MATHUSLA Kalman Filter Tracking Algorithm

John Paul Chou (Rutgers), Ali Garabaglu (Rutgers),
Stephen Greenberg (Rutgers), Keegan Humphrey (UofT),
Jiahao Liao (UofT)

Speaker: Keegan Humphrey

Motivation

Why was it necessary to overhaul the tracker?

- The Maximum Likelihood Tracker was not able to account for Multiple Scattering while fitting
  - This requires more aggressive track merging and a greedy algorithm to have good track efficiency
    - This strategy reduces the number of vertices formed especially in low mass signal where opening angles are smaller and two track muon events get merged into a single track

How does the Kalman Filter resolve this problem?

- The Kalman Filter allows for the inclusion of an arbitrary Gaussian Noise Covariance Matrix
  - A covariance matrix can be calculated and passed to the Filter to account for soft Multiple Coulomb Scattering
  - This means we can make tracks with only one hit per layer, not rely on aggressive merging, and still have good efficiency

Overview

- What is a Kalman Filter?
- Details of Multiple Coulomb Scattering
- Brief Description of the Algorithm
  - From simulated data to made tracks and vertices
- Visualisation
  - 10 and 2 GeV mumu events
- Performance
  - Made Track and Vertex Numbers
  - Vertexer Residual Plots
- Conclusions and Next Steps

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 $\longleftarrow m_k = H_k \ x_k \longrightarrow$ Filtered State

$[x_m, t_m, z_m]$

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

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 $\longleftarrow x_k^{k-1} = F_k \ x_{k-1} \longrightarrow$ 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!

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

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

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 $\beta = 1$

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

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

$$= \frac{(5.01 \text{ rad MeV})}{p} = \sqrt{Var(\theta_1)} = \sqrt{Var(\theta_2)}$$

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

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

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.

Outline of the Tracking Algorithm - Preparing

- **DIGITIZE** simulated (Geant4) hits (as in the Maximum Likelihood Linear Tracker)
    - Average close hits (within $20\mathrm{ns}$ and the same scintillator) weighted by their deposited energy
    - Smear time and position information according to geometric position uncertainty and a timing uncertainty of $1\mathrm{ns}$
- **SEED** (as in the Maximum Likelihood Linear Tracker)
    - Look for pairs of hits close to lightlike separated, we require $\frac{|\mathrm{ds}^2|}{c^2} \leq 5\mathrm{ns}^2$

After preparing the digitized hits and seeds, we loop over seeds and run the tracker on them until we run out of hits, or we run out of seeds (since do not replace used hits in the hit pool).

Outline of the Tracking Algorithm - Tracking

1. **FIND** hits by Kalman filtering in both directions from seed
    - Look for digitized hits that contribute the least to the χ^2 (provided $\chi_p^2 \leq 200$)
        - If no hits satisfy this, we propagate to the next layer
2. **FILTER** and **SMOOTH** over all hits found in previous step
3. **DROP** hits based on χ^2 increment and β best estimate
    - Hits with $\chi_s^2 \geq 150$ get dropped
    - Hits must also satisfy $0.9 \leq \beta \leq 1.1$
4. **FIT:** Run filter and smoother without altering hits
    - **VETO** a track if
        - $\chi^2/\mathrm{ndof} \geq 15$
        - The track has fewer than 3 layers with hits

Outline of the Tracking Algorithm - Vertexing

1. **MERGE** tracks that have a similar opening angle and have a close approach distance
    - We merge two tracks if
        - The opening angle satisfies $\cos\theta \geq 0.998$
        - And closest approach distance is $\leq 25\mathrm{cm}$
2. **VERTEXER** used is a Maximum Likelihood fitter (unchanged)
    - Only use lowest hit and velocity to represent the track
    - **SEED** with pairs of tracks that have a closest approach distance $\leq 100\mathrm{cm}$
        - The seed vertex position is taken to be the midpoint of the particle positions at the time of closest approach
    - **FIT** for vertex position using closest approach position
        - Closest approach cut is $100\mathrm{cm}$, add any other tracks that make the cut
        - **VETO** the vertex if the $\chi^2/\mathrm{ndof} \geq 15$

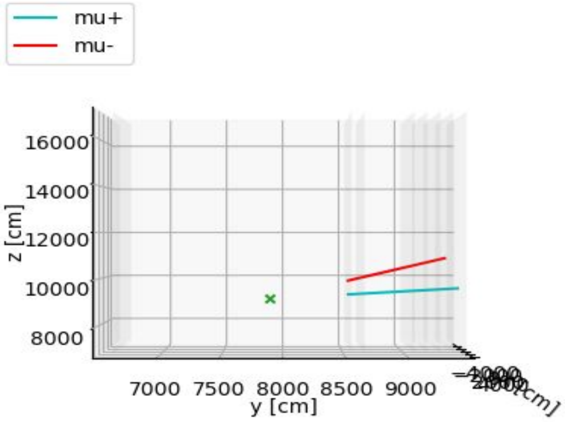
Algorithm Output - Made Tracks and Vertices

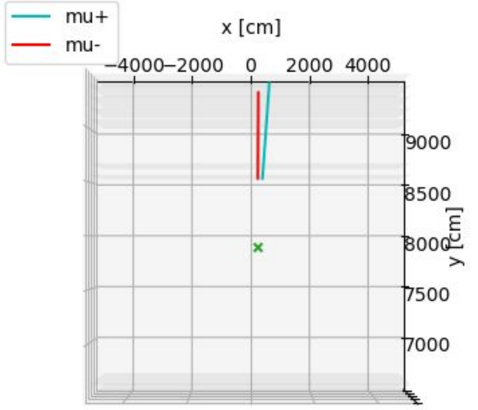
The following numbers were run using the update geometry which includes both the IP facing wall and hermetic floors

Algorithm	10 GeV mumu (3875 events)	2 GeV mumu (1007 events)	10 GeV uubar (360 events)	W Background (9436 events)
Kalman Tracks	4245	1016	3741	5254
Kalman Vertices	785	224	625	159

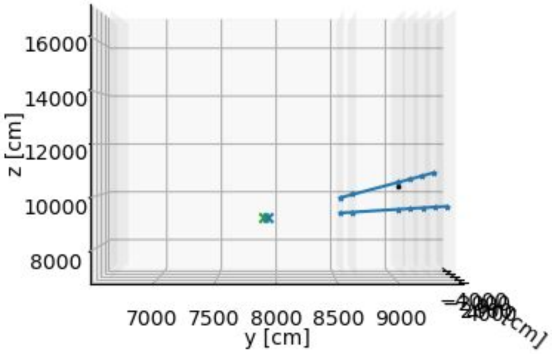
*All signal lifetimes are $c\tau = 50\mathrm{m}$

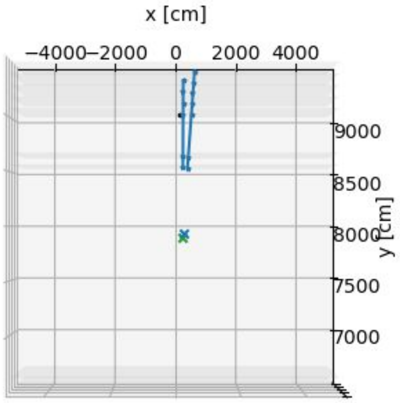
Visualisations of Signal Events - 10 GeV mumu





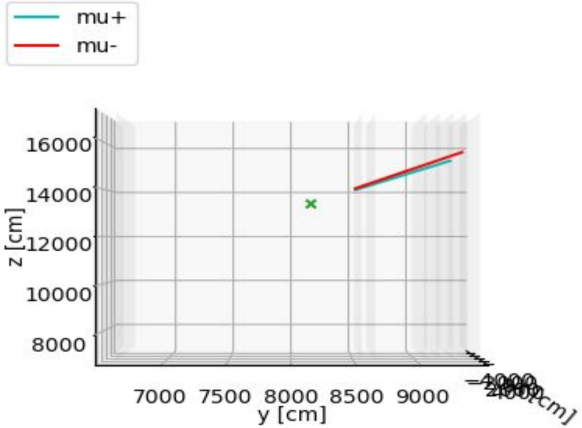
Simulated Tracks
and Decay Location

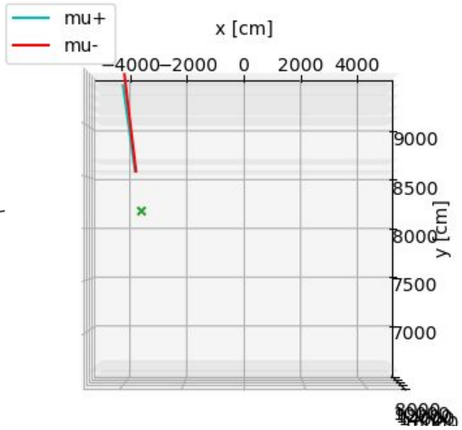




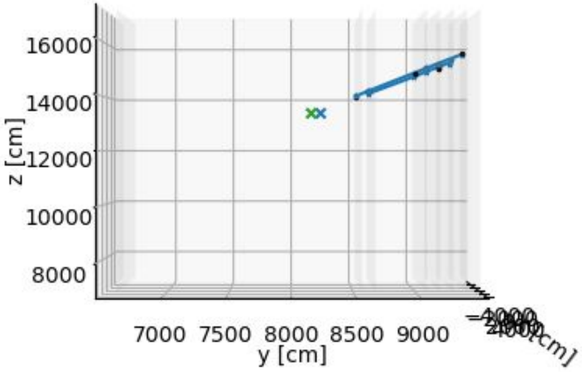
Reconstructed
Tracks and Vertex

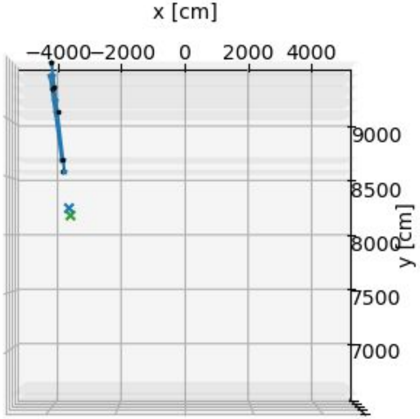
Visualisations of Signal Events - 2 GeV mumu





Simulated Tracks and Decay Location





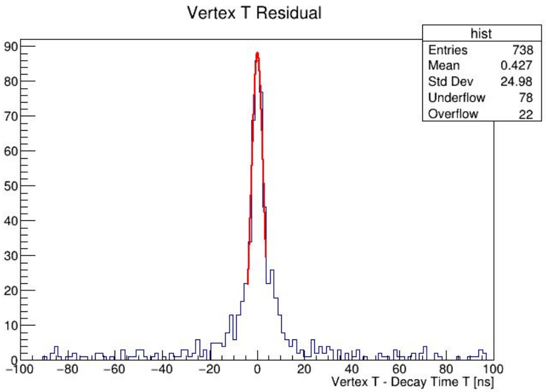
Reconstructed Tracks and Vertex

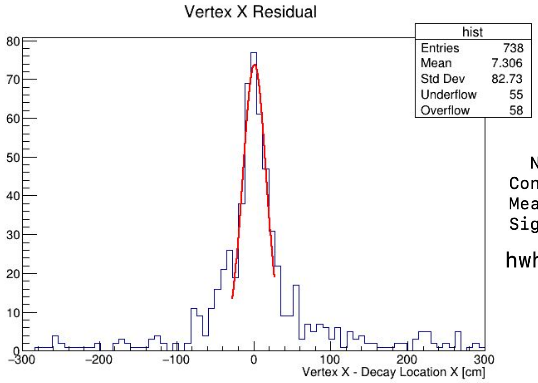
Vertexer Residual Plots - 10 GeV mumu

$$\sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} = 52.86\text{cm}$$

NAME	VALUE	ERROR
Constant	8.81631e+01	6.90536e+00
Mean	2.38688e-01	1.84612e-01
Sigma	2.47311e+00	2.24909e-01

hwhm is 2.36



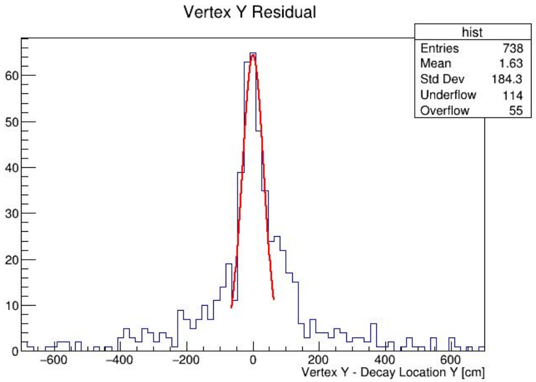


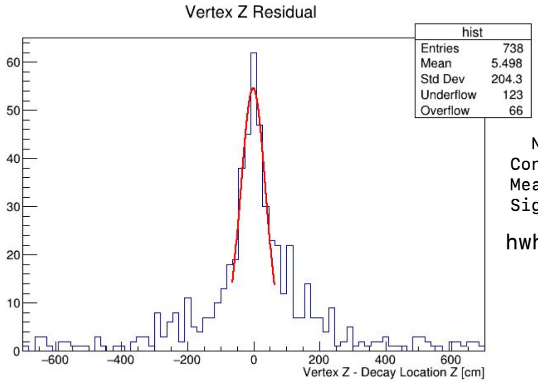
NAME	VALUE	ERROR
Constant	7.38243e+01	5.47066e+00
Mean	1.52132e+00	1.05898e+00
Sigma	1.54787e+01	1.16393e+00

hwhm is 11.68

NAME	VALUE	ERROR
Constant	6.44837e+01	5.19707e+00
Mean	1.46190e+00	2.45490e+00
Sigma	3.29230e+01	2.37035e+00

hwhm is 36.36





NAME	VALUE	ERROR
Constant	5.45335e+01	4.71654e+00
Mean	-4.33025e-01	2.99534e+00
Sigma	3.83507e+01	3.67359e+00

hwhm is 27.27

References

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

Conclusions and Next Steps

Updating the tracker to account for multiple coulomb scattering improves the quality of made tracks, especially for low mass signal.

Next we will carry out cutflow analysis and a variety of studies to characterise the performance of the algorithm

Detail on Initialising the Filter

To initialise the first state and covariance we use our two seed hits. If our two hits are m^1 and m^2 ($m_y^1 < m_y^2$), with errors ϵ^1 and ϵ^2 then we initialise the filter with

$$x_0 = \left[m_x^1, m_t^1, m_z^1, \frac{m_x^2 - m_x^1}{m_t^2 - m_t^1}, \frac{m_y^2 - m_y^1}{m_t^2 - m_t^1}, \frac{m_z^2 - m_z^1}{m_t^2 - m_t^1}\right]$$

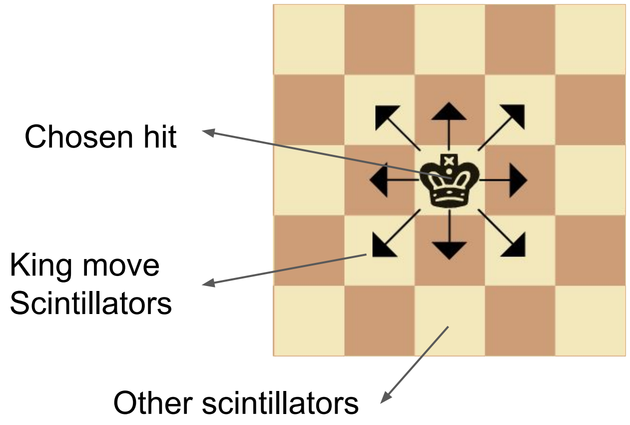
We then propagate these errors to initialise the covariance by using the Jacobian approximation we used to calculate Q.

Removing Duplicate Tracks

We have two main ways to remove duplicate tracks and avoid fake vertices

- "King Moves" Algorithm
  - If we think of a scintillator layer as a chessboard, then we remove at most 1 hit from the hit pool that could have been added to the track and are within 1 "king move" of the scintillator of the chosen hit
    - This is a way to try to remove delta ray hits that could form duplicate tracks

- Track Merging Algorithm
  - We also merge if tracks have a small closest approach distance and opening angle



Model Details - Matrix Particulars

$$F_k = \begin{pmatrix} 1 & 0 & 0 & \frac{\Delta y}{v_y} & 0 & 0 \\ 0 & 1 & 0 & 0 & \frac{\Delta y}{(v_y)^2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{\Delta y}{v_y} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$