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

Track reconstruction 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 (χ^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.

Track Fitting

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**

  χ^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

The Kalman filter

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

Three test statistics:

- **Incremental χ^2 of the filter**

  Can be used for discrimination between competing hits, little power in the starting phase

- **Total χ^2 of the filter**

  Can be used to test track hypothesis

- **Incremental χ^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

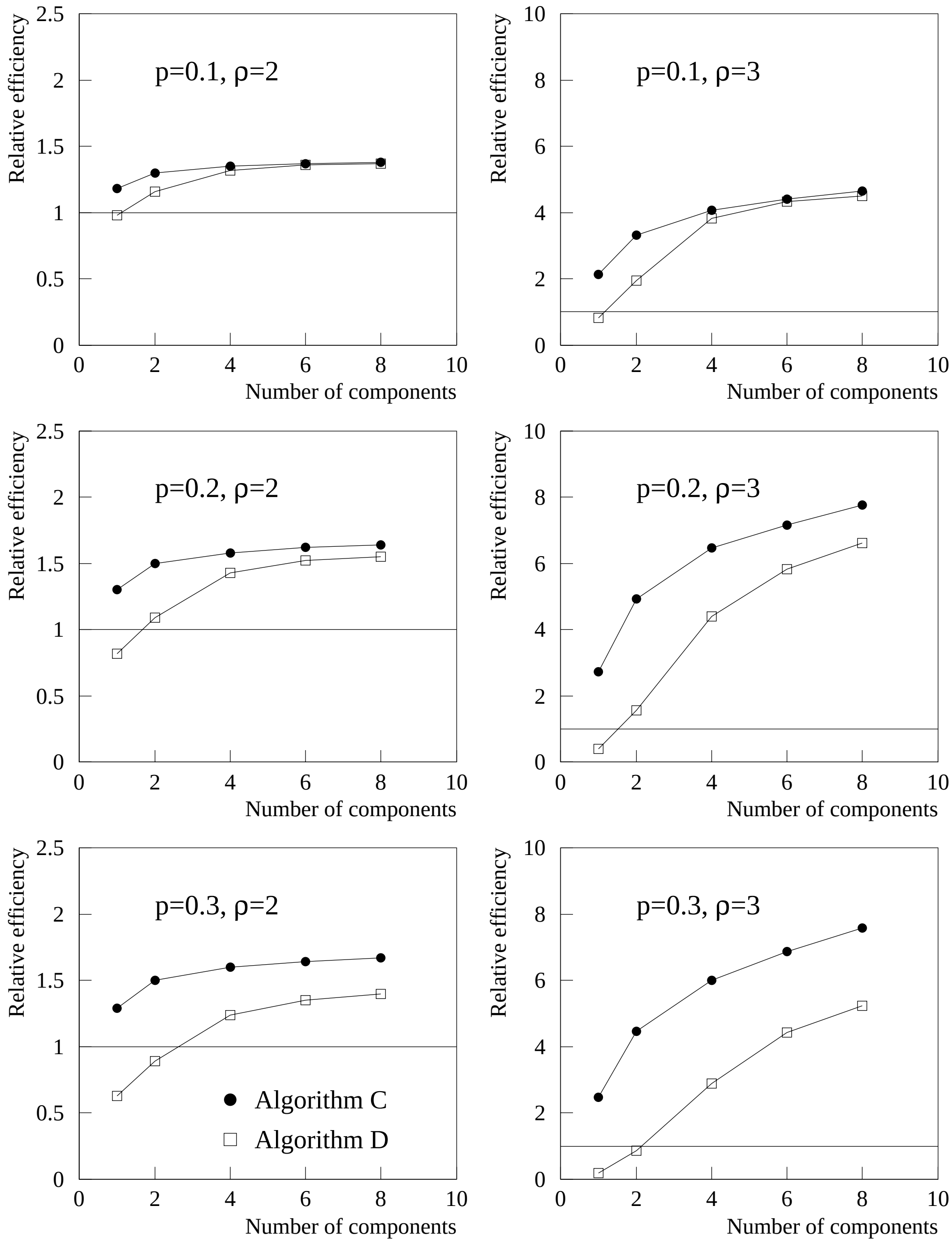
$$f(\boldsymbol{x}) = \sum_{i=1}^{n} p_i\, \varphi(\boldsymbol{x}; \boldsymbol{\mu}_i, \boldsymbol{V}_i), \quad \sum_{i=1}^{n} p_i = 1$$

- 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

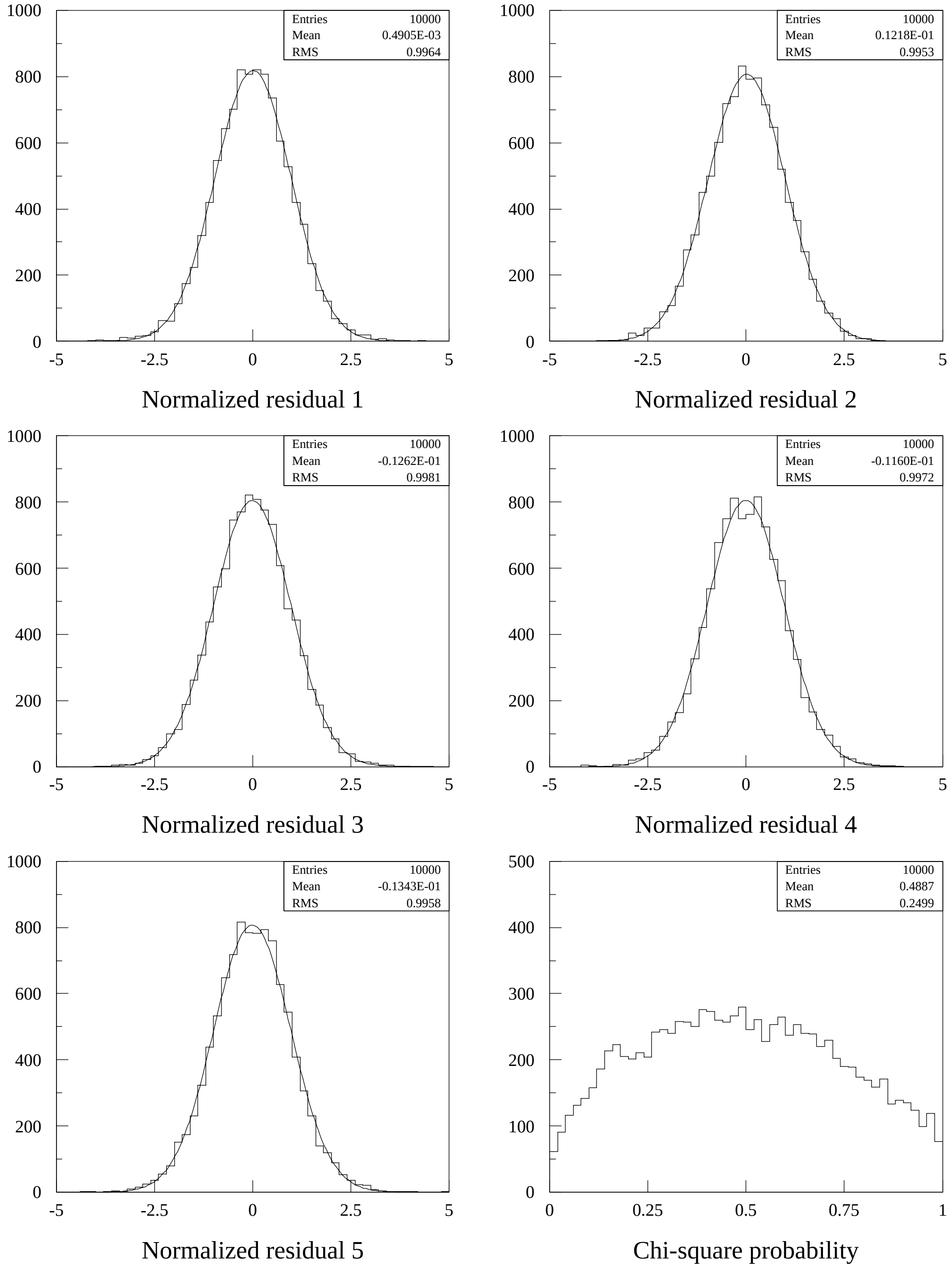
The Gaussian-sum filter

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 χ^2-distribution
- Execution time proportional to number of components
- Mixture model of measurement errors must be available



Relative precision of the GSF



Normalized residuals and χ^2-probability of the GSF

The Gaussian-sum filter

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

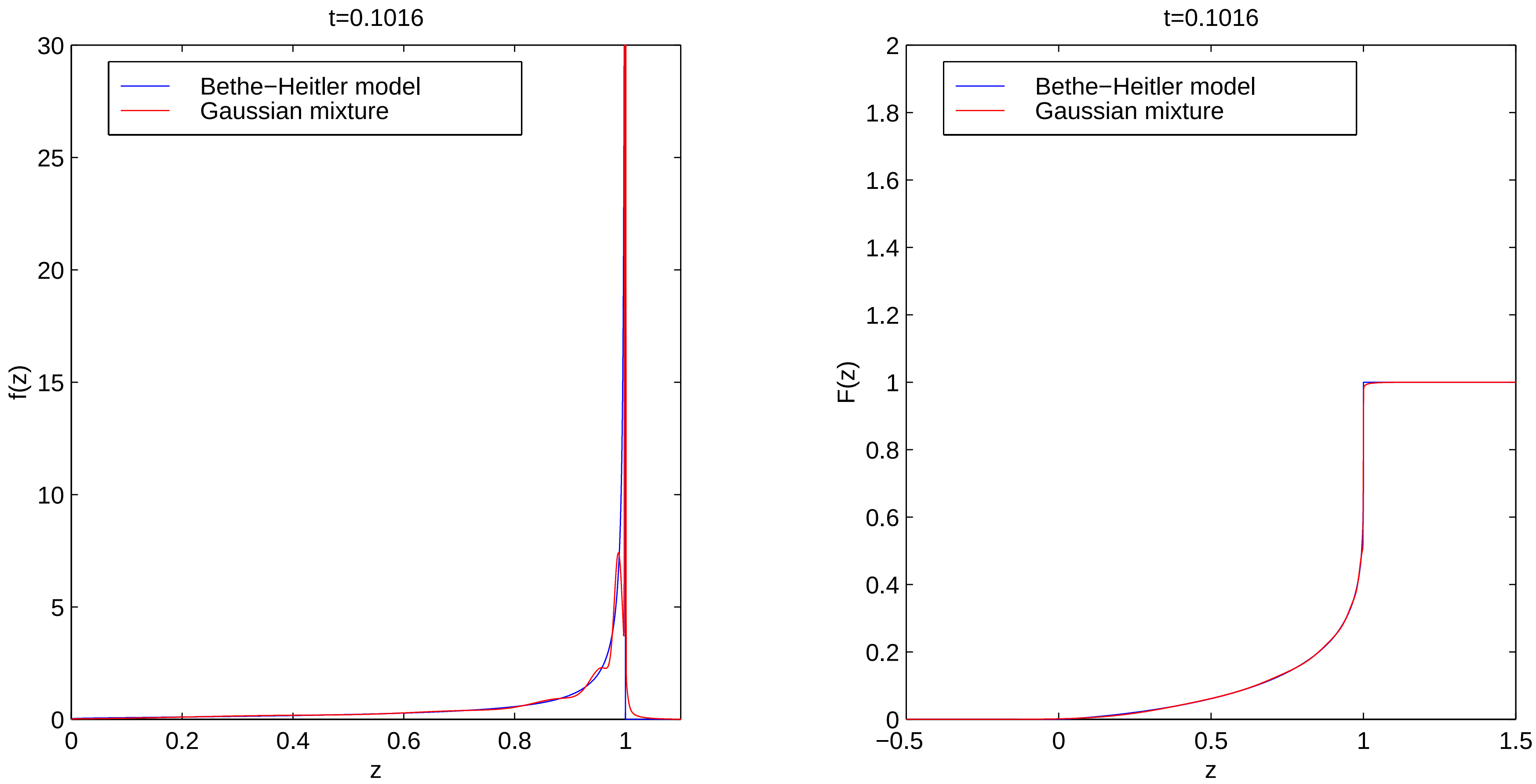
- Energy loss distribution approximated by mixture of 6 Gaussians
- Gaussian-sum filter compared to Kalman filter and Metropolis-Hastings algorithm
- Appreciable gain in precision of the momentum measurement, especially for thin layers

The Gaussian-sum filter

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 $z = p_1/p_0$

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:

$$p_i = \frac{\varphi(\boldsymbol{m}_i; \boldsymbol{H}\boldsymbol{x}, \alpha \boldsymbol{V}_i)}{c(\alpha) + \sum_j \varphi(\boldsymbol{m}_j; \boldsymbol{H}\boldsymbol{x}, \alpha \boldsymbol{V}_j)}$$

- φ measures the distance of the measurement i from the track. Usually, but not necessarily, Gaussian. α 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!

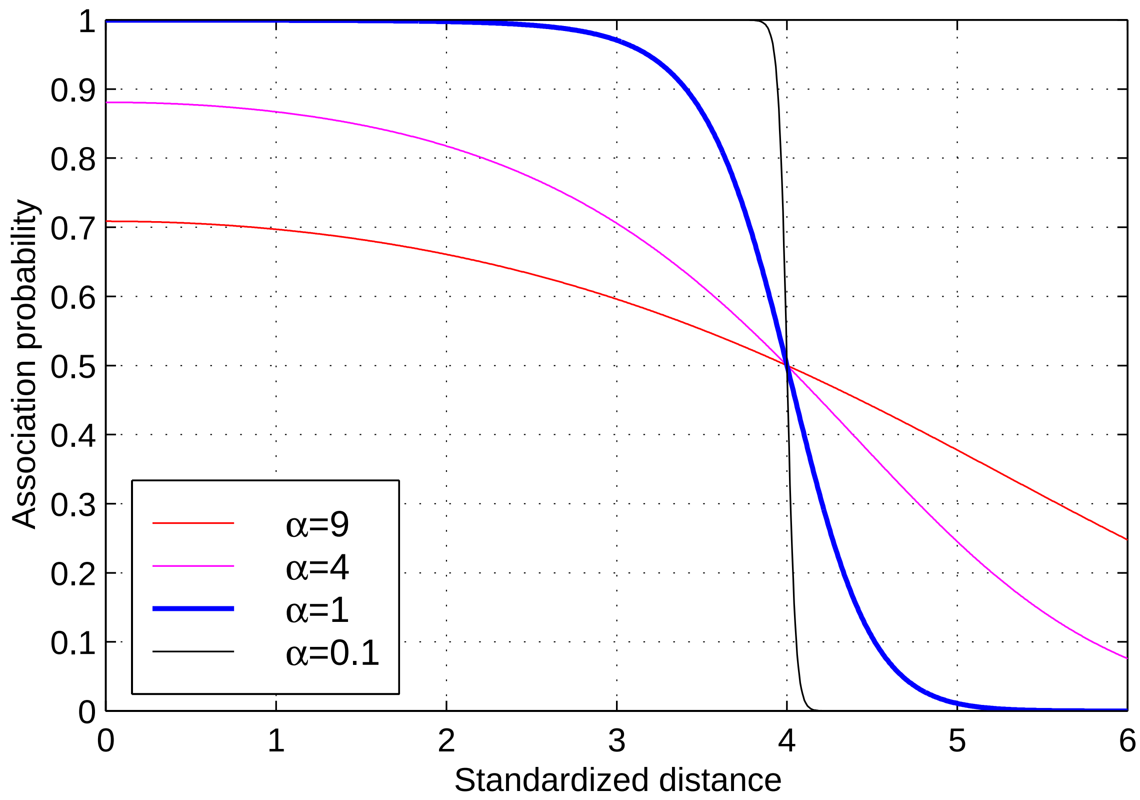
The Deterministic Annealing Filter

Deterministic Annealing helps to reach the global optimum:

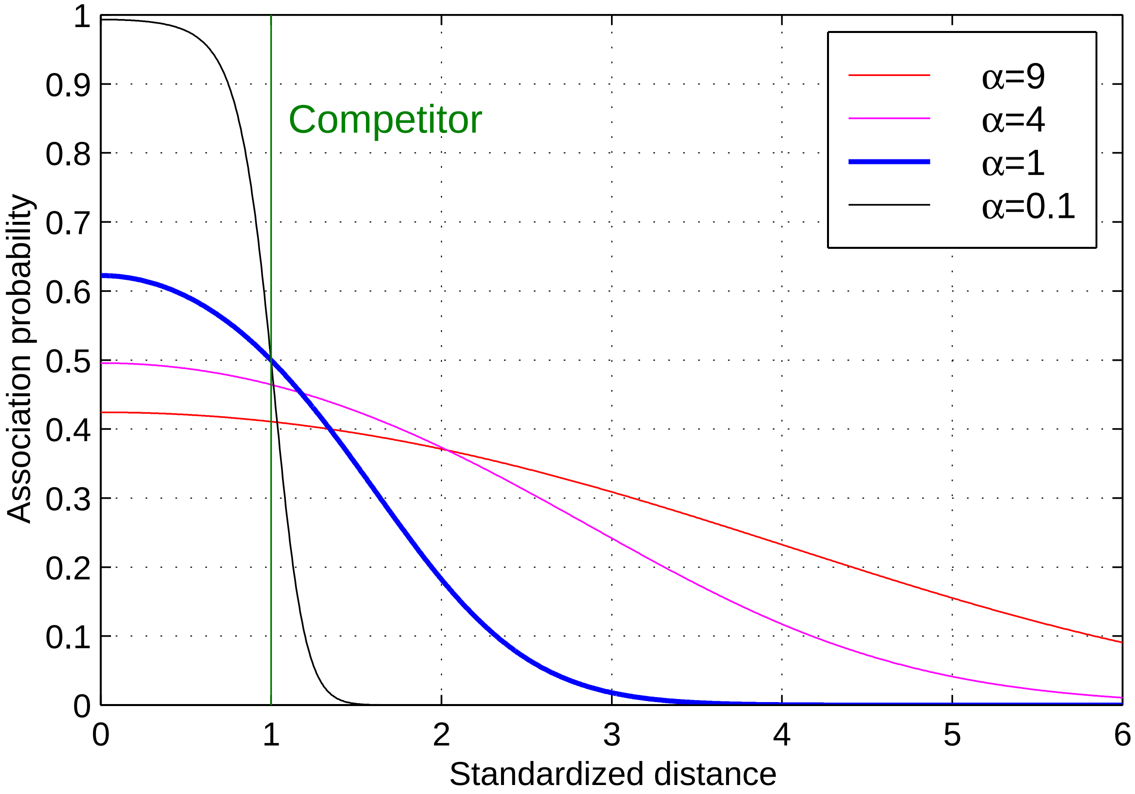
- $\alpha > 1$ at the start
- Stepped down to 1 in the course of the iterations

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



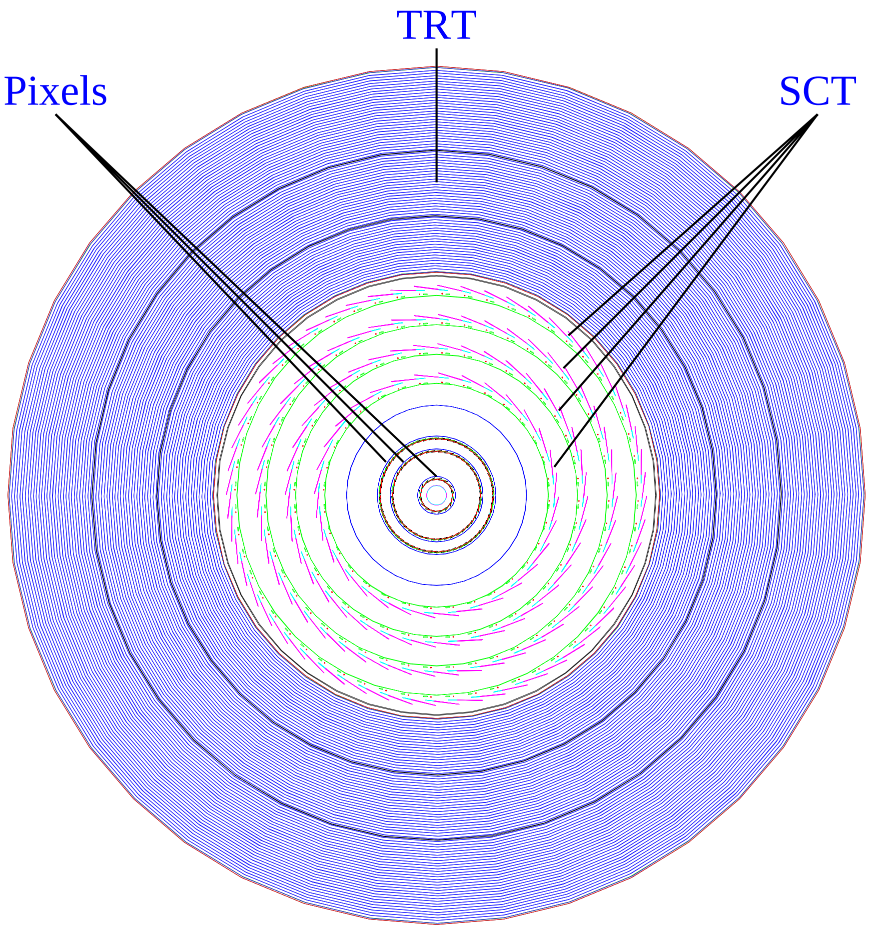
Weight function of an observation with competition

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!

Method	V_{rel}	t_{rel}
DAF	1.00	1.00
GSF	1.00	0.44
EAA with GD	8.83	1.49
EAA with DFP	1.03	1.63
ETA-G with GD	1.79	0.73
ETA-G with DFP	1.05	1.41
ETA-L with GD	125.6	0.65
ETA-L with DFP	1.07	1.67
KF	1.00	0.07

The relative generalized variance, tracks w/o mirror hits

Method	V_{rel}	t_{rel}
DAF nominal	1.54	1.21
DAF frozen	1.74	1.41
GSF all	1.59	7.04
GSF best	1.78	7.04
EAA nominal	1.56	2.12
EAA frozen	1.71	2.44
ETR	3.51	2.87
KF	~ 1500	0.08

The relative generalized variance, tracks with mirror hits

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

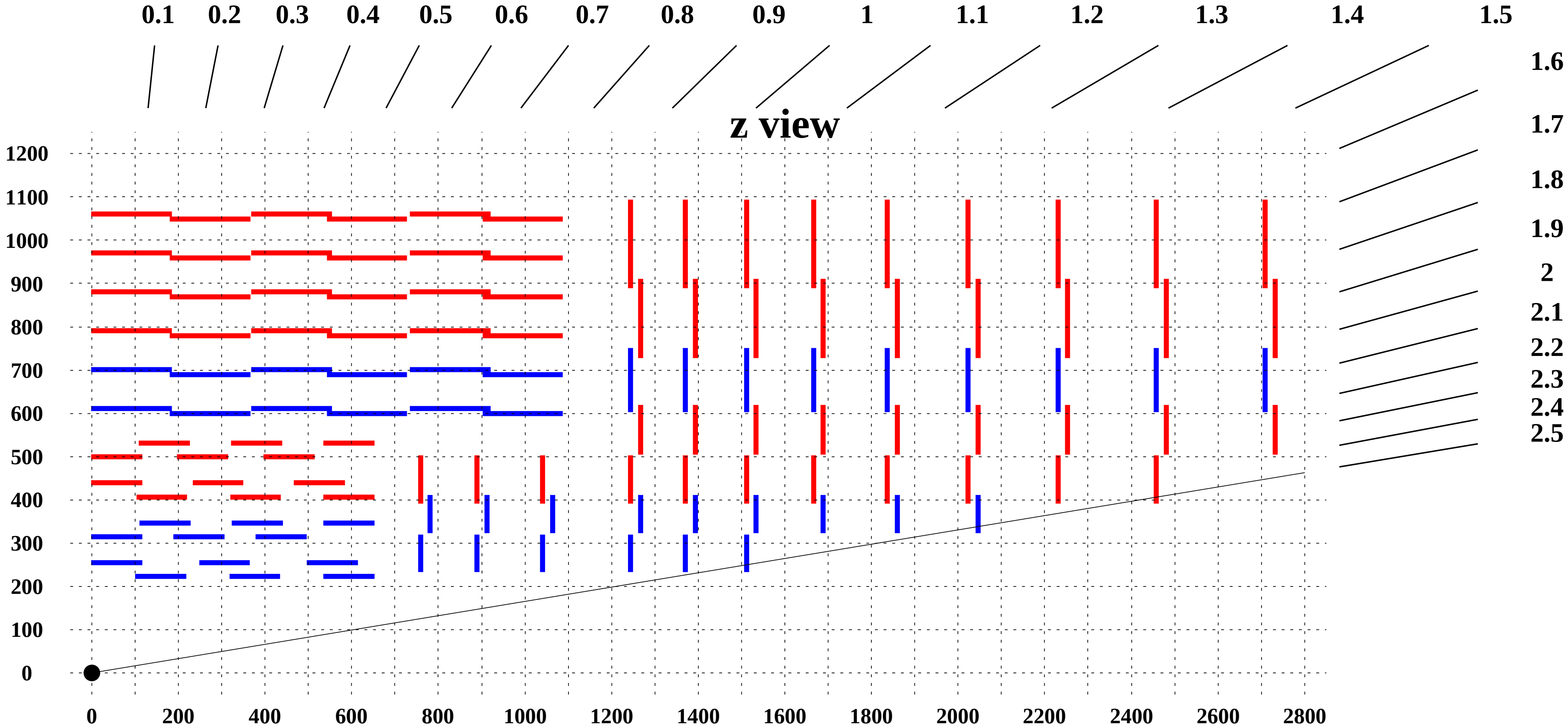
The Deterministic Annealing Filter

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.

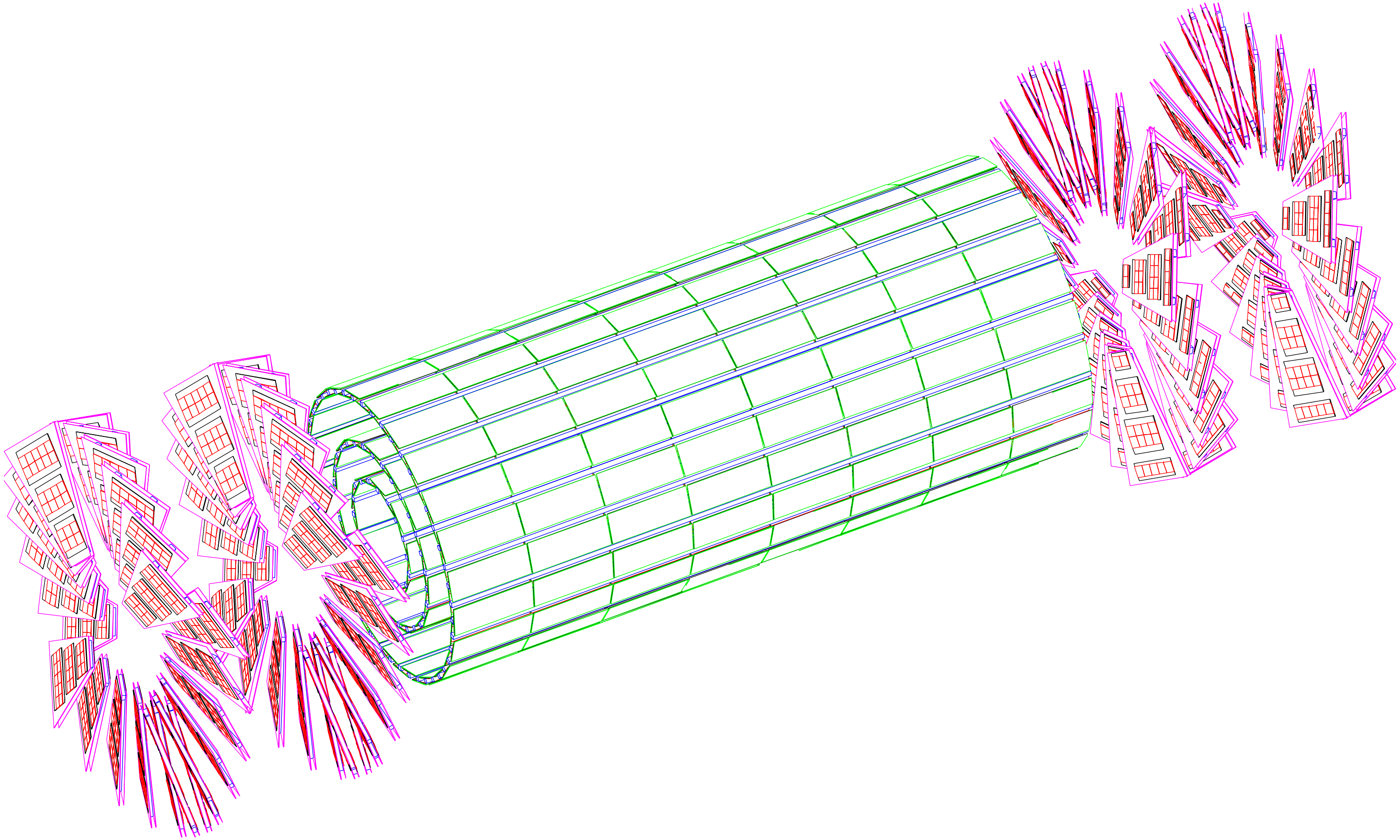
The Deterministic Annealing Filter



Silicon strip detector part of the CMS tracker

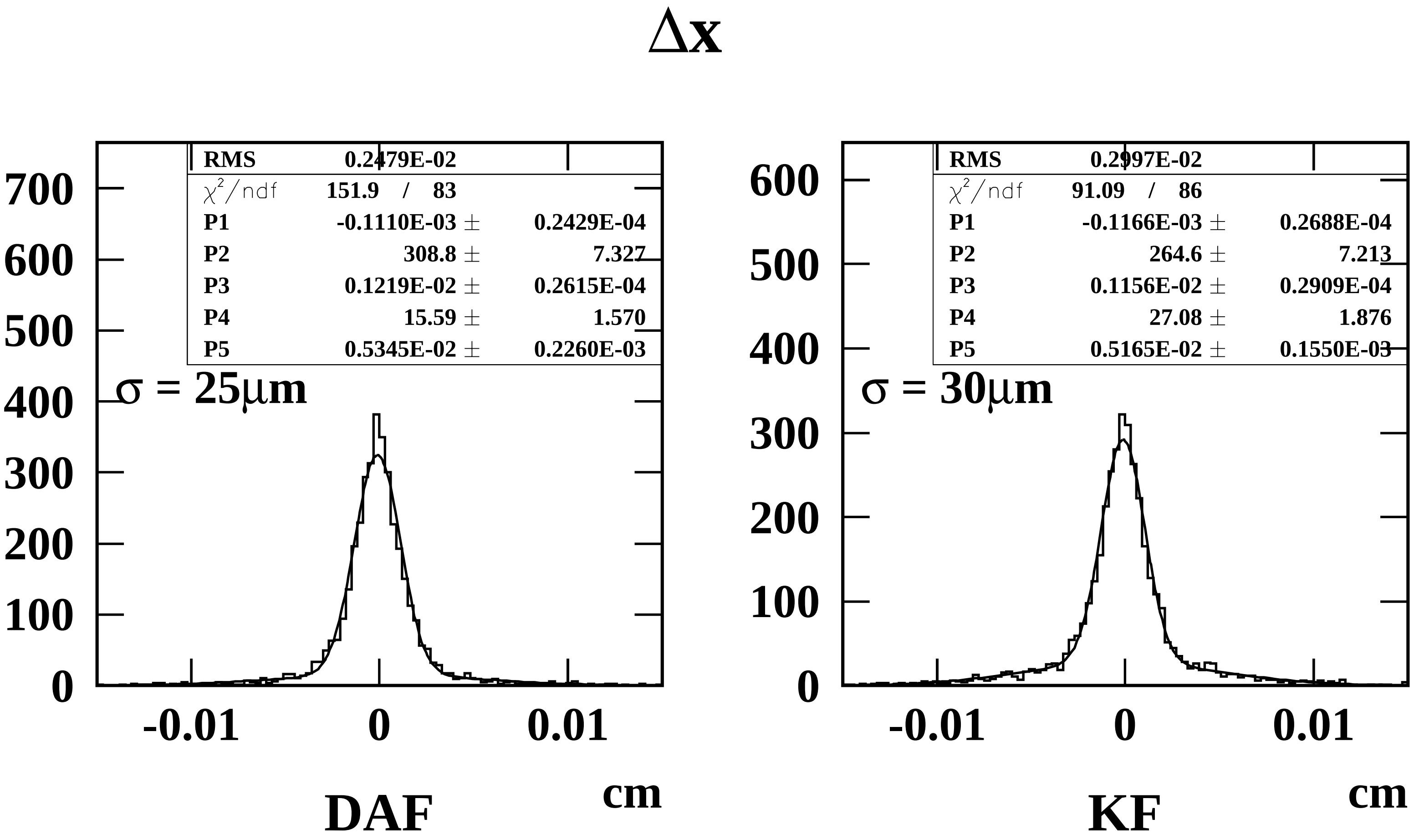
Red: single-sided, blue: double-sided

The Deterministic Annealing Filter



Pixel detector part of the CMS tracker

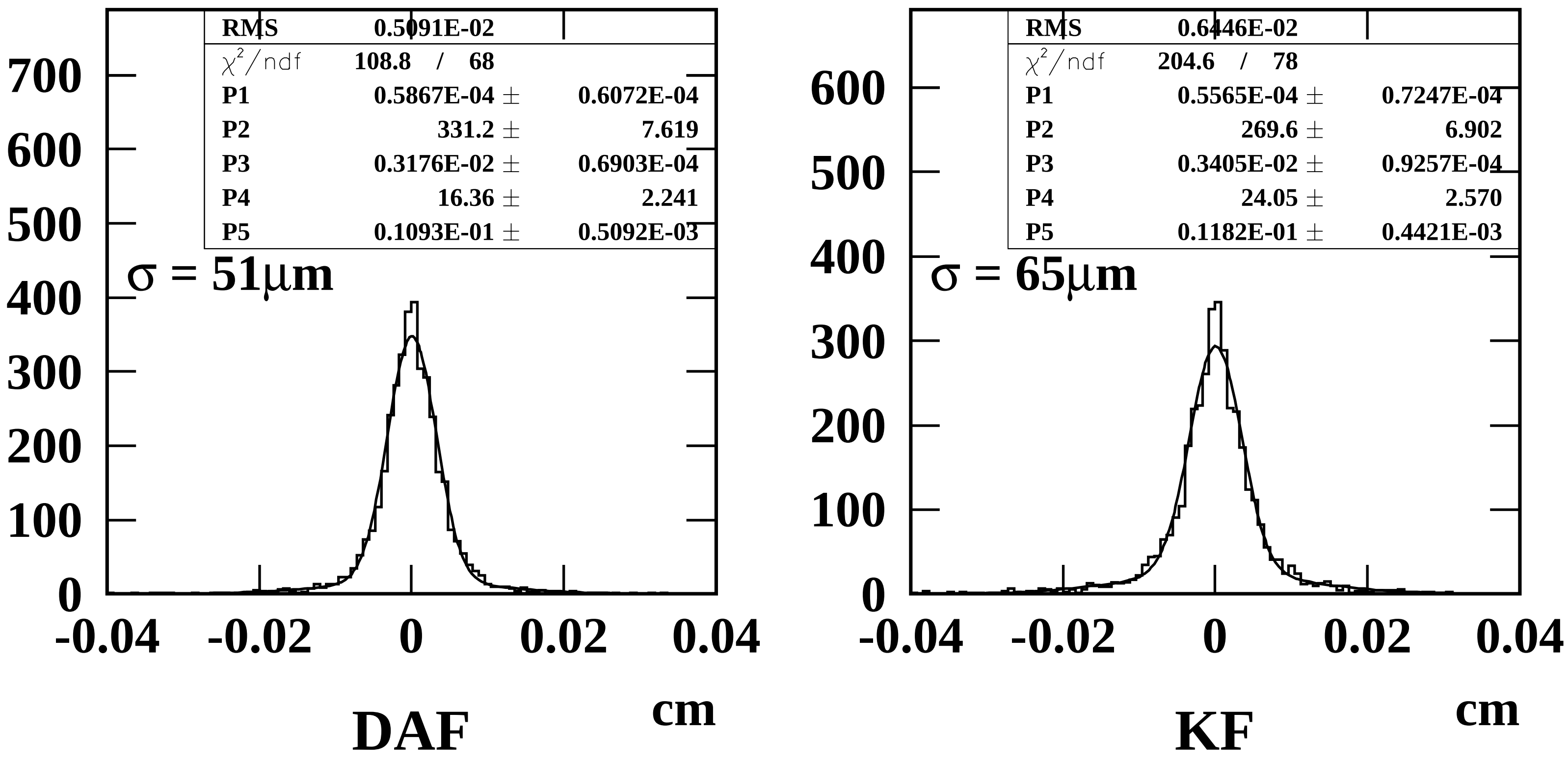
The Deterministic Annealing Filter



Transverse impact parameter resolution of tracks with $p_T > 15$ GeV/c in $E_T = 200$ GeV b-jets with $0 < |\eta| < 0.7$

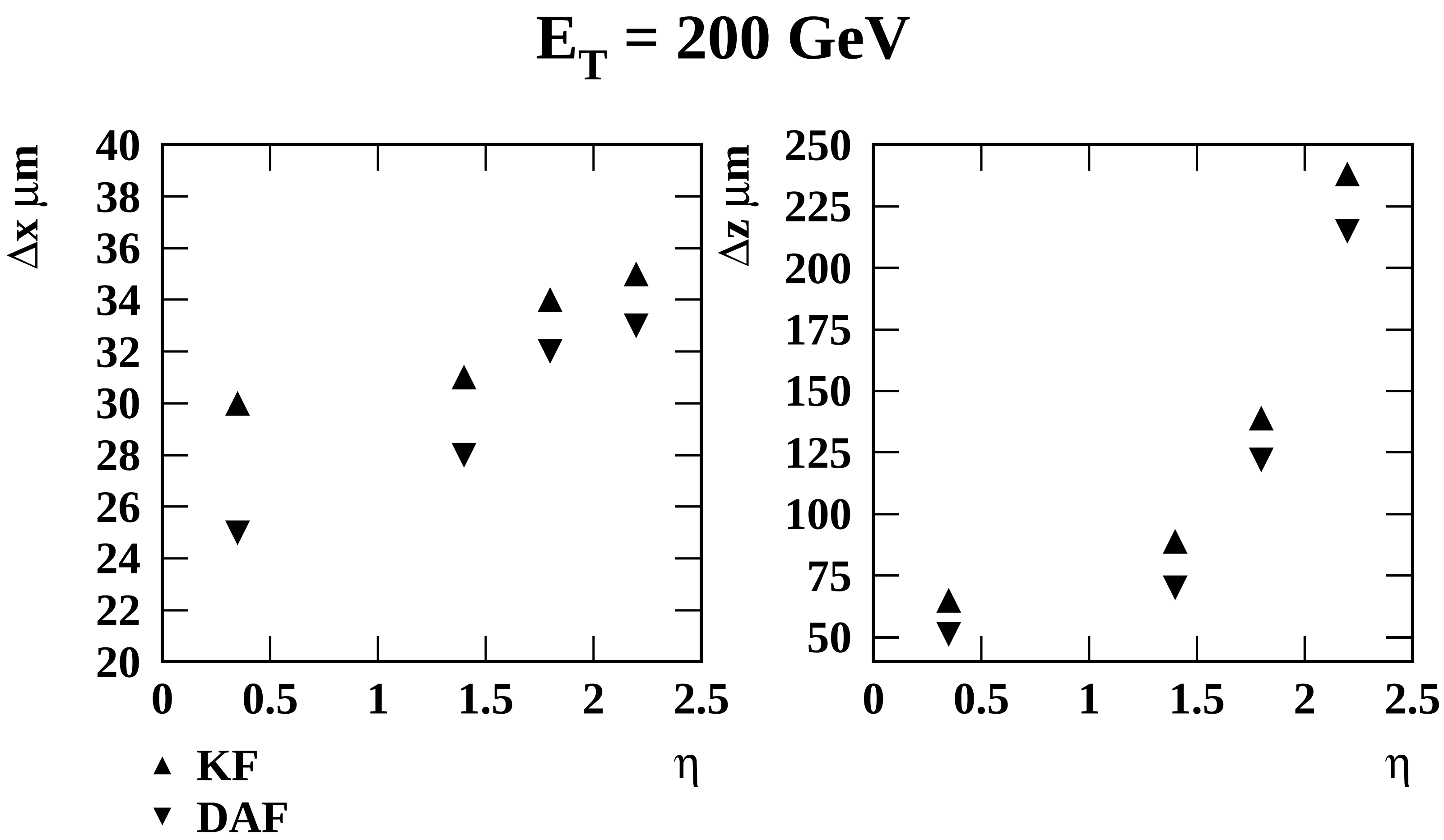
The Deterministic Annealing Filter

Δz

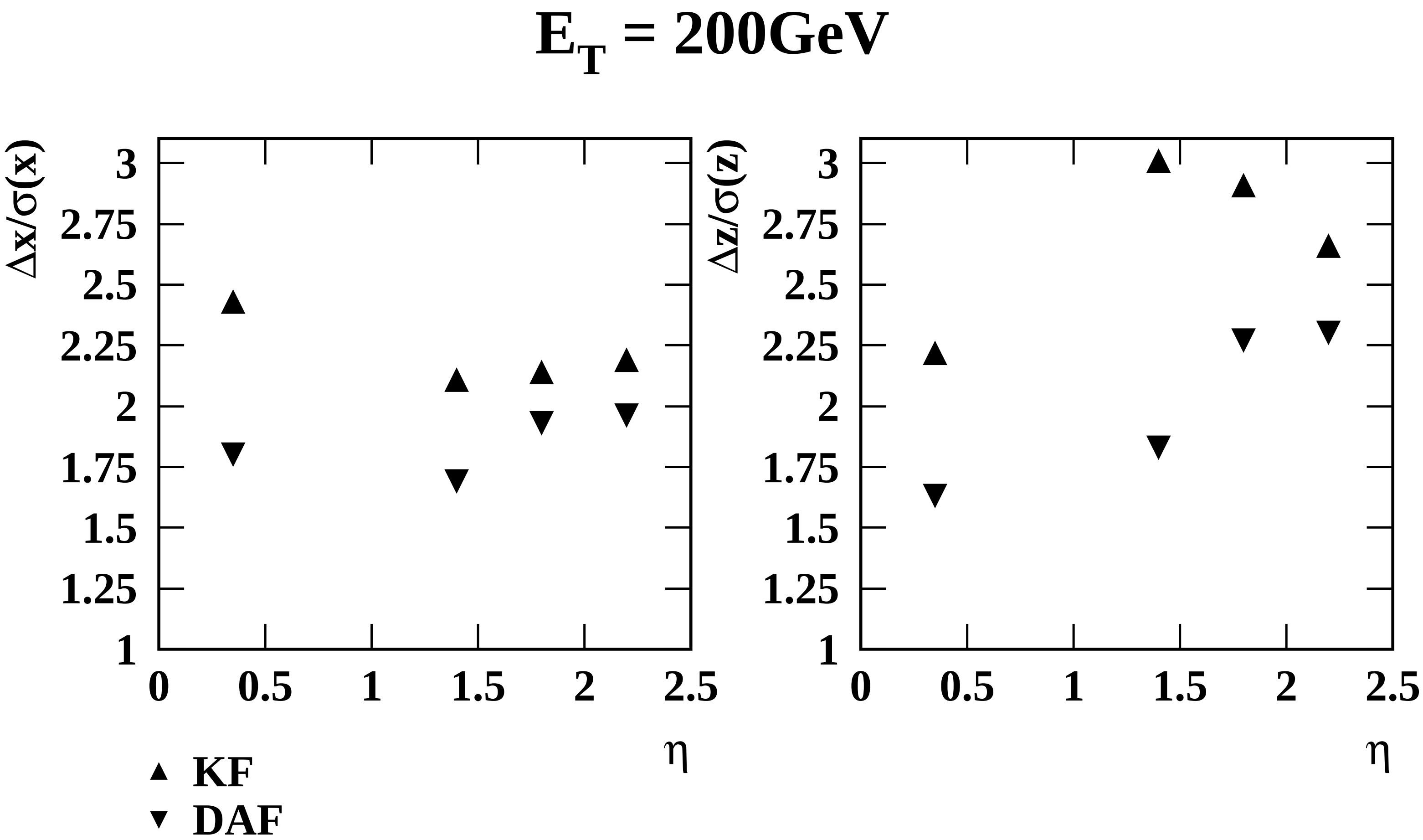


Longitudinal impact parameter resolution of tracks with $p_T > 15$ GeV/c in $E_T = 200$ GeV b-jets with $0 < |\eta| < 0.7$

The Deterministic Annealing Filter

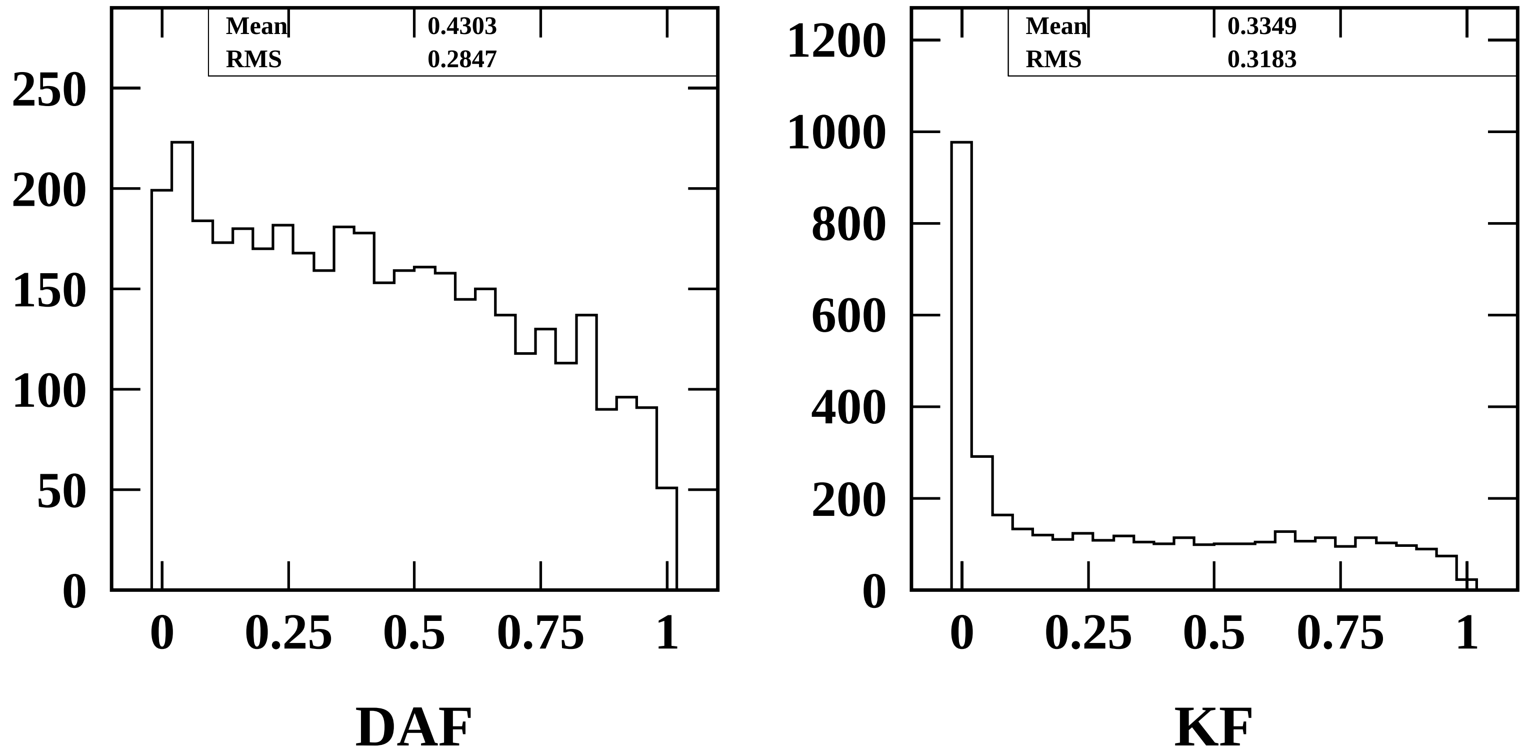


Impact parameter resolution versus η of tracks with $p_T > 15$ GeV/c in $E_T = 200$ GeV b-jets

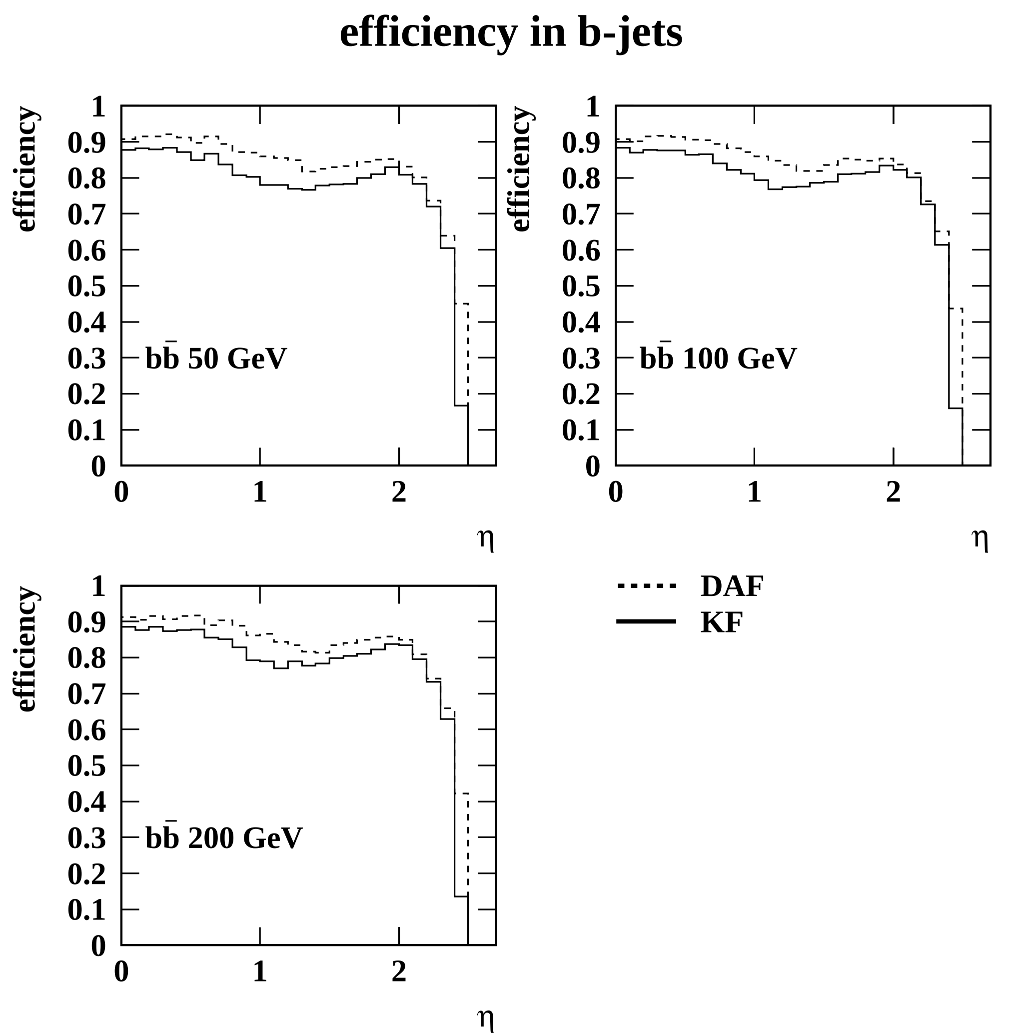


Impact parameter pulls versus η of tracks with $p_T > 15$ GeV/c in $E_T = 200$ GeV b-jets

χ^2 probability

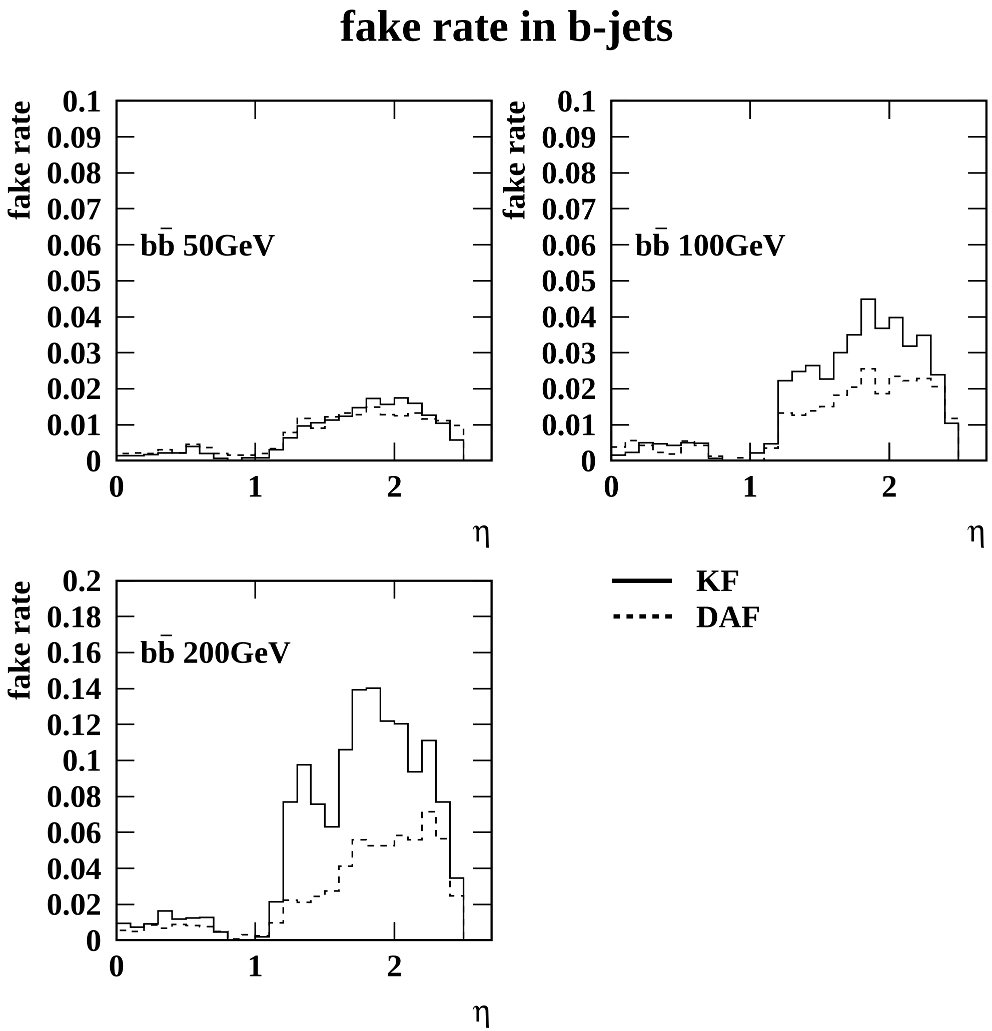


χ^2-probability of tracks with $p_T > 15$ GeV/c in $E_T = 200$ GeV b-jets with $0 < |\eta| < 0.7$



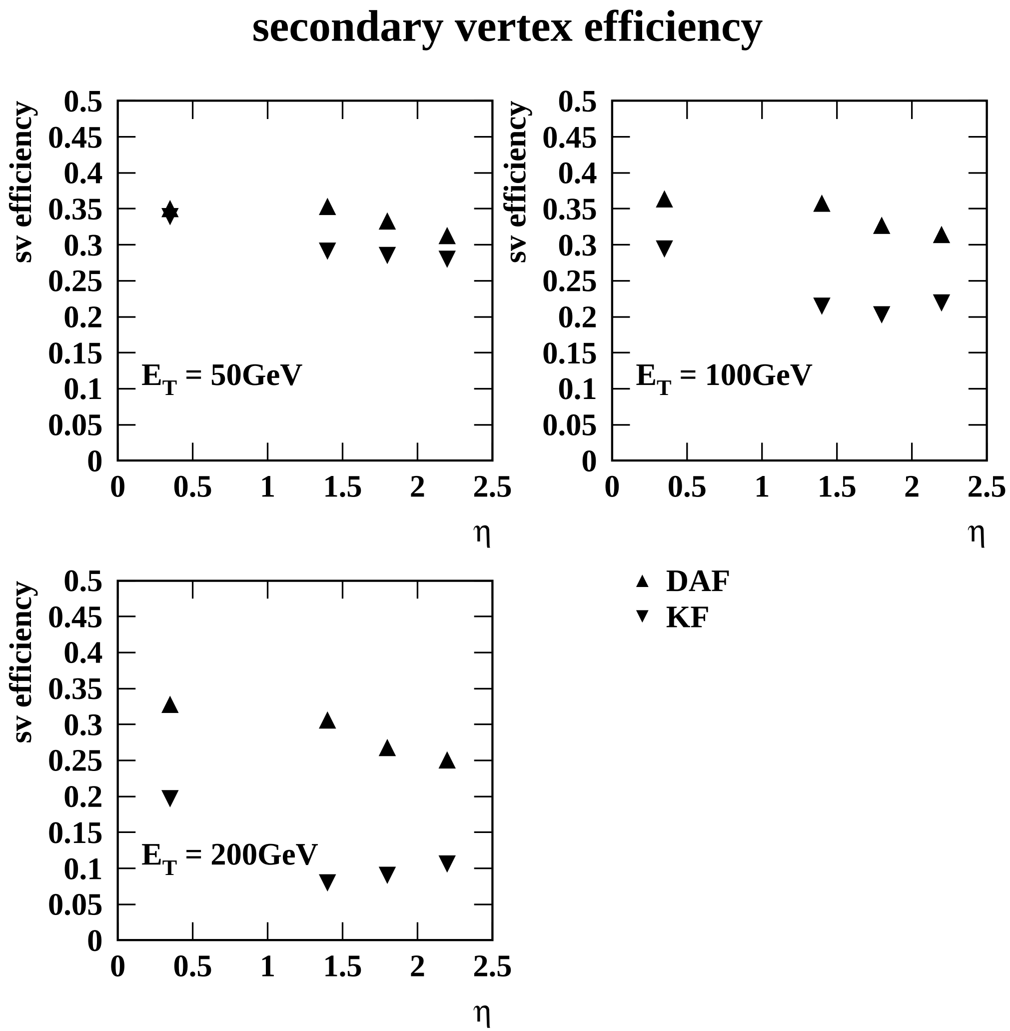
Overall track reconstruction efficiency in *b*-jets

The Deterministic Annealing Filter

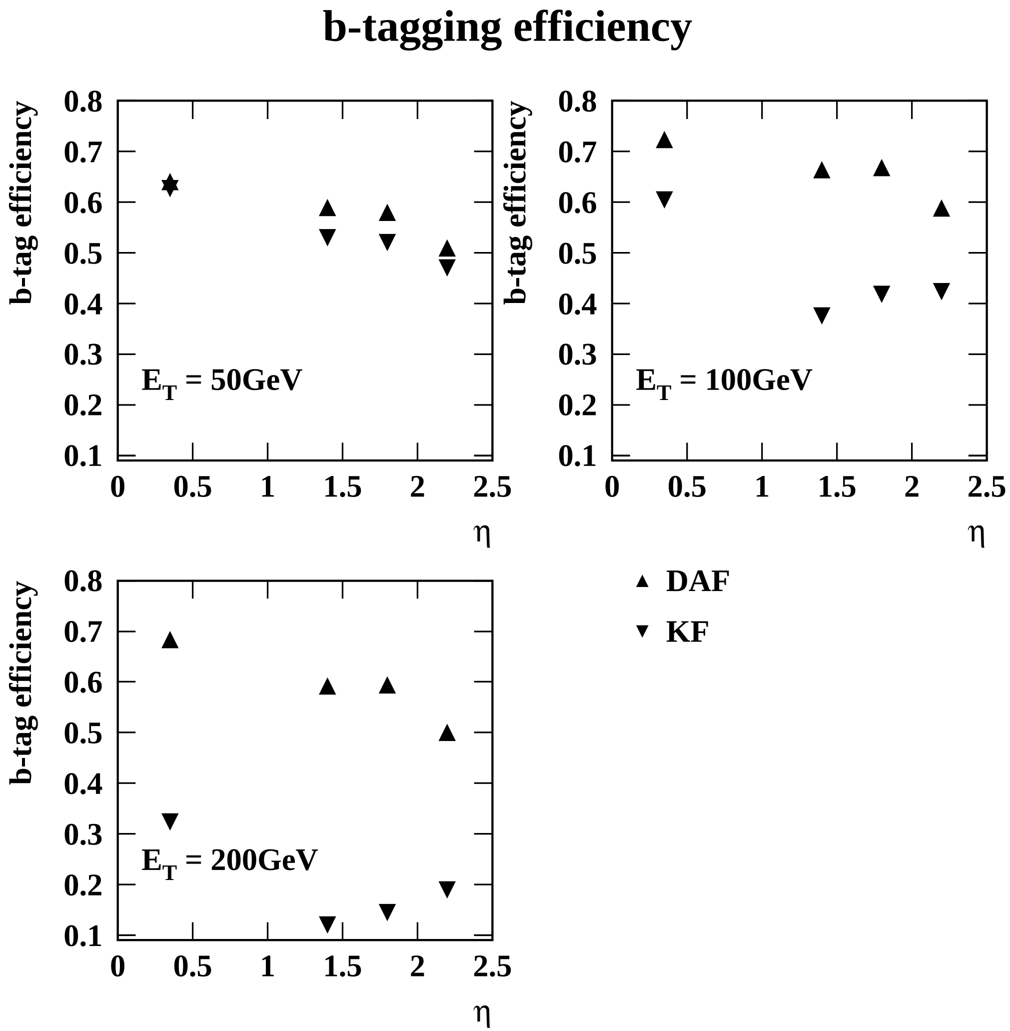


Overall track reconstruction fake rate in *b*-jets

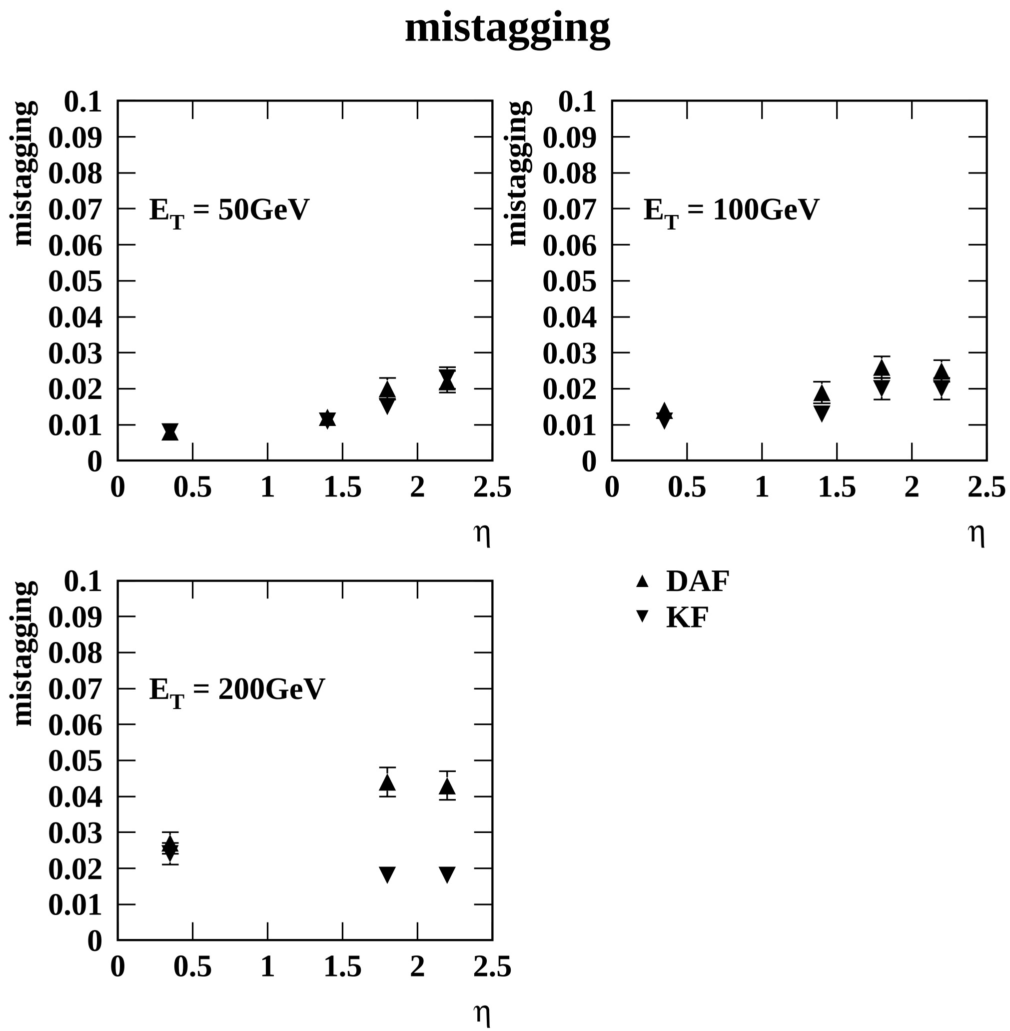
The Deterministic Annealing Filter



Secondary vertex finding efficiency in b-jets



b-tagging efficiency in L2 jets



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:**

$$(\Phi)_{ij} = \varphi_{ij} = \varphi(\boldsymbol{y}_i; \boldsymbol{H}\boldsymbol{x}_j, \alpha\boldsymbol{V}_i),$$

- **Compute the weights, normalizing by all competitors:**

$$p_{ij} = \frac{\varphi_{ij}}{c(\alpha) + \sum_k \varphi_{kj} + \sum_l \varphi_{il} - \varphi_{ij}}.$$

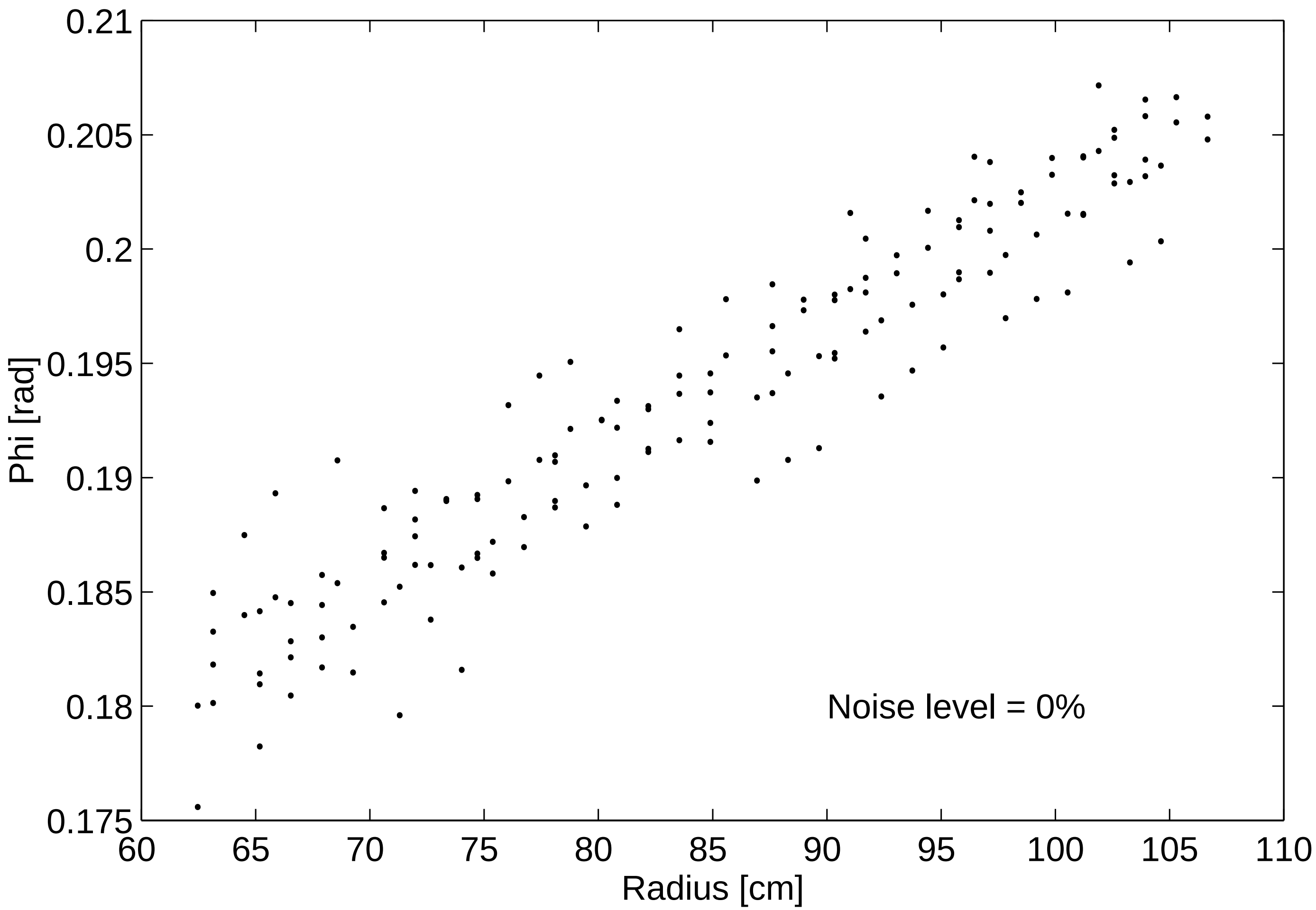
The Multi-track filter

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

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

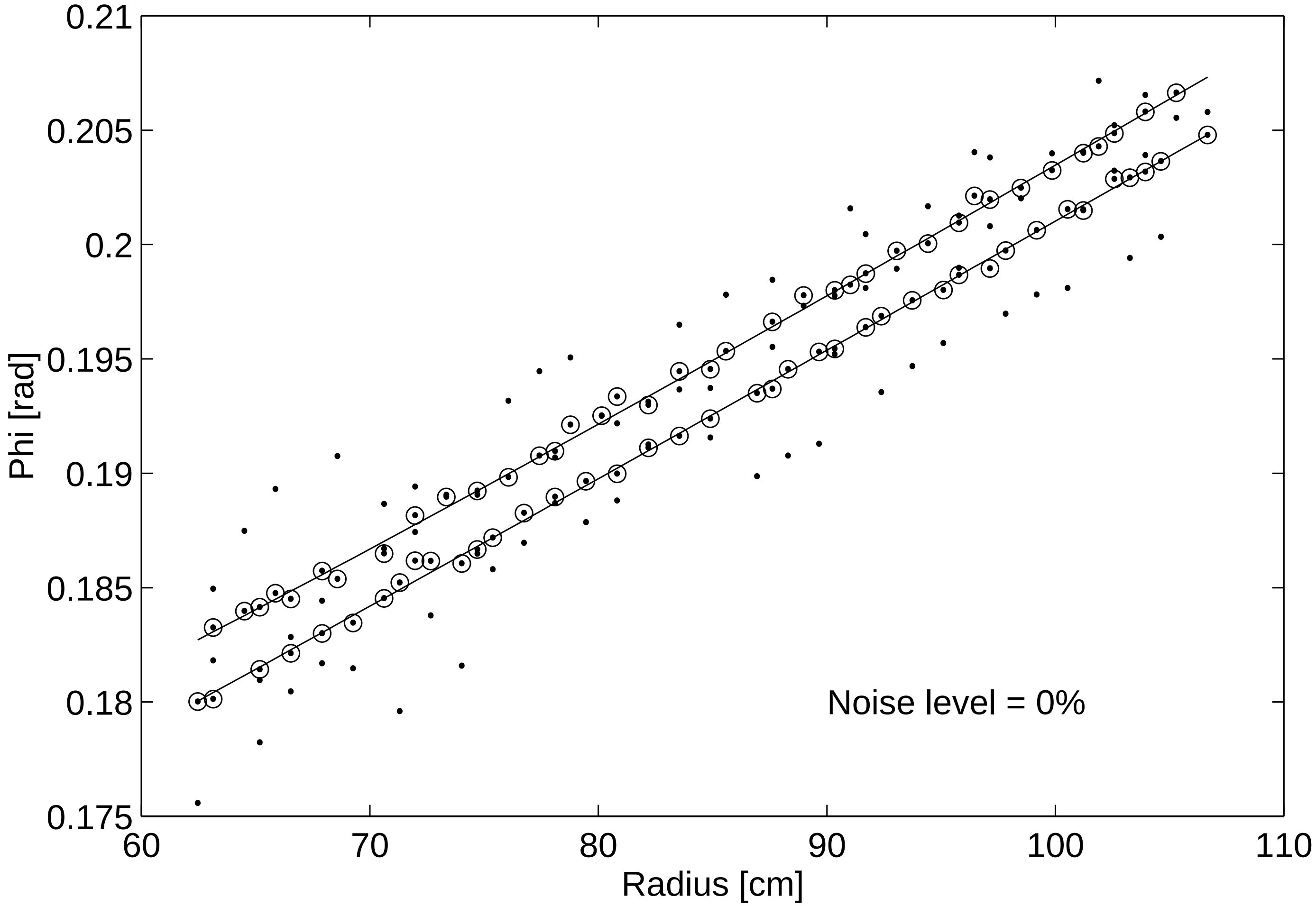
Best result from MTF, first iterations with modified EAA weights (competition between tracks and between mirror hits), followed by iterations with proper MTF weights.

The Multi-track filter



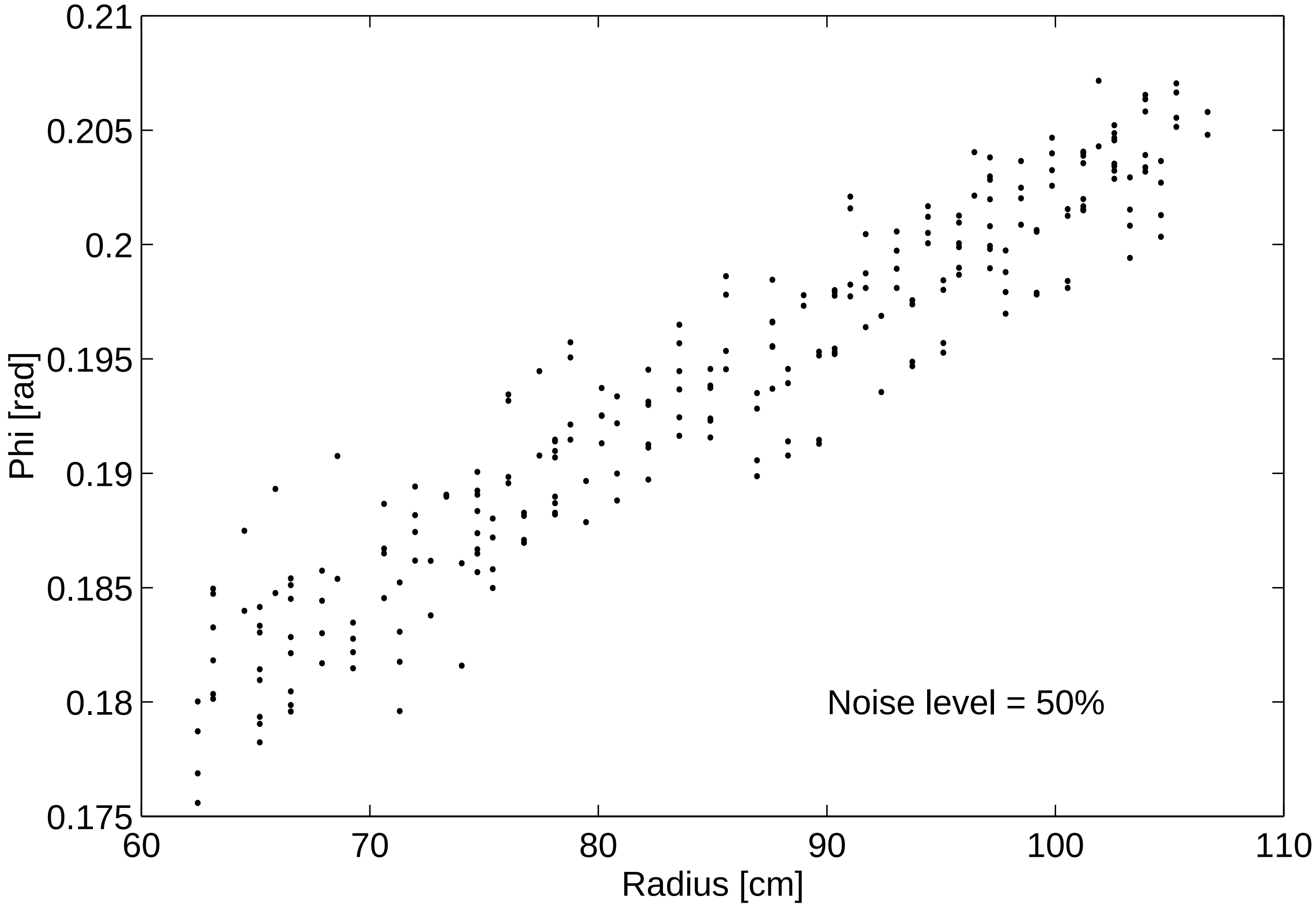
A track pair in the $R\Phi$-projection. Mirror hits, no noise.

The Multi-track filter



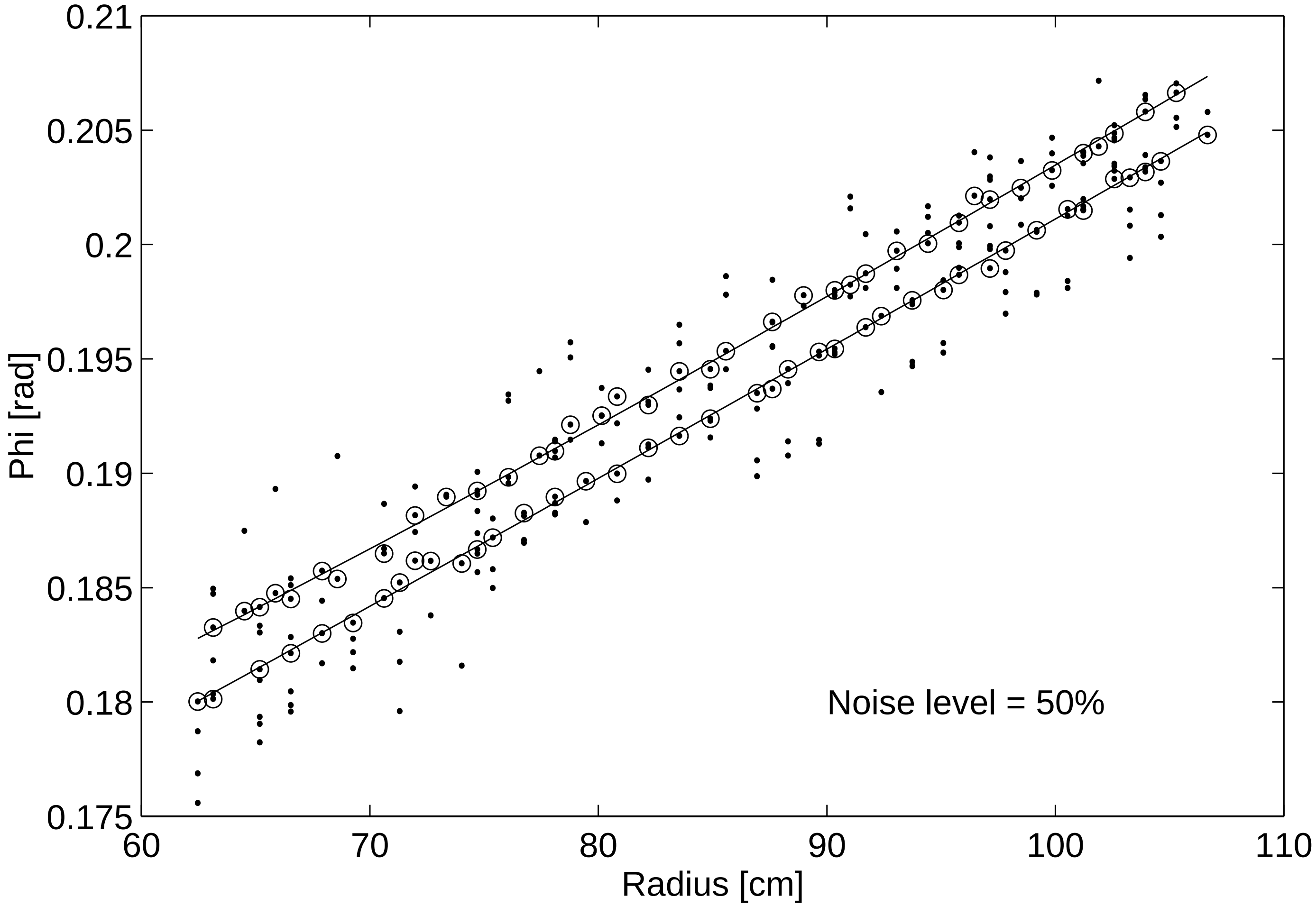
A track pair in the $R\Phi$-projection. Mirror hits, no noise.

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).

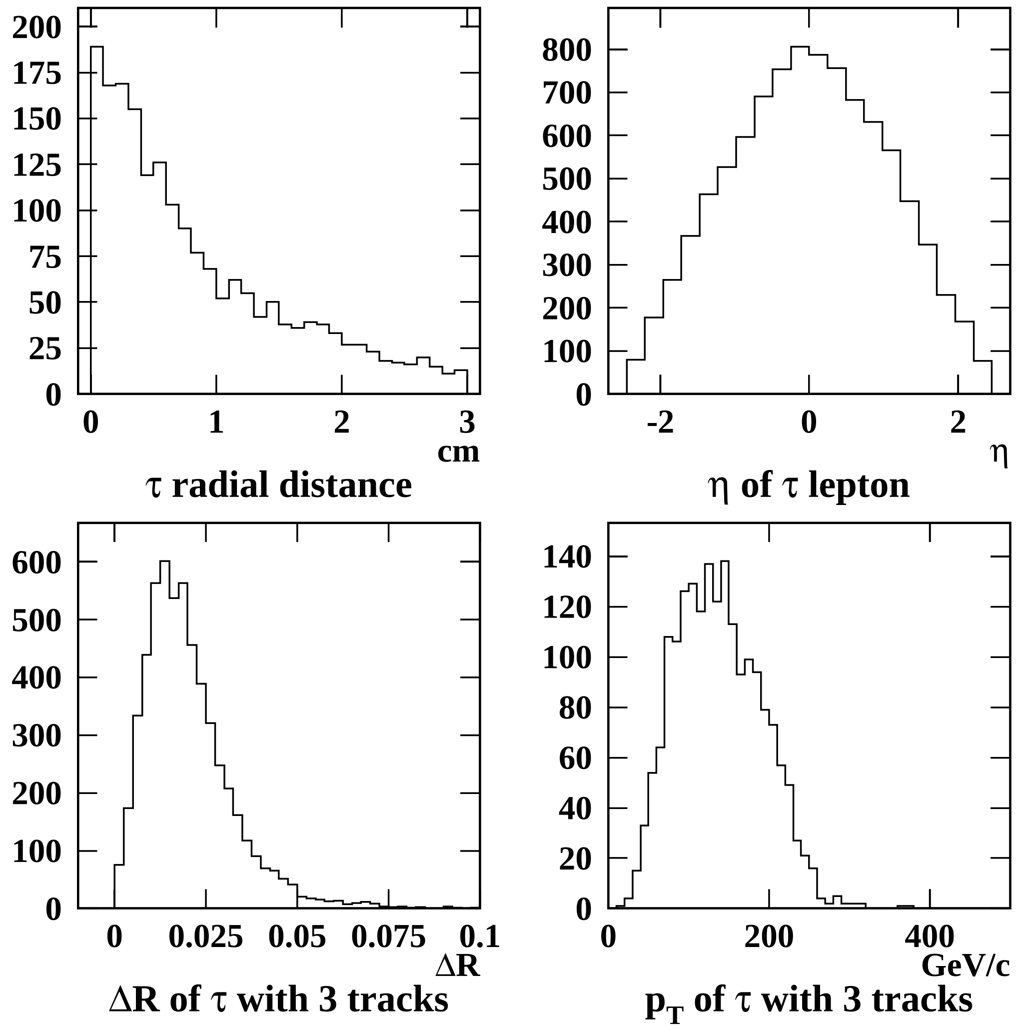
- Based on the building blocks of the DAF in ORCA
- Track finding by a combinatorial Kalman filter
- MTF used for smoothing
- Extensively verified on track pairs
- Physics study with 3-prong τ-decays

The Multi-track filter

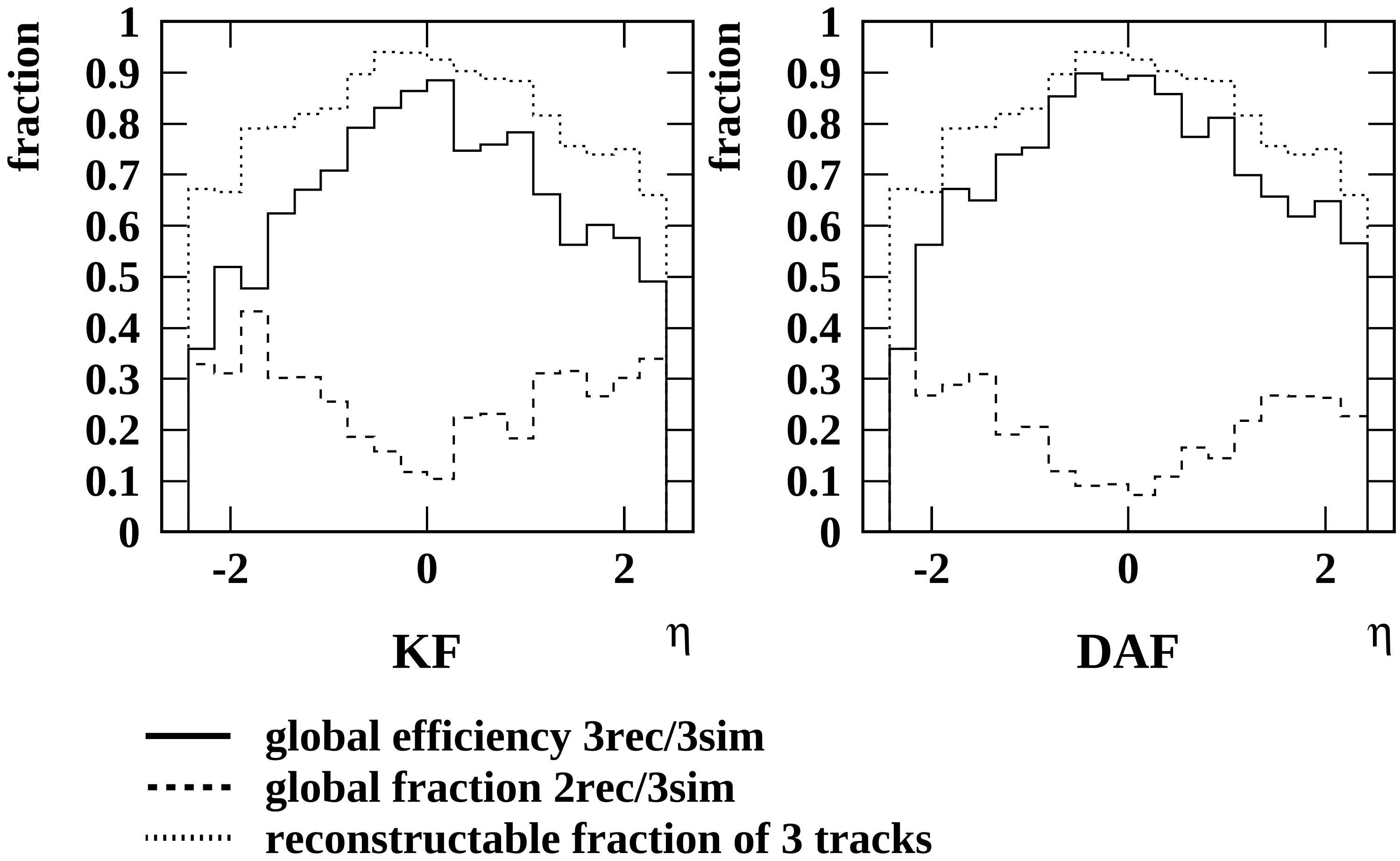
Results of a study of 3-prong τ-decays with the KF, the DAF, and the MTF:

- 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 χ^2-probability

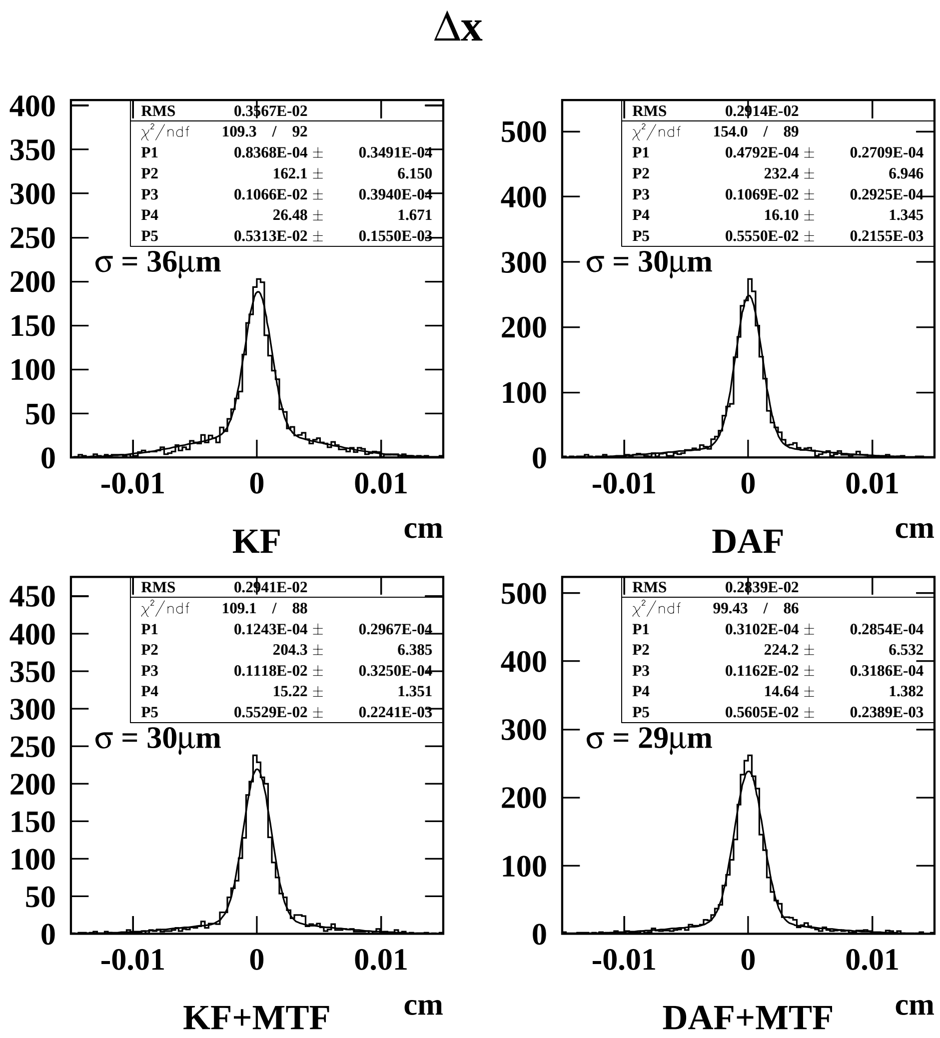
The Multi-track filter



Characteristics of simulated $H^0 \longrightarrow \tau^+\tau^-$ sample

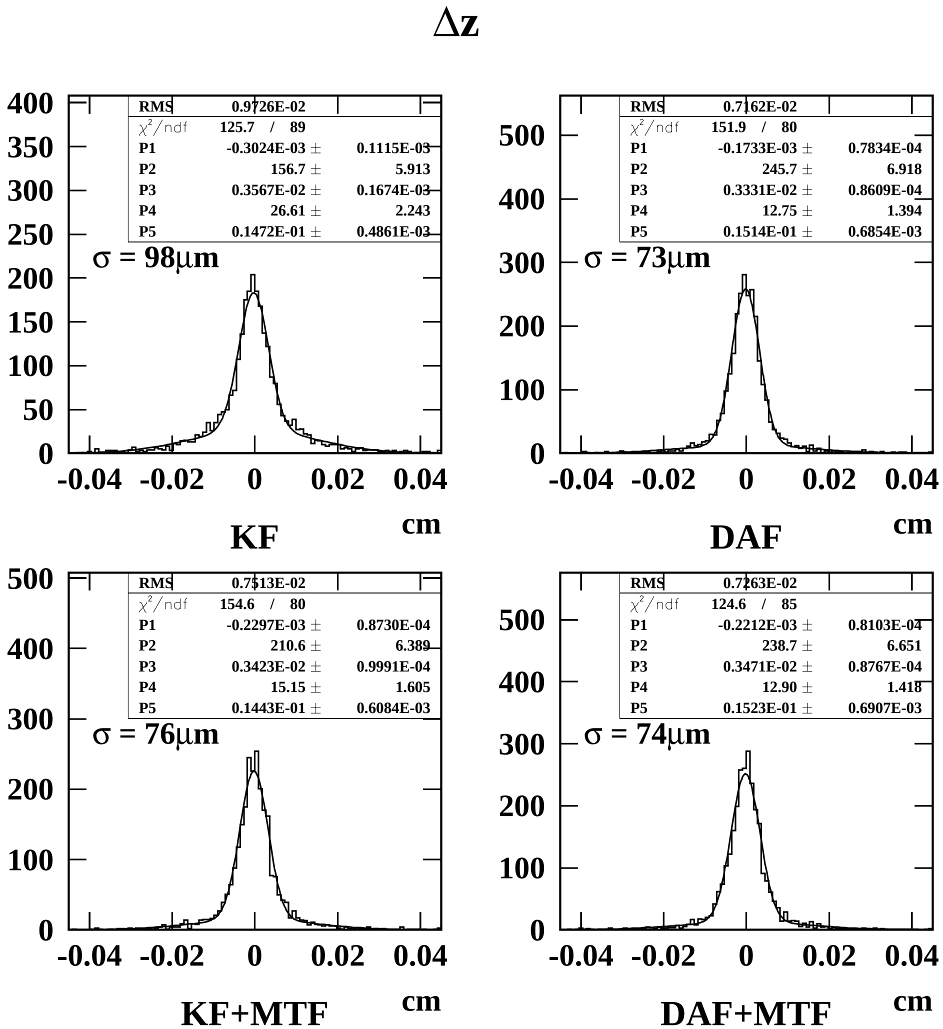


Efficiency of reconstructing three tracks in 3-prong τ-decays



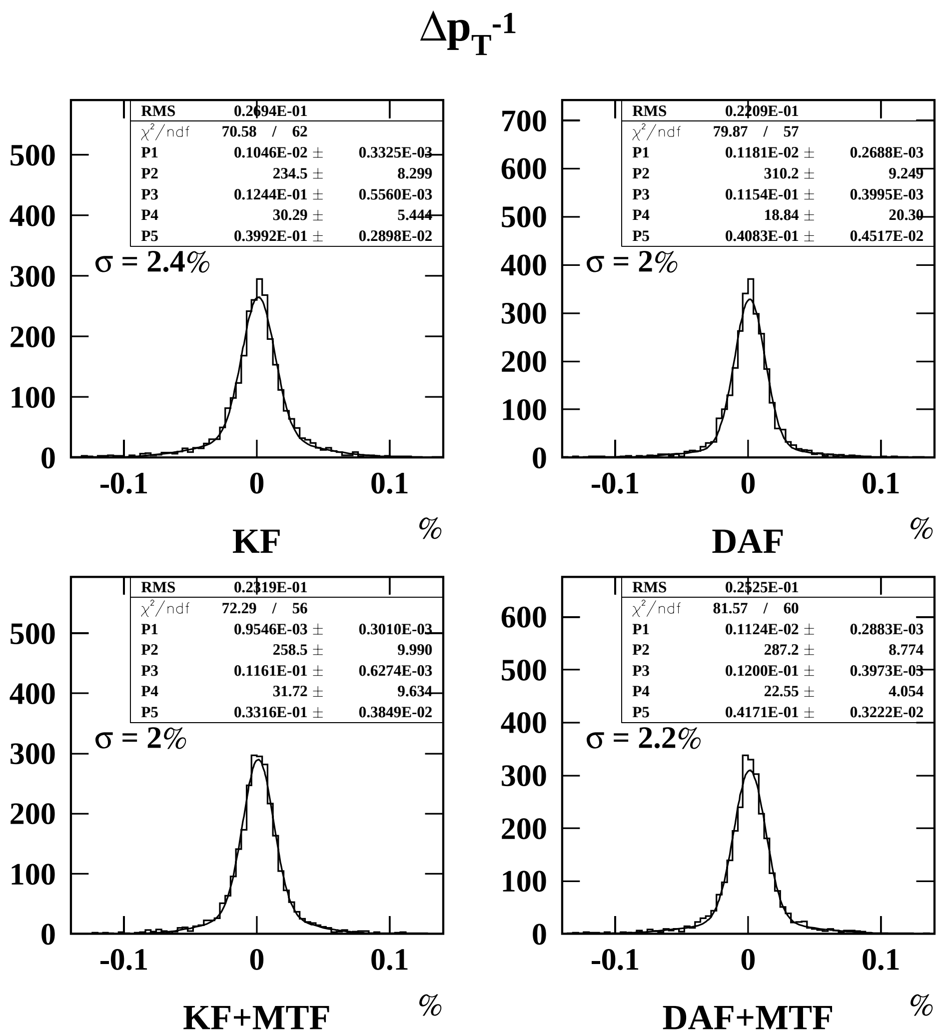
Transverse impact parameter resolution at the τ vertex

The Multi-track filter



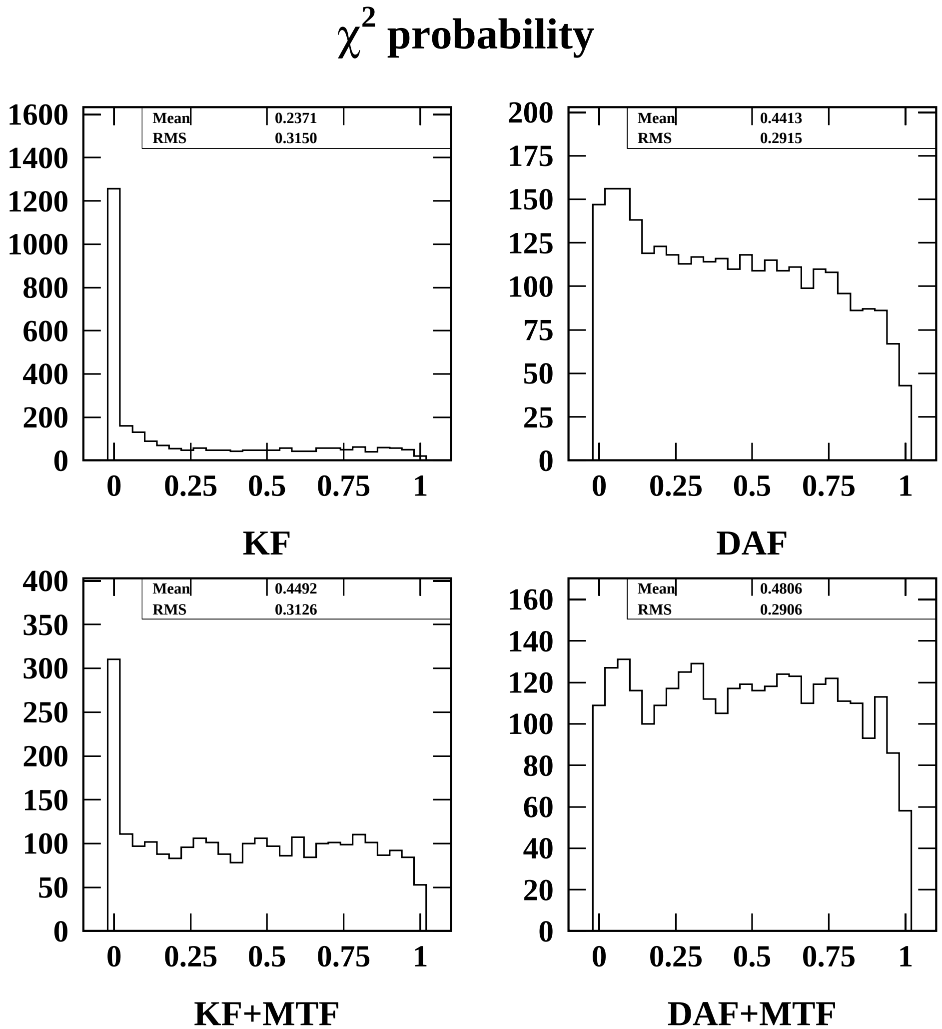
Longitudinal impact parameter resolution at the τ vertex

The Multi-track filter



Transverse momentum resolution at the τ vertex

The Multi-track filter



$\chi 2$-probability distributions

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 (χ^2)

- **Test of vertex hypothesis and final assignment**

  Use information from all vertices

Vertex Fitting

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

Linear vertex fit

Vertex can be interpreted as dynamic system: add a track to an existing vertex.

- **System equation (linear):**

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1}, \quad k = 1, \ldots, m$$

$\boldsymbol{v}_k$: vertex with k tracks, no "process noise"!

Linear vertex fit

- Measurement equation (non-linear):

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{v}, \boldsymbol{q}_k) + \boldsymbol{\epsilon}_k, \quad k = 1, \ldots, n$$

$\boldsymbol{m}_k$: estimated parameters (5-dim) of track k

$\boldsymbol{h}_k$: track model (helix)

$\boldsymbol{q}_k$: momentum at the vertex $\boldsymbol{v}$ (3-dim)

$\boldsymbol{\epsilon}_k$: covariance matrix of $\boldsymbol{m}_k$

- The state vector of the system is augmented by $\boldsymbol{q}_k$ 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:

- **χ^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

- **χ^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:**

$$\chi_k^2 = (\boldsymbol{m}_k - \boldsymbol{H}\boldsymbol{v})^T (\alpha \boldsymbol{V}_k)^{-1} (\boldsymbol{m}_k - \boldsymbol{H}\boldsymbol{v})$$

$$p_k = \frac{\exp(-\chi_k^2/2)}{\exp(-\chi_{\mathrm{cut}}^2/2) + \exp(-\chi_k^2/2)}$$

χ_k^2: **distance between track $\boldsymbol{m}_k$ and vertex $\boldsymbol{v}$**

χ_{cut}^2: **distance cut**

α: **Optional annealing factor**

Robust single-vertex fit

Deterministic annealing helps to reach the global optimum:

- $\alpha > 1$ at the start
- Decrease α 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 $\longrightarrow$ 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

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

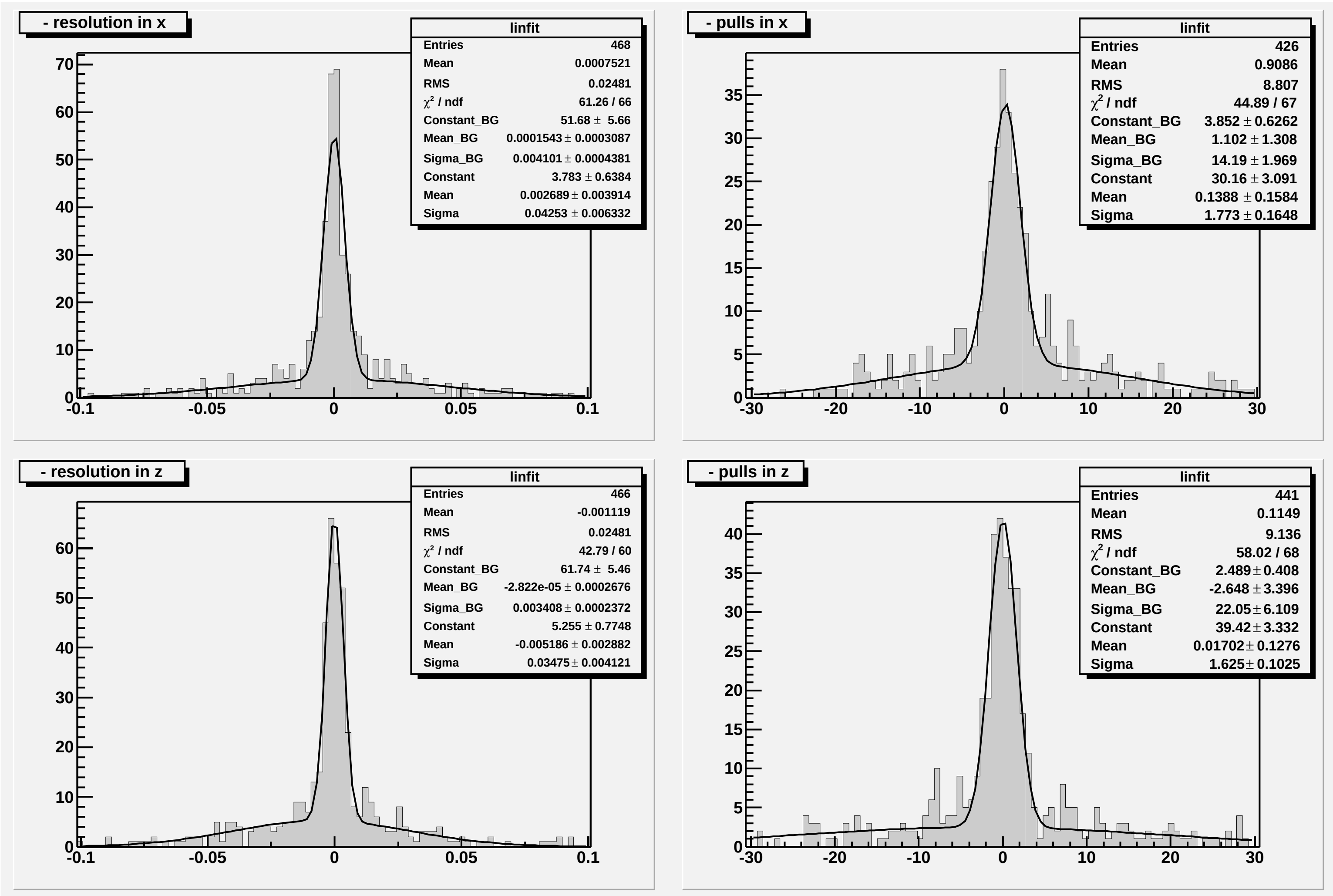
Precision and timing of $q\bar{q}$ vertices

Robust single-vertex fit

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

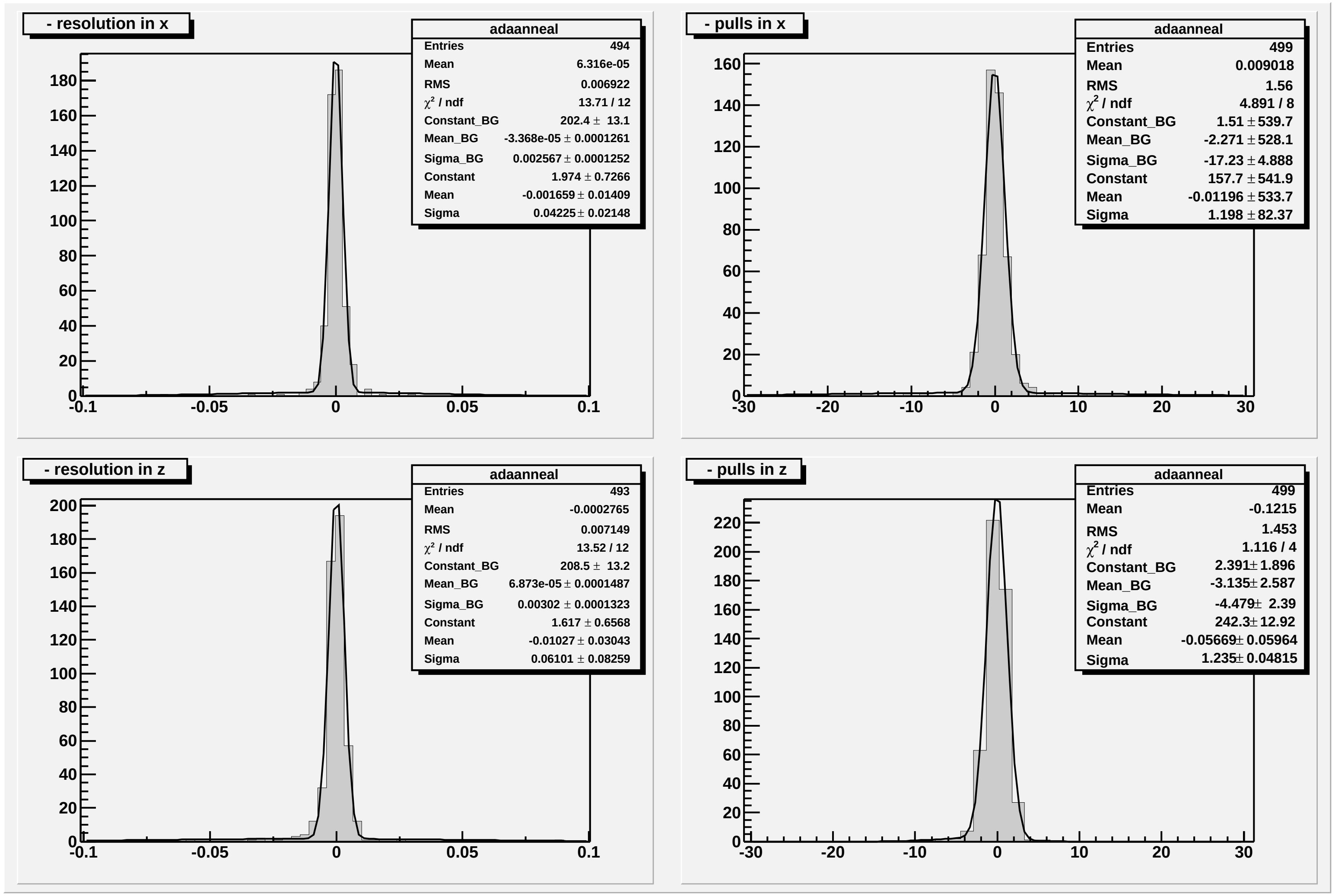
Precision and timing of $c\bar{c}$ vertices

Robust single-vertex fit



Linear Fitter

Robust single-vertex fit



Adaptive Fitter with annealing

Adaptive Multi-vertex fit

- Concurrent fit of several vertices with competition

  A track can belong to at most one vertex.

- Computation of the weights:

$$\chi^2_{ki} = (\boldsymbol{m}_k - \boldsymbol{H}\boldsymbol{v}_i)^T (\alpha \boldsymbol{V}_k)^{-1} (\boldsymbol{m}_k - \boldsymbol{H}\boldsymbol{v}_i)$$

$$p_{ki} = \frac{\exp(-\chi^2_{ki}/2)}{\exp(-\chi^2_{\text{cut}}/2) + \sum_i \exp(-\chi^2_{ki}/2)}$$

  χ^2_{ki}: **distance between track $\boldsymbol{m}_k$ and vertex $\boldsymbol{v}$.**

- 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)

Special thanks to

Are Strandlie

Matthias Winkler

Wolfgang Waltenberger

Many thanks to

Rainer Mankel

for inviting me to this seminar

References

[1] G. Kitagawa, Comput. Math. Appl. 18 (1989) 503

[2] R. Frühwirth, Comp. Phys. Comm. 100 (1997) 1

[3] R. Frühwirth and S. Frühwirth-Schnatter, Comp. Phys.Comm. 110 (1998) 80

[4] R. Frühwirth and M. Regler, Nucl. Instr. Meth. A 456 (2001) 369

[5] R. Frühwirth and A. Strandlie, Comp. Phys. Comm. 120 (1999) 197

[6] M. Ohlsson, C. Peterson and A. Yuille, Comp. Phys. Comm. 71 (1992) 77

[7] M. Gyulassy and M. Harlander, Comp. Phys. Comm. 66 (1991) 31

[8] M. Winkler, PhD thesis, Vienna University of Technology (2002)

[9] A. Strandlie and R. Frühwirth, Comp. Phys. Comm. 133 (2000) 34