Recent developments in track and vertex reconstruction

R. Frühwirth, CMS

Institute for High Energy Physics of the Austrian Academy of Sciences
Vienna, Austria

DESY, Hamburg, January 13, 2003
Outline

❖ Introduction

❖ Track Fitting

✧ Track fitting with Gaussian noise: The Kalman Filter

✧ Track fitting with non-Gaussian noise: The Gaussian-sum Filter

✧ Track fitting with high background: The Deterministic Annealing Filter

✧ Track fitting in narrow jets: The Multi-track Filter
Outline

❖ **Vertex Fitting**

✧ **Linear vertex fit**

✧ **Robust single-vertex fit**

✧ **Adaptive Multi-vertex fit**

❖ **Summary and Outlook**
Track fitting typically involves the following steps:

- **Raw data conversion**
  - Use calibration constants; produce channel hits

- **Cluster reconstruction**
  - Use alignment constants; produce space points

- **Local track finding**
  - Use magnetic field and detector geometry; produce track elements

- **Global track finding**
  - Use magnetic field and detector geometry; produce track candidates
Track Fitting

- **Track fitting**
  Use field, geometry and material; produce track parameters, errors and test statistics ($\chi^2$)

- **Test of track hypothesis and track selection**
  Use global track information; produce best subset of tracks

The exact sequence is of course detector dependent. Some steps may be iterated.

Example: Find high momentum tracks first, clean up, then find low momentum tracks.
In the first part I concentrate on the track fitting part:

- **Estimation of track parameters**
  Statistically optimality, speed

- **Computation of error matrices**
  Representative for actual errors

- **Computation of test statistics**
  $\chi^2$-statistics, pulls, ...
The Kalman filter

Today KF is widely used. Advantages:

- **Recursive**
  No inversion of large matrices, small steps

- **Estimates stay close to the track**
  Local information for clustering, multiple scattering, energy loss, …

- **Can be augmented by smoother**
  Optimal estimates anywhere, important for robustification

- **Easily generalized**
  Some extensions will be dealt with later

- **Optimal in the linear model with Gaussian noise**
How does it work? Basically only two different steps have to be iterated:

- **Prediction step**
  Extrapolate local estimate and local error matrix to the next detector element. Increment error matrix by process noise, e.g. multiple scattering.

- **Filter or updating step**
  Compute new local estimate by combining the prediction and the local measurement according to the LS principle (weighted average).

Repeat ad lib
The Kalman filter

Careful initialization is important!

- Use preliminary knowledge from track finding or seeding
- Use large error matrix or small weight matrix
- Sometimes needs delicate numerical considerations

Two implementations possible:

- Covariance filter
  Propagates error matrices, fast in material layers

- Information filter
  Propagates weight matrices, fast in measurement layers, numerically more stable
The Kalman filter

Only the estimate in the last detector element is based on full information. By smoothing it is propagated back to all estimates. Two ways:

- **Smoother algorithm**
  
  Uses Jacobians from filter, numerically delicate

- **Backward filter**
  
  Run filter in opposite direction, combine forward and backward filter by weighted average; numerically more stable
The Kalman filter

Three test statistics:

- **Incremental $\chi^2$ of the filter**
  
  Can be used for discrimination between competing hits, little power in the starting phase

- **Total $\chi^2$ of the filter**
  
  Can be used to test track hypothesis

- **Incremental $\chi^2$ of the smoother**
  
  Can be used for discrimination between competing hits or for searching outliers, full power for single outliers
The Gaussian-sum filter

With non-Gaussian noise the KF is suboptimal. Possible instances:

- **Measurement errors**
  - Gaussian core, long tails

- **Process noise**
  - tails in the multiple scattering distribution
  - non-Gaussian distribution of energy loss and bremsstrahlung

In some cases a non-linear filter can do better than the Kalman filter! Prime candidate: Gaussian-sum filter.
The Gaussian-sum filter

Invented by Kitagawa (1989). Main features:

- At every stage, the distribution of the track parameters is a mixture of Gaussians

\[
    f(x) = \sum_{i=1}^{n} p_i \varphi(x; \mu_i, V_i), \quad \sum_{i=1}^{n} p_i = 1
\]

- Implemented by parallel Kalman filters

- If the predicted density has \( n \) components, and the measurement density has \( m \) components, the updated density has \( m \times n \) components

- Number of components must be kept manageable
The Gaussian-sum filter

First tested for track fitting by Frühwirth (1997), with long-tailed measurement errors (mixture of two Gaussians).

- Two trimming algorithms tested (Collapse and Drop)
- For long tails appreciable gains in precision (see figure)
- Smoothing possible by combining two GSF
- Representative error matrix
- Distorted $\chi^2$-distribution
- Execution time proportional to number of components
- Mixture model of measurement errors must be available
The Gaussian-sum filter

Relative precision of the GSF
Normalized residuals and $\chi^2$-probability of the GSF
The Gaussian-sum filter

Further test by Frühwirth and Frühwirth-Schnatter (1998), with bremsstrahlung of electron tracks.

- Energy loss distribution approximated by mixture of 6 Gaussians
- Gaussian-sum filter compared to Kalman filter and Metropolis-Hastings algorithm
- Appreciable gain in precision of the momentum measurement, especially for thin layers
Another possible application is treatment of tails in multiple Coulomb scattering.

- Mixture model of multiple scattering is now available (Frühwirth and Regler, 2001)
- Ready to be used by the Gaussian-sum filter
- Gain in precision can be expected for thin layers
- CMS tracker is nice testbed, many thin layers
- Evaluation in progress
The Gaussian-sum filter

Distribution of \( z = \frac{p_1}{p_0} \)
The Deterministic Annealing Filter

Adaptive filter designed to cope with high background noise (Frühwirth and Strandlie, 1999). Main features:

- Equivalent to Elastic Arms Algorithm
- Implemented as iterated Kalman filter
- No complicated minimization necessary
- Can be combined with annealing
- Error matrix readily available
- Multiple scattering and energy can be incorporated
The Deterministic Annealing Filter

How does it work? Basically two steps have to be iterated:

- **Full track fit**
  Compute smoothed estimates $x$ in all measurement layers, by Kalman filter/smoother or any other method, using the association probabilities computed in the previous step for downweighting.

- **Weight computation**
  In each layer, compute the association probabilities of all measurements to the track.

The iteration stops when the weights have converged.
The Deterministic Annealing Filter

The DAF weights:

\[
p_i = \frac{\varphi(m_i; Hx, \alpha V_i)}{c(\alpha) + \sum_j \varphi(m_j; Hx, \alpha V_j)}
\]

- \(\varphi\) measures the distance of the measurement \(i\) from the track. Usually, but not necessarily, Gaussian. \(\alpha\) is an optional annealing factor, \(c(\alpha)\) controls the cut value.

- The measurements compete for inclusion in the fit. If there is a good measurement, noise is automatically suppressed. The final weight depends on the competitors: adaptive!
Deterministic Annealing helps to reach the global optimum:

- $\alpha > 1$ at the start
- Stepped down to 1 in the course of the iterations

Well-known technique in global optimization, e.g. of neural networks. Cooling close to $\alpha = 0$ gives a “hard” association. This is not the optimal approach (see below).
The Deterministic Annealing Filter

Weight function of an observation with no competition
The Deterministic Annealing Filter

Weight function of an observation with competition

R. Frühwirth

HEPHY Vienna
The Deterministic Annealing Filter

First study of DAF in ATLAS Transition Radiation Tracker (Frühwirth and Strandlie, 1999).

- 75 layers of straw tubes in the barrel part
- About 35 hits per track
- Mirror hit for every hit
- At least 50% noise!
The Deterministic Annealing Filter

Cross section of the barrel TRT in ATLAS
The Deterministic Annealing Filter

The DAF has been compared to its immediate competitors:

- **EAA** = Elastic Arms (Ohlsson, Peterson and Yuille, 1992)
- **ETR** = Elastic Tracking (Gyulassy and Harlander, 1991)
- **GSF** = Gaussian-sum filter
- **KF** = Kalman filter
- **Kalman filter without mirror hits (baseline)**

With mirror hits, the DAF is both the best and the fastest one!
The Deterministic Annealing Filter

<table>
<thead>
<tr>
<th>Method</th>
<th>$V_{rel}$</th>
<th>$t_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAF</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>GSF</td>
<td>1.00</td>
<td>0.44</td>
</tr>
<tr>
<td>EAA with GD</td>
<td>8.83</td>
<td>1.49</td>
</tr>
<tr>
<td>EAA with DFP</td>
<td>1.03</td>
<td>1.63</td>
</tr>
<tr>
<td>ETA-G with GD</td>
<td>1.79</td>
<td>0.73</td>
</tr>
<tr>
<td>ETA-G with DFP</td>
<td>1.05</td>
<td>1.41</td>
</tr>
<tr>
<td>ETA-L with GD</td>
<td>125.6</td>
<td>0.65</td>
</tr>
<tr>
<td>ETA-L with DFP</td>
<td>1.07</td>
<td>1.67</td>
</tr>
<tr>
<td>KF</td>
<td>1.00</td>
<td>0.07</td>
</tr>
</tbody>
</table>

The relative generalized variance, tracks w/o mirror hits
The relative generalized variance, tracks with mirror hits

<table>
<thead>
<tr>
<th>Method</th>
<th>$V_{rel}$</th>
<th>$t_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAF nominal</td>
<td>1.54</td>
<td>1.21</td>
</tr>
<tr>
<td>DAF frozen</td>
<td>1.74</td>
<td>1.41</td>
</tr>
<tr>
<td>GSF all</td>
<td>1.59</td>
<td>7.04</td>
</tr>
<tr>
<td>GSF best</td>
<td>1.78</td>
<td>7.04</td>
</tr>
<tr>
<td>EAA nominal</td>
<td>1.56</td>
<td>2.12</td>
</tr>
<tr>
<td>EAA frozen</td>
<td>1.71</td>
<td>2.44</td>
</tr>
<tr>
<td>ETR</td>
<td>3.51</td>
<td>2.87</td>
</tr>
<tr>
<td>KF</td>
<td>$\sim 1500$</td>
<td>0.08</td>
</tr>
</tbody>
</table>
The Deterministic Annealing Filter

The DAF has been implemented in the CMS reconstruction program ORCA (M. Winkler, 2002).

- Based on the building blocks of the Kalman filter in ORCA
- Track finding by a combinatorial Kalman filter
- DAF used for smoothing
- Extensively verified on single tracks
- Various physics studies
Results of a study of $b$-jets with KF and DAF:

- DAF has better impact parameter resolution
- DAF has better error estimates
- DAF has better track reconstruction efficiency at comparable fake rates
- DAF gives better secondary vertex finding efficiency

As expected, the improvements are largest for the highest jet energies.
The Deterministic Annealing Filter

Silicon strip detector part of the CMS tracker
Red: single-sided, blue: double-sided
The Deterministic Annealing Filter

Pixel detector part of the CMS tracker
Transverse impact parameter resolution of tracks with $p_T > 15 \text{ GeV/c}$ in $E_T = 200 \text{ GeV}$ $b$-jets with $0 < |\eta| < 0.7$
The Deterministic Annealing Filter

\[ \Delta z \]

<table>
<thead>
<tr>
<th></th>
<th>RMS</th>
<th>( \chi^2/\text{ndf} )</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAF</td>
<td>0.591E-02</td>
<td>108.8 / 68</td>
<td>0.5867E-04</td>
<td>0.6072E-04</td>
<td>0.3176E-02</td>
<td>16.36</td>
<td>0.1093E-01</td>
</tr>
<tr>
<td>KF</td>
<td>0.646E-02</td>
<td>204.6 / 78</td>
<td>0.5565E-04</td>
<td>0.7247E-04</td>
<td>0.3405E-02</td>
<td>24.05</td>
<td>0.1182E-01</td>
</tr>
</tbody>
</table>

\[ \sigma = 51 \mu m \]

\[ \sigma = 65 \mu m \]

Longitudinal impact parameter resolution of tracks with \( p_T > 15 \text{ GeV/c} \) in \( E_T = 200 \text{ GeV} \) \( b \)-jets with \( 0 < |\eta| < 0.7 \)
Impact parameter resolution versus $\eta$ of tracks with $p_T > 15$ GeV/$c$ in $E_T = 200$ GeV $b$-jets
The Deterministic Annealing Filter

$E_T = 200\text{GeV}$

Impact parameter pulls versus $\eta$ of tracks with $p_T > 15 \text{ GeV}/c$ in $E_T = 200 \text{ GeV} \ b$-jets

R. Frühwirth  
HEPHY Vienna
The Deterministic Annealing Filter

$\chi^2$ probability

![Graph showing $\chi^2$ probability for DAF and KF]

$\chi^2$-probability of tracks with $p_T > 15 \text{ GeV/c}$ in $E_T = 200 \text{ GeV} \ b$-jets with $0 < |\eta| < 0.7$

R. Frühwirth

HEPHY Vienna
The Deterministic Annealing Filter

efficiency in b-jets

Overall track reconstruction efficiency in b-jets
The Deterministic Annealing Filter

Overall track reconstruction fake rate in $b$-jets
Secondary vertex finding efficiency in $b$-jets

R. Frühwirth

HEPHY Vienna
The Deterministic Annealing Filter

**b-tagging efficiency**

- **$E_T = 50\text{GeV}$**
- **$E_T = 100\text{GeV}$**
- **$E_T = 200\text{GeV}$**

*DAF, KF*

**$b$-tagging efficiency in L2 jets**

R. Frühwirth

HEPHY Vienna
The Deterministic Annealing Filter

Mistagging rate in L2 jets
The Multi-track filter

Track fitting in dense jets poses particular problems.

- Merging of hits (clusters)
- Association of hits to tracks uncertain
- Biassed estimation

The Multi-track filter (MTF) is designed to cope with such problems (Strandlie and Frühwirth, 2000).

- Global competition of all hits for all tracks by a modified computation of the weights.
- Several DAFs running in parallel
- Has to be initialized and annealed very carefully
The Multi-track filter

The MTF weights:

- Compute the distance matrix:

\[(\Phi)_{ij} = \varphi_{ij} = \varphi(y_i; Hx_j, \alpha V_i),\]

- Compute the weights, normalizing by all competitors:

\[p_{ij} = \frac{\varphi_{ij}}{c(\alpha) + \sum_k \varphi_{kj} + \sum_l \varphi_{il} - \varphi_{ij}}.\]
The Multi-track filter

First study of MTF in ATLAS Transition Radiation Tracker (Strandlie and Frühwirth, 2000).

- Artificial track pairs
- Mirror hits, additionally up to 50% noise hits

Best result from MTF, first iterations with modified EAA weights (competition between tracks and between mirror hits), followed by iterations with proper MTF weights.
The Multi-track filter

A track pair in the $R\Phi$-projection. Mirror hits, no noise.
The Multi-track filter

A track pair in the $R\Phi$-projection. Mirror hits, no noise.
The Multi-track filter

A track pair in the $R\Phi$-projection. Mirror hits, 50% noise.
The Multi-track filter

A track pair in the $R\Phi$-projection. Mirror hits, 50% noise.
The Multi-track filter

The MTF has been implemented in ORCA (M. Winkler, 2002).

- Based on the building blocks of the DAF in ORCA
- Track finding by a combinatorial Kalman filter
- MTF used for smoothing
- Extensively verified on track pairs
- Physics study with 3-prong $\tau$-decays
The Multi-track filter

Results of a study of 3-prong $\tau$-decays with the KF, the DAF, and the MTF:

- DAF and MTF show significant improvement over KF
- MTF and DAF give the same track parameter resolution
- MTF gives slightly better error matrix
- MTF gives a nearly perfect $\chi^2$-probability
The Multi-track filter

Characteristics of simulated $H^0 \rightarrow \tau^+ \tau^-$ sample
The Multi-track filter

Efficiency of reconstructing three tracks in 3-prong $\tau$-decays
The Multi-track filter

Δx

Transverse impact parameter resolution at the τ vertex

R. Frühwirth

HEPHY Vienna
The Multi-track filter

\[ \Delta z \]

KF

DAF

KF+MTF

DAF+MTF

Longitudinal impact parameter resolution at the \( \tau \) vertex

R. Frühwirth

HEPHY Vienna
The Multi-track filter

Transverse momentum resolution at the $\tau$ vertex

R. Frühwirth

HEPHY Vienna
The Multi-track filter

$\chi^2$ probability

KF

DAF

KF+MTF

DAF+MTF

$\chi^2$-probability distributions

R. Frühwirth

HEPHY Vienna
Vertex Fitting

Vertex reconstruction typically involves the following steps:

- **Vertex finding**
  
  Use magnetic field and detector geometry; produce vertex candidates

- **Vertex fitting**
  
  Use magnetic field and material; produce vertex parameters, errors and test statistics ($\chi^2$)

- **Test of vertex hypothesis and final assignment**
  
  Use information from all vertices
Vertices in CMS, especially secondary ones, can be contaminated by extraneous tracks (outliers). Estimation should be robust (not influenced by outliers).

Two classes of estimators:

- Modified Kalman filter
  Least trimmed sum of squares, adaptive filter

- High breakdown-point estimators
  Least median of squares, Half sample mode
Vertex can be interpreted as \textit{dynamic system}: add a track to an existing vertex.

- System equation (linear):

\[ v_k = v_{k-1}, \quad k = 1, \ldots, m \]

\[ v_k: \text{vertex with } k \text{ tracks, no "process noise"!} \]
Linear vertex fit

- Measurement equation (non-linear):

\[ m_k = h_k(v, q_k) + \epsilon_k, \quad k = 1, \ldots, n \]

- \( m_k \): estimated parameters (5-dim) of track \( k \)
- \( h_k \): track model (helix)
- \( q_k \): momentum at the vertex \( v \) (3-dim)
- \( \epsilon_k \): covariance matrix of \( m_k \)

- The state vector of the system is augmented by \( q_k \) for track \( k \).
Linear vertex fit

The best linear estimator is again the Kalman filter.

- Iterative, fast
- Can be complemented by smoother
- Can be used to find outliers
- Easy to generalize
- Not robust
Linear vertex fit

Iteration of two steps:

- **Prediction:** Extrapolation of last estimate + cov. matrix
  
  Trivial because of very simple system equation

- **Update:** Weighted mean of prediction and estimated track
  
  Requires linear expansion of the measurement equation

Good initialization important.
Linear vertex fit

Three test statistics:

- $\chi^2$-increment of the filter
  tests the compatibility of the track with the vertex; little power in the beginning

- total $\chi^2$ of the filter
  global test of the vertex hypothesis

- $\chi^2$-increment of the smoother
  search for outliers; full power only for single outliers
Robust single-vertex fit

Adaptive vertex fit, automatic suppression of outliers (Frühwirth and Strandlie, 1999).

Characteristics:

- Iterated Kalman filter (EM algorithm)
- Equivalent to the minimization of a (complicated) ANN energy function
- Can be combined with annealing
- Covariance matrix available
Robust single-vertex fit

Iteration of two steps:

- **Kalman filter + smoother**
  - Optimal estimation of position $v$, tracks are downweighted by their association probabilities

- **Computation of the weights**
  - Compute the association probabilities for all tracks with respect to the current vertex position

The iteration is stopped as soon as the weights have stabilized.
Robust single-vertex fit

- **There is no competition between the tracks**
  Several tracks may (and do) belong to the same vertex.

- **Computation of the weights:**

  \[ \chi^2_k = (m_k - H v)^T (\alpha V_k)^{-1} (m_k - H v) \]

  \[ p_k = \frac{\exp(-\chi^2_k / 2)}{\exp(-\chi^2_{\text{cut}} / 2) + \exp(-\chi^2_k / 2)} \]

  \( \chi^2_k \): distance between track \( m_k \) and vertex \( v \)

  \( \chi^2_{\text{cut}} \): distance cut

  \( \alpha \): Optional annealing factor
Robust single-vertex fit

Deterministic annealing helps to reach the global optimum:

- \( \alpha > 1 \) at the start
- Decrease \( \alpha \) after each iteration until 1 is reached.

Well-known technique of global optimization, e.g. with ANN. If \( \alpha > 0 \) the association is “soft”. Cooling down to \( \alpha = 0 \) yields “hard” association not optimal!.

In all cases good initial position required \( \rightarrow \) need robust estimators with high breakdown point.
Robust single-vertex fit

Non-adaptive estimators:

- **Least trimmed sum of squares (LTS)**
  Minimizes the sum of the $h$ smallest squared residuals \([n/2] + 1 < h < n\). Breakdown point approximately \(1 - h/n\).

- **Minimum Covariance Determinant (MCD)**
  Mean of the $h$ observations \([n/2] + 1 < h < n\) minimizing the sample covariance matrix. Computation similar to the LTS estimator.
  Breakdown point approximately \(1 - h/n\).
Robust single-vertex fit

Robust one-dimensional estimators can be employed coordinate-wise:

- **Least median of squares (LMS)**
  Minimizes the median of the squared residuals. Middle point of the shortest interval covering 50% of the observations.

- **Shorth**
  Closely related to LMS. Mean of the data in the shortest interval covering 50% of the observations.

- **Half-sample mode (HSM)**
  Iterated LMS.
Robust single-vertex fit

- Several fitters have been implemented in ORCA (W. Waltenberger)
  - Adaptive, LTS, MCD, LMS, HSM
- Evaluation in progress
- Only preliminary results available
- Adaptive fitter performs very well, in particular with annealing
Robust single-vertex fit

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>t</th>
<th>Res(x)</th>
<th>Pulls(x)</th>
<th>Res(z)</th>
<th>Pulls(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearFitter</td>
<td>18 msec</td>
<td>39µ</td>
<td>2.1</td>
<td>39µ</td>
<td>1.9</td>
</tr>
<tr>
<td>LTSFitter (20% trim)</td>
<td>67 msec</td>
<td>25µ</td>
<td>1.1</td>
<td>29µ</td>
<td>1.1</td>
</tr>
<tr>
<td>MCDFitter (20% trim)</td>
<td>67 msec</td>
<td>21µ</td>
<td>1.2</td>
<td>27µ</td>
<td>1.2</td>
</tr>
<tr>
<td>PCLMSFitter (err.mat)</td>
<td>12 msec</td>
<td>74µ</td>
<td>0.66</td>
<td>90µ</td>
<td>0.64</td>
</tr>
<tr>
<td>AdaptiveFitter</td>
<td>44 msec</td>
<td>21µ</td>
<td>1.1</td>
<td>28µ</td>
<td>1.1</td>
</tr>
<tr>
<td>Adaptive + anneal</td>
<td>24 msec</td>
<td>22µ</td>
<td>1</td>
<td>28µ</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Precision and timing of $q\bar{q}$ vertices

R. Frühwirth

HEPHY Vienna
### Robust single-vertex fit

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>t</th>
<th>Res(x)</th>
<th>Pulls(x)</th>
<th>Res(z)</th>
<th>Pulls(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinearFitter</td>
<td>12 msec</td>
<td>58µ</td>
<td>3.5</td>
<td>42µ</td>
<td>1.9</td>
</tr>
<tr>
<td>LTSFitter (20% trim)</td>
<td>45 msec</td>
<td>27µ</td>
<td>1.5</td>
<td>30µ</td>
<td>1.3</td>
</tr>
<tr>
<td>MCDFitter (20% trim)</td>
<td>45 msec</td>
<td>36µ</td>
<td>1.8</td>
<td>32µ</td>
<td>1.6</td>
</tr>
<tr>
<td>PCLMSFitter (err.mat)</td>
<td>8 msec</td>
<td>78µ</td>
<td>0.4</td>
<td>113µ</td>
<td>0.54</td>
</tr>
<tr>
<td>AdaptiveFitter</td>
<td>30 msec</td>
<td>26µ</td>
<td>1.2</td>
<td>26µ</td>
<td>1.3</td>
</tr>
<tr>
<td>Adaptive + anneal</td>
<td>18 msec</td>
<td>26µ</td>
<td>1.2</td>
<td>31µ</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Precision and timing of $c\bar{c}$ vertices

R. Frühwirth  
HEPHY Vienna
Robust single-vertex fit
Robust single-vertex fit

Adaptive Fitter with annealing

R. Frühwirth

HEPHY Vienna
Adaptive Multi-vertex fit

- Concurrent fit of several vertices with competition
  A track can belong to at most one vertex.

- Computation of the weights:
  \[
  \chi^2_{ki} = (m_k - Hv_i)^T (\alpha V_k)^{-1} (m_k - Hv_i)
  \]
  \[
  p_{ki} = \frac{\exp(-\chi^2_{ki}/2)}{\exp(-\chi^2_{cut}/2) + \sum_i \exp(-\chi^2_{ki}/2)}
  \]

  \(\chi^2_{ki}\): distance between track \(m_k\) and vertex \(v\).

- Implementation in CMS planned
Adaptive Multi-vertex fit

- Can be implemented by iterated, parallel Kalman filters
- Equivalent to “Elastic Arm” approach, but faster and safer
- Allows dynamic swapping of tracks between vertices
- Annealing can be used to help reaching the global optimum
- Will be implemented and evaluated in CMS
Summary and outlook

I have presented several extensions of the Kalman filter:

❖ The Gaussian-sum filter
   Parallel KF. Useful for dealing with non-Gaussian noise, if the appropriate mixture model is available. Can be used to solve ambiguities. Sensitive to outliers.

❖ The Deterministic annealing filter
   Adaptive, robustified, iterative Kalman filter. Outliers are downweighted, depending on the presence of other observations.

❖ The Multi-track filter
   Parallel DAF with global competition. All hits compete for inclusion in all tracks.
Summary and outlook

❖ All of those filters are definitely slower than the KF
❖ Important to use them only if profitable
❖ Difficult identification task, more research needed
❖ Study application of GSF to multiple scattering tails
❖ Study application of GSF to electron tracks in the CMS tracker (A. Strandlie, W. Adam)
❖ Study feasibility of combining DAF and GSF
❖ Application of GSF and DAF to single and multi-vertex fitting under study (W. Waltenberger)
Summary and outlook

Special thanks to

Are Strandlie

Matthias Winkler

Wolfgang Waltenberger
Summary and outlook

Many thanks to

Rainer Mankel

for inviting me to this seminar
References