Analysis of test beam data by global optimization methods

R. Frühwirth[1], T. Bergauer[1], M. Friedl[1], H. Gjersdal[2], C. Irmler[1], T. Spielauer[1], A. Strandlie[2,3], M. Valentan[1]

[1]Institute of High Energy Physics, Austrian Academy of Sciences, Vienna, Austria

[2]University of Oslo, Oslo, Norway

[3]Gjøvik University College, Gjøvik, Norway

VCI 2013, Feb 14, 2013

Outline

- Motivation
- Algorithms
- Examples
  - ATLAS pixel test beam
  - Belle II test beam 2010
  - Belle II test beam 2012
- Conclusions

Motivation

- Track reconstruction in a silicon tracker requires
  - Estimation of alignment constants (shifts, rotations, . . .)
  - Calibration of hit resolution, depending on angle and cluster size
  - Knowledge of material traversed
- Solutions for full-scale trackers have been developed
  - Track-based alignment: Millepede, HIP, Kalman alignment, . . .
  - Track fit: GENFIT, KalTest, . . .
  - Sensor resolution: Manual or automatic tuning
  - Material: detector description database, manual tuning
- Full machinery is an overkill in a simple test beam environment
- Have developed **stand-alone software** for the concurrent estimation of alignment parameters, hit resolutions and material thickness
- Principal functionalities available in C++ (Oslo/Gjøvik) and Matlab (Vienna)

Algorithms

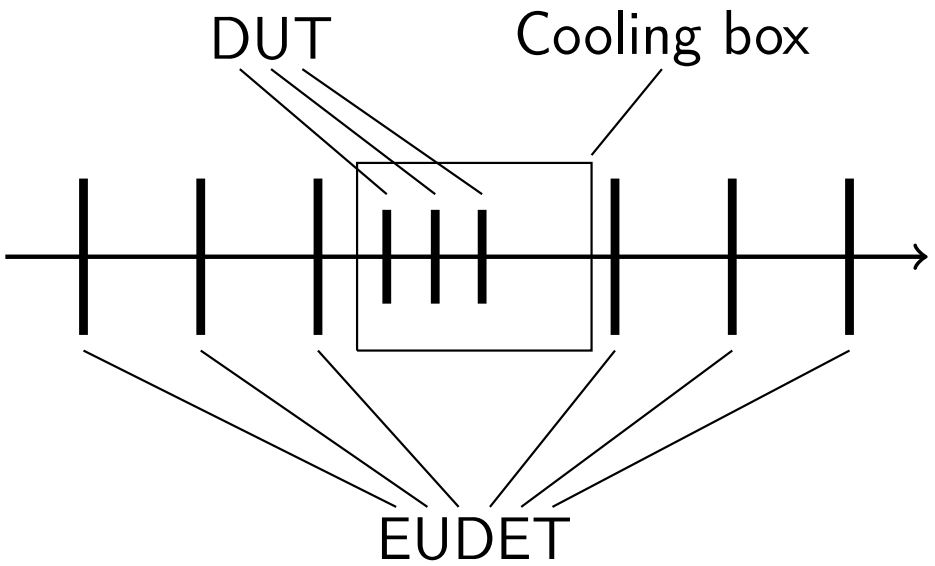
- Track model in a test beam setup is usually a straight line
- Estimation of track parameters either by Kalman filter (recursive least-squares) or by global fit (weighted least squares)
- Estimation produces quality indicators such as local and global χ^2 statistics, standardized residuals (pulls) etc.
- Standardized residuals must have mean 0 and rms 1, χ^2 must have mean equal to ndf, also for non-Gaussian errors
- **Principal idea**: optimize an objective function based on the quality indicators as a function of a set of parameters
- Parameters are alignment constants, hit resolutions, material thickness

Algorithms

- The objective function can be tailored to the problem at hand (see examples later)
- Requires multivariate optimization, preferably without computation of gradients
- Algorithms must be able to find the **global** optimum
- We have tried two optimizers:
  - Simplex algorithm by Nelder and Mead: one of the standard methods (Matlab: fminsearch)
  - VXQR1 by A. Neumaier: recent development at Univ. Vienna (Matlab: vxqr1 package)
- Simplex more prone to end in a local minimum, can be cured by restarting several times
- VXQR1 finds region of global optimum more reliably, but takes longer to converge to the actual optimum

Example 1: ATLAS pixel test beam

- EUDET telescope: six pixel sensors, 18.5 µm × 18.5 µm, $\sigma \approx 4$ µm
- DUT: three pixel sensors, 50 µm × 400 µm, $\sigma \approx 15$ µm/115 µm
- Total length ≈ 1 m



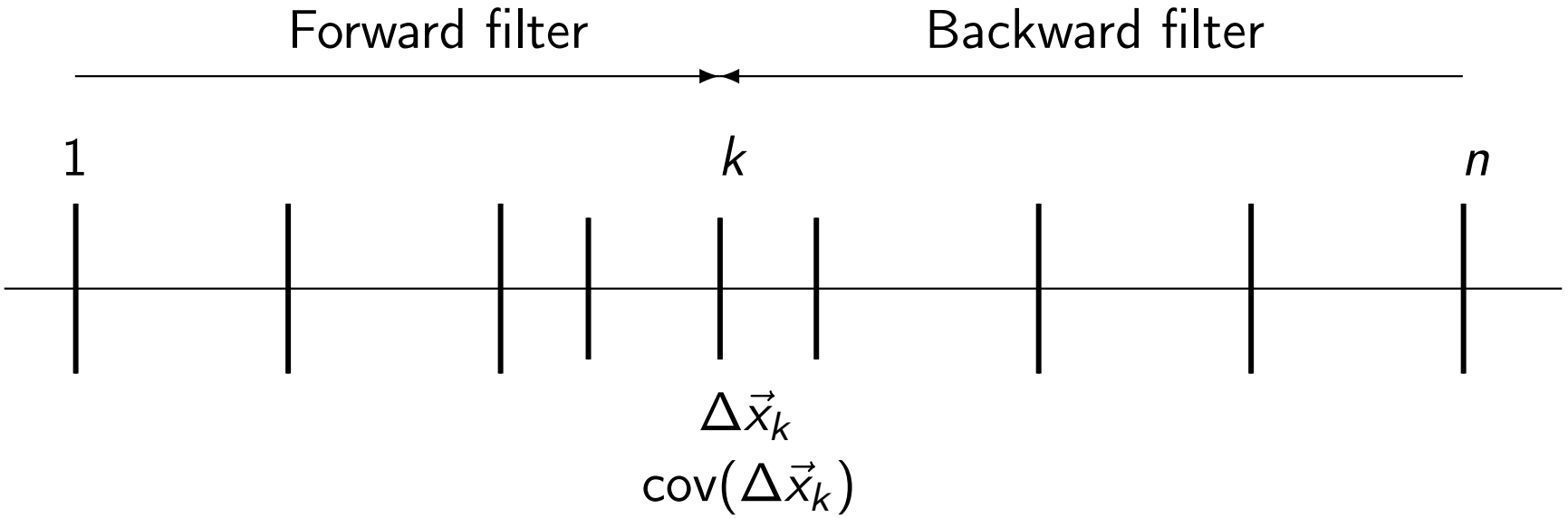
The experimental setup, consisting of ATLAS pixel planes (DUT), and EUDET Mimosa26 planes (EUDET).

Example 1: ATLAS pixel test beam

- **Aim**: estimate as many resolutions and thicknesses as possible
- Assume that material is concentrated in sensor modules
- Assume that measurement errors are uncorrelated
- **Data**: about 40000 tracks (π^+) at $p = 120\,\mathrm{GeV}/c$
- Optimize several objective functions based on standardized hit residuals and state differences of forward and backward filters: force mean to 0 and rms to 1
  - Each call to the objective function requires a fit of the track sample
  - Run forward and backward Kalman filter on all tracks, compute hit residuals and state differences
- Cross-check with simulated data, also for lower beam energies
- Results published in H. Gjersdal et al., JINST 8 P01009 (2013)

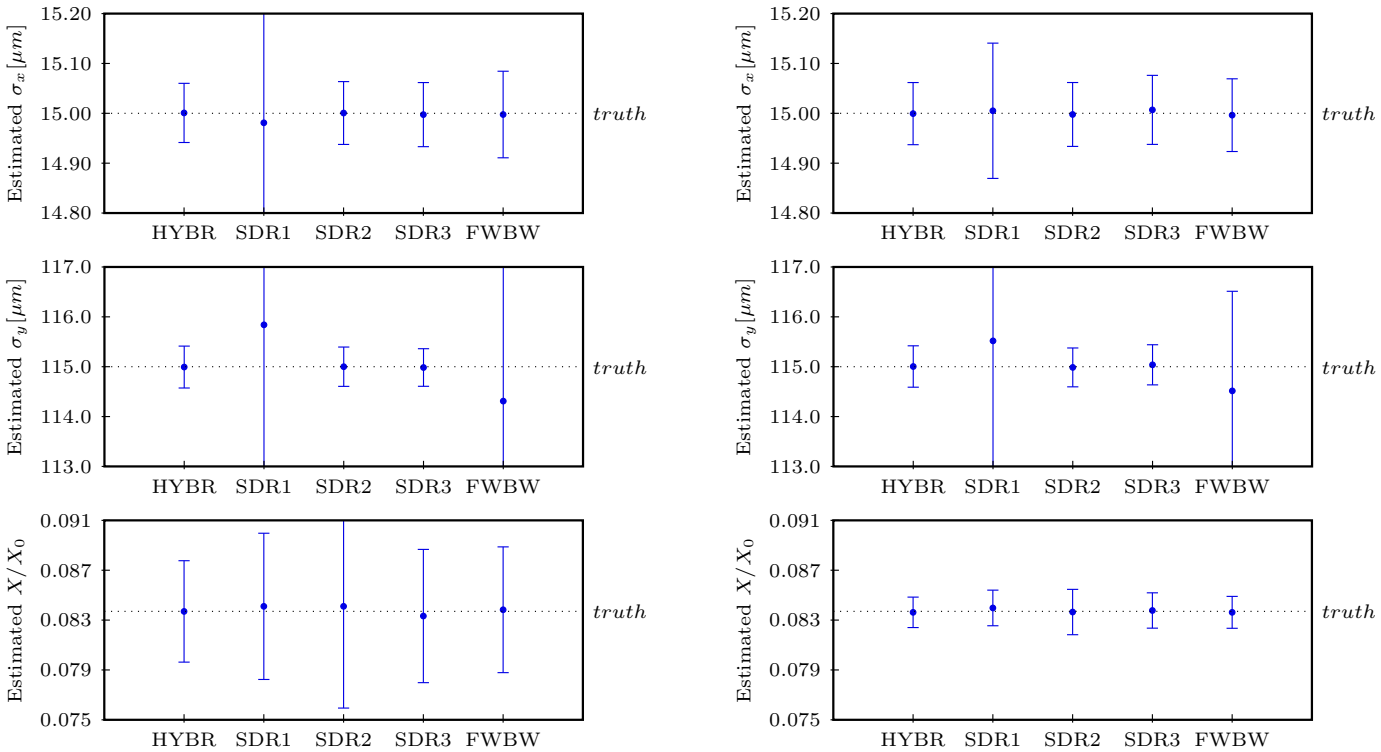
Example 1: ATLAS pixel test beam

- State difference $\Delta\vec{x}_k$ in layer k equals predicted state (forward) minus updated state (backward) or vice versa
- Covariance matrix of $\Delta\vec{x}_k$ is the sum of the respective covariance matrices
- Can be computed in all layers except first and last one
- Material thickness cannot be estimated in first and last layer



Example 1: ATLAS pixel test beam

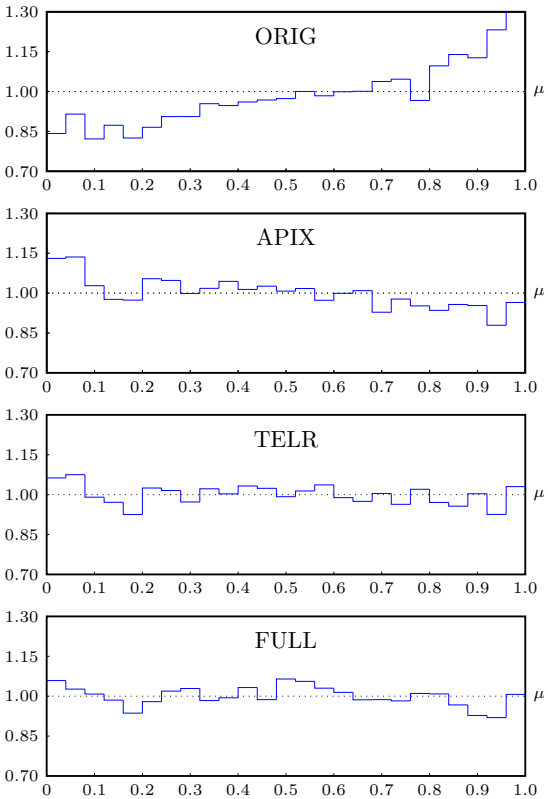
- Simulated data, 5 objective functions, 2 beam momenta



Mean and standard deviation of resolution and thickness estimates from 500 simulation experiments, DUT plane 3, $p = 120\,\mathrm{GeV}/c$ (left) and $p = 40\,\mathrm{GeV}/c$ (right)

Example 1: ATLAS pixel test beam

- Real data



Initial guesses

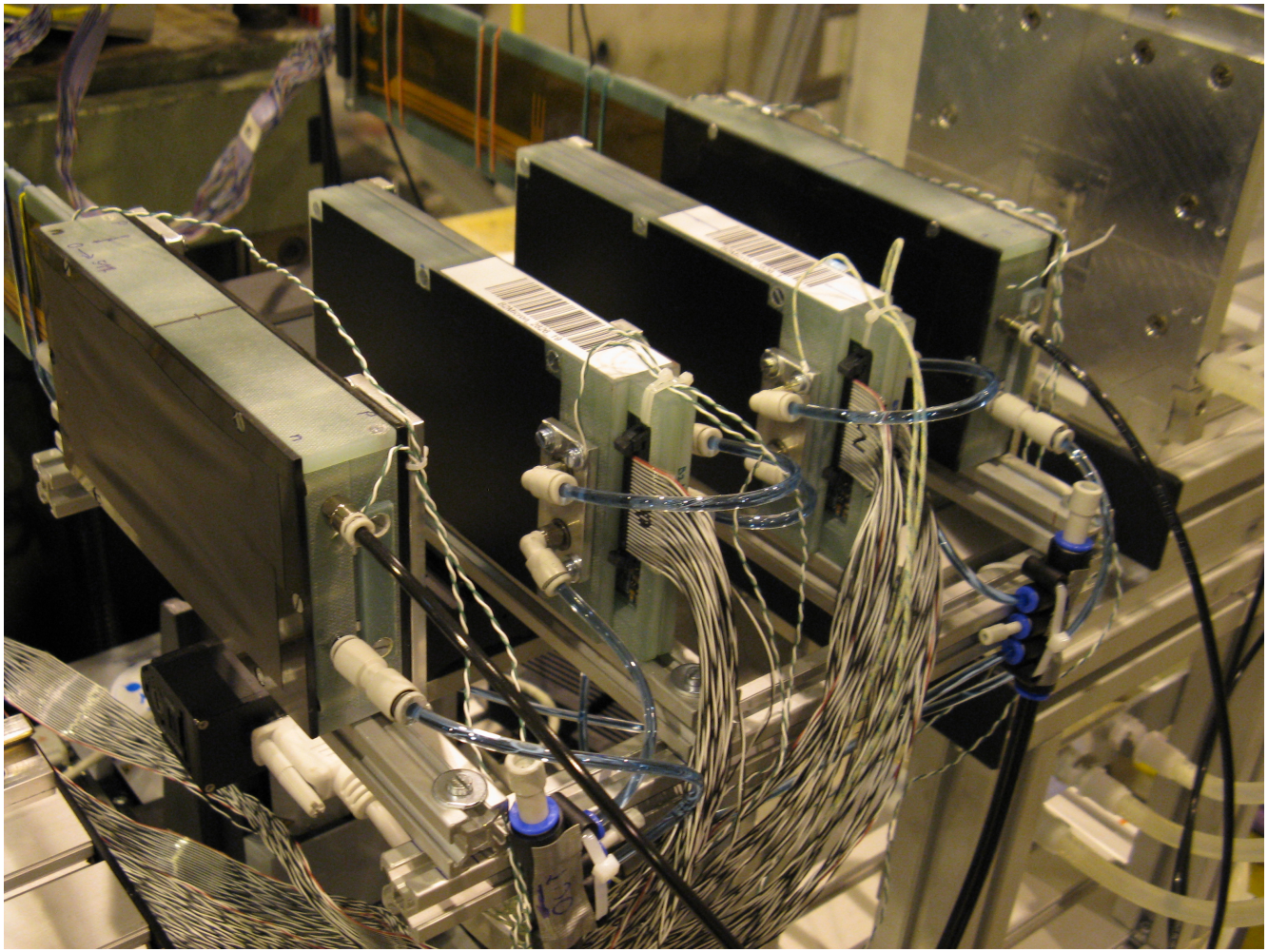
Estimate resolution and material in DUT

Estimate resolution and material in DUT, resolution in EUDET

Estimate resolution and material in DUT, resolution and material in EUDET

P-values of the track χ^2 for an increasing number of free model parameters. The histograms are scaled such that the average bin-content is 1.

Example 2: Belle II strip sensor test beam 2010

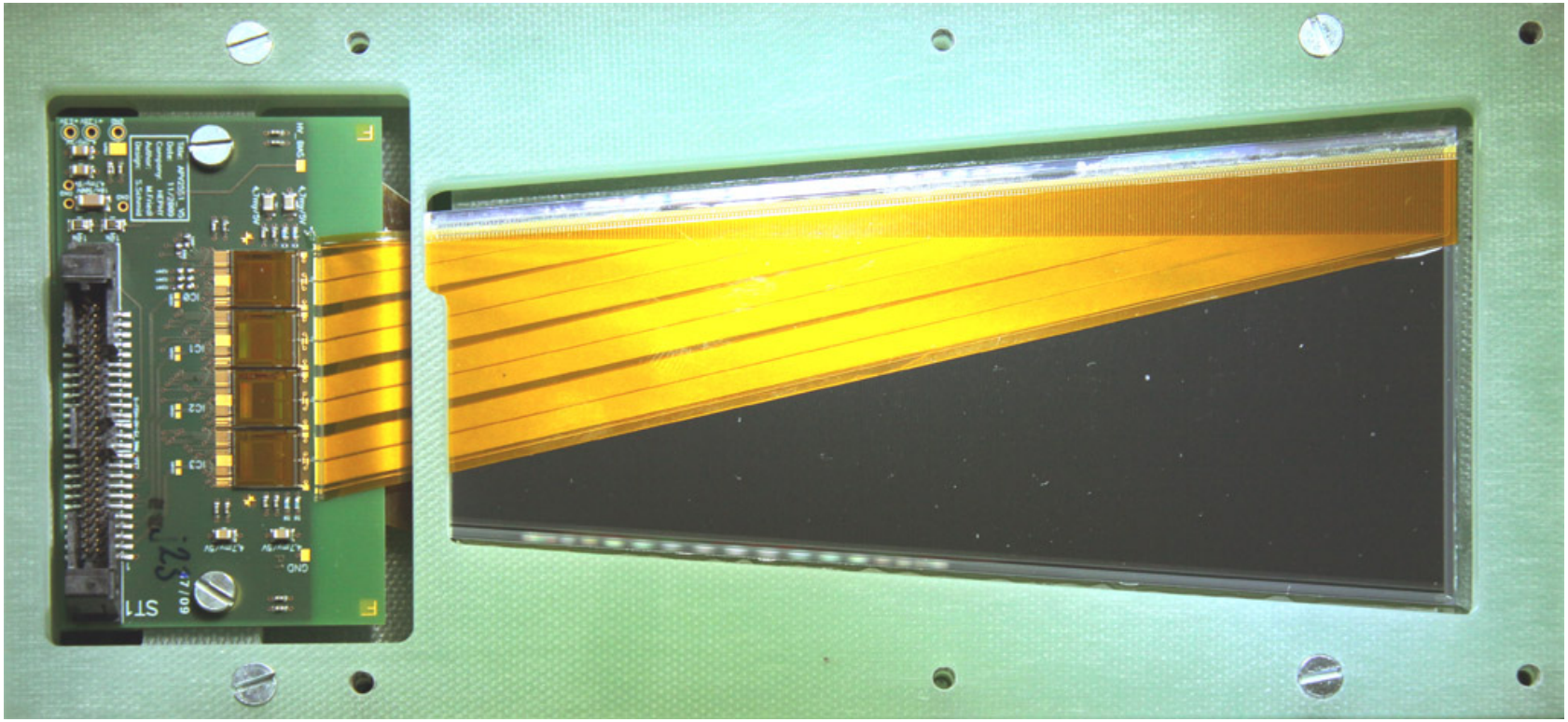


Test beam setup 2010

Example 2: Belle II strip sensor test beam 2010

- Stack of four double-sided sensors, n-side measures x, p-side measures y
- Total length of stack is $\approx 27\,\text{cm}$
- Two rectangular sensors (first and last):
  - n-side: 240 µm pitch with intermediate strips, measures x
  - p-side: 75 µm pitch with intermediate strips, measures y
- Two trapezoidal sensors:
  - n-side: 240 µm pitch with intermediate strips, measures x
  - p-side: 50 to 75 µm pitch with intermediate strips, measures y

Example 2: Belle II strip sensor test beam 2010



Trapezoidal sensor, n-side

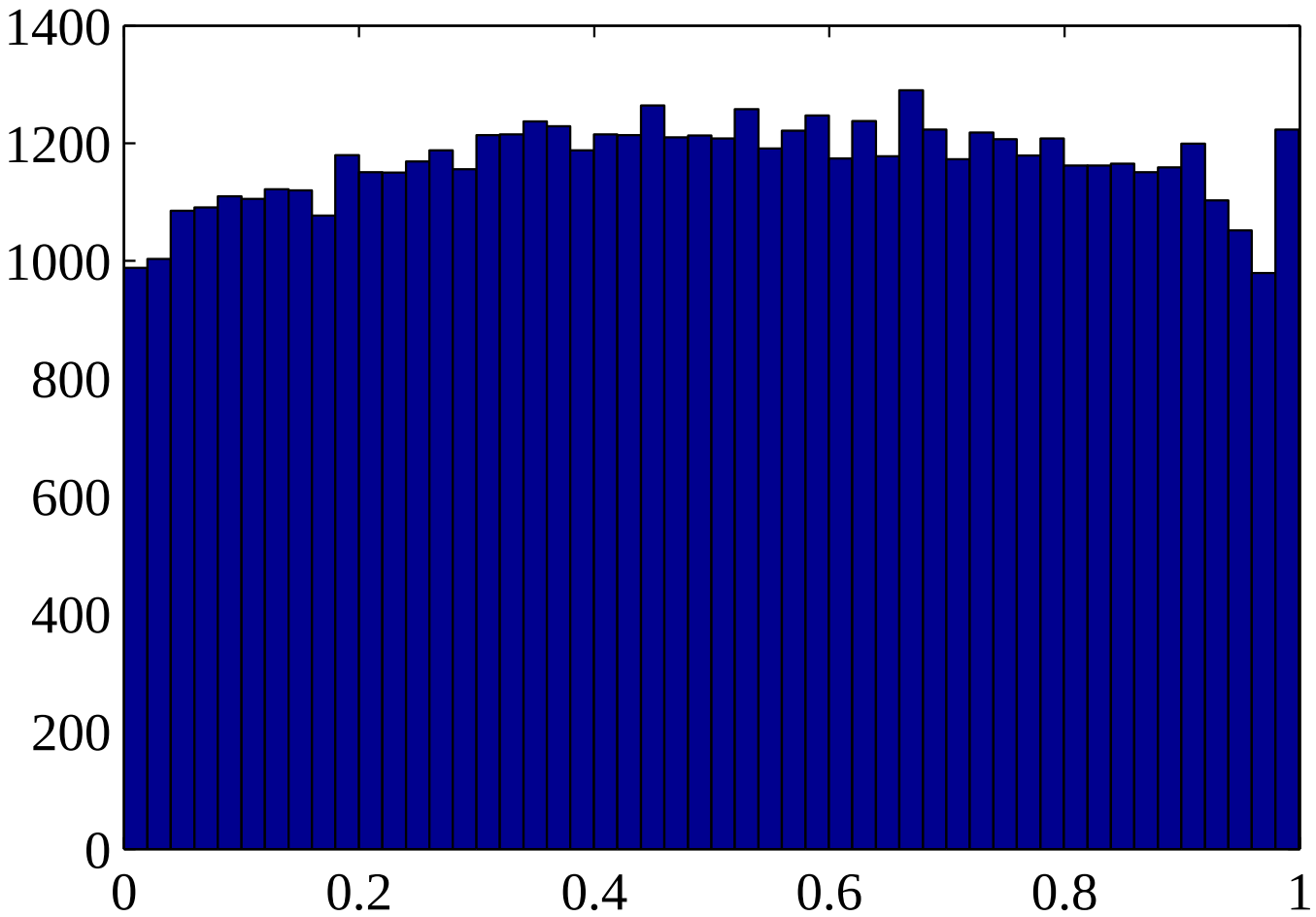
Example 2: Belle II strip sensor test beam 2010

- **Aim**: estimate seven alignment parameters (four shifts and three rotation angles) and eight sensor resolutions (x and y)
- Short lever arm and very thin sensors $\Longrightarrow$ not enough sensitivity to multiple scattering to estimate thickness
- **Data**: about 58000 tracks (π^+, p, K^+) at $p = 120\,\mathrm{GeV}/c$
- Optimize an objective function based on standardized hit residuals: force mean to 0 and rms to 1
  - Each call to the objective function requires a fit of the track sample
  - Run forward and backward Kalman filter on all tracks, compute hit residuals
  - Can use median and MAD in place of mean and rms — suppress influence of outliers

Example 2: Belle II strip sensor test beam 2010

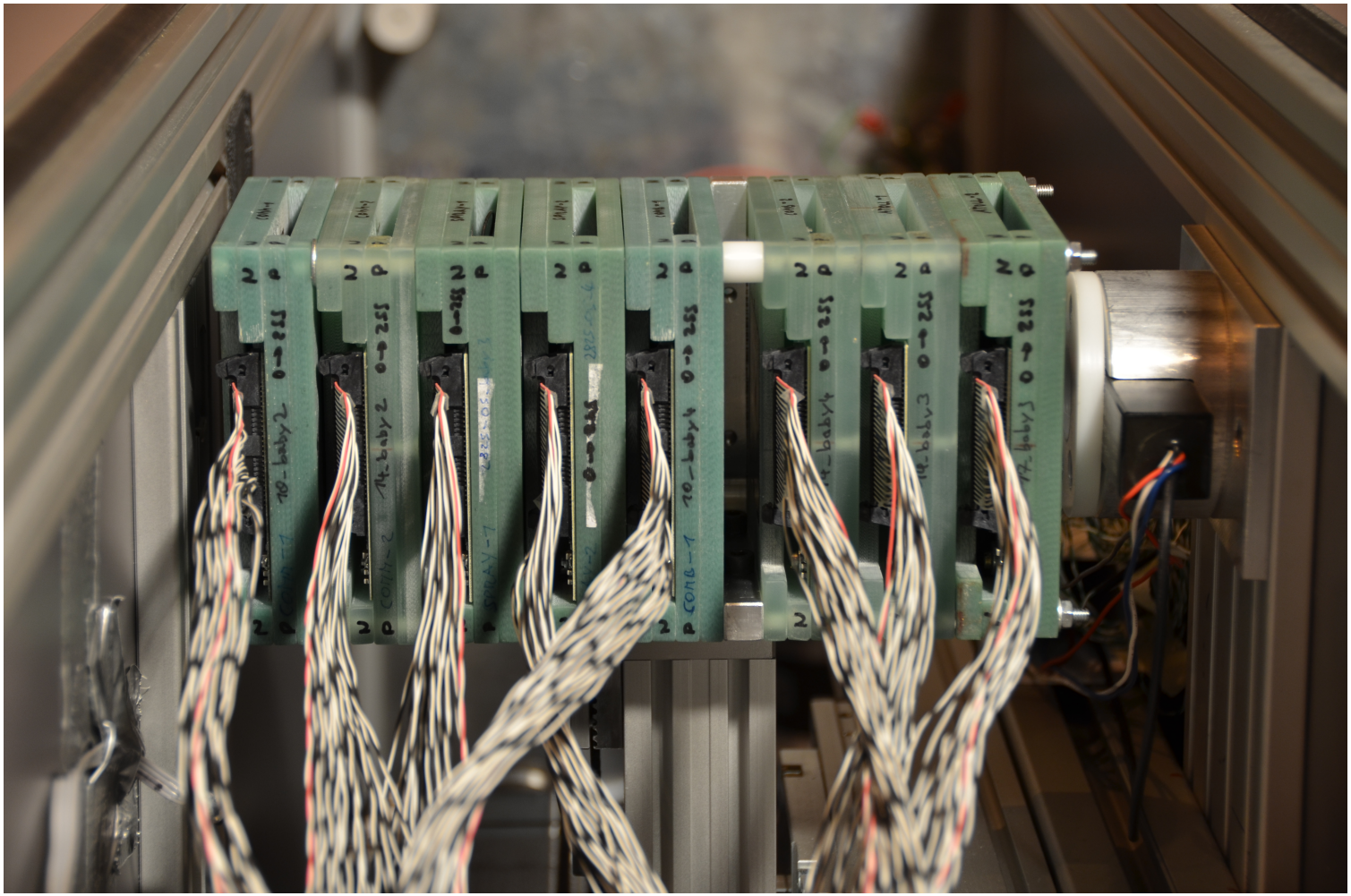
- Implementation in Matlab
- Kalman filter matrices are precomputed, reconstruction of entire sample takes about 400 msec
- Speed of convergence: ≈ 150 fits for alignment, ≈ 500 fits for resolution
- Estimated resolutions (simplex algorithm, in µm):
  - x (n-side): 25 44 40 23
  - y (p-side): 13 11 9 12
- Differences in n-side resolution partially explained by different S/N ratio — correlations on n-side?
- Results of VXQR1 agree within 2 µm
- Final distribution of P-values shows that errors are close to, but not quite Gaussian

Example 2: Belle II strip sensor test beam 2010



P-values of the track χ^2 with final alignment and resolutions.
Mean: 0.504, rms: 0.283.

Example 3: Belle II strip sensor test beam 2012



Test beam setup 2012

Example 3: Belle II strip sensor test beam 2012

- Stack of eight single-sided rectangular sensors, n-side measures y, p-side not read out
- Pitch on n-side is 100 µm
- Total length of stack is ≈ 18 cm
- Sensors 2 and 5 have large areas of inefficiency and are discarded in the analysis
- High percentage of clusters of size 1, between 62% and 73%
- Because of small distance between layers and parallel strips, **strong correlations** between hits with cluster size 1
- η-correction, being deterministic function, does not damp correlations significantly
- Clusters with size > 1 are assumed to be uncorrelated

Example 3: Belle II strip sensor test beam 2012

- **Aim**: estimate four shifts and sensor resolutions separately for clusters with size 1 and larger clusters
- Short lever arm and very thin sensors $\Longrightarrow$ not enough sensitivity to multiple scattering to estimate thickness
- **Data**: about 60000 tracks (π^+, p, K^+) at $p = 120\,\mathrm{GeV}/c$
- Track sample is divided into up to 30 subsamples according to their cluster patterns (size=1 or size>1)
- Most frequent pattern is cluster size 1 only
- Number of tracks used is 37600

Example 3: Belle II strip sensor test beam 2012

- Optimize an objective function based on hit residuals: force mean to 0 and rms to 1
  - Each call to the objective function requires a fit of the track sample
  - Fit implemented by weighted linear regression, multiple scattering is neglected
  - Regression matrices are precomputed for all possible cluster size patterns
  - Resulting fit is considerably faster than Kalman filter: ≈ 120 msec for fit of 30 patterns (37600 tracks).
- Alignment converges very quickly, not affected by correlations
- Estimation of resolutions is affected by correlations

Example 3: Belle II strip sensor test beam 2012

- **But**: Full correlation matrix cannot be determined uniquely from the linear LS fit residuals, as residuals are confined to subspace of lower dimension
- Ridge regression and shrinkage estimators do not give reasonable correlations
- Possible solution: use nonlinear fit instead, for example L1 fit
- As tracks are nearly parallel, approximate L1 fit its just median of hit corrdinates
- Have computed approximate correlation matrix in the following way:
  - Shift each track such that median of hits is zero
  - Compute correlation matrix of all hits

Example 3: Belle II strip sensor test beam 2012

- Have checked this procedure with simulated tracks
- Generate 40000 tracks with a mixture model of errors
  - 70% of hits have correlated uniform errors ($\sigma = 29\,\mu\text{m}$) $\leftrightarrow$ cluster size 1
  - The rest has uncorrelated normal errors ($\sigma = 15\,\mu\text{m}$) $\leftrightarrow$ cluster size >1
  - Correlations are generated randomly, uniformly in $[-0.5, 0.5]$
- Compare results with and without estimated correlation matrix

Example 3: Belle II strip sensor test beam 2012

- Results of simulation study
- With estimated correlations :

Est. σ cluster size 1								Est. σ cluster size 2							
26	×	29	29	×	27	29	28	14	×	17	17	×	15	16	15

- With zero correlations:

Est. σ cluster size 1								Est. σ cluster size 2							
32	×	29	30	×	28	31	31	17	×	13	15	×	13	14	13

- Results not perfect, but estimation of correlations seems to help somewhat

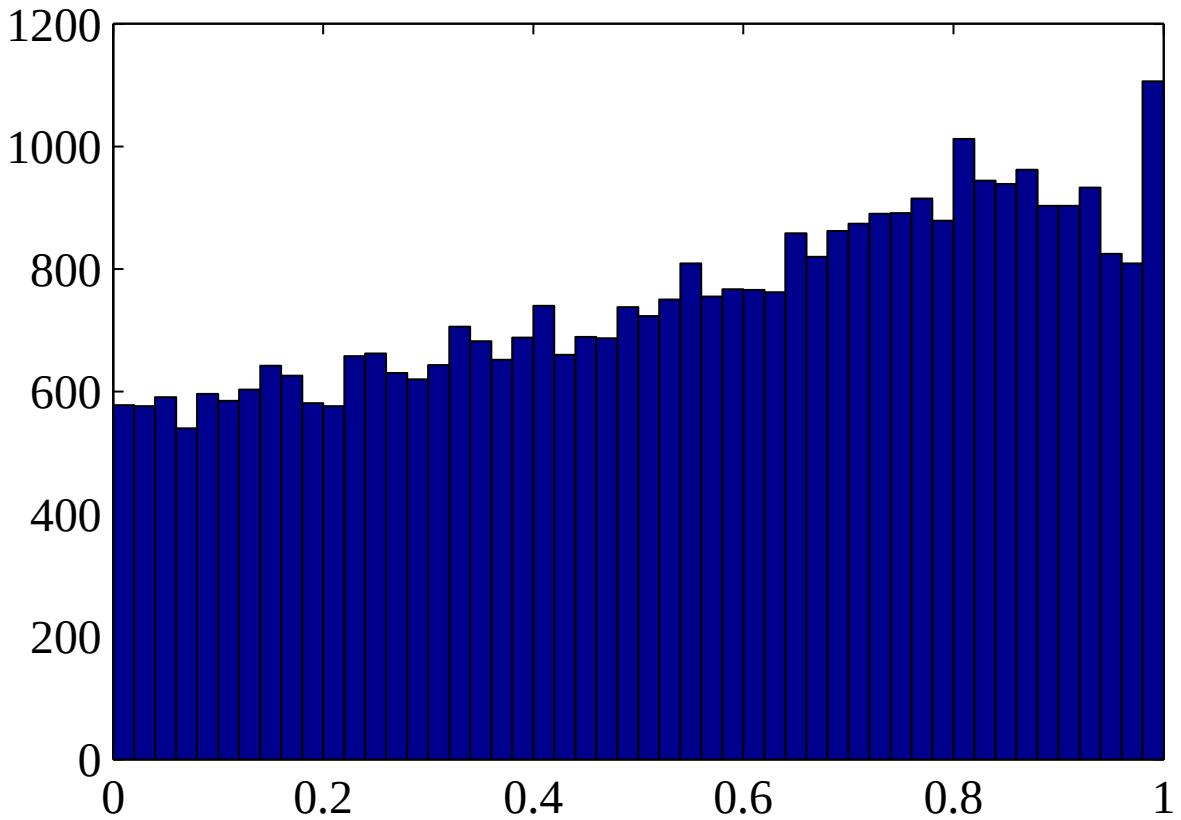
Example 3: Belle II strip sensor test beam 2012

- Under the same procedure we get the following resolutions for the real data:

Estimated σ, cluster size 1							Estimated σ, cluster size > 1						
23	$\times$	24	23	$\times$	27	32	23	15	$\times$	17	17	$\times$	24	26	19

- Actually this looks very reasonable
  - Layers 3 and 4 are identical and give indeed the same resolution
  - Layer 7 has a noise problem and has indeed the worst resolution
  - Layer 1 probably a bit too good
- Distribution of P-values is not that bad either, but clearly shows that the estimated correlations are not the true ones
- Will continue to work on this problem

Example 3: Belle II strip sensor test beam 2012



P-values of the track χ^2 with final alignment and resolutions.
Mean: 0.549, rms: 0.286.

Conclusions

- ☺ Have developed simple stand-alone program for integrated alignment and calibration of silicon sensors in test beam environments
- ☺ Under suitable conditions, sensor thickness (and other material) can be estimated as well
- ☺ Calculation time not more than a couple of minutes for ≈ 50000 tracks on a single core
- ☺ Reconstruction of larger samples can be easily parallelized
- ☺ Individual components of the error distribution can be estimated, e.g. for a mixture of different cluster sizes
- ☺ Multiple scattering decorrelates measurements and enables estimation of module material and possibly other material in the beam

Conclusions

- ☹ Global optimization limited to a fairly small number of variates, doubtful whether approach can be applied to a full scale tracker — could nevertheless be interesting to try it
- ☹ Problems arise if the observation errors are strongly correlated across layers — we have proposed a solution to this problem, which is, however, not perfect

Conclusions

- Should try to minimize the correlations:
  - Reduce number of size 1 clusters by improving S/N ratio
  - Introduce stereo angles between layers
  - Slant modules relative to beam
  - Increase distance between layers
  - Use beam with lower energy so that multiple scattering is no longer negligible
  - Put additional material next to sensors (just kidding!)
- Will have another test beam with low energy electrons early next year — how to handle bremsstrahlung?

Conclusions

- Should try to minimize the correlations:
  - Reduce number of size 1 clusters by improving S/N ratio
  - Introduce stereo angles between layers
  - Slant modules relative to beam
  - Increase distance between layers
  - Use beam with lower energy so that multiple scattering is no longer negligible
  - Put additional material next to sensors (just kidding!)
- Will have another test beam with low energy electrons early next year — how to handle bremsstrahlung?

Thank you!