A Kalman Filter for Track-based Alignment

E. Widl, R. Frühwirth, W. Adam

Institute for High Energy Physics

Austrian Academy of Sciences, Vienna, Austria

CHEP06, 13-17 February 2006

Outline

- Introduction
- The Kalman filter algorithm
- Simulation experiments in a simplified setup
- Results from the CMS Inner Tracker
- Conclusions and outlook

Introduction

- Iterative method for global alignment using charged tracks.
- Alignment parameters are updated after each track.
- Update is not restricted to the detector units that are crossed by the track.
- Update is restricted to detector units that have significant correlations with the ones in the current track.
- No inversion of large matrices.
- Certain amount of bookkeeping is required.

Introduction

- Requires an already aligned and fixed reference system.
- All updates are relative to this reference system.
- Possible to use prior information about the alignment obtained from mechanical and/or laser measurements.
- Possible to fix the position of certain detector units by giving them small prior uncertainty.
- Several detectors can be forced to move along with each by giving them large prior correlations.

The Kalman filter algorithm

- Assume we have estimated alignment parameters $\boldsymbol{d}$ with a variance-covariance matrix $\boldsymbol{D}$. They can come from previous measurements or a first guess either.
- The observations $\boldsymbol{m}$ of a track depend on the (true) track parameters $\boldsymbol{x}_{\mathrm{t}}$ and on the (true) alignment parameters $\boldsymbol{d}_{\mathrm{t}}$:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{x}_{\mathrm{t}}, \boldsymbol{d}_{\mathrm{t}}) + \boldsymbol{\varepsilon}, \quad \mathrm{cov}(\boldsymbol{\varepsilon}) = \boldsymbol{V}.$$

  The stochastic vector $\boldsymbol{\varepsilon}$ contains the effects of the observation error and of multiple scattering. Its variance-covariance matrix $\boldsymbol{V}$ can be assumed to be known. Energy loss is taken care of by the track model $\boldsymbol{f}$.

The Kalman filter algorithm

- The track model is linearized by a first-order Taylor approximation:

$$\boldsymbol{m} = \boldsymbol{c} + \boldsymbol{A}\boldsymbol{d}_\mathrm{t} + \boldsymbol{B}\boldsymbol{x}_\mathrm{t} + \boldsymbol{\varepsilon} = \boldsymbol{c} + \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_\mathrm{t} \\ \boldsymbol{x}_\mathrm{t} \end{pmatrix} + \boldsymbol{\varepsilon},$$

$$\boldsymbol{A} = \partial \boldsymbol{m} / \partial \boldsymbol{x}_\mathrm{t} \big|_{\boldsymbol{x}_\mathrm{e}}, \quad \boldsymbol{B} = \partial \boldsymbol{m} / \partial \boldsymbol{d}_\mathrm{t} \big|_{\boldsymbol{d}_\mathrm{e}}, \quad \boldsymbol{c} = f(\boldsymbol{x}_\mathrm{e}, \boldsymbol{d}_\mathrm{e}) - \boldsymbol{A}\boldsymbol{d}_\mathrm{e} - \boldsymbol{B}\boldsymbol{x}_\mathrm{e}.$$

- The expansion point $\boldsymbol{d}_\mathrm{e}$ is the nominal sensor position.
- The expansion point $\boldsymbol{x}_\mathrm{e}$ is the result of a preliminary fit.
- $\boldsymbol{x}_\mathrm{e}$ is biased because of a lack of knowledge of the alignment parameters, and gets weight 0 in the update.

The Kalman filter algorithm

- Update equation of the alignment parameters:

$$\widehat{\boldsymbol{d}} = \boldsymbol{d} + \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{G}(\boldsymbol{m} - \boldsymbol{c} - \boldsymbol{A}\boldsymbol{d}), \quad \boldsymbol{G} = \boldsymbol{V}^{-1} - \boldsymbol{V}^{-1}\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{V}\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{V}^{-1}.$$

- Update of the covariance matrix:

$$\widehat{\boldsymbol{D}} = \left(\boldsymbol{I} - \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{G}\boldsymbol{A}\right)\boldsymbol{D}\left(\boldsymbol{I} - \boldsymbol{A}^T\boldsymbol{G}\boldsymbol{A}\boldsymbol{D}\right) + \boldsymbol{D}\boldsymbol{A}^T\boldsymbol{G}\boldsymbol{V}\boldsymbol{G}\boldsymbol{A}\boldsymbol{D}.$$

Both terms on the right hand side are positive definite, so the left hand side is garantueed to be positive definite as well.

The Kalman filter algorithm

- When taking a closer look on all matrices in the update equations it turns out that $\boldsymbol{DA}^T$ is the only large matrix.
- The size of this matrix decreases if the update is restricted to detector units which have significant correlations with the ones in the current track.
- For this reason, attach to each detector unit i a list L_i of the detector units which have significant correlations with i. L_i contains only i itself in the beginning and grows as more tracks are processed.
- The speed of the update depends on the size of the list L of all detector units which are correlated with the ones crossed by the current track: $L = \bigcup_{i \in I} L_i$.
- The computational complexity of the parameter update is of the order $|L| \cdot k$, where k is the number of detector units hit by the current track.
- The computational complexity of the covariance matrix update is of the order of $|L|^2$.

The Kalman filter algorithm

- Regarding the computational complexity it is of crucial importance to restrict the size of the list L to an acceptable number.

- Current proposal:

  - Define a relation "$\sim$" between two different detector units i and j

    $$i \sim j \iff i \text{ and } j \text{ have been crossed by the same track.}$$

  - Define a metrics on the basis of this relation:

    If $i \sim i_1 \sim i_2 \sim \cdots \sim i_n \sim j$ is the shortest chain connecting i to j, the distance is $d(i,j) = n+1$. In particular, if $i \sim j$, then $d(i,j) = 1$.

The Kalman filter algorithm

✧ Using this distance, the following algorithm for updating the lists L_i, $i \in I$, is proposed:

For all $i \in I$ do:

- For all $j \in I \setminus \{i\}$ do:

  For all $k \in L_j$ with $d(k, j) < d_{\max}$, add k to L_i and store $d(k, i) = d(k, j) + 1$.

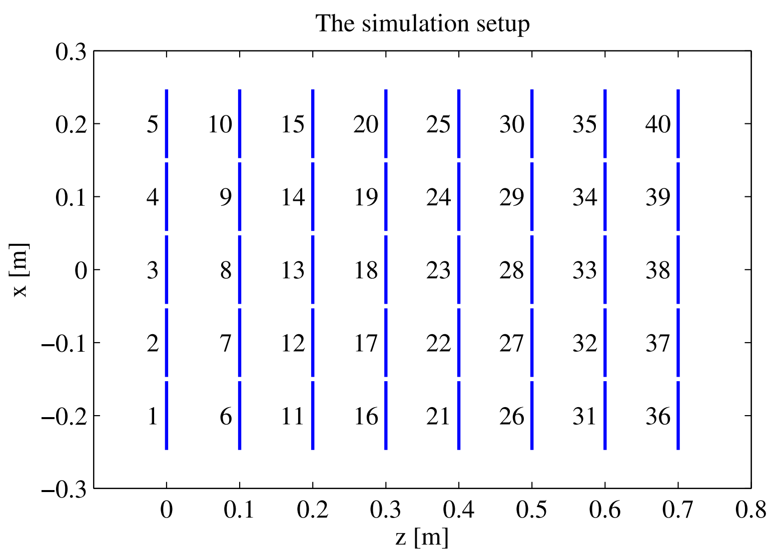
- If a detector k occurs several times in L_i, keep only the occurence with the smallest distance $d(k, i)$.

❑ The threshold $d_{\max}$ is the largest distance for which correlations are deemed to be significant.

❑ Needs to be tested and tuned on “real” tracks

Simulation experiments in a simplified setup

- Study of the basic properties of the method in a simple, small setup.
- Full covariance matrix is updated.
- Eight detector layers along the z-axis, with a spacing of 10 cm .
- In each layer, there is a row of five detector units, each $10 \times 10\,\mathrm{cm}^2$.



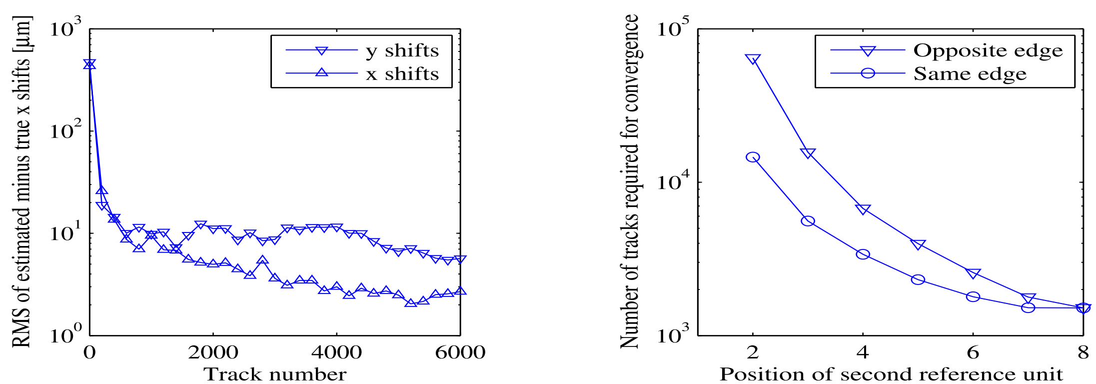
A schematic view of the simulation setup.

Simulation experiments in a simplified setup

- Straight tracks are simulated such that each track crosses all detector layers.
- The intersection points are smeared by a Gaussian resolution function.
- The standard deviation of the observation error is 50 μm both in x and in y.
- At least two detector units in different layers are required to fix the reference frame.
- All detector units apart from these two are misaligned by shifts in x and y.
- The shifts are generated randomly by drawing from a Gaussian distribution ten times as wide as the observation error.
- The positions of the reference units are fixed by giving them a very small prior uncertainty of the order of 0.1 μm.
- The prior uncertainty of the other units is set to 1 mm .

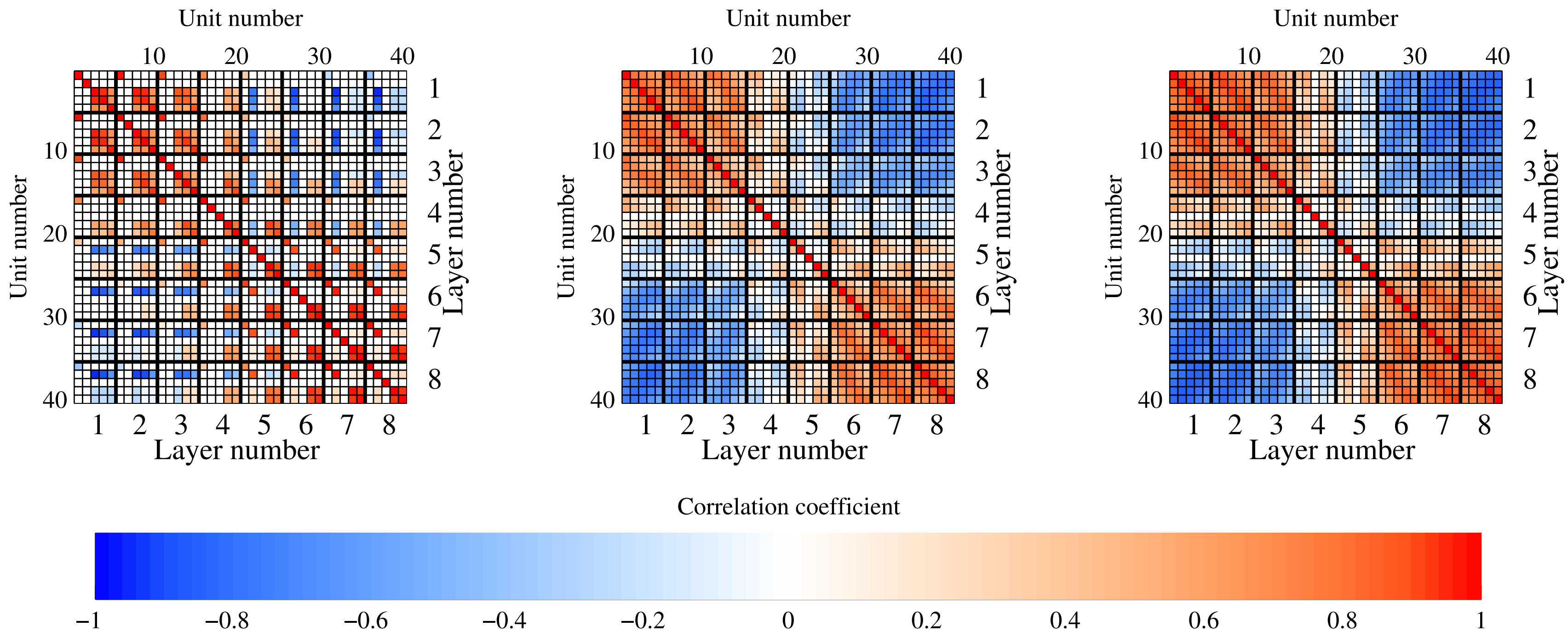
Simulation experiments in a simplified setup

- A quantitative assessment about the algorithm's precision can be made by computing the RMS of the difference $\boldsymbol{\delta}$ between true and estimated shifts.
- The speed of convergence is measured by the number of tracks required to bring the standard errors of all estimated shifts below a certain bound. In the following, we have used a bound of $10\,\mu$m.
- The number and relative position of the reference units has a large influence on the speed of convergence.



Left: An example of the evolution of the RMS of $\boldsymbol{\delta}$. **Right:** Number of tracks required for convergence as a function of the position of the second reference unit.

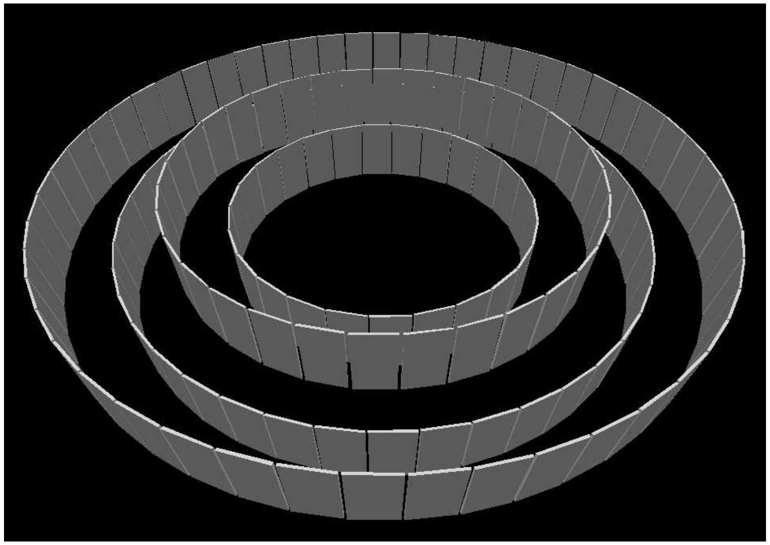
Simulation experiments in a simplified setup



Correlation matrix of the estimated x shifts, after 5 (left), 50 (centre) and 500 (right) tracks. The layers are separated by thick black lines. Units 18 and 23 are the reference units.

Results from the CMS Inner Tracker

- Study of the convergence and stability of the method in the CMS Inner Tracker using ORCA and its alignment interface.
- A wheel-like setup containing 156 modules from the Tracker Inner Barrel was used.



A schematic view of the (sub-)detector geometry.

Results from the CMS Inner Tracker

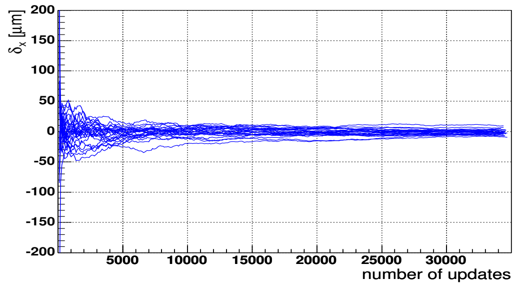
- Tracks of muons (μ^+, μ^-) in a homogeneous magnetic field (4T) were produced with a simplified fast simulation, simulated under the same hypotheses as used in the reconstruction (multiple scattering, energy loss, detector resolutions, etc.).
- The intersection points are smeared by a Gaussian resolution function (Pixels: x and y, Strips: only x).
- The standard deviations of the observation errors are according to their nominal values.
- Modules farther away from the interaction point are less frequently hit.

Results from the CMS Inner Tracker

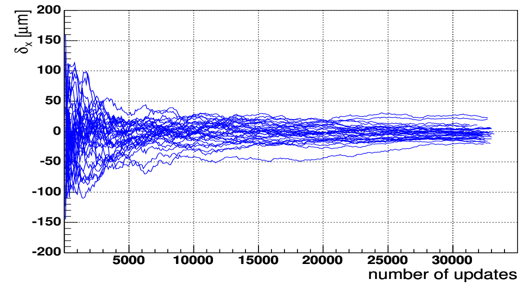
- The Pixel Detector is used as reference frame and is therefore not misaligned.
- All Silicon Strip Modules (1D as well as 2D) are misaligned by shifts in x and y direction.
- The shifts are generated randomly by drawing from a Gaussian distribution with $\sigma = 100\,\mu$m.
- The positions of the reference units are fixed by giving them a very small prior uncertainty of the order of 0.01 μm.
- The prior uncertainty of the other units is set to 0.5 mm in x and 0.5 cm in y;
- The concept of update lists has been applied here; the threshold of the update lists was set to $d_{\text{max}} = 6$.

Results from the CMS Inner Tracker

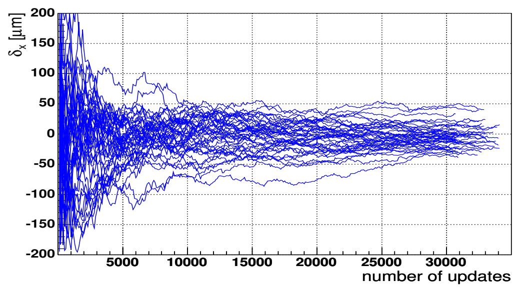
Layer 1



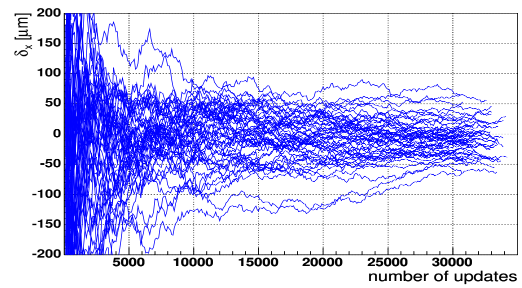
Layer 2



Layer 3

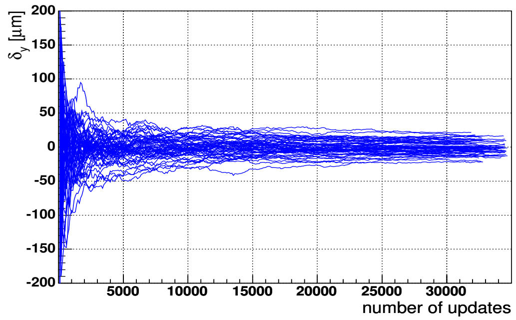


Layer 4

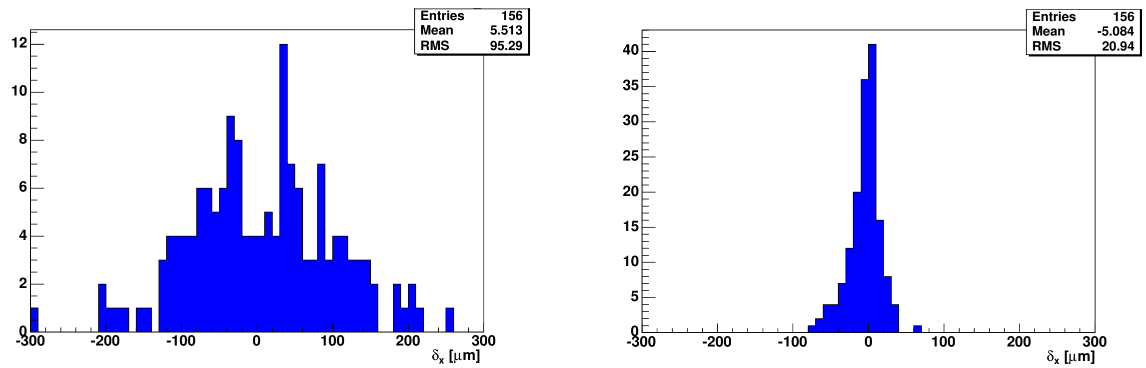


Convergence of estimates on the local x-shifts is quite good but depends on the the distance from the reference system.

Results from the CMS Inner Tracker

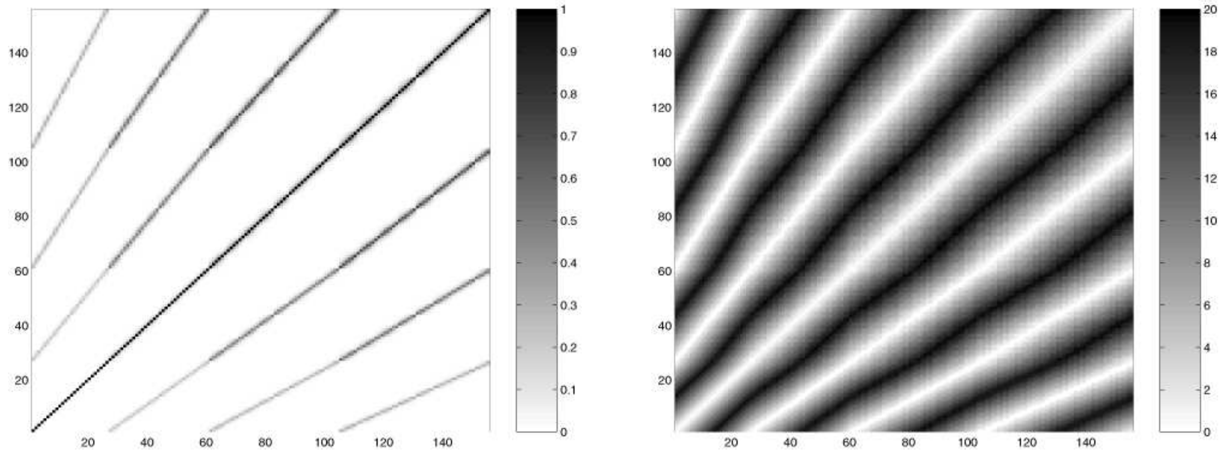


Convergence of estimates on the local y-shifts.



Distribution of the local x-shifts before alignment and after 100,000 processed tracks.

Results from the CMS Inner Tracker



Grayscale-coded visualization of the metrics (left) and the correlation matrix (right) for all modules in the wheel-like setup. Comparing these figures shows that the choice of $d(i,j) \leq 6$ doesn't exclude modules with relevant correlations during update. The modules are ordered by layer and increasing (global) polar angle and are indexed from 1 to 156.

$d_{\max}$	1	2	3	4	5	6
σ [μm]	24.75	21.38	20.97	20.95	20.94	20.94
T [s]	472	604	723	936	1152	1319

Precision and computing time as function of $d_{\max}$.

Conclusions and Outlook

Although the method has been shown to work in principle, clearly more development, testing and tuning is required to meet the challenge of a full alignment of the CMS Tracker.

- The distance cut d_{max} has to be optimized; this is particularly important in view of the influence of the maximal distance on the computation time.
- The scaleability of the algorithm has to be studied on a larger number of modules.
- The simplified fast track simulation has to be replaced by a full simulation.
- In view of the slower convergence for modules in the outer layers, alternatives to using single tracks are desirable. Using constrained muon pairs from Z- or J/ψ-decays is an auspicious possibility.