Available on CMS information server

CMS NOTE 2002/008

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

March 1, 2002

Estimation of Alignment Parameters, using the Kalman Filter with annealing

R. Frühwirth

Institute of High Energy Physics, Vienna, Austria

T. Todorov

IN2P3, Strasbourg, France

M. Winkler

CERN, Geneva, Switzerland

Abstract

Detector alignment is an essential step in the track reconstruction and analysis process. Alignment with tracks is one possible strategy to estimate positions and orientations of components of a track detector. We present a method for the estimation of alignment constants during track reconstruction, in parallel with the usual track parameters. The formalism is related to the standard Kalman Filter and uses annealing in order to avoid suboptimal solutions. The algorithm has been implemented in the Object oriented Reconstruction for Cms Analysis (ORCA) framework and has been tested with a simulated test-beam like setup for silicon detectors.

1 Introduction

Alignment of tracking detectors is an essential step in the task of track reconstruction. Without proper alignment it is impossible to reach the ultimate position and momentum resolution. Besides finding the alignment of the tracking detectors it is also important to constantly monitor the alignment and to update the alignment constants whenever required.

In the CMS Tracker, this task is particularly challenging because of the large number of independent silicon sensors, about 20000, and their high resolution, between 10 and 40 microns. To exploit fully the resolution of the tracker the sensors must be aligned to a precision significantly better than their intrinsic resolution. This precision can only be achieved in an alignment procedure based on charged tracks, since the expected precision from the mechanical mounting and laser beam alignment is significantly worse than the intrinsic sensor resolutions.

The large number of alignment parameters implies a very large number of tracks needed to align the tracker. Even at nominal LHC luminosity and the expected trigger rate of the order of 10 Hz for high P_t muons the time to accumulate the necessary statistics is of the order of a week. It is therefore important to make efficient use of the track information in the alignment algorithm.

In this note we present a unified framework for the simultaneous unbiased estimation of the alignment parameters of several detectors, which uses fully the track information. The method is a straightforward extension of the standard Kalman Filter [1]. Since the dimension of the parameter space can be large the occasional convergence to local minima cannot be excluded. We will present below strong empirical evidence that this actually occurs. We solve this problem by introducing annealing, i.e. by gradually turning on the weights of the observations in the course of the estimation process. The resulting algorithm closely resembles the Deterministic Annealing Filter invented for robust track reconstruction in the presence of noise and ambiguities [2].

For the sake of simplicity we have studied the proposed alignment procedure using a simple track and detector model, neglecting material effects and pattern recognition issues. It should be stressed, however, that the method is completely general and can be applied with any kind of track and detector model.

2 Formalism

The problem of aligning a track detector can be stated in the following generic way. The observation $\mathbf{m}$ recorded by the detector depends both on the vector of track parameters (track state) $\mathbf{p}$ of the track crossing the detector and on the vector of alignment parameters (alignment state) $\mathbf{a}$. The alignment state may contain both shifts and rotations. The dependence can be written down in a generalized measurement equation:

$$\mathbf{m} = \mathbf{f}(\mathbf{p}, \mathbf{a}) + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon}$ is the vector of measurement errors. The covariance matrix V of $\boldsymbol{\epsilon}$ is assumed to be known.

The function $\mathbf{f}$ may be linear or non-linear in either argument. In the presence of rotations in the alignment state, $\mathbf{f}$ is non-linear in $\mathbf{a}$. While the track parameters $\mathbf{p}$ are different from event to event, the alignment parameters $\mathbf{a}$ of a detector are common for all tracks. In the linear approximation the measurement equation can be written as:

$$\mathbf{m} = \mathbf{c} + \mathbf{H}\mathbf{p} + \mathbf{D}\mathbf{a} + \boldsymbol{\epsilon},$$

where $\mathbf{H}$ and $\mathbf{D}$ are the Jacobians:

$$\mathbf{H} = \frac{\partial \mathbf{f}}{\partial \mathbf{p}}(\mathbf{p}_0, \mathbf{a}_0)$$
$$\mathbf{D} = \frac{\partial \mathbf{f}}{\partial \mathbf{a}}(\mathbf{p}_0, \mathbf{a}_0).$$

We now assume that there is a predicted track state $\mathbf{p}_0$ along with its covariance matrix $\mathbf{C}_0$, as well as a predicted alignment state $\mathbf{a}_0$ along with its covariance matrix $\mathbf{E}_0$. The recipe for updating both the track state and the alignment state can be derived in a manner which is analogous to the derivation of the standard Kalman Filter. The resulting update formulas are:

$$\mathbf{a}_1 = \mathbf{a}_0 + \mathbf{E}_0\mathbf{D}^{\mathrm{T}}\mathbf{W}\left[\mathbf{m} - \mathbf{f}(\mathbf{p}_0, \mathbf{a}_0)\right]$$
$$\mathbf{E}_1 = \mathbf{E}_0 - \mathbf{E}_0\mathbf{D}^{\mathrm{T}}\mathbf{W}\mathbf{D}\mathbf{E}_0$$

for the alignment state plus covariance matrix, and

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{p}_0 + \mathbf{C}_0\mathbf{H}^{\mathrm{T}}\mathbf{W}\left[\mathbf{m} - \mathbf{f}(\mathbf{p}_0, \mathbf{a}_0)\right] \\ \mathbf{C}_1 &= \mathbf{C}_0 - \mathbf{C}_0\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H}\mathbf{C}_0\end{aligned}$$

for the track state plus covariance matrix. The following auxiliary matrix needs to be computed:

$$\mathbf{W} = \left[\mathbf{V} + \mathbf{H}\mathbf{C}_0\mathbf{H}^{\mathrm{T}} + \mathbf{D}\mathbf{E}_0\mathbf{D}^{\mathrm{T}}\right]^{-1}$$

Note that the update formulas for the track and alignment parameters decouple into two separate ones. We will refer to this again in section 5. In a similar way it is also possible to decouple the estimation of shift and rotation parameters, although this is tantamount to neglecting the correlations between the two (see Appendix).

Information about $\mathbf{a}$ is accumulated continually, increasing after each track. In addition, at each step in the alignment procedure the full covariance matrix $\mathbf{E}$ of $\mathbf{a}$ is known, which can be used as a criterion for stopping the alignment procedure. The convergence of $\mathbf{a}$ clearly depends on the precision of its starting value. It can be taken from mechanical measurements, from laser beam alignment, or from a previous alignment with tracks.

Smoothing is done in the usual way, by running two filters in opposite directions and combining the results on each detector surface. Both in the forward and in the backward filter the alignment states are not updated, but the knowledge of the current alignment state is used for the update of the track state vector (see above). The alignment parameters are updated in the smoothing step, which combines the predictions from the forward and the backward filter with the measurements. This means that at each update of $\mathbf{a}$ both the full information about the track and the current alignment of all other detectors enter into the update.

The predicted alignment states contain the current knowledge of the alignment parameters, based on the tracks already processed. The predicted track states, on the other hand, are based on the observations form the current track only, but depend also on the current alignment. As long as the alignment parameters are not known to sufficient precision the predicted track states are therefore biased, especially if several detectors are aligned simultaneously. As a consequence, it may happen that alignment parameters converge to a suboptimal solution (local minimum) of the alignment problem. This can be prevented by introducing annealing in a manner similar to the Deterministic Annealing Filter developed for robust track reconstruction [2]. This means that the observations in all detectors to be aligned are downweighted by a large factor in the beginning. This factor is then gradually decreased, until it reaches 1 after a prescribed number of tracks. Below we shall present strong empirical evidence that this annealing procedure solves entirely the problem of convergence to local minima.

3 OO design and implementation for a test-beam like setup

The object oriented design for the implementation of the track and alignment reconstruction is strongly related to the existing implementation of the track reconstruction environment for the CMS Tracker in ORCA [3]. Wherever possible, existing classes (e.g. the RecHit) are used directly, or provided base class interfaces are implemented (e.g. the DetLayer or the DetUnit). Therefore the alignment specific algorithms are isolated in specific classes for the minimization (fitter, smoother), similar to the way it is done in track reconstruction only for the CMS Tracker.

3.1 Overview

The task of track reconstruction (with or without alignment) can be split in four major components:

- a tracker geometry
- an alignment geometry
- tracker hits and detector readout
- a track reconstructor

This global layout (modularity) allows that one can change the track reconstructor while keeping the geometry the same or vice versa. Alternatively, one might want to study a given geometry first with simulated hits and then switch to real test beam data, only by changing the way hits are produced.

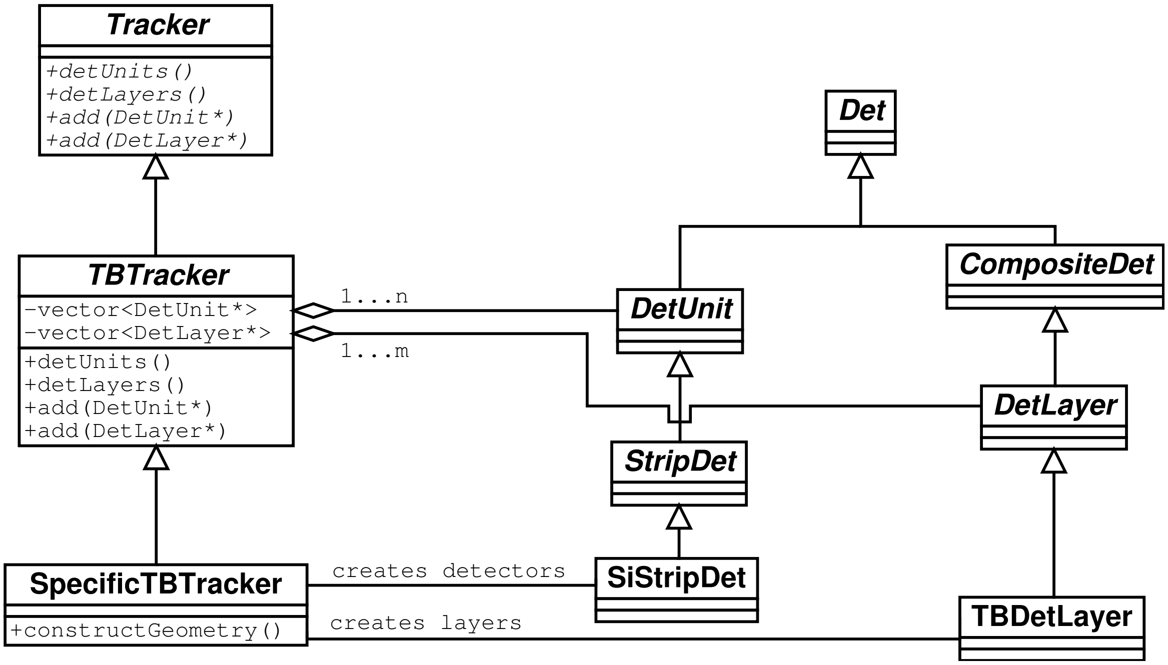


Figure 1: Object model of the Tracker Geometry. The TBTracker class implements the interface of the abstract base class Tracker and keeps control of the whole geometry. A SpecificTBTracker reads configuration files and constructs DetLayers and DetUnits which are then added to the TBTracker.

3.2 Tracker geometry

A common base class Tracker defines an interface for accessing specific tracker geometries (CMS or test beam) in a unique way. The requirement of a tracker geometry is to provide access to its sub-components, enabling the reconstruction of tracks. Therefore the Tracker base class interface prescribes a method that allows the access to the DetLayers. A second method allows the access to all sensitive volumes in a tracker — the DetUnits. In the case of a test beam a DetUnit most likely represents also the surface of a DetLayer. The object model with its basic components can be seen in Figure 1.

3.3 Alignment geometry

The geometrical structure used for track finding (Tracker geometry) is different from the geometrical structure for aligning components. An alignment geometry has to provide access to the specific alignment structures and, in the case of a demand, act on the current Tracker geometry by moving and rotating parts. This mechanism can be used for both aligning and misaligning, if one wants to introduce misalignment into an existing Tracker geometry.

In the case of our simple test beam example the alignment components coincide with the ones used in the track reconstruction — each DetUnit has its associated AlignableDetUnit and thus can be aligned. The current AlignmentStates (AlignmentParameters and AlignmentErrors) are stored within the AlignableDetUnit. In addition, the alignment geometry has to know about the reference detectors and the alignable detectors (see Figure 2).

3.4 Tracker hits and detector readout

Each DetUnit is connected to a detector readout. It reads out the Digis (digi = signal of one strip or pixel) of a detector event by event. These Digis are then used by a Clusterizer in order to create the RecHits — the fundamental measurements needed by the track reconstruction. However, when simulating tracks one does not have a “real” readout and one needs to simulate the detector response. In ORCA this is done by a Digitizer which creates Digis from the information it gets from simulated hits (the SimHits) and thus fills the readout. In addition, the readouts have to be reset at the beginning of each event, cleaning up the Digis and SimHits from the previous event. In our simulation experiment we use a HitPopulatorFromGun in order to create SimHits (see Figure 3).

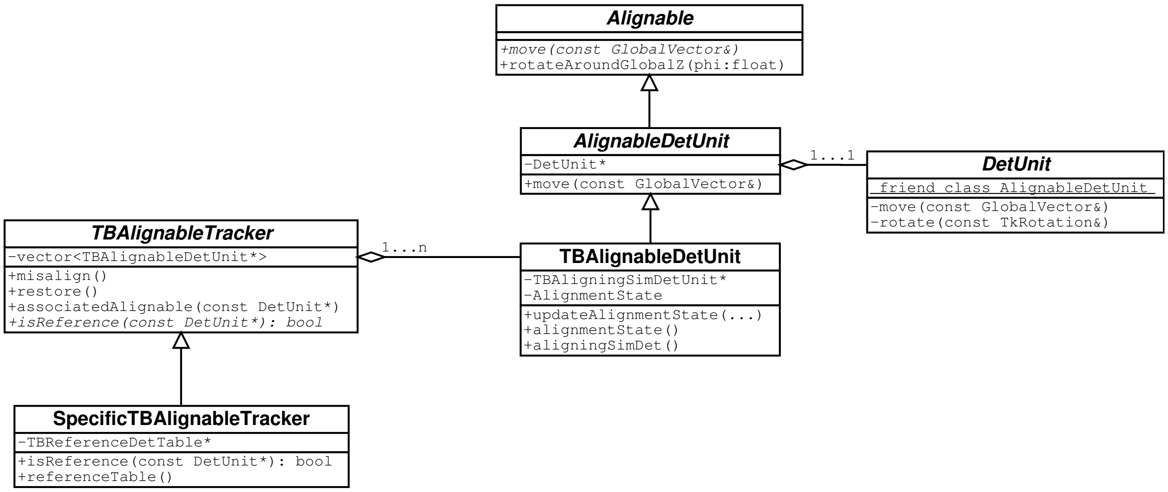

Alignable
+move(const GlobalVector&)
+rotateAroundGlobalZ(phi:float)
AlignableDetUnit
-DetUnit*
+move(const GlobalVector&)
1...1
DetUnit
friend class AlignableDetUnit
-move(const GlobalVector&)
-rotate(const TkRotation&)
TBAlignableTracker
-vector<TBAlignableDetUnit*>
+misalign()
+restore()
+associatedAlignable(const DetUnit*)
+isReference(const DetUnit*): bool
1...n
TBAlignableDetUnit
-TBAligningSimDetUnit*
-AlignmentState
+updateAlignmentState(...)
+alignmentState()
+aligningSimDet()
SpecificTBAlignableTracker
-TBReferenceDetTable*
+isReference(const DetUnit*): bool
+referenceTable()

Figure 2: Object model of the alignment geometry. In the case of our test beam like setup the alignment geometry is simple, as the structure of the track reconstruction geometry matches the structure of the alignment geometry.

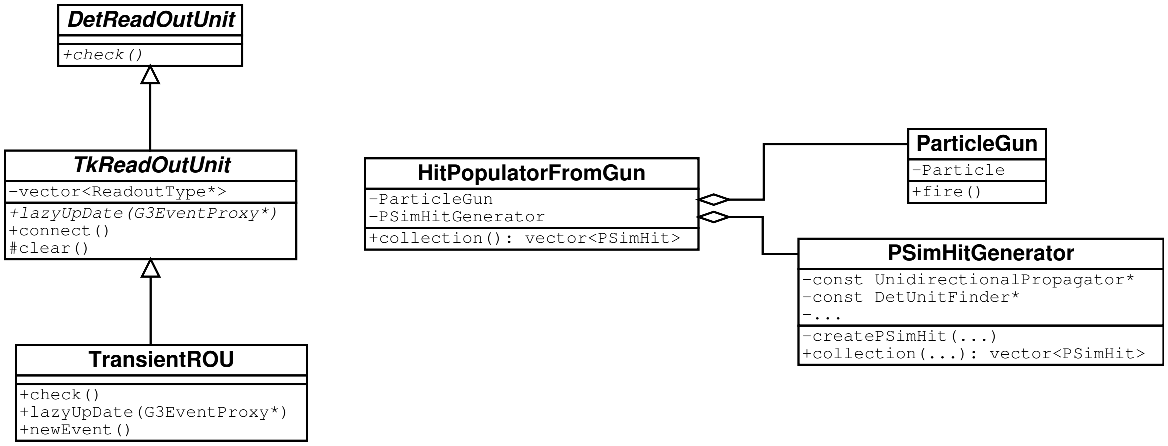


Figure 3: Class collaboration diagram for the detector readout and the HitPopulatorFromGun. In the case of a simulation, the Digis (signal per strip) of a detector have to be created from a detector response simulation. This is done by a Digitizer, which creates Digis based on the information provided by SimHits.

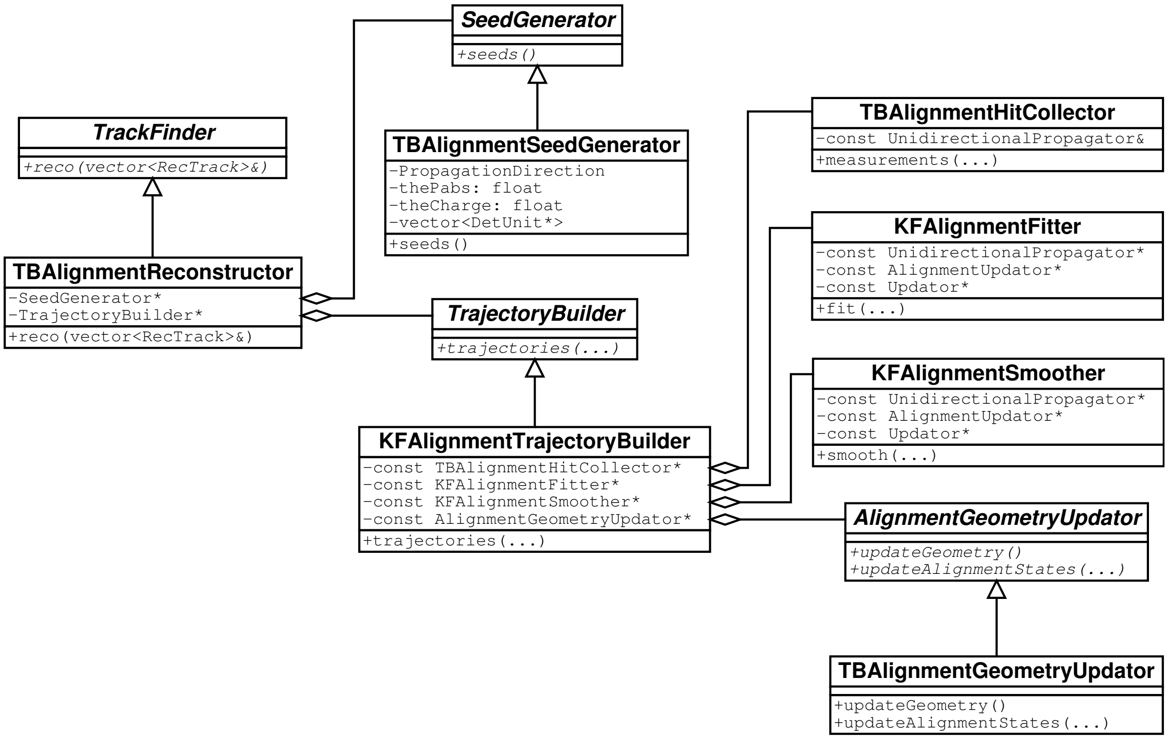


Figure 4: Object model of the track reconstructor. The most important components are the SeedGenerator and the TrajectoryBuilder.

3.5 Track reconstructor with alignment

It is required to reconstruct tracks on the base of the available RecHits taking into account misalignment of detectors. Therefore the reconstruction of tracks together with the estimation of alignment parameters is performed by a dedicated high level object. The track reconstructor contains a track SeedGenerator and a TrajectoryBuilder with alignment. The SeedGenerator creates initial starting values for potential track candidates, e.g. by combining pairs of RecHits. The AlignmentTrajectoryBuilder creates a Trajectory starting from a seed and estimates both track and alignment parameters. An overview is given in Figure 4.

4 Results of a simulation experiment

We have tested and verified the algorithm in a simulation experiment. The setup is a somewhat simplified model of a typical test beam configuration.

4.1 The detector model

We have used a layout with five consecutive pairs of x-y silicon strip detectors along the global z-(beam-)axis. Each detector has a pitch of $60\,\mu$m and a strip length of 6 cm. The RecHits are gaussian smeared SimHits with a standard deviation corresponding to the pitch of the detectors (pitch/$\sqrt{}12$). The observation along the strip is always set to 0 in the local frame of the detector (center of the strip); its standard deviation is set to the strip length divided by $\sqrt{}12$.

The first and the last pair of detectors define the reference frame, assuming that their true positions and orientations are known. The three intermediate pairs of detectors are misaligned (see Figure 5).

4.2 Global and local frame

The global coordinate system or frame has been chosen such that the global z-axis points into the direction of the beam (see Figure 5). The definition of the local coordinate frame is shown in Figure 6. The origin of the local frame is set in the middle of the detector volume, at half-width, half-length and half-depth of the detector. The local z-axis is perpendicular to the detector surface, with the strips sitting on the positive z-side. The local x- and

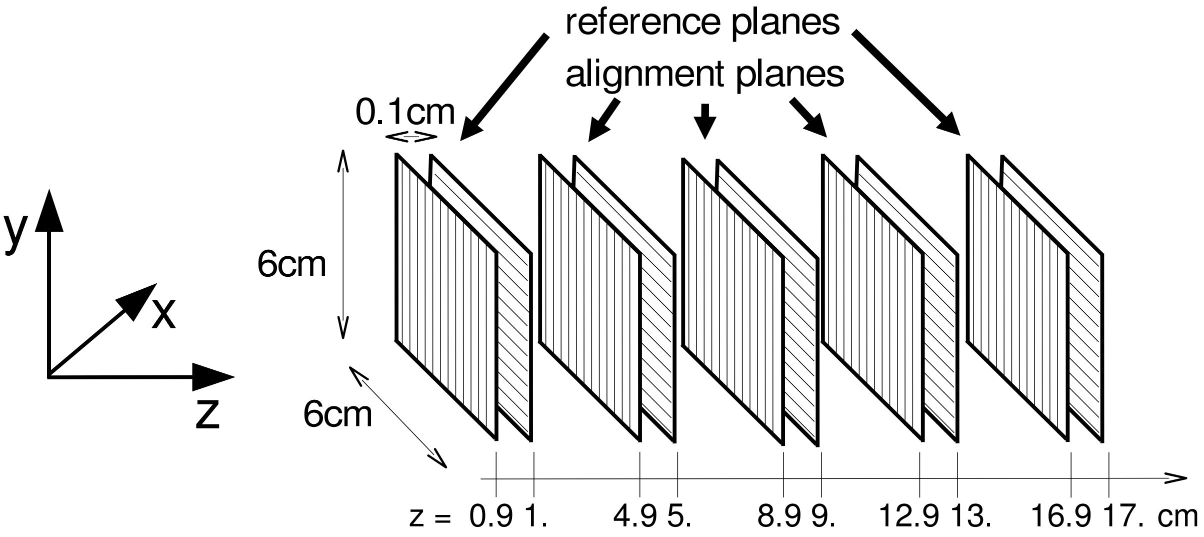


Figure 5: Layout of the simulated test beam tracker. The layout consists of five consecutive pairs of x-y strip detector planes with 4 cm distance between pairs and 1 mm distance within a pair. The strip detectors have 1000 strips each, with a pitch of 60 μm (= 6 cm total width of the detector) and a strip length of 6 cm. The first pair at $z = 0.9$ cm/1 cm and the last pair at $z = 16.9$ cm/17 cm define the reference frame with respect to which the three inner pairs are misaligned in the simulation and then aligned in the reconstruction. The particle beam is generated at $z = 0$ and distributed uniformly over the surface.

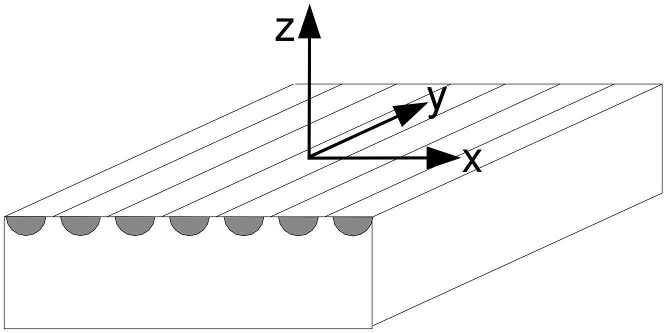


Figure 6: Definition of the local coordinate frame for a strip detector.

y-axis are both parallel to the detector surface, with local x being perpendicular to the strips, local y parallel to the strips.

In the case of our simulation experiment, global and local z coincide.

4.3 Track simulation and track model

The beam is generated at $z = 0$ covering the full 6 cm × 6 cm area of sensitive detector volumes. Positively charged muons with a momentum of 100 GeV are simulated. The beam direction is defined by two angles, θ and ϕ, where

$$\tan\theta = \frac{p_{\mathrm{T}}}{p_z} \quad \text{and} \quad \tan\phi = \frac{p_y}{p_x}$$

θ has been generated in the range $0 < \theta < 0.014$ in order to avoid tracks being perfectly parallel with z, and ϕ has been generated in the range $0 \leq \phi < 2\,\pi$.

In the absence of a magnetic field the track model is very simple. We have used the following set of track parameters for a fixed z:

$$\mathbf{p} = \begin{pmatrix} t_x \\ t_y \\ x \\ y \end{pmatrix}, \quad \text{with} \quad t_x = \frac{dx}{dz} = \frac{p_x}{p_z} = \tan\theta\cos\phi,\ t_y = \frac{dy}{dz} = \frac{p_y}{p_z} = \tan\theta\sin\phi$$

Track propagation from $z = z_1$ to $z = z_2$ is then described by a simple matrix:

$$\mathbf{p}|_{z=z_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ z_2 - z_1 & 0 & 1 & 0 \\ 0 & z_2 - z_1 & 0 & 1 \end{pmatrix} \mathbf{p}|_{z=z_1}$$

4.4 The misalignment model

Misalignment can be seen as the difference between the assumed ideal position and orientation of a given detector with respect to its true parameters in global space. Therefore each transformation from the local detector coordinate system into global space and vice versa has errors if the assumed ideal parameters are not the true ones. We introduce misalignment by moving and rotating the detector planes before we shoot particles, and we restore the assumed ideal (but now wrong) positions and rotations before we start reconstructing tracks. While the local position of a hit on a detector stays unchanged (the strip of the hit does not change), its global position will be wrong as the transformation from the local to the global frame is wrong. The task of the alignment procedure to estimate the relative offset and rotation with respect to the assumed position and rotation.

Out of the six possible alignment parameters — three shifts and three rotations — we simulate and estimate two shifts and one rotation in the plane of the detector (2-dimensional space), those three being the ones to which track reconstruction is most sensitive. Figure 7 illustrates the two shifts and the rotation. The transformation of a point $\mathbf{m}(x, y) \to \mathbf{m}'(x', y')$ from the global frame into the global misaligned frame is therefore:

$$\begin{aligned} \mathbf{m}'(x', y') &= R(\Delta\phi)(\mathbf{m}(x, y) - \mathbf{s}_{\text{global}}) \\ &= R(\Delta\phi) \left(\mathbf{m}(x, y) - \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right) \end{aligned}$$

with

$$\mathbf{s}_{\text{global}} = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

being the shift in $\Delta x, \Delta y$ and $R(\Delta\phi)$ a small rotation around z.

The following ranges for the shifts Δx and Δy and the rotation $\Delta\phi$ in the global frame have been chosen for the simulation:

shift in global x-direction:	$-0.2\,\text{cm} < \Delta x < 0.2\,\text{cm}$
shift in global y-direction:	$-0.2\,\text{cm} < \Delta y < 0.2\,\text{cm}$
rotation around global z-axis:	$-0.02\,\text{rad} < \Delta\phi < 0.02\,\text{rad}$

For each run the alignment parameters are randomly generated assuming a flat distribution in the above ranges.

It should be noted that instead of estimating $(\Delta x, \Delta y, \Delta\phi)$ directly we have chosen a slightly different set of alignment parameters:

$$\mathbf{a} = \begin{pmatrix} \Delta x' \\ \Delta y' \\ \Delta\phi \end{pmatrix}, \quad \text{with} \begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix} = R(\Delta\phi) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

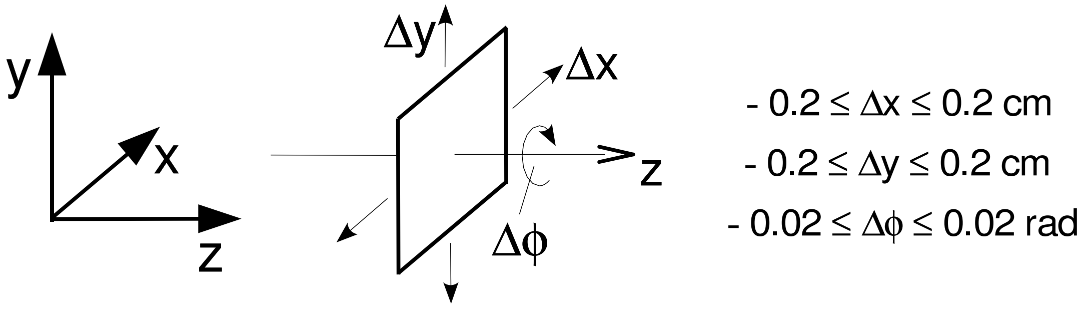


Figure 7: Illustration of shifting and rotating the planes for misalignment.

where we estimate the shift rotated by $\Delta\phi$ rather than the simulated global shift. In the following we refer to

$$\mathbf{s}' = \begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix}$$

as the global rotated shift. The reason for this choice will be explained after the derivation of the Jacobians.

The measurement model can then be formulated in the most general way:

$$\begin{aligned} \mathbf{m}_{\text{local}} &= \mathbf{f}(\mathbf{p}, \mathbf{a}) + \boldsymbol{\epsilon} \\ &= \mathbf{T}(\phi)_{\text{global}\to\text{local}} \mathbf{R}(\Delta\phi) \left[\mathbf{P}\mathbf{p}_{\text{global}} - \mathbf{s}_{\text{global}} \right] + \boldsymbol{\epsilon} \\ &= \mathbf{R}(\Delta\phi)\mathbf{P}\mathbf{p}_{\text{local}} - \mathbf{T}(\phi)\mathbf{s}' + \boldsymbol{\epsilon} \end{aligned}$$

where $\mathbf{m}$ is the observation (hit) in the local frame, $\mathbf{p}$ is the track state ($\mathbf{p}_{\text{global}}$ in the global and $\mathbf{p}_{\text{local}}$ in the local frame), $\mathbf{s}'$ is the global rotated shift, and $\boldsymbol{\epsilon}$ is the measurement error in the local frame. We will use the last expression of our measurement equation for deriving the essential Jacobians.

With our choice of track parameters, the matrix $\mathbf{P}$ is a simple projection:

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathbf{T}$ is the transformation from the global to the "ideal" local frame; in our case it is either the identity or a rotation by 90 degrees around the global z-axis. $\mathbf{R}$ is a small rotation around the z-axis in addition to the "ideal" one:

$$\mathbf{R} = \begin{pmatrix} \cos(\Delta\phi) & \sin(\Delta\phi) \\ -\sin(\Delta\phi) & \cos(\Delta\phi) \end{pmatrix}$$

4.5 The estimation procedure

For the estimation procedure we assume that the assignment of hits to tracks has already been done.

Track fitting is performed by a Kalman Filter plus smoother in order to get optimal predicted track states $\mathbf{p}_0$ in all detectors. The current state $\mathbf{a}_0$ of the alignment parameters in all detectors is used in the fit of the track. Note that the alignment parameters are estimated independently in all detectors. In order to avoid cluttering the notation the index to the detector is dropped.

Using the misalignment model described above, the derivative matrices $\mathbf{H}$ and $\mathbf{D}$ can be computed without difficulty in any particular detector:

$$\begin{aligned} \mathbf{H} &= \frac{\partial \mathbf{f}}{\partial \mathbf{p}}(\mathbf{p}_0, \mathbf{a}_0) = \mathbf{R}\mathbf{P} \\ \mathbf{D} &= \frac{\partial \mathbf{f}}{\partial \mathbf{a}}(\mathbf{p}_0, \mathbf{a}_0) = (-\mathbf{T} \mid \mathbf{R}'\mathbf{P}\mathbf{p}_0) \end{aligned}$$

where $\mathbf{R}'$ ist the derivative of the current rotation matrix:

$$\mathbf{R}' = \begin{pmatrix} -\sin(\Delta\phi_0) & \cos(\Delta\phi_0) \\ -\cos(\Delta\phi_0) & -\sin(\Delta\phi_0) \end{pmatrix}$$

and $\mathbf{p}_0$ is the predicted track state in the local system.

It can be seen that the third column of $\mathbf{D}$ depends on the rotation angle $\Delta\phi$, thus making the equation non-linear in $\Delta\phi$. By choosing the rotated global shifts $\Delta x'$, $\Delta y'$ as alignment parameters, we avoid that also the other elements of $\mathbf{D}$ depend on $\Delta\phi$. In order to simplify the notation the rotated global shifts will henceforth be denoted by Δx and Δy.

From this we compute the matrix $\mathbf{W}$, using the current annealing factor $\alpha^{(k)}$ which depends on the current track number k:

$$\mathbf{W} = \left[\alpha^{(k)}\mathbf{V} + \mathbf{H}\mathbf{C}_0\mathbf{H}^{\mathrm{T}} + \mathbf{D}\mathbf{E}_0\mathbf{D}^{\mathrm{T}} \right]^{-1}$$

The update of the track state and of the alignment state then proceeds as described in section 2.

We have investigated three annealing schedules:

Schedule	$\alpha^{(1)}$	$\alpha^{(n)}$	$\alpha^{(k)}$
A	1	1	1
B	10000	10000	10000
C	10000	1	$10000^{\frac{n-k}{n-1}}$

Schedule A is the standard Kalman Filter, schedule B is a Kalman Filter which effectively uses only the predictions of the tracks from the reference detectors, and schedule C is a Deterministic Annealing Filter [2] with a geometric cooling schedule. In all cases the initial values of the alignment parameters have been taken to be zero with sufficiently large errors.

4.6 Sensitivity and convergence of alignment parameters

Not all alignment parameters can be reliably determined as not all of the observations are equally affected by shifting and rotating the detector. The sensitivity of the alignment parameters with respect to the observations is given by the matrix $\mathbf{E}_0\mathbf{D}^{\mathrm{T}}\mathbf{W}$. The last factor $\mathbf{W}$ is dominated by the information content of the observations. Therefore in a detector measuring only x a shift Δy in y cannot be estimated very reliably because the sensitivity is dominated by the information content of the y-measurement. In our case this information content is smaller by a factor of 1 million [square of (length /pitch)] than the information content of the x-measurement. It should be noted, however, that this effect is irrelevant in track reconstruction as long as the y-coordinate of the prediction is sufficiently precise so that the precision of the x-coordinate is not spoiled by the rotation correction.

The different sensitivities of the alignment parameters are clearly visible if their evolution is plotted as a function of the number of tracks. As an example, Figure 8 shows the development of Δx, Δy, and $\Delta\phi$ in an x-detector, for a run with 2000 tracks. The behaviour depends on the annealing schedule. Convergence of Δx is satisfactory with all schedules, although with schedule B it is somewhat slower. Estimates of Δy are biased with all schedules. The differences between the three schedules can be seen most clearly in the convergence of $\Delta\phi$. Schedule A converges very quickly, but is off by about 0.2 mrad. Schedule B converges much more slowly and is off by 0.4 mrad after 2000 tracks. Schedule C is clearly the best, being off by less than $20\,\mu$rad after 2000 tracks in this particular run.

4.7 Precision of the estimated alignment parameters

Due to statistical fluctuations a single run cannot determine the absolute performance of a single schedule. To this end we have generated 2000 runs with random misalignment configurations and performing alignment over 2000 tracks for each run, which means that a total amount of 4 million tracks is generated and reconstructed per annealing schedule. The resulting histograms of residuals for schedule C are shown in Figure 9.

Figure 10 shows the standard deviations of the residuals (estimated alignment parameters minus true ones) for all annealing schedules. The shifts are shown only for the precise coordinates (Δx in x-detectors, Δy in y-detectors). Schedule A suffers from occasional convergence to local minima, whereas schedule B only uses information from the reference detectors. Schedule C (annealing) gives the best results in all cases. With runs of 2000 tracks each, the standard deviations of the shifts are about $0.6\,\mu$m, and the standard deviations of the rotations angles are below $50\,\mu$rad.

Finally, we have checked whether the computed errors on the alignment parameters correspond to the actual spread around the true values. The resulting histograms of standardized residuals (pulls) for schedule C are shown in Figure 11.

Figure 12 shows the standard deviations of the pulls for all annealing schedules. With schedule C, the standard deviations are indeed reasonably close to 1, whereas they are far too large with schedule A and far too small with schedule B. The mean values are compatible with 0 in all cases.

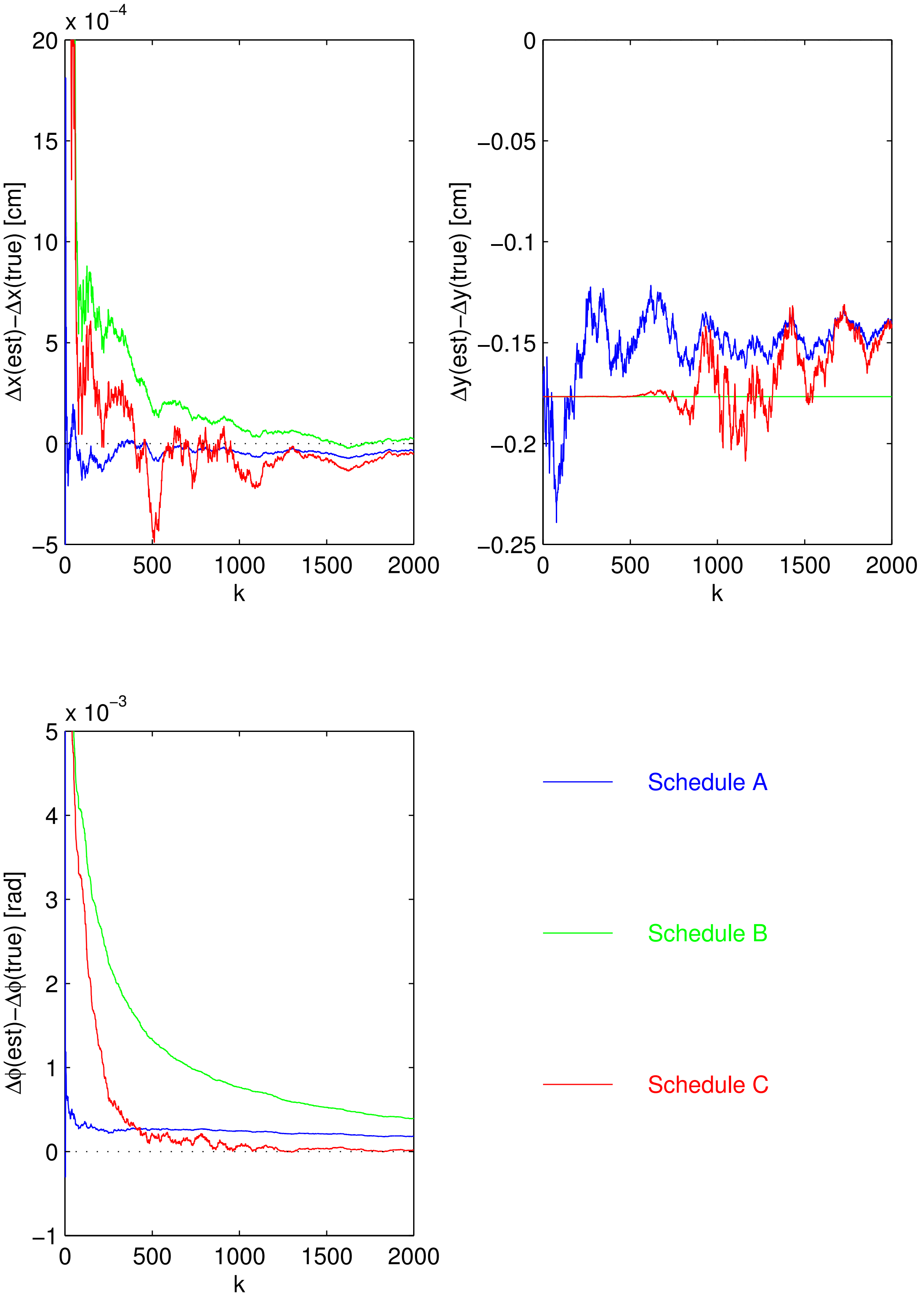


Figure 8: Development of estimated alignment parameters as a function of the track number for an x-detector. While convergence of Δx (shift in precise coordinate, scale in micron) is satisfactory for all three schedules, there is no convergence for Δy (shift in unprecise coordinate, scale in mm). The convergence of $\Delta\phi$ (scale in mrad) is best for schedule C.

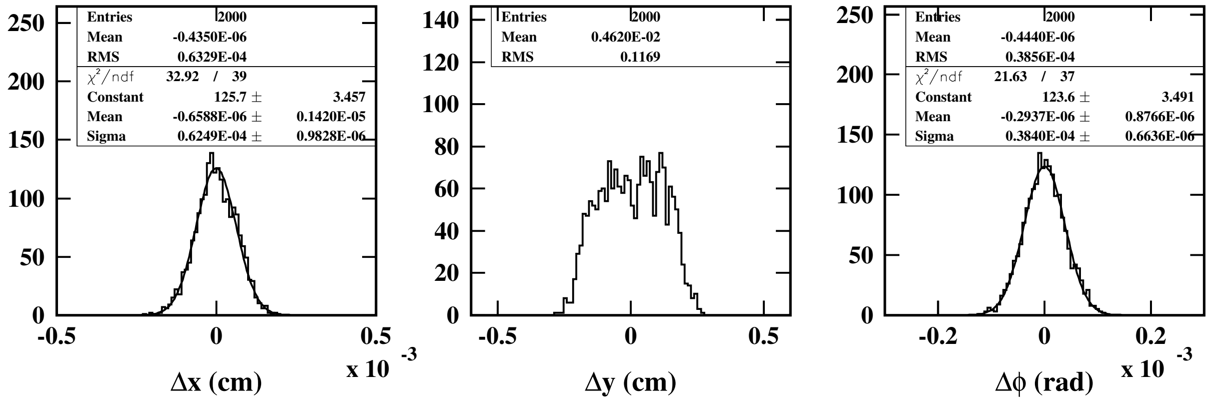


Figure 9: Residual histograms of the three alignment parameters for an x-detector and for annealing schedule C. As one can see, the shift along the unprecise coordinate cannot be estimated reliably.

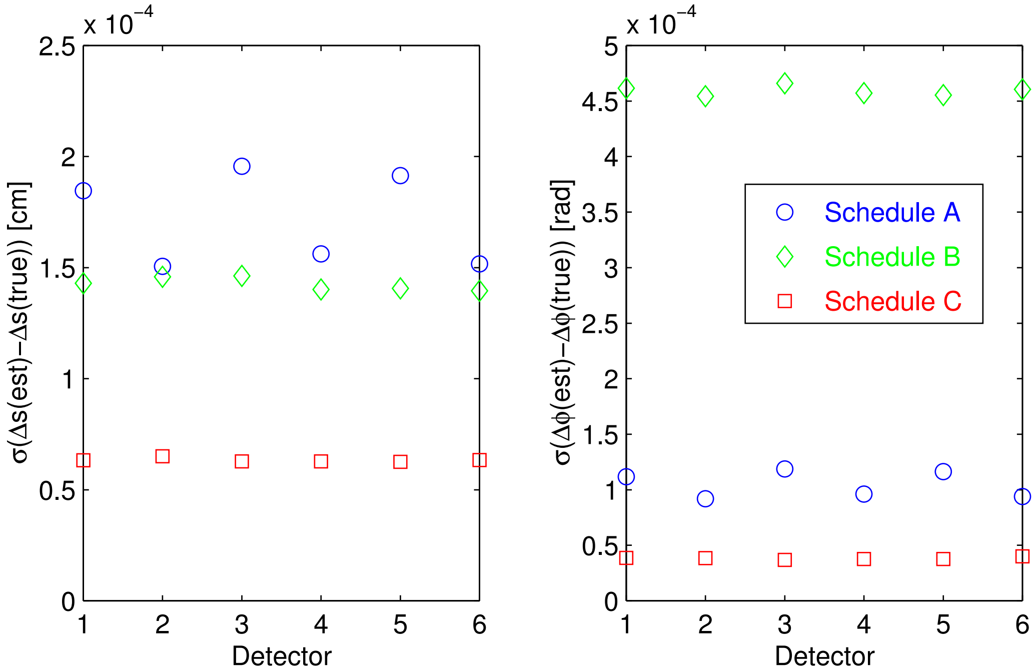


Figure 10: Standard deviation of the residuals (estimated alignment parameters minus true ones). Left hand side: shift in precise coordinate, right hand side: angle of rotation. The scale of the residuals of the shifts is in micron. The shifts are of the order of about $0.6\,\mu$m for schedule C, and 1.5–$2\,\mu$m for schedule A and B. The scale of the residuals of the rotation angle $\Delta\phi$ is tenths of a millirad. Schedule C achieves resolutions of about $50\,\mu$rad, schedule A about $100\,\mu$rad and schedule B about $450\,\mu$rad.

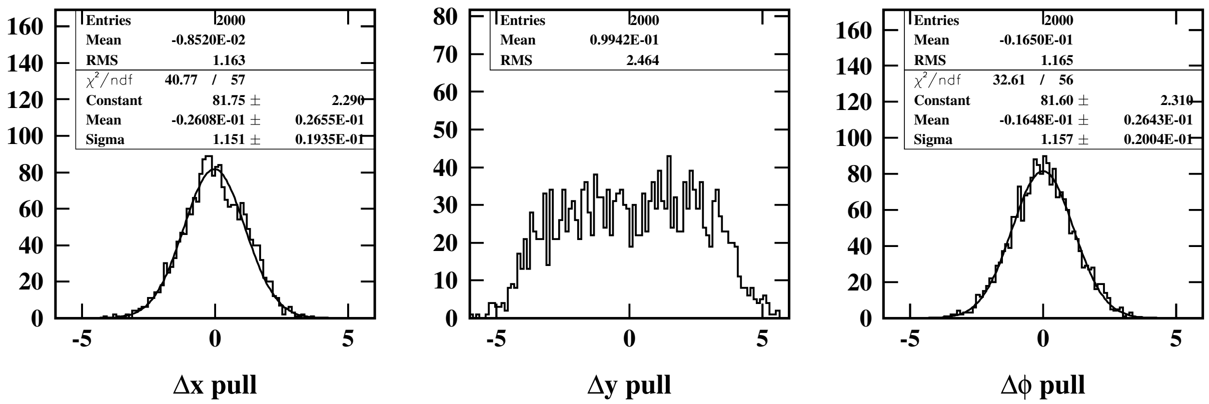


Figure 11: Pull histograms of the three alignment parameters for an x-detector and for annealing schedule C. The standard deviation of the pulls is reasonably close to 1 for the shift Δx and for the rotation $\Delta\phi$, whereas it is too large for the shift in Δy.

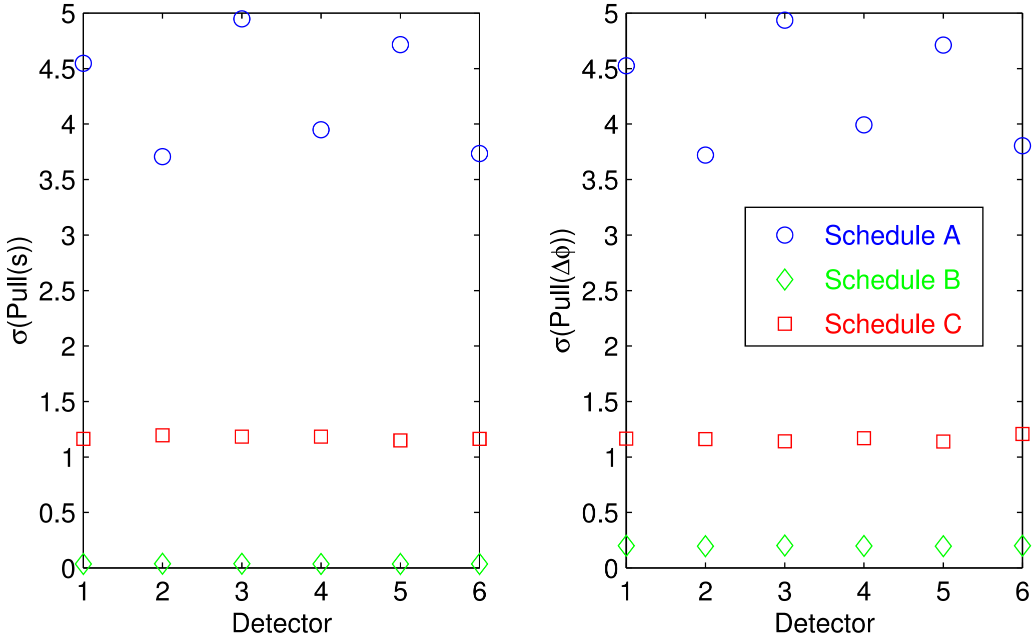


Figure 12: Standard deviation of the standardized residuals (pulls). Left hand side: shift in precise coordinate, right hand side: angle of rotation. Only in the case of schedule C the calculated errors of the reconstructed alignment parameters correspond reasonably good to the actual spread around the true values. In the case of schedule A these errors are too small, in the case of schedule B they are too big.

4.8 Track reconstruction after alignment

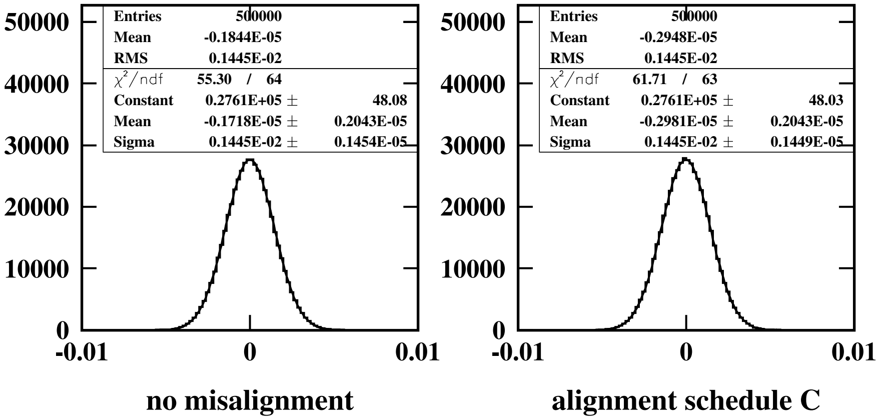
In order to quantify the quality of the different alignment strategies, we generate for each misalignment configuration 2500 tracks, out of which we use the first 2000 tracks for estimating the alignment parameters. Then we "freeze" the current alignment state and continue for the last 500 tracks where we only estimate the track parameters. For each annealing schedule we do this with 1000 different random misalignment configurations, resulting in a total amount of 2.5 million simulated tracks out of which 0.5 million reconstructed tracks are used for track analysis.

4.8.1 Resolution of the track parameters

For the resolution in the four track parameters $(t_x, t_y, x, y)^T$ we obtain the following results:

Schedule	x [μm]	y [μm]	$dx/dz \times 10^{-4}$	$dy/dz \times 10^{-4}$
no misalignment	14.45 ± 0.01	14.56 ± 0.01	1.372 ± 0.001	1.372 ± 0.001
A	14.55 ± 0.01	14.64 ± 0.01	1.37 ± 0.001	1.371 ± 0.001
B	16.13 ± 0.02	16.19 ± 0.02	1.448 ± 0.001	1.444 ± 0.001
C	14.45 ± 0.01	14.59 ± 0.01	1.37 ± 0.001	1.371 ± 0.001

Although alignment schedule C achieves best results for the estimation of the alignment parameters, this does not translate into a significantly better track parameter resolution compared with results from alignment schedule A. In both cases the parameter resolutions are equivalent to the ones achieved without misaligning the detectors. The differences in track position resolution between schedules A (Kalman Filter) and C (Deterministic Annealing Filter) are less than one tenth of a micron in favour of schedule C, and there is no difference at all in direction resolution. Track parameter resolution after strategy B is about 10% worse compared to the resolution obtained without misalignment (or strategy A and C). An example of two resolution distributions of the x coordinate is given in Figure 13.



no misalignment

alignment schedule C

Figure 13: Histograms of the position resolution x after track reconstruction without initial misalignment of detectors (left) and track reconstruction after alignment with annealing schedule C (right). As can be seen, full track parameter resolution is achieved with schedule C.

4.8.2 Pulls of the track parameters

For the standard deviations of the pulls of the four track parameters $(t_x, t_y, x, y)^T$ we obtain the following results:

Schedule	x	y	dx/dz	dy/dz
no misalignment	1 ± 0.001	1 ± 0.001	1.002 ± 0.001	1.001 ± 0.001
A	1.007 ± 0.001	1.006 ± 0.001	1 ± 0.001	1.001 ± 0.001
B	1.009 ± 0.001	1.016 ± 0.002	1.001 ± 0.001	1.002 ± 0.001
C	1 ± 0.001	1.002 ± 0.001	1 ± 0.001	1.001 ± 0.001

In all cases, the calculated errors of the reconstructed track parameters are compatible with the actual spread around the true values. The pull distributions are mean-value free with a standard deviation of 1 in all cases. The bad error estimation of alignment parameters for alignment schedules A and B seem to have almost no effect on the error estimation of the track parameters. Examples of two pull distributions of the x coordinate is given in Figure 14.

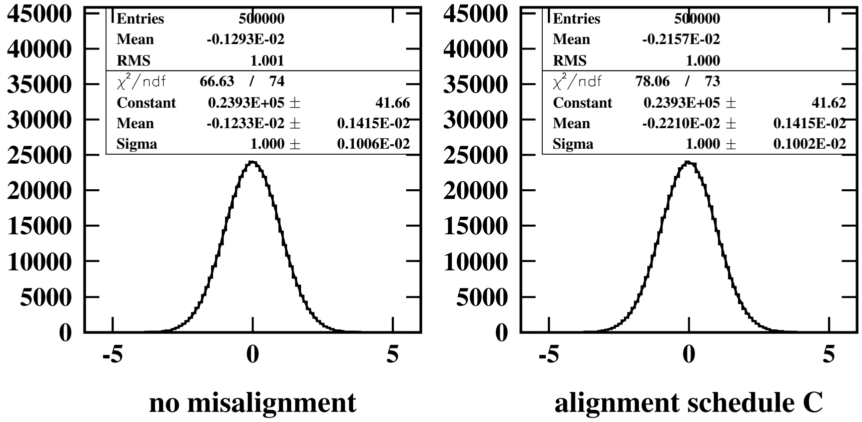


Figure 14: Histograms of the pulls of the x position coordinate after track reconstruction without initial misalignment (left picture) and alignment schedule C. The histograms are mean-value free and the standard deviations are exactly 1 in both cases.

5 Summary and Conclusion

We can summarize the results in the following way:

- We have presented a method for the simultaneous estimation of alignment parameters of several detectors with respect to a set of reference detectors in parallel to track reconstruction. The reference detectors are needed in order to fix the reference frame with respect to which the other detectors are aligned. This method (“alignment with tracks”) has been formulated in the Kalman Filter formalism which is already known for track reconstruction.

- We have investigated three different alignment strategies (annealing schedules). We have concluded from a simulation experiment that a Kalman Filter with annealing (schedule C) gives quantitatively and qualitatively the best results for the estimation of the alignment parameters. The necessary number of tracks and the annealing schedule can be taylored to the specific properties of the setup.

- Differences in track reconstruction after using alignment schedule A or C are negligible. In both cases the precision of track reconstruction is as good as for a perfectly aligned tracker. The performance of alignment schedule B can be expected to get worse with a more realistic track model using multiple scattering.

- The algorithms have been implemented and tested in ORCA for a test-beam like setup. The formulation of alignment reconstruction in the Filter formalism has made it easy to implement the algorithms in the existing logic and framework for track reconstruction in ORCA. In fact, this was one of our main motivations to formulate “alignment with tracks” in the Kalman Filter formalism for track reconstruction.

We therefore conclude that:

- Simultaneous estimation of track and alignment parameters with the Kalman Filter is possible.

- The method can be used with any kind of track model, provided that the magnetic field is known to sufficient precision. The result is a kind of “local” alignment, giving positions and orientations of a set of detector elements with respect to a fixed set of reference detectors.

- The algorithms have been verified in a simple and fully controlled simulation environment. This can be considered as a small but essential step towards an application in a realistic environment, e.g. for the CMS Tracker. Global alignment of the CMS Tracker is of course much more complex, requiring relative alignment of detector elements which are never crossed by one and the same track. In addition, the effects of wrong hit-to-track assignment and material effects on the convergence of the alignment parameters have to be investigated. This, however, is beyond the scope of this note.

- In our simulation experiment alignments with methods A and C are both precise enough to achive full (intrinsic) resolution in track parameters.
- The decoupling of the update formulas into one update for track parameters and one for alignment parameters has additional advantages. They can be used together or separately in the following way:
  - One can update both track and alignment parameters and still have a fully valid track with consistent error estimates after smoothing.
  - For a certain number of tracks one does only update the alignment parameters without using the track for further analysis after smoothing (standard alignment procedure).
  - After alignment over a certain number of tracks one freezes the current alignment states. For the following track reconstruction one uses the generalized update formulas derived in section 2 in order to take into account properly results from the preceding alignment. This is an alternative way of taking into account results from the alignment, without moving hits or detectors “by hand”. The shifts and rotations will enter statistically correctly via the alignment parameters in the update of the track state.

Appendix: Addendum to the Formalism

In the case of the estimation of all six alignment parameters — three shifts and three rotations — it might be of advantage to separate explicitly the shifts $\mathbf{s}$ and the rotations $\mathbf{r}$ in the linear approximation:

$$\mathbf{m} = \mathbf{f}(\mathbf{p}, \mathbf{s}, \mathbf{r}) + \boldsymbol{\epsilon}$$

In the linear approximation $\mathbf{m}$ can be expressed as

$$\mathbf{m} = \mathbf{c} + \mathbf{H}\mathbf{p} + \mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{r} + \boldsymbol{\epsilon}$$

with

$$\begin{aligned} \mathbf{H} &= \frac{\partial \mathbf{f}}{\partial \mathbf{p}}(\mathbf{p}_0, \mathbf{s}_0, \mathbf{r}_0) \\ \mathbf{A} &= \frac{\partial \mathbf{f}}{\partial \mathbf{s}}(\mathbf{p}_0, \mathbf{s}_0, \mathbf{r}_0) \\ \mathbf{B} &= \frac{\partial \mathbf{f}}{\partial \mathbf{r}}(\mathbf{p}_0, \mathbf{s}_0, \mathbf{r}_0) \end{aligned}$$

With $\mathrm{cov}(\mathbf{p}_0) = \mathbf{C}_0$, $\mathrm{cov}(\mathbf{s}_0) = \mathbf{S}_0$ and $\mathrm{cov}(\mathbf{r}_0) = \mathbf{U}_0$ and

$$\mathbf{W} = \left[\mathbf{V} + \mathbf{H}\mathbf{C}_0\mathbf{H}^{\mathrm{T}} + \mathbf{A}\mathbf{S}_0\mathbf{A}^{\mathrm{T}} + \mathbf{B}\mathbf{U}_0\mathbf{B}^{\mathrm{T}}\right]^{-1}$$

the update formulas are:

a) for the track parameters $\mathbf{p}$:

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{p}_0 + \mathbf{C}_0\mathbf{H}^{\mathrm{T}}\mathbf{W}\left[\mathbf{m} - \mathbf{f}(\mathbf{p}_0, \mathbf{s}_0, \mathbf{r}_0)\right] \\ \mathbf{C}_1 &= \mathbf{C}_0 - \mathbf{C}_0\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H}\mathbf{C}_0 \end{aligned}$$

b) for the shifts $\mathbf{s}$:

$$\begin{aligned} \mathbf{s}_1 &= \mathbf{s}_0 + \mathbf{S}_0\mathbf{A}^{\mathrm{T}}\mathbf{W}\left[\mathbf{m} - \mathbf{f}(\mathbf{p}_0, \mathbf{s}_0, \mathbf{r}_0)\right] \\ \mathbf{S}_1 &= \mathbf{S}_0 - \mathbf{S}_0\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\mathbf{S}_0 \end{aligned}$$

c) for the rotations $\mathbf{r}$:

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{r}_0 + \mathbf{U}_0\mathbf{B}^{\mathrm{T}}\mathbf{W}\left[\mathbf{m} - \mathbf{f}(\mathbf{p}_0, \mathbf{s}_0, \mathbf{r}_0)\right] \\ \mathbf{U}_1 &= \mathbf{U}_0 - \mathbf{U}_0\mathbf{B}^{\mathrm{T}}\mathbf{W}\mathbf{B}\mathbf{U}_0 \end{aligned}$$

References

[1] R. Frühwirth, Nucl. Instrum. Meth. A262 (1987) 444.

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

[3] http://cmsdoc.cern.ch/orca