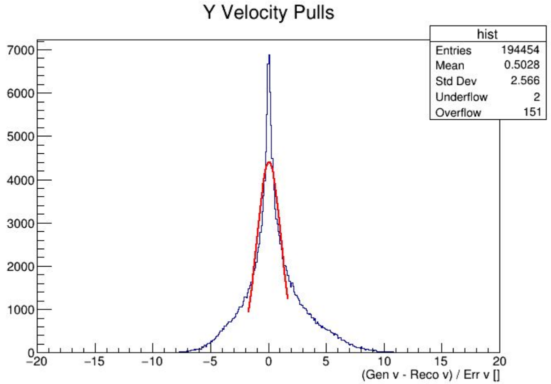
Y	Vertical
Z	Beamline
X	Other

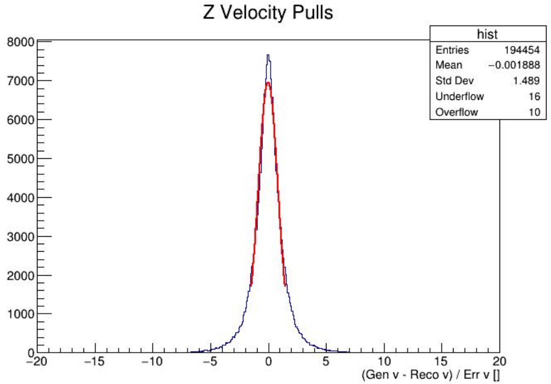




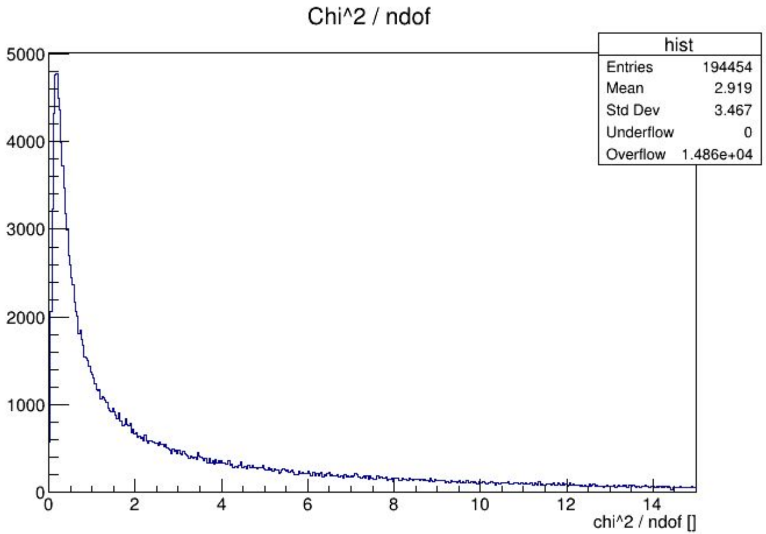


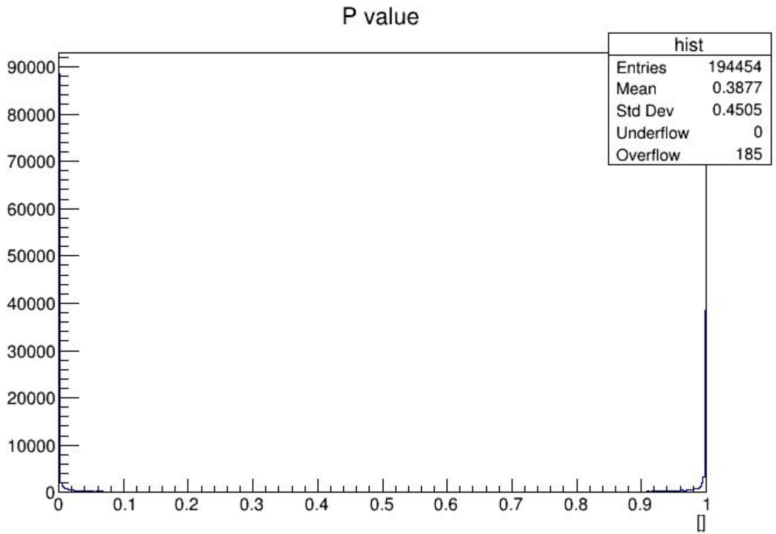






Generated vs Reconstructed Chi Square Distribution





Long tails we saw in the pulls appear in the chi^2 / ndof distribution

Follow Up Studies - Track Efficiency

- **Denominator** is number of events where the generated muon is ***predicted*** (using the generated velocity) to pass through 5 layers with at least a 50 cm **distance** between the predicted position and the **edge of the module**
- **Numerator** is **subset of denominator** events with a reconstructed track
- We also enforce same selection as pull and residual and ***at least* 5 layers with hits**
- *We find an efficiency of: 90.2 %*

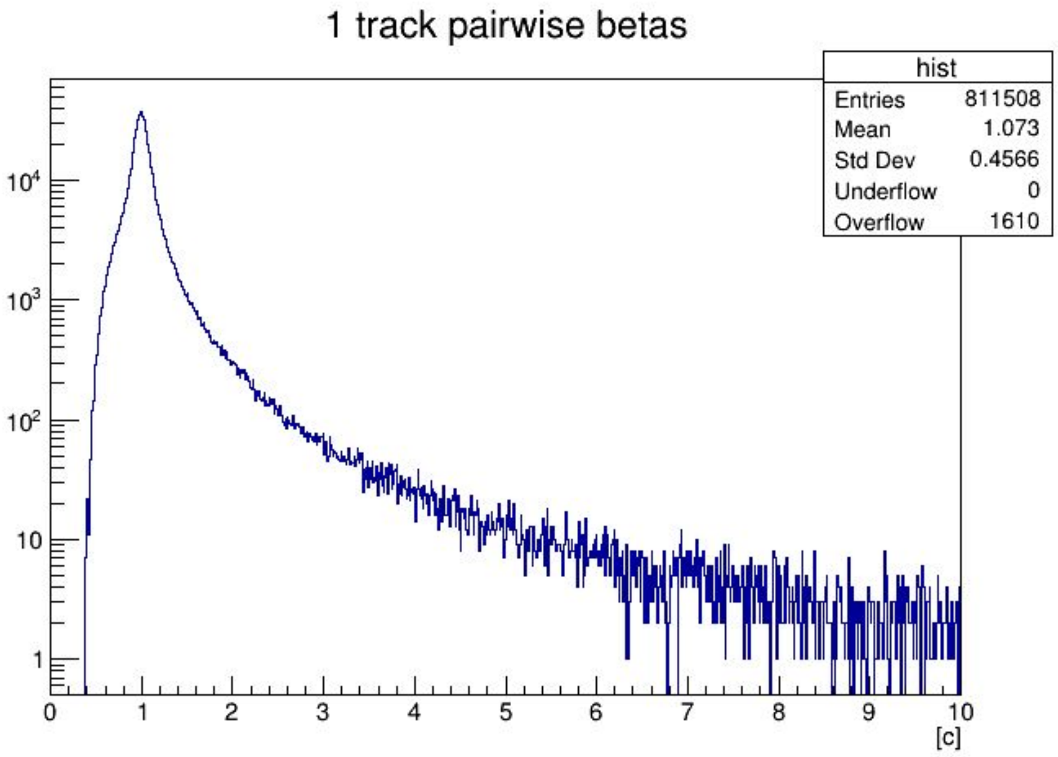
Physics Story about the Room for Improvement

- Current fit variables are: $x_{\text{state}} = [x, t, z, v_x, v_y, v_z]$
  - This means beta is unconstrained before we cut on it
- If the beta of the track drifts from 1 during hit selection, the tracker will tend to select incorrect hits
  - We expect this to cause
    - Inefficiency in track reconstruction
    - Track betas that differ significantly from 1
    - Tracks with higher fake rates
- We expect that constraining beta during track reconstruction will effectively address this issue
  - This amounts to removing one degree of freedom and changing variables

Conclusion

- Our algorithm introduces an inefficiency and inaccuracy in the tracker
- Constraining beta = 1 has the potential to effectively address these issues
- We plan to implement this upgrade in time for the CDR

Backup - Pairwise Betas Between Hits on Successive layers



Restricted to
1 hit per layer

Backup - What is a Kalman Filter?

A Kalman Filter is a linear, recursive, flexible fitting algorithm that provides the optimal fit given gaussian uncertainties. Following [1] we can make a filter with these steps.

- Describe the Measured Data
  - Choose a Measurement Matrix H_k (for us this projects out the velocity in the filtered state vector)

Measurement of a hit ← $m_k = H_k \; x_k$ → Filtered State

$[x_m, t_m, z_m]$ $\qquad$ $x_{\text{state}} = [x, t, z, v_x, v_y, v_z]$

k indexes the detector layers with chosen hits

- Predict
  - Choose a Prediction Matrix F_k (for us this propagates the state to the next layer)
  - Predict the next state vector and covariance based on the current state vector and covariance
    - Use this prediction to choose which data to add to the fit (for us these are the digitized hits m_k)

Predicted State ← $x_k^{k-1} = F_k \; x_{k-1}$ → Filtered State

$$C_k^{k-1} = F_{k-1} C_{k-1} F_{k-1}^T + Q_{k-1}$$

Predicted Covariance

Filtered Covariance

Process Noise Covariance

This is where we include scattering!

Backup - What is a Kalman Filter? - Filtering and Smoothing

- Filter
  - Update the predicted state to include the newly chosen data in the fit

Filtered Covariance

$$C_k = [I - C_k^{k-1} H_k^T (V_k + H_k C_k^{k-1} H_k^T)^{-1} H_k] C_k^{k-1}$$

Filtered State

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

Covariance of m_k

- Smooth
  - After ALL data has been chosen and filtered, propagate later states information to earlier ones

Smoothed State

$$x_k^n = x_k + A_k (x_{k+1}^n - x_{k+1}^k)$$

Smoothed Covariance

$$C_k^n = C_k + A_k (C_{k+1}^n - C_{k+1}^k) A_k^T$$

Smoother Gain Matrix

$$A_k = C_k F_k^T (C_{k+1}^k)^{-1}$$

Where n is the number of measurements

Backup - Computing χ^2s

To compute the contribution to the χ^2 for a chosen hit $(\chi_p^2,\ \chi_f^2,\ \chi_s^2)$, generically called χ_+^2, we use the following formulas. These assume there are no correlations between states on successive layers.

χ_+^2 calculated at prediction

$$\chi_p^2 = (r_k^{k-1})^T (V_k + H_k C_k^{k-1} H_k^T)^{-1} r_k^{k-1}$$

$$r_k^{k-1} = (m_k - H_k x_k^{k-1}) \qquad \chi_f^2 = \chi_p^2$$

residual

χ_+^2 calculated at filter

χ_s^2, the χ^2 increment calculated at each smoothing step, carries over with $k-1 \rightarrow n$.

The total χ^2 used to calculate the χ^2/ndof and pass or veto the track is the sum over all χ_s^2

Backup - Multiple Scattering - Covariance Matrix Computation

To calculate Q (following [3]) we parameterize the scattering by two orthogonal uncorrelated angles θ_1 and θ_2, and fix $\beta = 1$

$$\sigma(\theta_{\text{proj}}) = \frac{13.6}{p}\sqrt{\frac{L_{rad}}{\sin\phi}}\left[1 + 0.038\ln\left(\frac{L_{rad}}{\sin\phi}\right)\right] \qquad L_{rad} \equiv \sum_i \frac{X_i}{X_{0,i}}$$

We choose $p = 500\text{MeV}$ for high acceptance

ϕ is the inclination angle of the track

$$= \sqrt{Var(\theta_1)} = \sqrt{Var(\theta_2)}$$

For two covariance matrices, whose variables are related by the functions f_i, we can approximate as

$$Q_{ij}(\hat{y}) \approx \left[\frac{\partial f_i}{\partial y_n}\frac{\partial f_j}{\partial y_m}V_{nm}\right]_{\hat{y}}$$

Letting V be the covariance matrix for the scattering angles and $P_i \equiv (x_k^{k-1})_i$, we find

$$Q_{ij} = \langle P_i, P_j\rangle = \sigma^2(\theta_{\text{proj}})\left(\frac{\partial P_i}{\partial\theta_1}\frac{\partial P_j}{\partial\theta_1} + \frac{\partial P_i}{\partial\theta_2}\frac{\partial P_j}{\partial\theta_2}\right)$$

Backup - Multiple Scattering - Explicit Components

Letting Δy be the difference in y between the current layer and the one we are predicting to

$$x_{\text{state}} \doteq P_i = \left[\frac{\Delta y\,\alpha_3}{\beta_3} + x_0, \frac{\Delta y}{c\,\beta_3} + t_0, \frac{\Delta y\,\gamma_3}{\beta_3} + z_0, c\,\alpha_3, c\,\beta_3, c\,\gamma_3\right]_i$$

$$\alpha_3 \equiv \frac{v_x}{c} \qquad \beta_3 \equiv \frac{v_y}{c} \qquad \gamma_3 \equiv \frac{v_z}{c}$$

Recall that we parametrise by y instead of t since it is the most precise parameter we have.

$$Q = \sigma^2(\theta_{\text{proj}}) \begin{pmatrix} \frac{\Delta y^2\left({\beta_3}^2+{\alpha_3}^2\right)}{{\beta_3}^4} & \frac{\Delta y^2\,\alpha_3}{c\,{\beta_3}^4} & \frac{\Delta y^2\,\alpha_3\gamma_3}{{\beta_3}^4} & \frac{c\,\Delta y}{\beta_3} & -\frac{c\,\Delta y\,\alpha_3}{{\beta_3}^2} & 0 \\ \frac{\Delta y^2\,\alpha_3}{c\,{\beta_3}^4} & \frac{\Delta y^2\left(1-{\beta_3}^2\right)}{c^2\,{\beta_3}^4} & \frac{\Delta y^2\,\gamma_3}{c\,{\beta_3}^4} & \frac{\Delta y\,\alpha_3}{\beta_3} & -\frac{\Delta y\left(1-{\beta_3}^2\right)}{{\beta_3}^2} & \frac{\Delta y\,\gamma_3}{\beta_3} \\ \frac{\Delta y^2\,\alpha_3\gamma_3}{{\beta_3}^4} & \frac{\Delta y^2\,\gamma_3}{c\,{\beta_3}^4} & \frac{\Delta y^2\left({\gamma_3}^2+{\beta_3}^2\right)}{{\beta_3}^4} & 0 & -\frac{c\,\Delta y\,\gamma_3}{{\beta_3}^2} & \frac{c\,\Delta y}{\beta_3} \\ \frac{c\,\Delta y}{\beta_3} & \frac{\Delta y\,\alpha_3}{\beta_3} & 0 & c^2\,\left(1-{\alpha_3}^2\right) & -c^2\,\alpha_3\beta_3 & -c^2\,\alpha_3\gamma_3 \\ -\frac{c\,\Delta y\,\alpha_3}{{\beta_3}^2} & -\frac{\Delta y\left(1-{\beta_3}^2\right)}{{\beta_3}^2} & -\frac{c\,\Delta y\,\gamma_3}{{\beta_3}^2} & -c^2\,\alpha_3\beta_3 & c^2\,\left(1-{\beta_3}^2\right) & -c^2\,\beta_3\gamma_3 \\ 0 & \frac{\Delta y\,\gamma_3}{\beta_3} & \frac{c\,\Delta y}{\beta_3} & -c^2\,\alpha_3\gamma_3 & -c^2\,\beta_3\gamma_3 & c^2\,\left(1-{\gamma_3}^2\right) \end{pmatrix}$$

While it was convenient to fix $\beta = 1$ for computing the scattering angle variance, in the rest of the algorithm we let it float and cut on it instead.

References for Backup Slides

1. Frühwirth, R. (1987). Application of Kalman filtering to track and vertex fitting. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 262(2-3), 444-450. doi:10.1016/0168-9002(87)90887-4
2. Lynch, G. R., Dahl, O. I. (1991). Approximations to multiple Coulomb scattering. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, 58(1), 6-10.doi:10.1016/0168-583x(91)95671-y
3. E. J. Wolin and L. L. Ho, Nucl. Instrum. Meth. A329, 493-500 (1993) doi:10.1016/0168-9002(93)91285-U