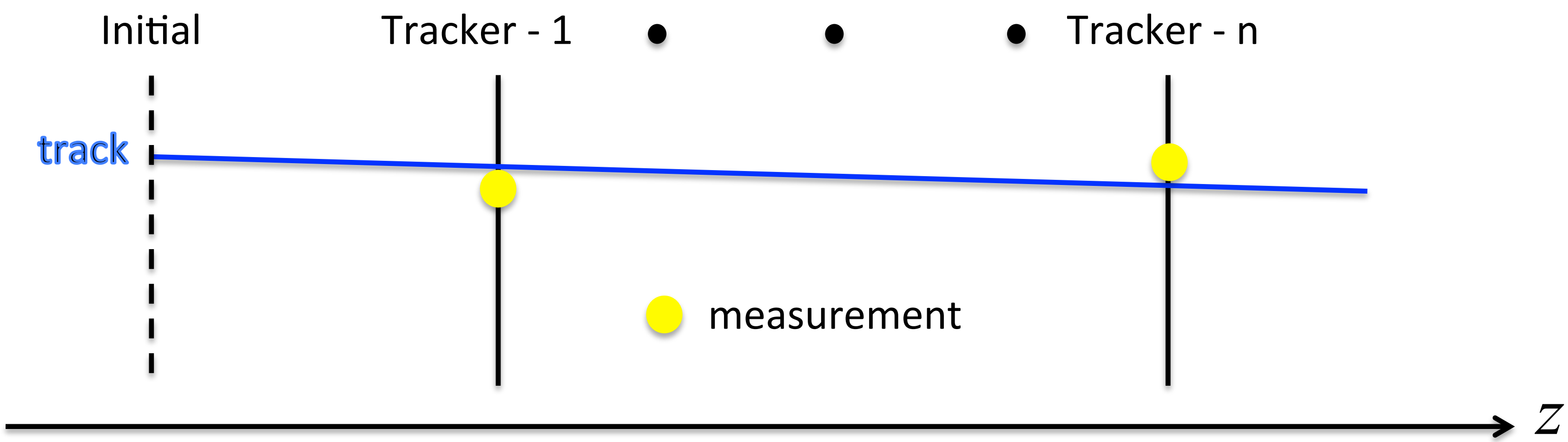
Tracking Templates for Plane-Tracker Detection Systems in the Absence of Magnetic Field

Tongtong Cao

Outline

- Introduction to tracking templates
- Physics happen in the passage of a track
- Algorithms:
  - Kalman Filter (KF)
  - Smoother
  - Seed determination
  - Treatment for multi-hits on trackers: minimized χ^2; Deterministic Annealing Filter (DAF)
  - Selection of candidate tracks
- Two version of template packages
- The third version of template
- Package explanations
- Preliminary tests
- Summary

Introduction to Tracking Templates



- Tracking templates are independent of detection systems, and can be applied to reconstruct particles' trajectories through hits' measurements on plane trackers for detecting systems in the absence of magnetic field.
- For a specified detection system, users just need to set up some parameters in a header file for configuration, such as number of trackers, position of trackers in z axis, particles' properties, properties for materials in the path of tracks, etc, and define input and output in the main function.

Physics

- Mean energy loss
  - Due to ionization $K = 4\pi N_A r_e^2 m_e C^2 = 0.307075 \frac{MeV \cdot cm^2}{g}$
    - Heavy particle: Bethe-Bloch formula

$$-\frac{dE}{ds} = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} (\frac{1}{2} ln(\frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2}) - \beta^2) \qquad \beta\gamma \in [0.1, 1000]$$

    - Lighter particle: formulas are from Computer Physics Communications, 79:157-164, 1994
      - For e^+

$$-\frac{dE}{ds} = \frac{1}{2} K \frac{Z}{A} (2ln\frac{2m_e c^2}{I} + 4ln\gamma - 2) \qquad \beta\gamma \in [0.1, 1000]$$

      - For e^-

$$-\frac{dE}{ds} = \frac{1}{2} K \frac{Z}{A} (2ln\frac{2m_e c^2}{I} + 3ln\gamma - 1.95) \qquad \beta\gamma \in [0.1, 1000]$$

  - Due to radiation (only for $e^{\pm}$)

$$E' = E_0 e^{-\frac{s}{X_0}}$$

- Multi-scattering

$$\theta_{ms} = \frac{13.6}{\beta c p} z \sqrt{\frac{s}{X_0}} (1 + 0.038 ln \frac{s}{X_0})$$

Effective Material Properties of Trackers

A tracker consists of several components, which properties are different. The effective material properties of the tracker is calculated as mass-weight averages.

Weight: $p_i = \frac{m_i}{\Sigma m_i} = \frac{v_i \rho_i}{\Sigma v_i \rho_i}$

ρ_{eff}: $\frac{1}{\rho_{eff}} = \Sigma \frac{p_i}{\rho_i}$

A_{eff}: $A_{eff} = \Sigma p_i A_i$

X_{0eff}: $\frac{1}{\rho_{eff} X_{0eff}} = \Sigma \frac{p_i}{\rho_i X_{0,i}}$

Z_{eff}: $Z_{eff} = \Sigma p_i Z_i$

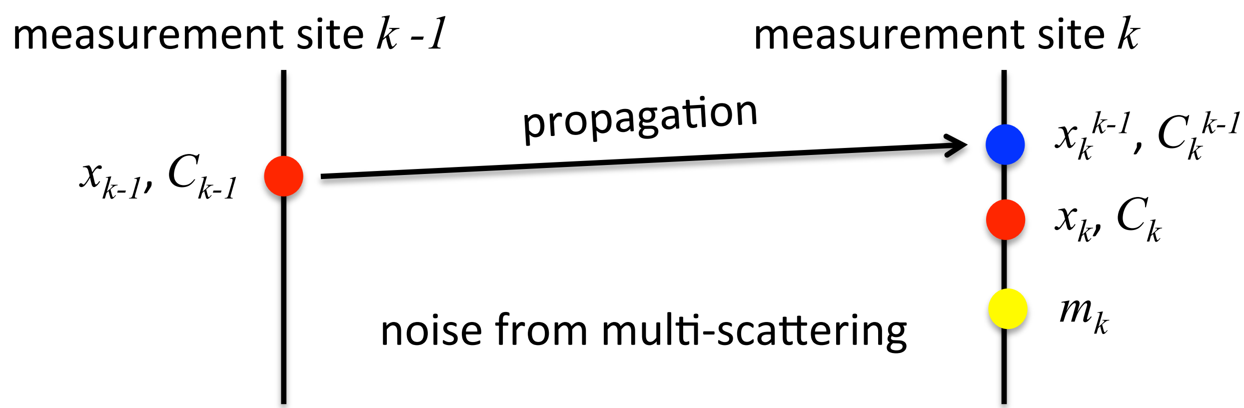
I_{eff}: $I_{eff} = exp(\frac{\Sigma p_i \frac{Z_i}{A_i} lnI_i}{\Sigma p_i \frac{Z_i}{A_i}})$

Composition	Z	A	I (MeV)	X_0 (m)	ρ (g/cm^3)	Vol. (cm^3)
# 1	?	?	?	?	?	?
…	?	?	?	?	?	?
# n	?	?	?	?	?	?
tracker	effective ?	effective ?	effective ?	effective ?	effective ?	effective ?

In the processing of tracking, candidate tracks pass through front, middle (measurement), and back planes of tracks in turns.

KF: Basic Idea and Key Problem

- Basic idea: the Kalman filter proceeds progressively from one measurement to the next and improves the knowledge about the particle trajectory by updating the track parameters with each new measurement
- Key problem: the state vector and its covariance at the measurement site $k\text{-}1$ is given, estimated state vector and its covariance at the site k can be calculated by propagation. What is the best solution to update state vector with consideration of noise from multi-scattering using measurement, estimated stated vector and its covariance?



KF: Answer to the Key Problem

The Kalman filter proceeds progressively from one measurement to the next and improves the knowledge about the particle trajectory by updating the track parameters with each new measurement.

- The state-space system from measurement site k-1 to k:

  $\tilde{x}_k = F_{k-1} x_{k-1} + w_{k-1}$ and $m_k = H_k \tilde{x}_k + \nu_k$

  - $F_{k\text{-}1}$ is propagation matrix with consideration of Lorentz force;
  - $w_{k\text{-}1}$ denotes random process noise with consideration of multi-scattering, which is regarded as Gaussian distribution with zero mean
  - H_k is projector matrix
  - v_k denotes measurement noise
- Estimated state at site k by propagation : $x_k^{k-1} = F_{k-1} x_{k-1}$
- Estimated covariance matrix at site k by propagation :

$$C_k^{k-1} = E[(\tilde{x}_k - x_k^{k-1})(\tilde{x}_k - x_k^{k-1})^T] = F_{k-1} C_{k-1} F_{k-1}^T + Q_{k-1} \quad (Q_{k-1} \equiv w_{k-1} w_{k-1}^T)$$

KF: Answer to the Key Problem

- The updated state by the filter: $x_k = x_k^{k-1} + K_k(m_k - H_k x_k^{k-1})$
  - m_k - $H_k x_k^{k-1}$ is called the innovation term, which is residual of the real measurement and the expected measurement
  - K_k is called the Kalman gain matrix, which represents a measurement by how much the innovation improves the expected state.
- The sum of the variances of the estimation errors: $J_k = TrC_k$, where $C_k = E[(\tilde{x}_k - x_k)(\tilde{x}_k - x_k)^T]$
- To minimize J_k, $\partial J_k / \partial K_k = 0$, then

$$K_k = C_k^{k-1} H_k^T (H_k C_k^{k-1} H_k^T + V_k)^{-1} \quad (V_k \equiv \nu_k \nu_k^T)$$

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

KF: Equations

Prediction equations:

$$x_k^{k-1} = F_{k-1} x_{k-1}$$

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

k represents index of measurement site

Filter equations:

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

$$x_k = x_k^{k-1} + K_k (m_k - H x_k^{k-1})$$

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

x: state vector
F: propagation matrix
C: covariance matrix of state
Q: covariance matrix of process noise

K: Kalman gain matrix
V: covariance matrix of measurement noise
H: projector matrix
m: measurement vector

KF: Track Representation, F and Q

- Reference: z
- State vector: $x = (x, y, t_x, t_y)^T$
- F and Q between two sites

$$F = \begin{bmatrix} 1 & 0 & \Delta z & 0 \\ 0 & 1 & 0 & \Delta z \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & Q_{t_x t_x} & Q_{t_x t_y} \\ 0 & 0 & Q_{t_y t_x} & Q_{t_y t_y} \end{bmatrix}$$

$$Q_{t_x t_x} = (1 + t_x^2)(1 + t_x^2 + t_x^2)\theta_{ms}^2$$
$$Q_{t_y t_y} = (1 + t_y^2)(1 + t_x^2 + t_x^2)\theta_{ms}^2$$
$$Q_{t_x t_y} = Q_{t_y t_x} = t_x t_y (1 + t_x^2 + t_x^2)\theta_{ms}^2$$

- If there are multi-sites (or multi-steps in propagation) between two measurements, then

$$F = F_1 F_2 \cdots \qquad Q = Q_1 + Q_2 + \cdots$$

- Discussion for propagation between two sites
  - Propagation for F only need to be implemented by one step
  - Propagation for momentum and Q are designed to be implemented by n steps, where n depends on the setup of step size, so that momentum and Q can be updated step by step

KF: m, V and H

$$m = (x, y)^T$$

$$V = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

KF: χ^2

In case of the Kalman filter, there are two residuals to consider: The first one is the residual between the expected state vector x_k^{k-1} after propagation from measurement *k-1* to *k* and the state vector x_k after filtering. The second one is the residual of the actual measurement m_k and the measurement expected from the filtered estimate $h_k(x_k)$.

$$\chi^2_k = \chi^2_{k-1} + (x_k - x_{k-1})^T (C_k^{k-1})^{-1} (x_k - x_{k-1}) + (m_k - h_k(x_k))^T (V_k)^{-1} (m_k - h_k(x_k))$$

Smoother

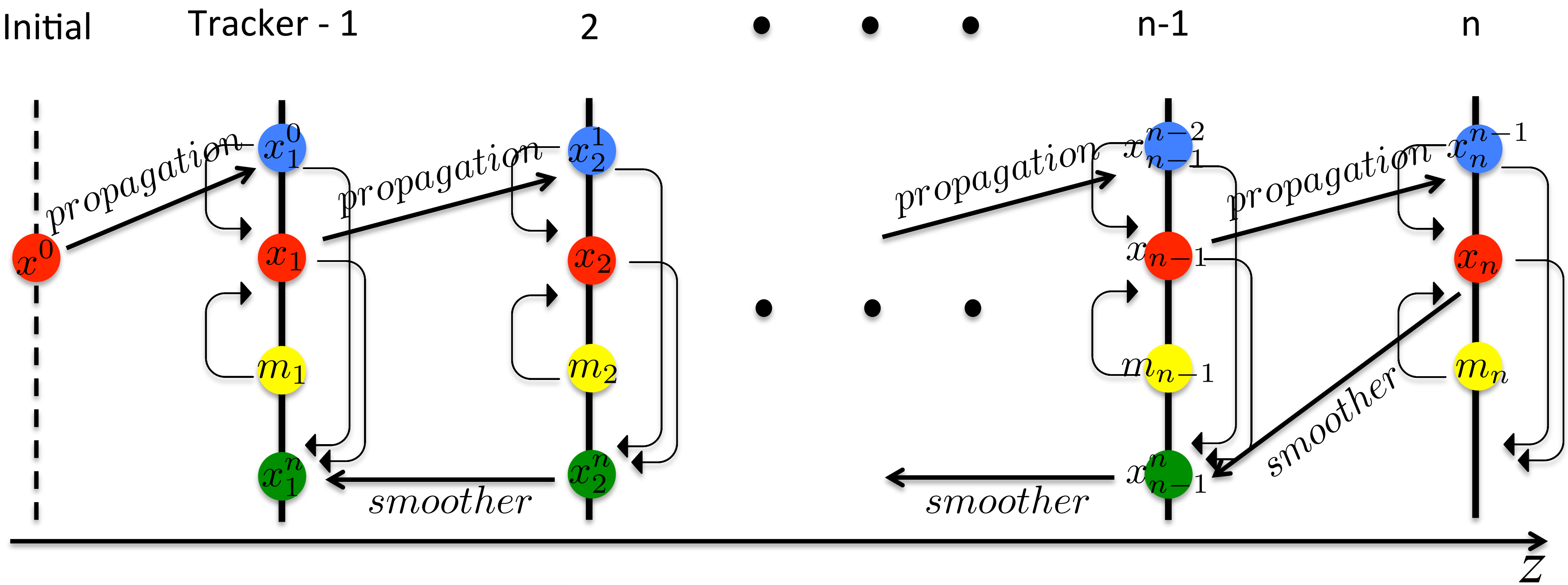
- Due to the recursive nature of the Kalman filter approach the computed state vector x_k is based on the k measurements collected so far. It is unaffected by subsequent estimates. A smoother allows to further improve this estimate using information from any subsequent measurements as well.
- The smoothing is also a recursive procedure which proceeds step by step in the direction opposite to that of the filter with the smoother equations:

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

$$A_k = C_k F_k^T (C_{k+1}^k)^{-1} \qquad k \in \{1, \cdots, n-1\}$$

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

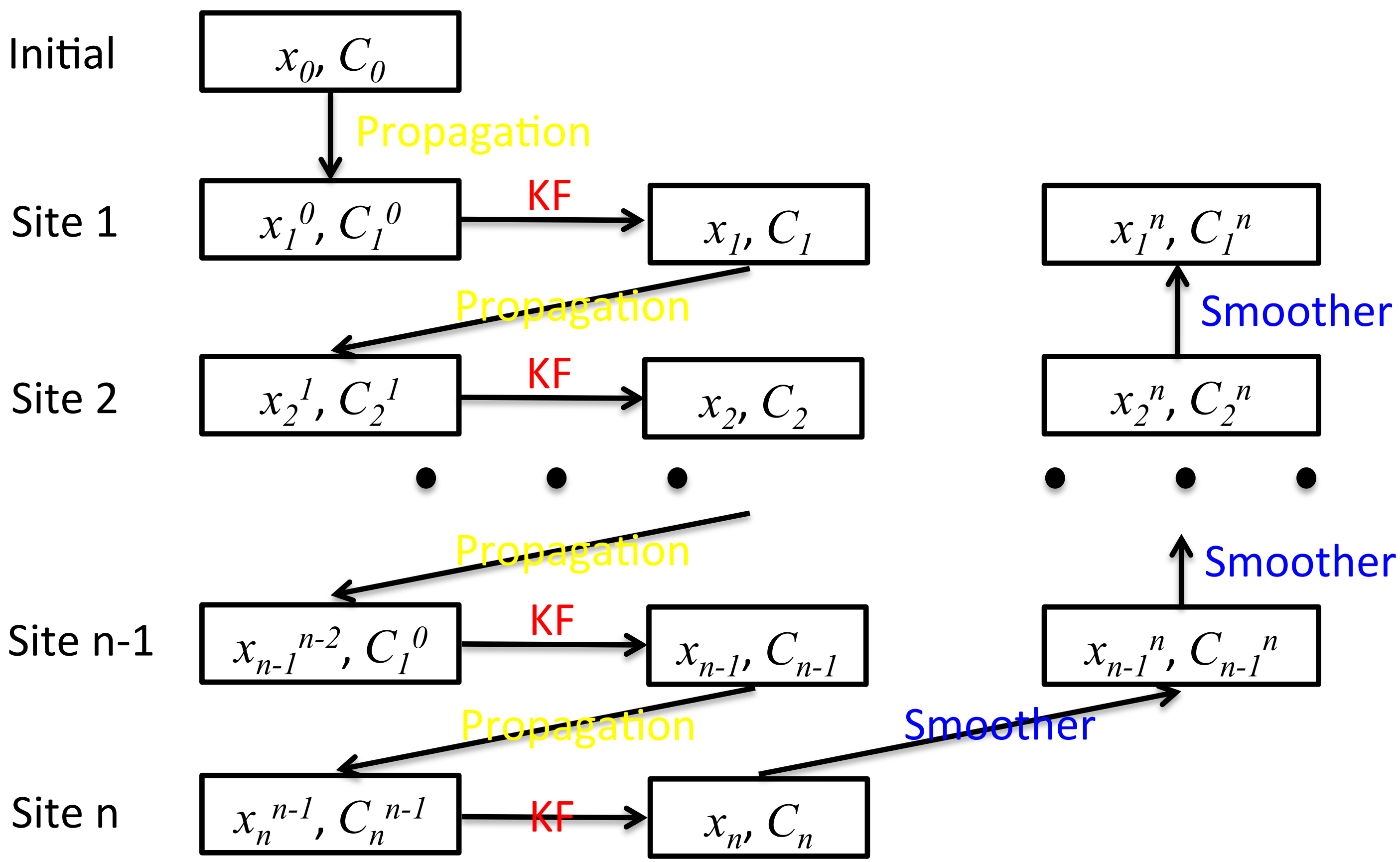
Scheme of KF + Smoother



- Predicted state vector
- Filtered state vector
- Measurement
- Smoothed state vector

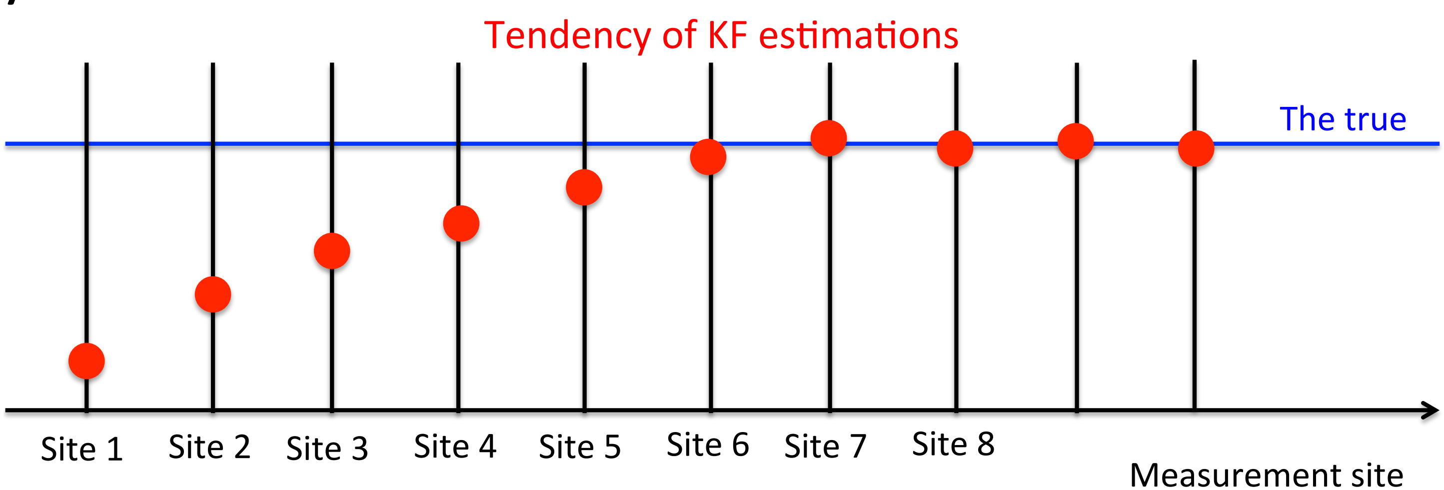
- The prediction makes an estimate of the future state of the system.
- The filter estimates the system state in the present.
- The smoother allows to estimate the state of the system in the past.

Scheme of KF + Smoother Programing



Initial State Vector and Its Covariance Matrix

- For general applications of KF with enough measurements, the setup of initial state vector and its covariance is not very restrictive, and estimated states tends to the true site by site.



- For the application of physics experiments, the setup of initial state vector and its covariance is required to be relatively close to the true situation since there are no enough measurement sites.

Seed Determination

- Initial state vector and its covariance matrix, are determined by two hits from two different trackers. Any two hits can build a seed for tracking.
- Two trackers with minimized number (>0) of hits are chosen to build seed(s) so as to minimize number of candidate tracks.
- Determination of a seed:

$$\begin{cases} x_0 = (z_2 - z_0)x_1/(z_2 - z_1) - (z_1 - z_0)x_2/(z_2 - z_1) \\ y_0 = (z_2 - z_0)y_1/(z_2 - z_1) - (z_1 - z_0)y_2/(z_2 - z_1) \\ t_{x_0} = (x_2 - x_1)/(z_2 - z_1) \\ t_{y_0} = (y_2 - y_1)/(z_2 - z_1) \end{cases}$$

$$\begin{cases} var(x_0) = ((z_2 - z_0)/(z_2 - z_1))^2\sigma_{x_1}^2 + ((z_1 - z_0)/(z_2 - z_1))^2\sigma_{x_2} \\ var(y_0) = ((z_2 - z_0)/(z_2 - z_1))^2\sigma_{y_1}^2 + ((z_1 - z_0)/(z_2 - z_1))^2\sigma_{y_2} \\ var(t_{x_0}) = \sigma_{x_1}^2/(z_2 - z_1)^2 + \sigma_{x_2}^2/(z_2 - z_1)^2 \\ var(t_{y_0}) = \sigma_{y_1}^2/(z_2 - z_1)^2 + \sigma_{y_2}^2/(z_2 - z_1)^2 \end{cases}$$

$$x_0 = (x_0, y_0, t_{x_0}, t_{y_0})^T$$

factor: variance of initial state parameters should be somewhat larger than calculated values by two hits since effects of multi-scattering

$$C_0 = factor^2 \begin{bmatrix} var(x_0) & 0 & 0 & 0 \\ 0 & var(y_0) & 0 & 0 \\ 0 & 0 & var(t_{x_0}) & 0 \\ 0 & 0 & 0 & var(t_{y_0}) \end{bmatrix}$$

Multi-hit Treatment on Trackers

(not used to construct initial state)

- Sources for multi-hits
  - From multi-tracks: for a specified track, hits from other tracks are regarded as noise
  - From noise:
    - random noise uncorrelated with the passage of a track
    - track correlated noise like corsstalk, delta rays, and cluster decays
- Two ways
  - Minimized χ^2: compare χ^2 for all hits on a tracker and choose the one with minimized χ^2
  - DAF: An effective hit is calculated by all hits with respective weights, and then applied in tracking

DAF: Determination of Weights and Effective Hit

DAF consists of two steps: First, tracking with the application of KF and smoother is run on fixed hits, which are chosen to construct an initial state, and effective hits for trackers, which are not used to construct initial state(s) with application of their all hits and respective weights. Then the weights are recalculated according to the predicted residuals of the hits (χ^2).

- An effective hit:

$$\bar{\boldsymbol{m}}_k = \bar{V}_k \left(\sum_{i=1}^{n} p_k^i \left(V_k^i\right)^{-1} \boldsymbol{m}_k^i \right)$$

$$\bar{V}_k = \left(\sum_{i=1}^{n} p_k^i \left(V_k^i\right)^{-1} \right)^{-1}$$

- k: index of a tracker
- i: index of a hit on the tracker
- r: dimension of measurements

- Weights: $x^n{}_k$ is estimated state from the smoother

$$p_k^i = \frac{\Phi_k^i}{\sum_{j=1}^{n} \Lambda_k^j + \Phi_k^j}$$

$$\Phi_k^i = \frac{1}{(2\pi)^{\frac{r}{2}} \sqrt{T \cdot |V|}} \cdot \exp\left(-\frac{1}{2T} \left(\boldsymbol{m}_k^i - \boldsymbol{h}_k\left(\boldsymbol{x}_k^n\right)\right)^T \left(V_k^i\right)^{-1} (\dots) \right)$$

$= \chi^2$

$$\Lambda_k^i = \frac{1}{(2\pi)^{\frac{r}{2}} \sqrt{T \cdot |V|}} \cdot \exp\left(-\frac{\chi^2_{\text{cut}}}{2T} \right)$$

DAF: Discussion for Parameters

- Cut-off parameter χ^2_{cut}: control the cut value; equivalent to a χ^2 cut for low T
- Annealing factor T: basically inflates the measurement errors
- Effects of two parameters:

Weight function of an observation with no competition

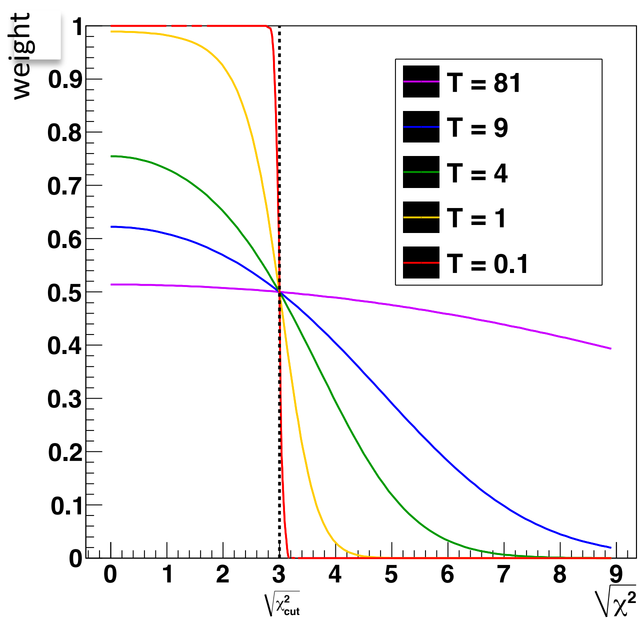


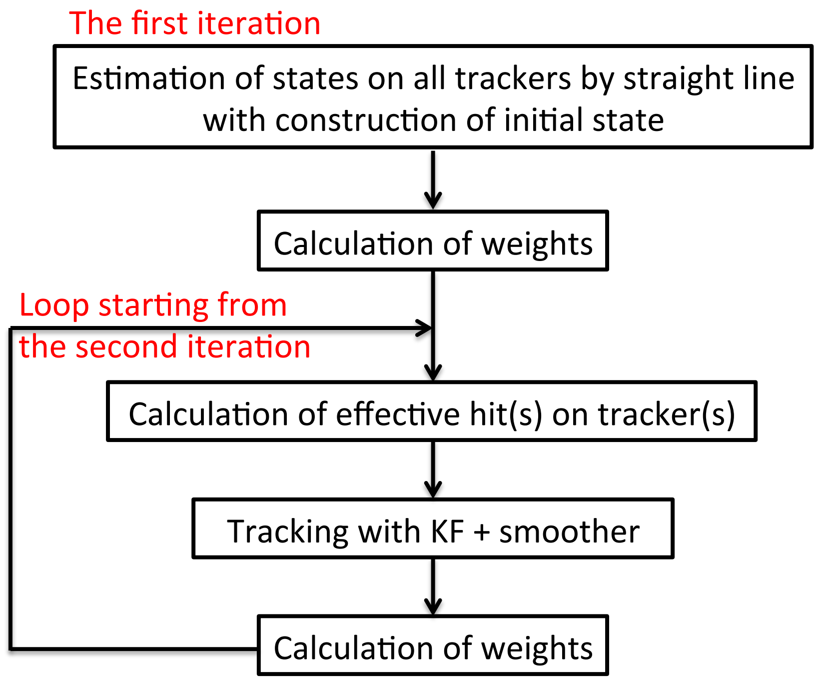
Figure is from Erik Krebs's master thesis

- χ^2 has little effect on the weight for very high T so that even measurements that are far away are still taken into account.
- The lower T the higher the weight is accepted by χ^2 and DAF is forced to make a decision whether to accept or reject the measurement.
- Cooling close to $T = 0$ gives a "hard" association. This is not the optimal approach.

DAF: Iteration and NDF

- Iteration: A problem when optimizing measurement weights is to avoid local optima since information in the initial phases of the filter may be insufficient. Therefore, a process is introduced where DAF with two steps (tracking and weight recalculation) are iterated several times and T is lowered in each iteration.

Scheme of DAF Iteration



- NDF: Since the measurements contribute according to their weights, NDF is $\sum_k \hat{V}_k \sum_i p_k^i (V_k^i)^{-1} r - s$.

Selection of Candidate Tracks

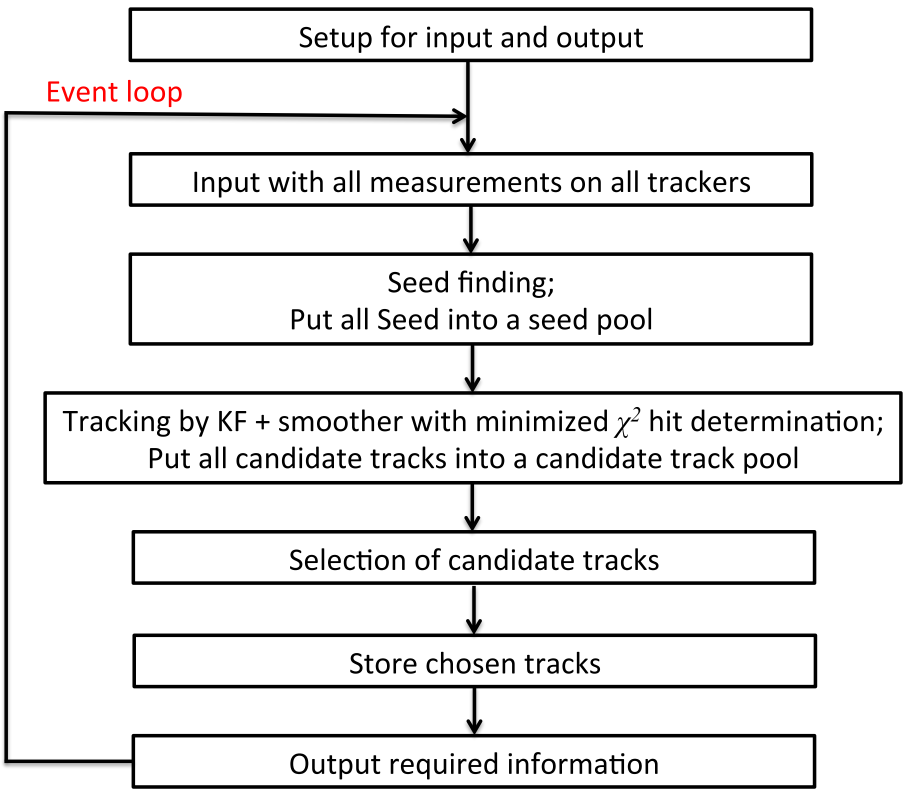
Two Steps:

1. Take an order of candidate tracks:
   - First priority: number of hits used for the track reconstruction is from large to small
   - Second priority: χ^2/NDF is from small to large

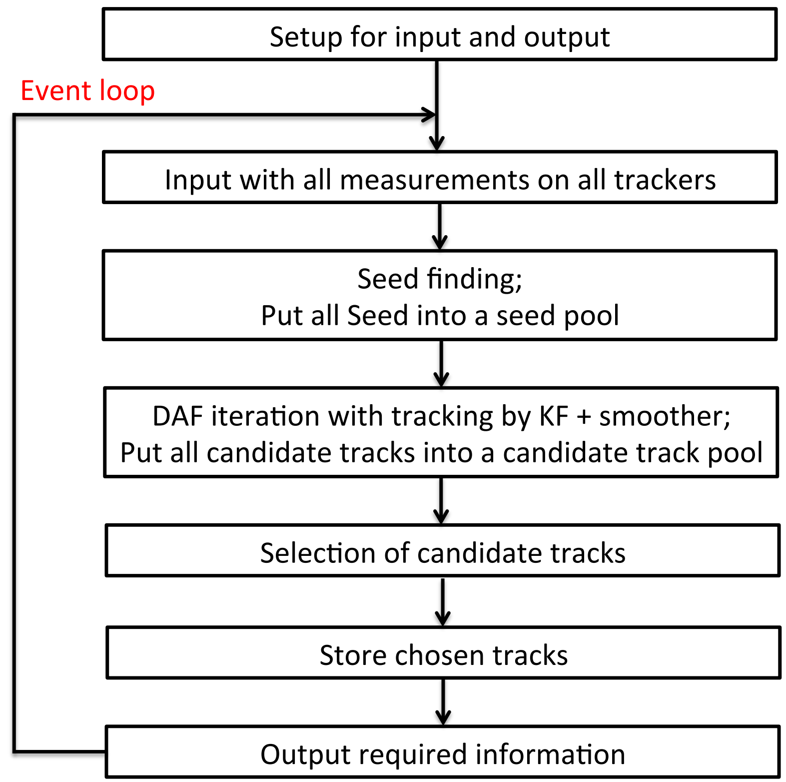
2. Select good tracks:
   - Firstly, the first track in the order is chosen and store its respective hit measurements for each tracker into a pool.
   - Then, check if the second track has different hit measurements from measurements of the first track for all trackers. If yes, the second track is chosen and store its respective hit measurement for each tracker into the pool; If no, cancel this track.
   - Continue to check if the next track has different hit measurements from all of stored measurements in the pool for all trackers. If yes, the track is chosen and store its respective hit measurement into the pool. If no, cancel this track;
   - Continue Until the last candidate track.

Scheme of Package Programming

Version 1: minimized χ^2



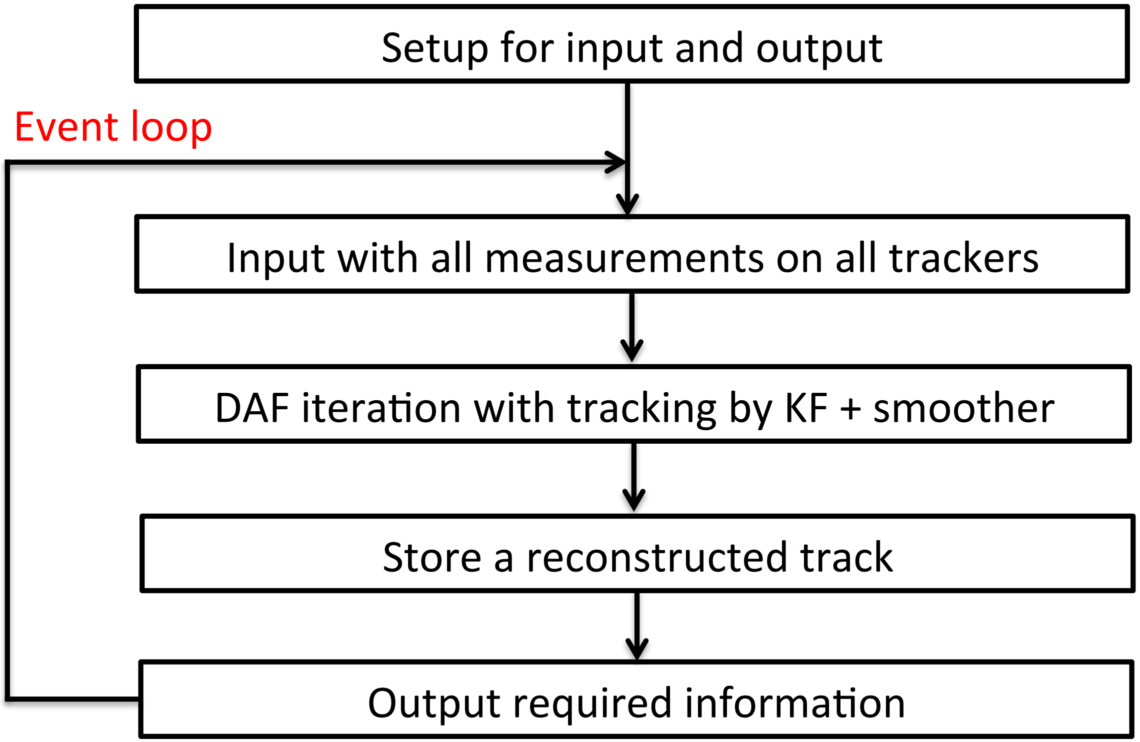
Version 2: DAF



- The DAF is a better way to treat multi-hits on trackers, especially for trackers with hit(s) caused by track correlated noise.
- Running time for V2 is several times than V1.
- Tracking for both V1 and V2 starts from the seed determination.

Version 3 with DAF for Focused Beams

- For the beam reconstruction, if a beam is focused and only 1 track is concerned for each event, the setup for initial state and its covariance can be set as common values instead of construction by seed
- Comparison to the version 2:
  - Starting from the initial site with common state and covariance, so no seed finding
  - Only 1 reconstructed track, so no selection for candidate tracks



Packages: Overview

- Environment: C++ and ROOT
- Build tool: Make
- Files in the package:
  - Main function: templateTracking.cxx
  - Configuration setup: utility.h
  - Classes:
    - Particles' properties: particle.h, particle.cxx
    - Plane and hit on plane: plane.h, plane.cxx, planeHit.h, planeHit.cxx
    - Trackers: tracker.h, tracker.cxx
    - Propagation: stepper.h, stepper.cxx
    - Matrix definition: matrix.h, matrix.cpp
    - Tracking by Kalman filter and smoother: trackSystem.h, trackSystem.cxx, trackSite.h, trackSite.cxx, trackState.h, trackState.cxx
    - Information storage for reconstructed track: track.h, track.cxx
    - Tracking engine: trackFinder.h, trackFinder.cxx
  - To build a dictionary: template_LinkDef.h
  - Makefile

Packages: Unit

- Length: mm
- Time: ns
- Momentum: MeV/c
- Material properties:
  - I: MeV
  - X_{0}: m
  - ρ: g/cm^3
  - Volume: cm^3

Packages: How to Run

1. In utility.h, assign values to parameters according to a detection system's configuration
2. In templateTracking.cxx, set input and output according to formats of input and output files
3. Make and run

General Setup for Tests

- 4 layers of trackers
- Assumed positions of starting and tracker sites in z axis: 0, 100, 200, 300, 400 (mm)
- Pseudo-measurements (mm): first, obtain true values for hits on sites with assumption of initial state vector; then, smear true values by Gaussian with $\sigma = 1$ (mm)
- Material in the passage of tracks: only air

Test for DAF - T in V2

- True hit locations and measurements on trackers:

Tracker	Real init: (0, 0, 0.1, 0.1)		Pseudo-measurement	
	x (mm)	y (mm)	x (mm)	y (mm)
# 1	10	10	10.5	9.7
# 2	20	20	20.9	19.2
# 3	30	30	29.7; 33.3	30.7; 27.2
# 4	40	40	39.5	40.6

- For tracker - 3, there are two measurements in both x and y. One is from track, and the other is noise. The difference is 3σ to 4σ.
- Four hits are constructed by 2 × 2 measurements. The first hit is from track, and others include measurement(s) from noise.

- Parameters: χ^2_{cut} = 4; T in 6 iterations: (81, 9, 4, 1, 1, 1)
- Weights for each iteration:

T	Weights			
81	0.123952	0.125298	0.122855	0.124189
9	0.147546	0.117531	0.108301	0.0862689
4	0.185312	0.103629	0.0842593	0.0471189
1	0.451163	0.0279704	0.010225	0.000633913
1	0.615149	0.00626075	0.00214672	2.18485e-05
1	0.620466	0.00543204	0.0020815	1.82231e-05
	hit - 1	hit - 2	hit - 3	hit - 4

- For the first iteration, weights almost have no difference among hits.
- In the processing of iterations, the weight for the first hit from the track is larger and larger, while weights for all noise hits are smaller and smaller, until negligible.
- No hits are removed since the threshold for weights is set as 0

Test for a Two-Track Event in V1 and V2

- True hit locations and measurements on trackers: two tracks cause two hits, i.e. two measurements for both x and y, on all trackers

Tracker	Real init: (0, 0, 0.1, 0.1)		Real init: (10, 10, -0.1, -0.1)		Pseudo-measurement	
	x (mm)	y (mm)	x (mm)	y (mm)	x (mm)	y (mm)
# 1	10	10	0	0	10.5; 0.6	9.7; -0.7
# 2	20	20	-10	-10	20.9; -10.7	19.2; -9.2
# 3	30	30	-20	-20	29.7; -19.5	30.7; -21.2
# 4	40	40	-30	-30	39.5; -30.5	40.6; -29.7

- Parameters for V2: χ^2_{cut} = 4; T in 6 iterations: (81, 9, 4, 1, 1, 1)
- Two tracks are successfully reconstructed from both V1 and V2
  - Results of V1:
    - tracker - 4 for track 1: (39.57, 40.62, 0.0962, 0.1037)
    - tracker - 4 for track 2: (-30.41, -29.96, -0.1026, -0.09828)
  - Results of V2:
    - tracker - 4 for track 1: (39.52, 40.69, 0.09606, 0.1038)
    - tracker - 4 for track 2: (-30.29, -30.13, -0.1025, -0.09834)

Test for Version 3

- Common initial state: (0, 0, 0, 0)
- Common uncertainties: (10, 10, 0.087, 0.087)
- True hit location and measurements on trackers:

Tracker	Real init: (8, 6, 0.05, 0.06)		Pseudo-measurement	
	x (mm)	y (mm)	x (mm)	y (mm)
# 1	13	12	12.5; 18.3	12.8; 20.7
# 2	18	18	16.8; 14.7	17.3; 22.2
# 3	23	24	23.9; 22.8	19.3; 27.2
# 4	28	30	27.2; 79.8	30.4; 80.9

- For all trackers, there are two measurements in both x and y. One is from track, and the other is noise.
- Four hits are constructed by 2 × 2 measurements. The first hit is from track, and others include measurement(s) from noise.

- Parameters: χ^2_{cut} = 4; T in 6 iterations: (81, 9, 4, 1, 1, 1)
- The track is successfully reconstructed with suppression of noise hits
  - Tracker 4: (27.68, 29.76, 0.05061, 0.05955)
  - Tracker 1: (12.44, 12.00, 0.05086, 0.05893)

Summary

- Three versions of tracking templates for plane-tracker detection systems are developed
  - V1: KF + minimized χ^2; seed finding; seed finding for initial setup
  - V2: KF + DAF; seed finding for initial setup
  - V3: KF + DAF; common initial setup specified for focused beams
- In the application of experimental data, the number of iterations and parameters' setup in DAF need to be explored to optimize efficiency.
- Very preliminary tests prove that the packages are available. More tests need to be implemented by experimental data or pseudo-data from simulation.