ATL-INDET-PUB-2007-009
25 June 2007

TRT alignment for SR1 cosmics and beyond

Andrea Bocci
Duke University, Durham NC, USA.

Wouter Hulsbergen
CERN, Geneva, Switzerland.

June 21, 2007

We present an algorithm for the alignment of the ATLAS Inner Detector, developed for the analysis of cosmics data taken with the ID barrel in SR1 in the summer of 2006. Our results include a comparison of the relative SCT-TRT alignment with survey measurements, a study of the alignment of SCT and TRT module positions and TRT module deformations, and a comparison of the 'local' and 'global' methods for alignment.

Contents

1 Introduction

In this note we describe an algorithm for the alignment of the ATLAS TRT. The algorithm was developed in the context of the analysis of data from the ID-barrel cosmics test that took place in SR1 in summer 2006. Though in its original form only applicable to the TRT the algorithm has been extended to the SCT and Pixel detectors as well. The emphasis in this note is on TRT alignment, but some results for SCT alignment with cosmics and SCT/Pixel/TRT alignment with Monte Carlo data are included.

The structure of this note is as follows. In section 2 the χ^2 formalism used for extracting alignment constants from a sample of reconstructed tracks is presented. In section 3 we derive expressions for the derivatives of residuals to alignment parameters in the atlas TRT, SCT and Pixel detectors. Section 4 briefly describes the implementation of the algorithm in the Athena framework. Validations and results obtained from SR1 data are reported in sections 5 and 6. In the latter section we also report on an analysis of TRT module deformations. We conclude with a summary and outlook in section 7.

This note is the first ATLAS note to propose an alignment algorithm for the entire ATLAS Inner Detector. Previous algorithms that deal only with SCT and Pixel were described in [1, 2, 3].

2 The minimum χ^2 alignment formalism

In this section we derive a minimum χ^2 formalism for an alignment or calibration based on reconstructed charged particle trajectories. We take a didactic approach and start with the determination of track parameters using a Newton-Raphson method for χ^2 minimization. Subsequently, we introduce the alignment problem as an extra set of parameters that affects a collection of tracks and we explain how those parameters can be extracted by minimizing the total χ^2 with respect to both track parameters and alignment parameters simultaneously. The formalism we derive is equivalent to that in references [4] and [1], but the derivation follows a different approach.

In the remaining paragraphs we briefly touch on the treatment of multiple scattering, the problem of solving a large system of linear equations, a method to identify weakly constrained alignment parameters, adding additional constraints in the alignment process and the comparison of the 'global' and 'local' methods for alignment.

2.1 Track fitting with Newton-Raphson

The procedure of extracting the parameters describing the trajectory of a charged particle in a detector from a collection of measured coordinates is usually referred to as *track fitting*. Most track fitting procedures determine the track parameters from the minimization of a χ^2. This χ^2 is schematically defined as

$$\chi^2 \;=\; \sum_i \left(\frac{m_i - h_i(x)}{\sigma_i} \right)^2 \quad , \tag{1}$$

where x is a vector of track parameters and $\{m_i\}$ is the set of measured coordinates each with uncertainty σ_i. The function $h_i(x)$ expresses the measured coordinate in terms of the track parameters and is referred to as the *measurement model*.

The choice of the track parameterization x depends both on the actual layout of the detector (e.g. magnetic field, central versus forward) and on the method that is used to minimize the χ^2. Different minimization methods exists but those used for track fitting have in common that they find the minimum by explicitly calculating the derivative of the χ^2 to x. To illustrate this we first simplify notations by rewriting Eq.1 in matrix notation

$$\chi^2 \;=\; (h(x) - m)^T V^{-1} (h(x) - m) \quad , \tag{2}$$

where m is now a vector of measurements and V is the corresponding covariance matrix, which is usually diagonal. The condition that the χ^2 is minimum with respect to x is expressed as

$$0 \;\equiv\; \frac{\mathrm{d}\chi^2}{\mathrm{d}x} \;=\; 2H^T V^{-1} (h(x) - m) \quad , \tag{3}$$

where the derivative $H \equiv \mathrm{d}h/\mathrm{d}x$ is sometimes called the *projection* matrix. The value of x for which Eq.3 is satisfied can be obtained with the Newton-Raphson method: One calculates the value and derivative of Eq.3 at a certain initial value x_0, [1]

$$\left.\frac{\mathrm{d}\chi^2}{\mathrm{d}x}\right|_{x_0} \;=\; -2H^T V^{-1} (h(x_0) - m) \tag{4}$$

$$\left.\frac{\mathrm{d}^2\chi^2}{\mathrm{d}x^2}\right|_{x_0} \;=\; 2H^T V^{-1} H \tag{5}$$

and obtains the solution for x with

$$x_1 = x_0 - \left(\frac{\mathrm{d}^2\chi^2}{\mathrm{d}x^2}\right)^{-1} \frac{\mathrm{d}\chi^2}{\mathrm{d}x} \quad . \tag{6}$$

One can show that the covariance matrix for x is given by

$$C \;=\; 2\left(\frac{\mathrm{d}^2\chi^2}{\mathrm{d}x^2}\right)^{-1} \;=\; \left(H^T V^{-1} H\right)^{-1} \tag{7}$$

If the derivative H depends on x, we call the problem ‘non-linear’. In this case x_1 is not necessarily the value that minimizes the χ^2, although it is closer to the solution than x_0 is. One can therefore recompute the derivatives at the updated value x_1 and insert those in Eq.6 to obtain a new estimate x_2. This procedure can be ‘iterated’ until a certain convergence criterion is met, usually defined by a minimum change in the χ^2.

[1] Note that in the calculation of $\mathrm{d}^2\chi^2/\mathrm{d}x^2$ we omit second order derivatives in the residual $r = m - h(x)$. In other words, we linearize the residual rather than $\mathrm{d}\chi^2/\mathrm{d}x$ itself. The reason for this is that terms proportional to r could give negative contributions to $\mathrm{d}^2\chi^2/\mathrm{d}x^2$, which is undesirable. By omitting these contributions the minimization with Newton-Raphson is equivalent to the least squares estimator for the linearized measurement model.

The procedure described above that minimizes the χ^2 with respect to all track parameters and all coordinate measurements simultaneously is usually called a *global* track fit. An alternative track fitting procedure that has gained in popularity over the last two decades is based on the Kalman filter [5, 6] and several implementations of this algorithm exist in the Atlas software. One advantage of the Kalman filter over the global fit is that it is fast because the parameters x can be determined without explicitly calculating all their correlations. Unfortunately, for the alignment procedure described here these correlations are important which is why we rely on a global track fit instead. In the Atlas software framework such a track fit exists in the form of the `TrkGlobalChi2Fitter` [7].

2.2 The global χ^2 method for alignment

In realistic applications of the track fit the measurement model $h(x)$ does not only depend on track parameters but also on calibration or alignment parameters α that are common to all reconstructed tracks. In this section we describe how these parameters can be extracted from a large sample of tracks by minimizing the sum of the χ^2 contributions of all tracks with respect to α. To simplify notation we rewrite the χ^2 of an individual track as

$$\chi^2 = r^T V^{-1} r \tag{8}$$

where r is the residual vector [2],

$$r(x, \alpha, m) = h(x) - m \quad . \tag{9}$$

On purpose we leave ambiguous whether the calibration parameters are seen as part of the measurement model $h(x)$ or used to 'correct' the measurements themselves: for the formalism described here it is sufficient that the derivatives $\mathrm{d}r/\mathrm{d}\alpha$ can be calculated.

We now define a *total* χ^2 as

$$\chi^2 = \sum_{\text{tracks}} r^T V^{-1} r \tag{10}$$

At first it may seem that one could minimize this χ^2 with the procedure outlined in the previous section, replacing x by a vector containing both the calibration parameters and the track parameters of all tracks in the sample. However, such a strategy becomes unfeasible for a track sample exceeding 10000 tracks or so, because of the computation time needed to invert the second derivative matrix. Therefore, we ask ourselves how a minimum-χ^2 solution for α can be obtained without explicitly minimizing this χ^2 simultaneously with respect to the large set of track parameters.

The solution exploits the fact that only the parameters α are common to the tracks and the parameters x are not. For a given value α_0 we compute for each track the solution $x(\alpha_0)$, and use its derivative to predict how the tracks change as a function of α_0. Formally this can be done in at least two ways: we could either explicitly write down the solution for x and insert

[2] Note that we choose an unconventional sign for the residual. In the literature the symbol H is usually reserved for the derivative of the measurement model projection matrix $H \equiv \partial h(x)/\partial x$. However, the derivative that truly matters is the that of the residual and we therefore define $H \equiv \partial r/\partial x$. We choose the sign of the residual such that the two definitions for H are equal.

this in the χ^2 derivative for α, as is for example done in [1], or we can replace the derivatives to α by total derivatives that include the contribution due to changes in x:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} = \frac{\partial}{\partial\alpha} + \frac{\mathrm{d}x}{\mathrm{d}\alpha}\frac{\partial}{\partial x} \tag{11}$$

The derivative $\mathrm{d}x/\mathrm{d}\alpha$ follows from the condition that, once at minimum χ^2, we do not want any change in the derivative of the track χ^2 with respect to x, which can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{\partial\chi^2}{\partial x} = 0 \tag{12}$$

and results in

$$\frac{\mathrm{d}x}{\mathrm{d}\alpha} = -\frac{\partial^2\chi^2}{\partial\alpha\partial x}\left(\frac{\partial^2\chi^2}{\partial x^2}\right)^{-1} . \tag{13}$$

Note that if the problem is linear, this derivative is independent of the actual value of x or α, such that it is even valid if the track χ^2 was not yet minimized with respect to x.

Defining

$$A \equiv \frac{\partial r}{\partial\alpha} \qquad H \equiv \frac{\partial r}{\partial x} \tag{14}$$

and neglecting second order derivatives in the residual (linearization), we can now compute $\mathrm{d}x/\mathrm{d}\alpha$ and obtain for any total derivative to α

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} = \frac{\partial}{\partial\alpha} - A^T V^{-1} H C \frac{\partial}{\partial x}. \tag{15}$$

In this expression we have substituted the covariance matrix for x, Eq.7. This expression is still valid if x and C are determined using more constraints than expressed by the χ^2 in the previous section, for example a vertex constraint, or hits in a sub-detector that we are not concerned with.

Further ignoring any second order derivatives of the residual, we obtain for the first and second derivatives of the χ^2 contribution of a single track

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} = 2A^T V^{-1}\left(V - HCH^T\right)V^{-1} r \tag{16}$$

$$\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2} = 2A^T V^{-1}\left(V - HCH^T\right)V^{-1} A. \tag{17}$$

Comparing these expressions to those of the track fit in the previous section, one observes that only the term HCH^T is new. This matrix represents the covariance of the track parameters in *measurement space*. The matrix

$$R = V - HCH^T \tag{18}$$

is the covariance matrix of the residuals of the track fit.[3]

[3] The matrix R is in general singular. Its rank is the number of degrees of freedom of the fit, *i.e.* the number of hits minus the number of track parameters.

An important simplification takes place if the initial track parameters x_0 are actually those that minimize the track's χ^2 for the current set of alignment constants. In this case, by virtue of Eq.3, $H^T V^{-1} r$ is zero and the first derivative reduces to

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} = 2A^T V^{-1} r \quad . \tag{19}$$

Consequently, if the measurements m are independent and V is diagonal, the derivative to a particular parameter α_i only receives contributions from residuals for which $\partial r / \partial\alpha$ does not vanish. In other words, if α_i is an alignment parameter of module X, only hits in module X contribute to the first derivative of the χ^2 to α_i. We shall label this important property by saying that the first derivative is *local*. This property is not self-evident from Eq.16 and has not been explicitly identified in [4, 1].

A consequence of the first derivative being local is that in order to compute the derivative it is not necessary to keep track of contributions to the χ^2 other than those for which $\partial r / \partial\alpha$ is non-zero. In particular, contributions from hits in other subdetectors, from vertex constraints or multiple scattering terms can be ignored.

Following Eq.19 the condition that the total χ^2 of a sample of tracks be minimal with respect to both track and alignment parameters can thus be expressed as

$$0 \equiv \frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} = 2 \sum_{\text{tracks}} \frac{\partial r}{\partial\alpha}^T V^{-1} r \quad . \tag{20}$$

For M alignment parameter, the minimum χ^2 condition consists of M coupled equations. In general, these equations are *not linear* in the alignment parameters α.

In analogy with the procedure introduced in the previous section, we now search for a solution by linearizing the minimum χ^2 condition. We evaluate the first and second derivative for an initial calibration α_0 and corresponding track parameters x_0, accumulate the result for a large sample of tracks and solve the *linear* system of M equations

$$\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\Delta\alpha = -\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} \tag{21}$$

for $\Delta\alpha$. If the residuals are non-linear in either track parameters or alignment parameters, several iterations may be necessary to minimize the χ^2. This method of obtaining the solution to Eq.20 is called the *global method for alignment*. The well-known `MILLIPEDE` algorithm [4] is an example of this alignment procedure.

In contrast with the global method, the *local method for alignment* searches for a solution to Eq.20 by ignoring the correlation between parameters corresponding to different alignable objects (or *modules*) in the detector. This effectively reduces the problem posed in Eq.21 from solving one big system to many small systems. This simplicity does not come for free: when correlations are ignored, iterations are necessary to solve the minimum Eq.20 even if the original condition was entirely linear. We present a discussion of the local method and a comparison to the global method in section 2.7.

Provided the second derivative matrix can be inverted, the covariance matrix for the alignment parameters in the global method becomes

$$\mathrm{Cov}(\alpha) = 2\left(\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\right)^{-1} . \tag{22}$$

Ignoring higher order derivatives in α, the change in the total χ^2 as the result of a change $\Delta\alpha$ in the alignment parameters is

$$\Delta\chi^2 \; = \; \Delta\alpha^T \frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} + \frac{1}{2}\Delta\alpha^T \frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\Delta\alpha \tag{23}$$

Substituting the result for the global method, one obtains

$$\Delta\chi^2 \; = \; -\frac{1}{2}\,\Delta\alpha^T \frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\,\Delta\alpha \; = \; -\Delta\alpha^T\,\mathrm{Cov}(\alpha)^{-1}\,\Delta\alpha^T. \tag{24}$$

Consequently, in the global method the change in the total χ^2 is equivalent to the *significance* of the alignment correction. In section 6 we exploit this result to compare the convergence of the different ‘solution’ methods.

2.3 Dealing with multiple scattering

In track fits multiple coulomb scattering (MCS) can be incorporated in the track model by allowing the track to ‘kink’ on a finite number of multiple scattering planes. Denoting the scattering angles (two for each plane) by θ_i, the χ^2 takes the form

$$\chi^2 \; = \; \sum_{\text{hits } i} \frac{(h_i(x,\theta) - m_i)^2}{V_{ii}} \; + \sum_{\text{scat.angles } j} \frac{(\hat{\theta} - \theta_j)^2}{\Theta_{jj}} \quad . \tag{25}$$

where the first term corresponds to the measured coordinates and the second term represents the contribution from the ‘expected’ scattering angles. The expectation value for the scattering angles, denoted by $\hat{\theta}$, is zero. Its uncertainty, denoted by $\sqrt{\Theta_{jj}}$ is a function of track momentum, particle type and the thickness of the scattering plane. Since the expectations for the scattering angles are independent, Θ_{jj} is diagonal.

In a global track fit the χ^2 is minimized for x and θ simultaneously. With the substitution

$$\begin{array}{rcc} x & \longrightarrow & (x,\theta) \\ m & \longrightarrow & (m,\hat{\theta}) \\ V & \longrightarrow & \begin{pmatrix} V & 0 \\ 0 & \Theta \end{pmatrix} \end{array} \tag{26}$$

the χ^2 minimization procedure for alignment discussed above remains entirely unchanged. In particular, the measurement covariance matrix V remains diagonal and the important property that the first derivative is local is preserved. The `TrkGlobalChi2Fitter` used in our implementation of the alignment algorithm follows exactly this procedure to incorporate MCS. There exists an alternative method for the treatment of MCS in global track fits in which MCS contributions are treated by introducing correlations between the measurements, which results in a non-diagonal matrix V. This method is used in reference [1] and briefly summarized in appendix B. The main disadvantage of formulating the problem in this way is that it hides the fact that the first derivative does not depend on the scattering angle residuals and hence remains local.

2.4 Solving a large system of linear equations

The main challenge in the application of the global method is to actually solve the large linear system in Eq.21. For the alignment problems studied in this note, with typically 1000 parameters, an explicit inversion of the second derivative matrix is feasible, such that $\Delta\alpha$ follows directly from

$$\Delta\alpha \;=\; -\left(\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\right)^{-1}\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} \tag{27}$$

We have successfully applied both `CLHEP` matrix inversion and `LAPACK` diagonalization (see also section 2.5) to obtain an alignment of the TRT and SCT with SR1 cosmics with 1400 parameters, using typically 1 minute of CPU time. The diagonalization with `LAPACK` is shown to be applicable to problems with about 10000 parameters as well, in particular by exploiting parallel computing [1, 8]. However, the computation time of the diagonalization or inversion grows approximately with the third power of the dimension of the problem making even the step from 10000 to the actual 40000 parameters in the ATLAS alignment nontrivial. In addition, the memory required to store the inverse of the second derivative is very large, over 3Gbyte for 40000 parameters.

There exist several algorithms that solve the system in Eq.21 not with an explicit inversion as in Eq.27, but by searching iteratively for a solution x that minimizes the ‘distance’,

$$d \;=\; \left|\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\Delta\alpha + \frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha}\right| \tag{28}$$

The first algorithm of this kind is (to our knowledge) `MINRES` [9] dating from the early 1970s. This algorithm is for example applied for the alignment of the CMS detector [10]: using `MINRES` they report a solution to Eq.21 with 10000 parameters in approximately 30 seconds. ATLAS collaborators [11] are currently investigating more modern algorithms such as MA27 [12], MA57 [13] and PARDISO [14] that are even faster and less sensitive to initial conditions. These algorithms can be combined with the ARPACK [15] algorithm to calculate a limited number of eigenvalues and eigenvectors of $\mathrm{d}^2\chi^2/\mathrm{d}\alpha^2$, useful for the suppression of weak degrees of freedom (see section 2.5). Besides the improvement in time by several orders of magnitudes, another advantage of the iterative algorithms is that they can make optimally use of the fact that $\mathrm{d}^2\chi^2/\mathrm{d}\alpha^2$ is often sparse, saving a considerably amount of memory.

2.5 Eigenvalues and such

We have seen earlier that finding the minimum χ^2 solution to the alignment problem is a matter of solving the linear system

$$\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2}\,\Delta\alpha \;=\; -\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha}$$

for $\Delta\alpha$. A particular interesting aspect of this problem is the eigenvalue distribution of $\mathrm{d}^2\chi^2/\mathrm{d}\alpha^2$, because that tells us something about the so-called ‘weak modes’, linear combinations of the alignment parameters that are relative poorly constrained.

To illustrate this we first introduce a simplified notation, defining

$$A \equiv \frac{1}{2}\frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2} \qquad b \equiv -\frac{1}{2}\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} \qquad x \equiv \Delta\alpha \tag{29}$$

such that the minimization problem becomes $Ax = b$. Any symmetric matrix A can be written as

$$A \;=\; UDU^T \tag{30}$$

where U is an orthogonal matrix ($U^{-1} = U^T$) whose columns are the eigenvectors of A and D is a diagonal matrix with diagonal entries that are the eigenvalues of A. The rank of A is the number of non-zero eigenvalues of A. The matrix A is invertible only if all eigenvalues are non-zero, *i.e.* if its rank is equal to its dimension N.

The diagonalization of A is a non-trivial problem, but assuming it can be done and provided all eigenvalues are non-zero, the solution to the set of equations becomes

$$x \;=\; A^{-1}b \;=\; UD^{-1}U^Tb \tag{31}$$

where the inverse of D follows trivially. In the χ^2 minimization problem, the inverse of A is also the covariance matrix for x, such that we have

$$C(x) \;=\; UD^{-1}U^T \tag{32}$$

Expressed in terms of the eigenvalues $d_j \equiv D_{jj}$ and eigenvectors $u_i^{(j)} \equiv U_{ij}$ the solution and covariance matrix are

$$x \;=\; \sum_j \frac{\langle u^{(j)}, b\rangle}{d_j}\, u^{(j)} \qquad C(x) \;=\; \sum_j \frac{1}{d_j} u^{(j)} \otimes u^{(j)} \tag{33}$$

where the $\langle , \rangle$ represents an inner product and the $\otimes$ a Kronecker product ($(a \otimes b)_{ij} = a_i b_j$). Finally, the expected change in the total χ^2 becomes

$$\chi^2 \;=\; -\sum_j \frac{\langle u^{(j)}, b\rangle^2}{d_j}. \tag{34}$$

This result, that the change in the total χ^2 is the sum of the contributions for the individual modes, is non-trivial. It is a consequence of the fact that the alignment corrections for the different modes are uncorrelated.

One observes in these equations that the reciprocal of the eigenvalue appears in the covariance matrix: small eigenvalues of A lead to large statistical uncertainty in x. We call the eigenvectors that have small associated eigenvalues the ‘weak modes’.

In the specific case that the system is underconstrained a subset of the eigenvalues is zero and a subset (or all) of the parameters x have infinitely large uncertainty. An example of this is a global transformation of the detector: Such a transformation leaves the total χ^2 invariant. If this degree of freedom is not removed from the system of linear equations, the number of solutions is infinite.

2.5.1 Rescaling the problem

An interesting method to remove unconstrained and underconstrained degrees of freedom is by omitting certain eigenmodes in the solution by limiting the sum in Eq.33 to eigenvalues above a certain threshold [1, 16]. This is less trivial than it seems. First, we need a recipe for choosing a sensible threshold. Second, we should realize that there is some freedom in adjusting the scale of the eigenvalues. The most trivial example of this is by 'mixing' the system of linear equations with a square matrix S, e.g.

$$SAx \;=\; Sb \tag{35}$$

Provided that the rank of S is N, the solution to the 'mixed' system is the same as that of the original system, but the eigenvalues of SA are not equal to those of A. More generally we could also mix the parameters with a square matrix with rank N and rewrite the system as

$$(SAT^{-1})\,(Tx) \;=\; Sb \quad . \tag{36}$$

The diagonalization is less work if the the matrix A is symmetric so we are mostly interested in reparameterizations that leave SAT^{-1} symmetric and choose $T^{-1} = S^T$. In other words, we look at transformations of the form

$$(SAS^T)\,(S^{T^{-1}}x) \;=\; Sb \quad . \tag{37}$$

Denoting the eigenvalues and eigenvectors of SAS^T as e_j and $v^{(j)}$ respectively, the solution becomes

$$x = \sum_j \frac{\langle S^T v^{(j)}, b\rangle}{e_j}\, S^T v^{(j)} \qquad C = \sum_j \frac{1}{e_j} (S^T v^{(j)}) \otimes (S^T v^{(j)}) \tag{38}$$

(Note that $\langle S^T v^{(j)}, b\rangle = \langle v^{(j)}, Sb\rangle$.) We now make the following observations:

- if S is a matrix that diagonalizes A, then $S^T v^{(j)} = v^{(j)}$, and this result reduces to the result derived above;
- although S is required to be invertible, we never explicitly need the inverse of S to rescale A or calculate x;
- since for every eigenvector of SAS^T

  $$(SAS^T)v^{(j)} = e_j v^{(j)} \iff A(S^T v^{(j)}) = e_j (S^{-1} v^{(j)}) \tag{39}$$

  the eigenvectors of SAS^T are eigenvectors of A only if S is orthogonal. However, even if S is not orthogonal, something special occurs for the eigenvectors with zero eigenvalue: If $v^{(j)}$ is an eigenvector of SAS^T with eigenvalue zero, then $S^T v^{(j)}$ is an eigenvector of A with eigenvalue zero.[4]

The last point implies for the alignment that the unconstrained degrees of freedom are independent of the scaling S.

[4] This important observation was made by P. Brückman in the alignment hypernews.

2.5.2 Choosing a suitable basis

We have seen in the paragraph above that the matrix A can be rescaled and that this allows us to scale the eigenvalues more or less arbitrarily. The question of 'choosing' the well-constrained eigenmodes therefore comes down to choosing *both* a suitable basis and a suitable eigenvalue threshold. We shall try to answer this question by studying the statistical uncertainties in the alignment parameters. Given a certain alignment parameter α_i, the contribution of the uncertainty in α_i to the uncertainty in the hit residual is

$$\sigma^2_{\text{tot}} \;=\; \sigma^2_{\text{hit}} + \frac{\mathrm{d}r}{\mathrm{d}\alpha_i}\sigma^2_{\alpha_i}\frac{\mathrm{d}r}{\mathrm{d}\alpha_i} \tag{40}$$

The requirement that the contribution of the alignment parameter be small, can then be formulated as the requirement that

$$\sigma^2_{\alpha_i} \;<\; \epsilon \left\langle \frac{\mathrm{d}r}{\mathrm{d}\alpha_i}\,\frac{1}{\sigma^2_{\text{hit}}}\,\frac{\mathrm{d}r}{\mathrm{d}\alpha_i} \right\rangle^{-1} \tag{41}$$

for a certain value ϵ. To keep notation short in what follows, we define the average projection of the hit error in the alignment coordinate

$$\rho^2_{\alpha_i} \;\equiv\; \left\langle \frac{\mathrm{d}r}{\mathrm{d}\alpha_i}\,\frac{1}{\sigma^2_{\text{hit}}}\,\frac{\mathrm{d}r}{\mathrm{d}\alpha_i} \right\rangle^{-1} . \tag{42}$$

The contribution from eigenmode j to the variance in alignment parameter α_i follows from Eq.38. Normalized to the hit resolution, this contribution is

$$\sigma^{2(j)}_{\text{norm}}(\alpha_i) \;=\; \frac{1}{\rho^2_{\alpha_i}}\,\frac{1}{e_j}\left(S^T v^{(j)}\right)^2_i \tag{43}$$

Therefore, a measure for the 'total contribution' of j to the hit uncertainty over all modules is

$$\sigma^{2(j)}_{\text{norm}} \;=\; \sum_i \frac{1}{\rho^2_{\alpha_i}}\,\frac{1}{e_j}\left(S^T v^{(j)}\right)^2_i \tag{44}$$

The meaning of this 'total contribution' may not be very well defined, but at least it gives us an idea how we could normalize the matrix A such that the eigenvalues are expressed in terms of detector resolution: If we choose S diagonal, with elements given by

$$S_{ii} \;=\; \rho^2_{\alpha_i} \tag{45}$$

then we have exactly

$$\sigma^2_{\text{norm}} \;=\; \frac{1}{e_j} \tag{46}$$

In other words, the contribution of eigenmode j to the total variance relative to the detector resolution for each module would be exactly $1/e_j$. Denoting the sum over the subset of eigenvalues with $\frac{1}{e_j} < \epsilon$ by $\sum^*$, we derive that the normalized variance of any alignment parameter is bounded from above by ϵ

$$\sigma^2_{\text{norm}}(\alpha_i) \;=\; \sum_j{}^{*} V_{ij}\frac{1}{e_j}V_{ij} \leq \epsilon \sum_j{}^{*} V_{ij}V_{ij} \leq \epsilon \sum_j V_{ij}V_{ij} \leq \epsilon \quad . \tag{47}$$

This leaves us with the question of how we could estimate $\rho^2_{\alpha_i}$. The simplest way to do this is to realize that the diagonal elements of A are equal to

$$A_{ii} \;=\; \sum_{\text{hits}\,j} \frac{\mathrm{d}r_j}{\mathrm{d}\alpha_i} w_j \frac{\mathrm{d}r_j}{\mathrm{d}\alpha_i} \tag{48}$$

where the sum runs over all hits in the module X to which this alignment parameter belongs and where the weight w_j given by

$$w_i \;\equiv\; \frac{1}{\sigma^2_{\text{hit},j}} \sigma^2_{\text{residual}\,j} \frac{1}{\sigma^2_{\text{hit},j}} \tag{49}$$

If the errors from the track parameters were negligible, we would simply have $w_j = 1/\sigma^2_{\text{hit}}$ and

$$A_{ii} \;=\; n_X \frac{\mathrm{d}r_X}{\mathrm{d}\alpha_i} \frac{1}{\sigma^2_{\text{hit}X}} \frac{\mathrm{d}r_X}{\mathrm{d}\alpha_i} \tag{50}$$

so the diagonal element of A itself, normalized by the number of hits, would be exactly what we need. In general this is not the case. Therefore, we have chosen to keep track of the average σ^2_{hit} and the average weight w_j in each module and compute the diagonal elements of S as

$$S_{ii} \;=\; \sqrt{\frac{n_X}{A_{ii}} \left\langle \frac{1}{\sigma^2_{\text{hit}}} \right\rangle_X^{-1} \left\langle w \right\rangle_X} \tag{51}$$

The eigenvalue e_j themselves can now be interpreted as the *effective number of hits* contributing to eigenmode j. The only remaining issue is to select a suitable value for ϵ. For the cosmics setup we have chosen $\epsilon = 0.025$, *i.e.* the equivalent of 40 hits. This value rejects all modes that contribute more than about 1 % to the detector resolution.

Note that in the derivation above we have ignored correlations between the different alignment parameters in a module. One solution to treat the correlations within a module properly is to diagonalize the submatrix of A corresponding to module X as part of the rescaling with S. Though this is conceptually not complicated, we have not further investigated it.

2.6 Additional constraints

Up to now we have only discussed constraints from single tracks. In order to remove poorly constrained degrees of freedom it is sensible to introduce other constraints. In appendix A we describe a simple procedure to constrain tracks to a common vertex.[5] Since vertex constraints are irrelevant for cosmics we have not yet implemented these.

Another important class of constraints are 'survey' constraints, additional information obtained from detector survey. Such constraints can be implemented as additional contributions to the χ^2. For example, if a survey measurement exist for alignment parameter α_i, with value $\hat{\alpha}_i$ and uncertainty σ_i, we add a term

$$\chi^2_{\text{survey}} \;=\; \left(\frac{\alpha_i - \hat{\alpha}_i}{\sigma_i} \right)^2 \tag{52}$$

[5] The procedure we describe does not require refitting the tracks, which makes it not just simple but also fast.

to the χ^2. The additional terms to the derivatives follow trivially.

In general survey constraints involve combinations of alignment parameters, for example on the difference between module positions. If the constraint is expressed as a constraint equation $g(\alpha)$ with covariance G, the χ^2 contribution becomes

$$\chi^2_{\text{survey}} \;=\; g^T G^{-1} g \tag{53}$$

and the derivatives now involve also the derivatives of g to α. For the analysis described in this note we have not made use of real survey constraints. However, as an example we have implemented a constraint to the *nominal* geometry, *i.e.* the geometry without alignment corrections. The uncertainties assigned to the nominal geometry were derived from the distribution of misalignments parameters extracted from data.

It is not clear how useful ATLAS survey constraints are for a real application in the alignment procedure. For example, SCT and Pixel survey measurements were recorded while the detector was at a much higher temperature than that at which it will be operated for data taking. Therefore, we expect that in practice survey measurements will only be used as a warning bell or for initialization of the alignment constants.

Another class of constraints are *exact* constraints. Exact constraints describe relations between parameters that we hold to be exactly correct. For example, we may decide that a certain rotation is the same for all TRT modules, simply because we have no sensitivity to it. Or we may define that the average position of all TRT modules is exactly at the origin in order to remove the associated unconstrained global degrees of freedom. Exact constraints can be imposed by parameter substitution, but often it is easier to use the method of Lagrange multipliers. If the constraint can be expressed as a function $g(\alpha)$, the χ^2 contribution becomes

$$\chi^2_{\text{exact}} \;=\; \lambda^T g(\alpha) \tag{54}$$

where λ is a (vector of) additional parameters. If the χ^2 derivative to α and λ vanish simultaneously, we have minimized the χ^2 under the condition that $g(\alpha) = 0$. In a true application this just involves adding λ to the list of parameters in the derivative calculation. We have not yet implemented any exact constraints in the algorithm discussed here.

2.7 The local χ^2 method for alignment

In the *local* χ^2 method for alignment, correlations between alignment parameters of different modules in the detector are ignored: The parameters for each module are calculated under the asumption that all other modules do not move. In practice this can be done defining a separate χ^2 for each module. Only the subset of hits in the module contribute to the χ^2 derivatives of this module. As we have seen in section 2.2, this really only affects the second derivative because the first derivative is already ‘local’, *i.e.* determined only by the residuals in the module under consideration. Consequently, the alignment expressions for the local method follow from those of the global method if ‘in-between-module’ terms are ignored in the second derivative. The expressions for the derivatives do not change and the alignment constants follow from Eq.21, when applied on a module-by-module basis. The number of parameters per module is small, making the inversion of the second derivative now trivial. In references [2, 3] the local method is formulated in terms of unbiased residuals. In appendix C we show that this formulation is equivalent to our formulation.

Because the Atlas alignment group included proponents of both the global and the local method for alignment, it is worthwhile to briefly summarize the differences. As explained above the global and local method for alignment both solve the *same* set of equations Eq.20. Therefore, independent of the procedure that is followed to obtain a solution to this condition, if the procedure converges such that Eq.20 is satisfied, the procedure is as good as it gets in terms of alignment parameters.

In the local method for alignment certain terms in the 2nd derivative (which is the first derivative of the condition equation) are ignored. Therefore, the calculation of the solution following Newton-Raphson cannot be expected to lead in a single step to the right answer, even if the problem itself is entirely linear (*i.e.* if higher order derivatives vanish). Since correlations between modules are in general large, several iterations of data processing and recomputation of the derivatives may be necessary to arrive at a solution that actually minimizes the total χ^2. Possible criteria to truncate the iterative procedure are the change in the total χ^2 or the size of the first derivative itself. The local method is particularly sensitive to poorly constrained degrees of freedom: in the worst case those can lead to poor convergence or oscillations. Another problem with the local method is that it does not give access to the covariance matrix for the alignment parameters. Uncertainties can be calculated locally, but those underestimate the true uncertainty.

In its most complete form, the global method for alignment has three main advantages: First, since the 2nd derivative is now complete, Newton-Raphson can be expected to lead to a single step solution if the problem is linear. Second, if the system of equations is solved by explicitly inverting the 2nd derivative matrix (for example by diagonalization), the full covariance matrix is available. Finally, the calculation of eigenvalues and corresponding eigenvectors allows the identification and removal of poorly constrained degrees of freedom.

In practice, the advantages of the global method are less striking. First, problems of this kind are not linear and as we shall see in section 6 the global method does not necessarily converge much faster than the local method does. Second, for alignment problems as large as that posed by the Atlas inner detector, the large system of equations can actually only be solved with iterative methods, as discussed in section 2.4. In this case neither the complete set of eigenmodes, nor the alignment covariance matrix are available.

3 Expressions for the derivatives to alignment parameters

In this section we derive expressions for the derivatives $\partial r/\partial\alpha$ that are needed for the calculation of the derivatives of the total χ^2 discussed in the previous section. These derivatives depend on the reference frame in which they are calculated. Therefore, we start with a reminder of the coordinate frames relevant for reconstruction and alignment.

3.1 Coordinate frames in Atlas

Three Atlas frames are of particular importance for alignment, namely the global frame, the local module frame and the measurement frame. The *global* coordinate frame is the frame in which the origin is in the center of the inner detector, the z-axis is along the beam direction, the y axis is the vertical axis and the x axis is the horizontal axis pointing towards the center of the accelerator. The *local module frame* is different for each TRT, SCT and Pixel detector element (table 1). In this frame the z-axis is perpendicular to the module plane. The y-axis is parallel to the straw direction in the TRT, parallel to the strip direction on the non-stereo side in the SCT and parallel to 'long side' of modules in the Pixel detector. (Therefore, for all detector types, the y-axis in the module frame is paralel to the z-axis of the global reference frame.) Finally, the x-axis is perpendicular to y and z.

	origin	y-axis	z-axis
TRT strawlayer	det-elem midpoint	‖ straws	⊥ strawlayer
SCT wafer	det-elem midpoint	‖ strips on non-stereo side	⊥ wafer
Pixel wafer	det-elem midpoint	‖ long side	⊥ wafer

Table 1. Local module frame coordinates

	origin	y-axis	z-axis
TRT	middle of straw wire	‖ straw	$\hat{x} \times \hat{y}$
SCT	det-elem midpoint	‖ strips on non-stereo side	⊥ wafer
Pixel	*see table 1*		

Table 2. Measurement frame coordinates

The *measurement frame* (tabel 2) is the coordinate frame in which the residual is calculated. It is unique for every hit on the track. In the Pixel detector the measurement frame is equal to the module frame. In the SCT the measurement frame shares the z-axis with the module frame, but the xy-plane is rotated such that the y-axis is parallel to the strip direction. (In the module frame the y-axis is parallel to the strip direction on the non-stereo side only.) In the TRT measurement frame the x-axis is perpendicular to both track and straw, the y-axis is parallel to the straw and the z-axis is perpendicular to x and y. The motivation for defining the TRT frame like this is that the drifttime of the TRT hit is a measurement of the distance between track and wire, *i.e.* $|x|$ in the measurement frame.

3.2 Transformations between coordinate frames

We consider the transformation between a global frame with coordinates x and local frame with coordinates x'. In the Athena framework the transformation from local to global coordinates is defined as

$$\mathbf{x} \;=\; T \;+\; R\,\mathbf{x}' \tag{55}$$

where T and R are respectively a translation and rotation. We call the combined operation of a rotation and translation a *transform* and label it with the symbol A. In Athena a transform is represented by a `CLHEP::HepTransform3D` object. To combine transformations we define a multiplication for transforms such that

$$A_1\,A_2\,\mathbf{x} \;=\; T_1 \;+\; R_1\,(T_2 \;+\; R_2\,\mathbf{x}), \tag{56}$$

Note that transforms do not commute.

3.3 Alignment transforms

The nominal position of a straw in the TRT is calculated with a sequence of transforms that schematically looks like this

$$A_{\text{nominal}} \;=\; A_{\text{system}}\,A_{\text{module}}\,A_{\text{straw}}. \tag{57}$$

In this expression A_{system} is a global transform different for barrel-A, barrel-C, endcap-A and endcap-C, A_{module} is the transform from the module frame to the system frame and A_{straw} is the transformation from straw to module frame.

Alignment corrections are applied by inserting *alignment transforms* in the transformation from global to straw coordinates. In the TRT alignment corrections are applied on a per-module basis and take the form

$$A_{\text{aligned}} \;=\; \Delta_{\text{system}}\,\Delta_{\text{module}}\,A_{\text{nominal}} \tag{58}$$

In other words both the module and the system correction are applied in the global frame.

The nominal position of a strip in the SCT or a pixel in the Pixel detector is calculated with transforms that look like

$$A_{\text{nominal}} \;=\; A_{\text{system}}\,A_{\text{layer}}\,A_{\text{module}}\,A_{\text{sensor}} \tag{59}$$

Alignment corrections can be applied at three levels and are

$$A_{\text{aligned}} \;=\; \Delta_{\text{system}}\,\Delta_{\text{layer}}\,A_{\text{system}}\,A_{\text{layer}}\,A_{\text{module}}\,\Delta_{\text{module}}\,A_{\text{sensor}} \tag{60}$$

Note that, in contrast with the situation in the TRT, the module-level alignment corrections in SCT and Pixel are defined in the module frame.

For the alignment procedure it is important to know in which frame alignment corrections are applied. However, for the calculation of derivatives described here, the frame is only of secondary importance since the correction can always be transformed to a different frame, *e.g.*

$$\Delta_{\text{local}} \;=\; A^{-1}_{\text{local-to-global}}\Delta_{\text{global}}A_{\text{local-to-global}} \tag{61}$$

In the following we will therefore assume that alignment corrections are calculated in the global reference frame.

3.4 Derivatives in the TRT

To calculate the derivative of the hit residual to alignment corrections we need to understand the way the *measurement* frame of a hit changes as a result of the alignment. The measurement frame in the TRT is special since it is constructed such that the x-axis is perpendicular to both the straw direction and the direction of the track leading to simple expressions for the derivatives in the measurement frame. We shall therefore start with the TRT and subsequently extend the formalism to SCT and Pixel.

We label coordinates in the measurement frame with a prime and choose the straw position as the origin in this frame. We denote the coordinate of the point where the track intersects the x-axis in this frame by $\mathbf{x}' = (x', 0, 0)$. Since the measurement frame was chosen such that its x-axis is perpendicular to both straw and track, the residual of the TRT is simply the value of x' (eventually with a minus sign depending on your preferred definition of a residual). The derivative of the residual to the coordinate of the track in the measurement frame is thus given by

$$\begin{pmatrix} \partial r/\partial x' \\ \partial r/\partial y' \\ \partial r/\partial z' \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{62}$$

The derivatives to the track direction in this frame all vanish, provided we use the point $\mathbf{x}'$ as the origin of the rotation.

If T and R are respectively the translation and rotation from the measurement frame of the TRT hit to global coordinates, $\mathbf{x}'$ is related to a global coordinate $\mathbf{x}$ by the inverse of the transformation in Eq.55, ie

$$\mathbf{x}' \;=\; R^T \,(\,\mathbf{x} \;-\; T\,). \tag{63}$$

We now consider the effect of an infinitesimally small transformation in the global frame, consisting of a rotation

$$\delta R = (1 + \epsilon) \tag{64}$$

and a translation

$$\delta T = \eta, \tag{65}$$

where η is an infinitesimally small vector and ϵ is an infinitesimally small rotation matrix. The transformation from local to global coordinates changes as follows

$$\mathbf{x} \;=\; \eta \;+\; (1 \;+\; \epsilon)\,(T \;+\; R\,\mathbf{x}') \;=\; \eta \;+\; (1 \;+\; \epsilon)\,T \;+\; (1 \;+\; \epsilon)\,R\,\mathbf{x}' \tag{66}$$

with the inverse transformation given by

$$\mathbf{x}' \;=\; R^T\,(1+\epsilon^T)\,(\mathbf{x} \;-\; \eta \;-\; (1+\epsilon)T\,) \;=\; R^T\,(\mathbf{x} - T) \;+\; R^T\,\epsilon^T\,\mathbf{x} \;-\; R^T\eta \;+\; \mathrm{O}(\eta\epsilon) \tag{67}$$

The small rotation can be parameterized in terms of rotations around the 3 major axis[6]

$$(1+\epsilon) \;=\; \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\epsilon_x \\ 0 & \epsilon_x & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \epsilon_y \\ 0 & 1 & \\ -\epsilon & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\epsilon_z & 0 \\ \epsilon_z & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{68}$$

[6] In Athena the corresponding rotation matrices can be created with `CLHEP::RotationX(double)`, etc..

The order of the rotations is irrelevant, since the rotations commute if we neglect terms of second order in ϵ. The derivatives of the residual to the rotation and translation can now be explicitely calculated. For the translation the result is

$$\begin{pmatrix} \partial r/\partial \eta_x \\ \partial r/\partial \eta_y \\ \partial r/\partial \eta_z \end{pmatrix} = \begin{pmatrix} -(R^T)_{xx} \\ -(R^T)_{xy} \\ -(R^T)_{xz} \end{pmatrix} \tag{69}$$

and for the rotation

$$\begin{pmatrix} \partial r/\partial \epsilon_x \\ \partial r/\partial \epsilon_y \\ \partial r/\partial \epsilon_z \end{pmatrix} = \begin{pmatrix} (R^T)_{xy}z - (R^T)_{xz}y \\ (R^T)_{xz}x - (R^T)_{xx}z \\ (R^T)_{xx}y - (R^T)_{xy}x \end{pmatrix} \tag{70}$$

In this expression the point (x, y, z) is the position of the point of closest approach of the track to the straw in *global* coordinates, the coordinate frame related to the measurement frame by R and T. However, the expressions remain equally valid in any other reference frame. For example, if we were interested in calculating the derivatives to a small rotation and translation in the module frame we could simply transform R, T and (x, y, z) to that frame. Therefore, these expressions give access to the derivatives to an alignment transform in any frame.

3.5 Derivatives in the SCT and Pixel

The expressions for the alignment of the SCT are of course similar to those for the TRT alignment, with the substitution of strips for straws. However, in the ATLAS tracking there is one small difference: in the SCT the residual is not evaluated in a frame perpendicular to track and silicon strip (which we shall label as the 'normal frame' in what follows), but in the silicon wafer plane. Therefore, the residual is not the smallest distance between track and strip and the derivative of the residual to the track position and direction in the measurement frame is not given by the simple expression in Eq.62.

The authors of the 'local-χ^2-method for SCT alignment' [2] have chosen to recompute the residuals and their uncertainties in the normal frame. Although this is an elegant solution, we shall here explore the alternative of expressing the derivatives in the measurement frame, which is conceptually not that hard either.

Define once more the position of the track in the measurement frame as given by $\mathbf{x}' = (x', 0, 0)$. Consider a small change in the position $(\delta_x, \delta_y, \delta_Z)$. If (n_x, n_y, n_z) is a vector parallel to the track direction in the measurement plane, the change in the x-position of the point at which the track intersects the xy-plane, becomes

$$\Delta x' = \delta_x - \delta_z n_x/n_z \tag{71}$$

Therefore, the derivative of the residual to a change in the position of the track in the measurement frame is given by

$$\begin{pmatrix} \partial r/\partial x' \\ \partial r/\partial y' \\ \partial r/\partial z' \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -n_x/n_z \end{pmatrix} \quad . \tag{72}$$

The derivatives to a rotation of the track in the measurement frame again vanish, provided $\mathbf{x}'$ is the origin of the rotation. Note that although the direction of the vector $\mathbf{n}$ has an arbitrary sign, the ratio n_x/n_z does not, because it is the tangent of an angle. Defining ψ as the angle of the track in the xz plane in the measurement frame, we have $n_x/n_z = \cot\psi$. Using these results the expressions for the derivative of the residual to the alignment translation become

$$\begin{pmatrix} \partial r/\partial\eta_x \\ \partial r/\partial\eta_y \\ \partial r/\partial\eta_z \end{pmatrix} = \begin{pmatrix} -(R^T)_{xx} + \cot\psi\,(R^T)_{zx} \\ -(R^T)_{xy} + \cot\psi\,(R^T)_{zy} \\ -(R^T)_{xz} + \cot\psi\,(R^T)_{zz} \end{pmatrix} \tag{73}$$

and the derivative to the rotation

$$\begin{pmatrix} \partial r/\partial\epsilon_x \\ \partial r/\partial\epsilon_y \\ \partial r/\partial\epsilon_z \end{pmatrix} = \begin{pmatrix} \Big((R^T)_{xy} - \cot\psi\,(R^T)_{zy}\Big)\,z - \Big((R^T)_{xz} - \cot\psi\,(R^T)_{zz}\Big)\,y \\ \Big((R^T)_{xz} - \cot\psi\,(R^T)_{zz}\Big)\,x - \Big((R^T)_{xx} - \cot\psi\,(R^T)_{zx}\Big)\,z \\ \Big((R^T)_{xx} - \cot\psi\,(R^T)_{zx}\Big)\,y - \Big((R^T)_{xy} - \cot\psi\,(R^T)_{zy}\Big)\,x \end{pmatrix} \tag{74}$$

Be aware that in this expression the point (x, y, z) refers to the position of the point of intersection of track and wafer in *global* coordinates while the tangent $\cot\psi$ is calculated in the measurement (wafer) frame. For $\cot\psi = 0$ these expressions are of course equal to those for the TRT.

Residuals in the Pixel are two-dimensional since a pixel hit measures both x and y in the local frame. The expressions for the derivatives of the x-residual are equal to those in the SCT. The expressions for the derivatives of the y-residual can be obtained in a similar fashion as the derivatives above. The result for the translation is

$$\begin{pmatrix} \partial r_y/\partial\eta_x \\ \partial r_y/\partial\eta_y \\ \partial r_y/\partial\eta_z \end{pmatrix} = \begin{pmatrix} -(R^T)_{yx} + \cot\omega\,(R^T)_{zx} \\ -(R^T)_{yy} + \cot\omega\,(R^T)_{zy} \\ -(R^T)_{yz} + \cot\omega\,(R^T)_{zz} \end{pmatrix} \tag{75}$$

and for the rotation

$$\begin{pmatrix} \partial r_y/\partial\epsilon_x \\ \partial r_y/\partial\epsilon_y \\ \partial r_y/\partial\epsilon_z \end{pmatrix} = \begin{pmatrix} \Big((R^T)_{yy} - \cot\omega\,(R^T)_{zy}\Big)\,z - \Big((R^T)_{yz} - \cot\omega\,(R^T)_{zz}\Big)\,y \\ \Big((R^T)_{yz} - \cot\omega\,(R^T)_{zz}\Big)\,x - \Big((R^T)_{yx} - \cot\omega\,(R^T)_{zx}\Big)\,z \\ \Big((R^T)_{yx} - \cot\omega\,(R^T)_{zx}\Big)\,y - \Big((R^T)_{yy} - \cot\omega\,(R^T)_{zy}\Big)\,x \end{pmatrix} \tag{76}$$

where $\cot\omega \equiv n_y/n_z$ is the cotangent of the angle of the track in the yz plane in the wafer frame.

4 TRTAlignAlg: Implementation of an algorithm for the alignment of Inner Detector modules

The alignment algorithm described in the previous sections has been implemented in the Athena package InnerDetector/InDetAlignAlgs/TRT_AlignAlgs. The 'algorithm' – as in the athena meaning of the word – is called TRTAlignAlg. Originally intended for the alignment of TRT modules only, the algorithm has been extended to SCT and Pixel modules as well. The implementation is reasonably well documented in the source code. Therefore, we limit the discussion here to a summary of the main features of the algorithm and a list of things that are missing or where improvements can be made. In the appendix we briefly describe some aspects of the C++ design and the most important subroutines.

The most relevant properties of the algorithm are:

- the algorithm implements the calculation of alignment corrections for six types of modules, namely TRT/SCT/Pixel and Barrel/Endcap, each represented by a C++ class, derived from a common ModuleBase base class. The base class implements the derivatives to rotations and translations. In the derived class a C++ specialization of the derivative calculation is possible, which allows for the implementation of derivatives to module deformations. As an example, we have implemented a twist correction for TRT barrel modules;

- the alignment degrees of freedom are defined in the 'alignment' frame which is a module dependent frame (see appendix). For each type of module each degree of freedom can be activated independently. The alignment is parameterized as a δ alignment, *i.e.* a small correction on top of existing alignment constants. This makes the assumption of commuting rotation derivatives valid;

- all levels of granularity supported by the geometry are also supported by the algorithm, namely 3 levels in the SCT and Pixel and 2 levels in the TRT;

- in its 'execute' method the algorithm processes the list of tracks on a given input list and computes the contribution to the full first and second derivative, according to expressions Eq.19 and Eq.17 respectively. To obtain the derivative of the residuals to the track parameters H and the track covariance matrix C, the algorithm internally calls TrackGlobalChi2Fitter to refit the tracks. There is no explicit calculation of derivatives to track parameters inside the algorithm;

- in its 'finalize' method the algorithm estimates new alignment corrections using either the 'global' or the 'local' method. For calculating the solution in the local method CLHEP is used to invert the (small) second derivative matrices. In the global method LAPACK is used for diagonalization, exploiting the interface provided by the SiGlobalChi2Algs/Al* classes;

- objects representing the total χ^2 derivative data can be persistified and added. The part of the algorithm calculating new alignment constants operates on these objects. This allows splitting the event processing across multiple jobs;

- during the event processing the χ^2 derivative data objects are created dynamically when first needed, including the ‘inter-module’ terms in the second derivative. Therefore, the sparsity of the second derivative is fully retained, saving memory and eventually diskspace. This sparsity is still lost in the ‘solution’ phase of the global method since neither CLHEP nor LAPACK can deal with sparse matrices.

The output of the algorithm consists of

- a ROOT file with histograms for each module, in particular distributions of residuals, residual-pulls and hit errors;
- a log file with information about the alignment for each module. We have used these log files to make figures in this note that show observables as a function of iteration;
- a pool file with the alignment constants for TRT/SCT/Pixel;
- a text file with TRT alignment constants to be read by TRTAlignDbTool;
- a text file with SCT/Pixel alignment constants to be read by InDetAlignDbTool
- a binary file with the total χ^2 derivatives

Several improvements are possible or even required before the algorithm can be used to align the complete ID:

- using LAPACK for diagonalization when running the global method is not going to work with 40k degrees of freedom. Faster methods have been explored by others in ATLAS [11]. We are waiting for a conclusion before implementing an interface to one of those methods;
- there is no support yet for vertex constraints. The mathematics describing how to deal with such constraints is described in appendix A;
- there is no support for using ‘generic’ survey constraints. Survey data could of course be loaded to initialize the geometry, but that is not the same as an actual constraint;
- the implementation is rather pragmatic. There is no clear distinction between data and interfaces or tools. A restructuring would definitely improve maintainability.

5 Test of derivatives of global method by identifying global degrees-of-freedom

As explained in section 2 in the global method for alignment a linear system of the form $Ax = b$ is solved, where A is the second derivative of the total χ^2 to the alignment parameters. If certain linear combinations of alignment parameters are not constrained, A has eigenvalues with value zero. The eigenvectors corresponding to these eigenvalues correspond to the unconstrained 'modes' of the system.

A trivial example of unconstrained degrees of freedom are the three global rotations and translations. If no other constraints are used, a given sample of reconstructed tracks does not constrain a global motion of the detector. Therefore, if all modules are aligned both in position and orientation, the eigenvalue spectrum must contain exactly 6 zero eigenvalues, 3 translations and 3 rotations. Identifying these zero eigenvalues is a useful test of the alignment algorithm: If the test succeeds, this means that the correlations between the residuals have been correctly computed and that the derivatives to the alignment parameters are correct — or at least consistently wrong.

We have performed this zero-eigenvalue test by running the alignment procedure on a sample of high-momentum multi-muons, generated in a perfectly aligned detector. These data contain pixel, SCT and TRT hits in both barrel and endcap. At the time of writing, our alignment algorithm does not yet implement a fast method for solving the linear system. Therefore, it is not possible to treat SCT and pixel alignment at the module level in these data: the number of parameters is simply too large. Instead, we have reduced the number of alignable objects to 3 in both the SCT and the pixel, namely for barrel, forward- and backward-endcap. In the TRT we have performed tests both with individual modules and with a 'combined' object as in pixel and SCT.

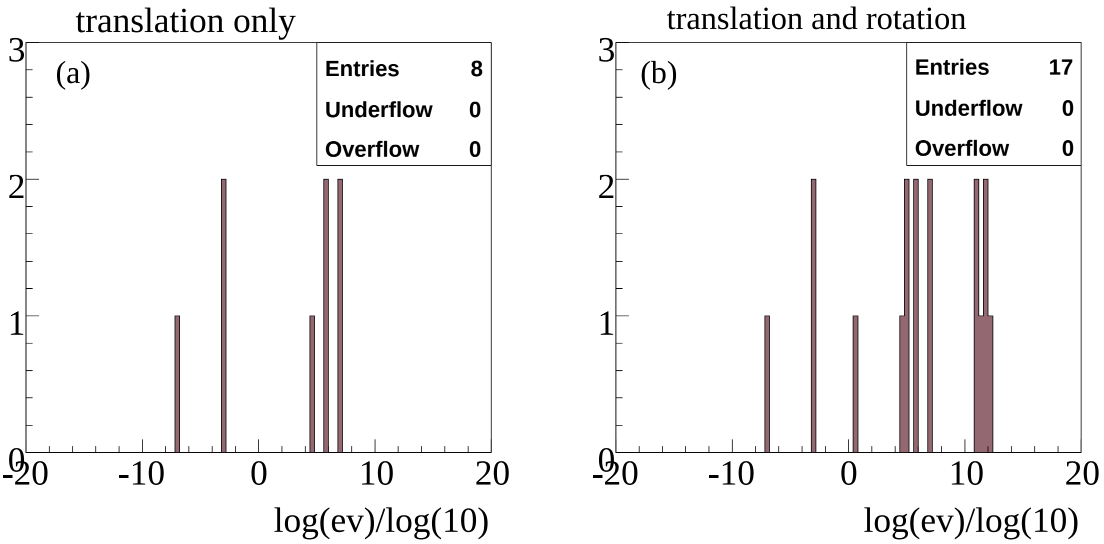

Figure 1. Eigenvalue spectra for A for the alignment of TRT, SCT and pixel barrel, aligning only translation (a) or translations and rotations (b). The matrix A was not rescaled.

To illustrate the procedure we start with the most simple scenario: We process tracks with barrel hits only and align only three objects, namely a single TRT barrel, SCT barrel and pixel barrel. Since all TRT barrel straws are parallel, the data do not constrain the motion of the TRT along the straw direction, *i.e.* the global z coordinate or the local y-coordinate. Therefore, there are only 2 independent translations in the TRT. At first, we only align for translations, leading to a total of $2 + 3 + 3 = 8$ parameters. The eigenvalue spectrum of A is shown in figure 1a. As one can see, the spectrum contains two distinct classes of eigenvalues: 3 eigenvalues are more than a factor 10^6 smaller than the remaining 5 eigenvalues. These are the 3 unconstrained translations. They are not exactly zero due to small numerical mistakes in the calculation of the derivatives.

Figure 1b shows the same results if the rotations are included. The total number of parameters is now 17. Clearly, this distribution does not show the same separation of scales as that for the translations only. However, with a bit of imagination one can subtract the left distribution from the right distribution to obtain the eigenvalues for the rotations only. Those show again the separation we seek: three eigenvalues for the rotations are much smaller than the remaining eigenvalues. Note that it is not possible to choose a scenario with only rotations: unless the *alignment frame* is the same for all alignable objects, the global rotation cannot be decoupled from a global translation.

The problem with the representation of the eigenvalue spectrum above is that the ‘scale’ for rotations and translations is not the same. In fact, it is arbitrary as discussed in section 2.5. For example, we could change the unit of a rotation from rad to mrad, which would shift all rotation eigenvalues by a factor 10^6 but would otherwise not have any effect on the alignment result. A solution to this problem is the scaling recipe discussed in section 2.5. By normalizing the eigenvalues to the detector resolution, the scale is fixed in a meaningful way. The result of this is shown in figure 2 for the same two scenarios. Figure 2b now shows a distinct set of 6 eigenvalues that represent the three global translations and the three global rotations. The eigenvalues themselves are the effective number of hits per alignment mode. For the unconstrained degrees of freedom this is typically much smaller than 1. Note that the number of hits was in fact not very large. (For this test we used only a 100 events.)

As a final example we show in figure 3 the eigenvalue spectrum if all modules in the TRT barrel are treated as separate alignable objects using 1000 events. The eigenvalues corresponding to the unconstrained degrees of freedom are still easy to identify. All results presented here were obtained without multiple scattering angles in the global track fit. We have verified that the results do not qualitatively change if multiple scattering is included.

Finally, figure 4 shows the pull distribution for the alignment parameters both without and with the treatment of multiple scattering in the track fit. For the high momentum muons in this sample the effect of multiple scattering is small. Note that neither of the pull distributions is consistent with unity. We can partially attribute this to a problem in the measurement uncertainty (V) assigned to clusters in the pixel-x direction: the width of the pull distribution for residuals in this coordinate is about 1.3. For the other measurements the pull is much closer to 1. We have not further investigated this problem.

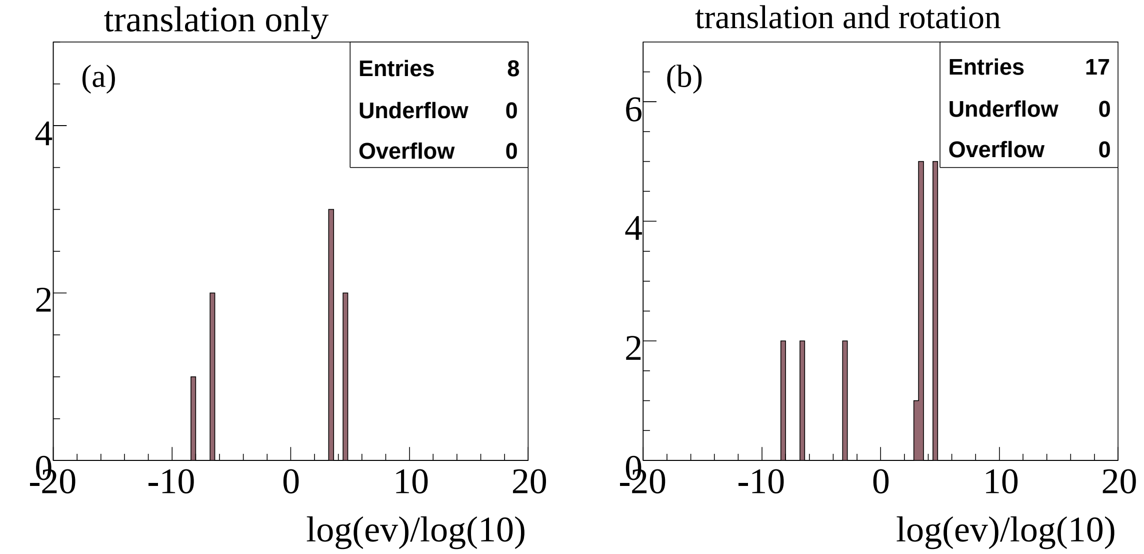

Figure 2. Eigenvalue spectra of *A* for the alignment of TRT, SCT and pixel barrel, aligning only translation (a) or translations and rotations (b). The matrix *A* was rescaled with the recipe described in section 2.5.

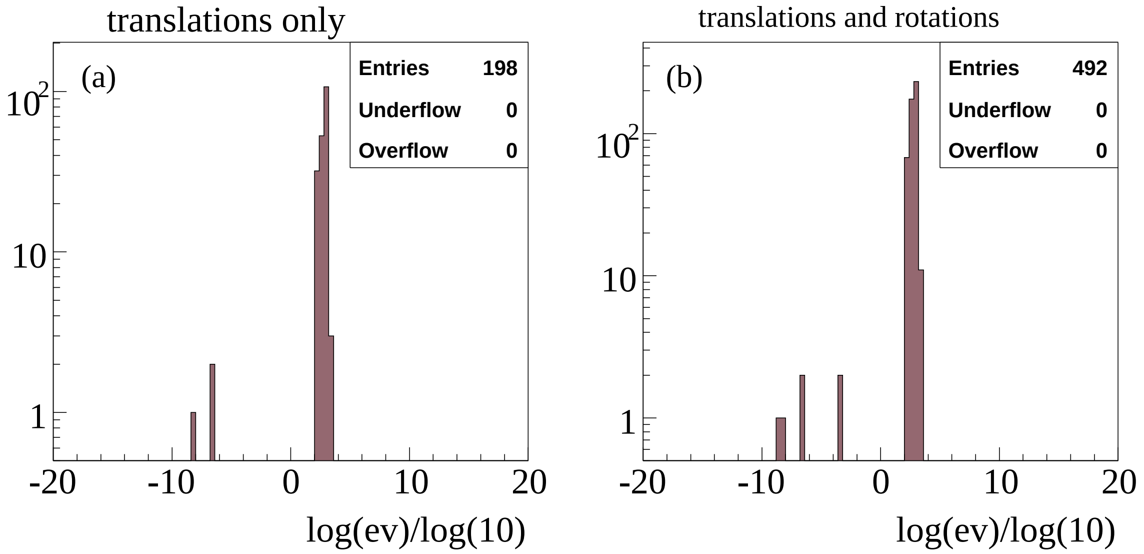

Figure 3. Eigenvalue spectra of *A* for the alignment of SCT and pixel barrel and TRT barrel modules , aligning only translation (a) or translations and rotations (b). The matrix *A* was rescaled with the recipe described in section 2.5.

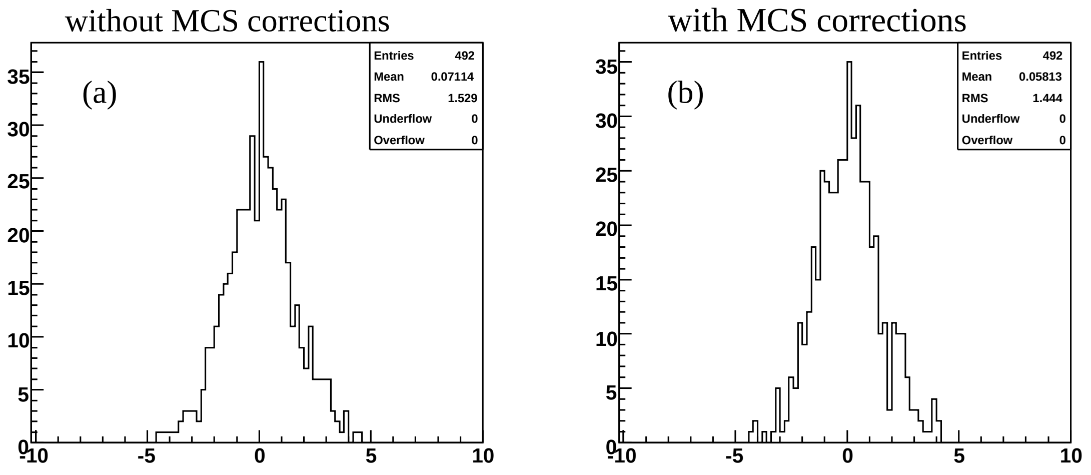

Figure 4. Pull of alignment parameters on perfectly aligned simulated multi-muon events without (a) and with (b) multiple scattering corrections in the track fit.

6 Results obtained on SR1 cosmics data

6.1 Setup of the SR1 cosmics barrel test

Figure 5 gives a schematic overview of the setup used for recording cosmics data with the ATLAS inner detector barrel in SR1. A cosmics trigger is created from a coincidence of two scintillators, providing a trigger rate of about 2.4 Hz [18]. A total of 4 TRT sectors are instrumented with readout electronics, namely sectors 6 and 7 on the 'top' and sectors 22 and 23 on the 'bottom'. Each sector consists of 3 TRT modules, labeled by their layer number (0–2). The straw sense wires are split in the middle and read out on the two ends of the barrel, named A and C respectively. In the bottom sectors, only the A end is instrumented.

Figure 6 (in a mirror view of figure 5) shows also the instrumented SCT sectors. A total of 504 SCT modules are instrumented. About 450 of these have sufficient overlap with the trigger scintillators and TRT modules to be exposed to muons with hits in both TRT and SCT. Several 100k events have been recorded with this setup in the summer of 2006. The analysis described in this note was mostly based on run 3099 which contains about 30k events.

A GEANT simulation of single muon cosmic events was used to study expected event rates and develop reconstruction, alignment and monitoring algorithms. Cosmic showers and the distribution of material surrounding the detector were not accounted for in this simulation. This prohibits a quantitative comparison between data and simulated events due to the difference in track multiplicity and the large uncertainties in the momentum distribution.

6.2 Event reconstruction

Most of the analysis described here was performed in Atlas release 12.0.4 using the `InDetCosmicRecExample` package as a base for track reconstruction. SCT-TRT tracks are reconstructed using an algorithm originally developed for the analysis of CTB data [7]. The tracks are fitted with the `GlobalChi2Fitter`. Note that tracks that are reconstructed in the TRT only do not provide information on track parameters outside the *xy*-plane. Therefore, we have relied exclusively on tracks with both TRT and SCT hits, except for the extraction of global alignment parameters, discussed in section 6.5.

The SR1 setup does not have a magnetic field. Consequently, the momentum of the muons traversing the detector cannot be measured. Each reconstructed track is fitted without accounting for multiple scattering, which is equivalent to fitting the track as if it had infinite momentum. As a result track χ^2 distributions and pull distributions look far from optimal and track parameter uncertainties are significantly underestimated. Since the uncertainties in alignment parameters are proportional to the uncertainties on track parameters, the former are also underestimated. We have briefly considered to gauge the uncertainties by deriving scale factors using simulated data, but such a procedure is a) non-trivial by itself and b) it is not clear that results derived on Monte Carlo apply to data, because of the uncertainty in the true momentum distribution in the data.

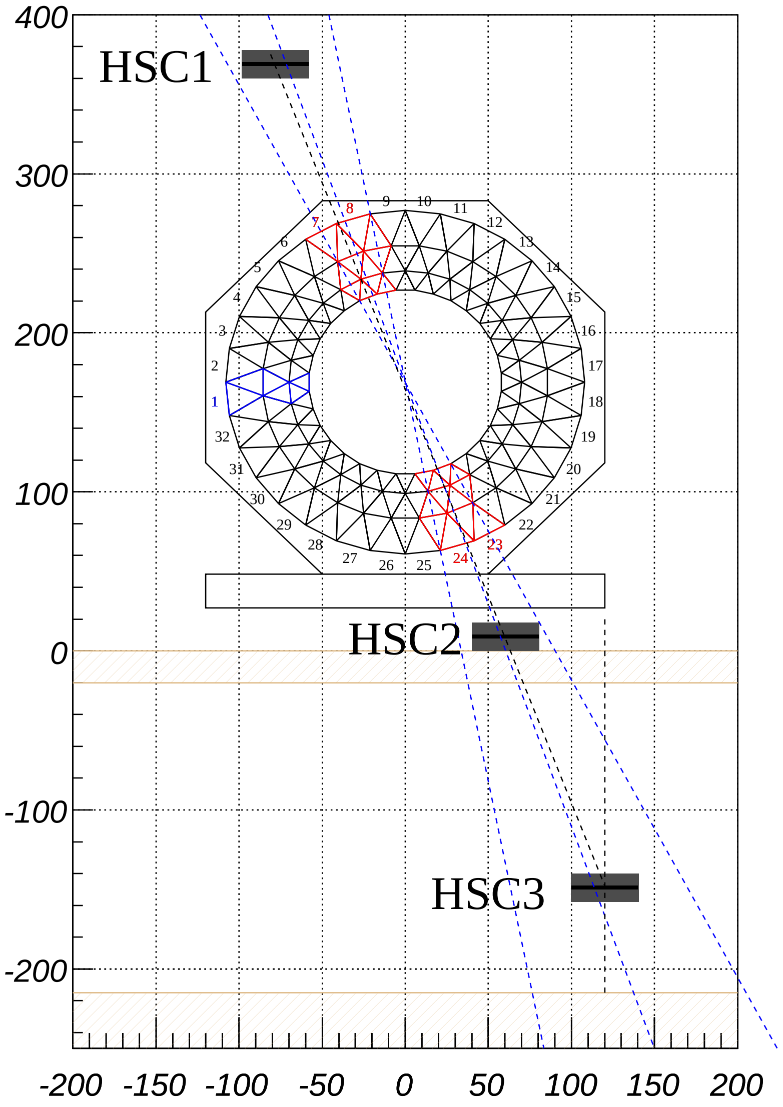

Figure 5. *xy*-View of the cosmic ray trigger setup. Only the top and middle scintillators were used in the trigger. The instrumented TRT sectors are displayed in red. The SCT is not shown. Dimensions are given in cm.

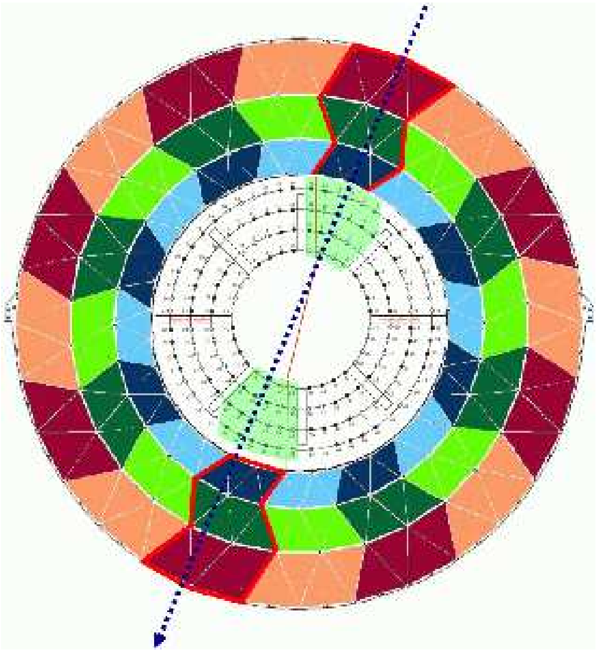

Figure 6. End-view of the barrel. The red polygons indicate instrumented TRT sectors. The instrumented SCT sectors are shown in green.

6.3 Track and hit selection

Tracks for alignment are selected by requiring at least 5 SCT hits, at least 20 TRT hits and a χ^2 per degree of freedom less than 20. Figure 7 shows the distributions for the variables for tracks from the `InDetCosmicTracks` track list in data and MC. The two bumps in the distribution of the number of TRT hits correspond to tracks that have been reconstructed only in the top or bottom sector or in both. Note that tracks in the data have more TRT hits and worse χ^2 than in the simulation MC. The higher number of TRT hits is most likely due to a higher noise level in the data. The difference in the average χ^2 could for example originate from remaining alignment problems in the data or a difference in momentum distribution. Understanding these differences is outside the scope of this document, but we shall assume that they are irrelevant for the purpose of using simulated data in this analysis, which is the validation of the alignment algorithm itself.

For the analysis of the cosmic data no selections are applied at the hit level.

6.4 Performance of the algorithm on simulated cosmics data

To test the performance of the alignment algorithm in a controlled environment we have applied it to simulated cosmics events. Events are generated with the nominal detector geometry. Misalignments are introduced by applying a non-zero alignment ‘correction’ in the event reconstruction. These corrections are currently read from text files, just as in real data. The alignment algorithm is run in iterations. After each iteration the alignment correction file is updated with the output of the alignment algorithm. The result obtained after convergence can directly be compared to the nominal alignment by studying the difference of the ob-

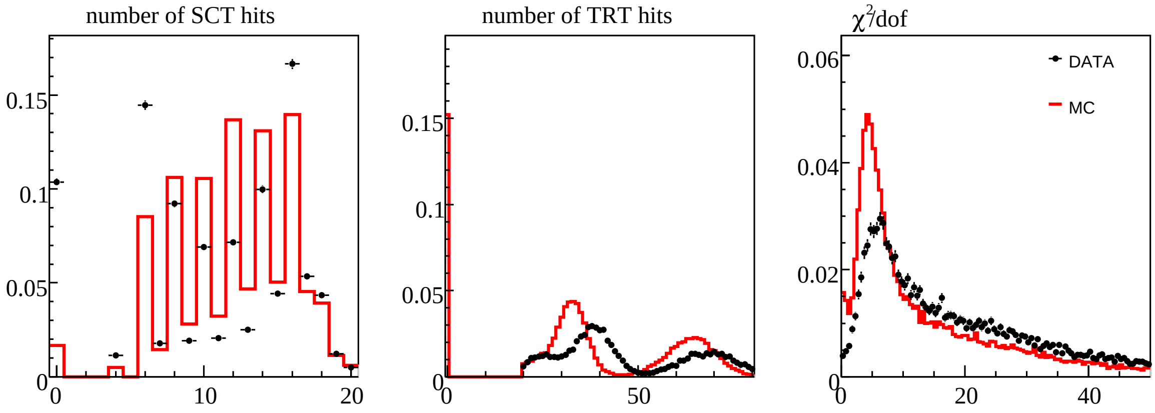

Figure 7. Distributions for selection variables from the `InDetCosmicTracks` track list on run 3099. All distributions have been normalized to a unit integral.

tained alignment corrections from zero. Besides providing a check for the convergence of the alignment procedure, this exercise also leads to some understanding of the accuracy of the estimated statistical uncertainties.

The TRT modules were displaced by $\pm 50\,\mu$m in the local x direction and rotated by 3 mrad around the module axis (the local y-axis). The SCT was kept in its nominal position. The alignment algorithm was run in the 'local' mode, aligning only for the local x and local rot-y module parameters. A total of 15K events were used, with the event selection described above.

In figure 8 the evolution of the two TRT alignment parameters Δx and Δrot-y is reported for each module as a function of iterations. From the given starting points they all converge fairly quickly to their "stable" values corresponding to the TRT geometry that minimizes the residuals of the given set of tracks. Due to statistical uncertainties, the position of a module that minimizes the residuals differs from the nominal position of the module. As can be seen in the figure, the alignment corrections look actually *significantly* different from zero. However, as explained in section 6.2, we expect that the statistical uncertainty assigned to the alignment corrections is underestimated, such that it cannot be directly used to judge the statistical significance.

Rather than interpreting the deviations from zero, we provide evidence that they are statistical in nature: If they are statistical, then the deviations should depend only upon the particular set of tracks chosen for the alignment and not on the alignment corrections used as a starting point for the alignment procedure. To show this we run again the TRT alignment algorithm with exactly the same setup but this time starting from the nominal geometry instead of the misaligned geometry. Comparing the result in figure 9 to that in figure 8 shows that the two procedures indeed converge to the same alignment. Although this does not prove that the alignment procedure does not introduce systematic misalignment affects, it does show that if these effects exist, they do not depend on the starting point of the alignment.

In Fig.10 the residual distributions before and after the alignment are compared. The misaligned geometry is shifted away from zero. After the alignment, the residual distribution

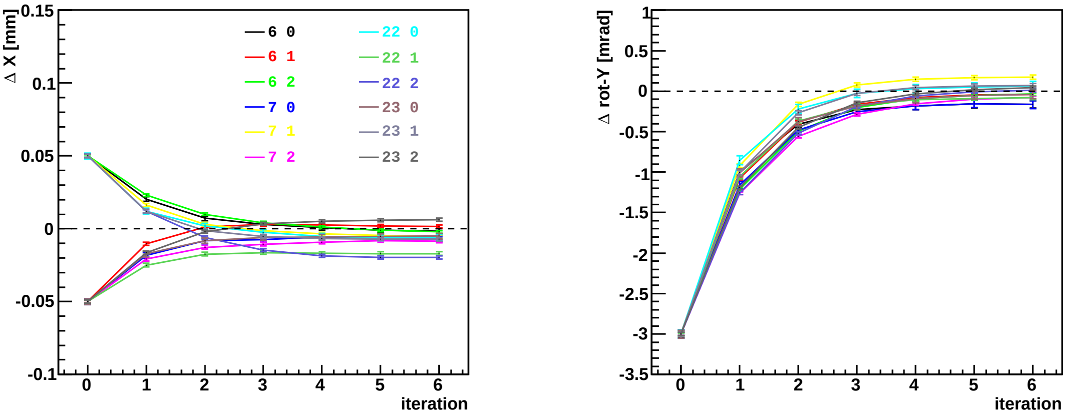

Figure 8. Module TRT alignment parameters local x (left) and local rot-y (right) as a function of iteration for the MC sample with the initial misalignment of ±50 μm in Δx and 3 mrad in Δrot-y.

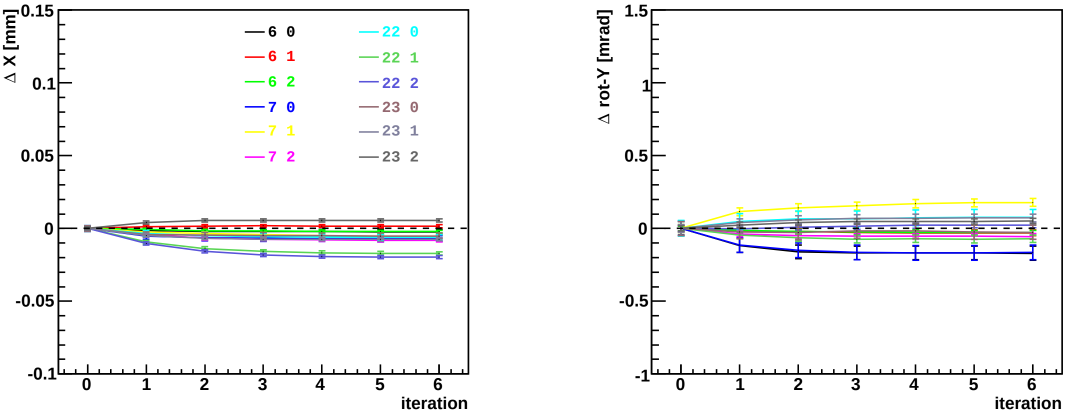

Figure 9. Module TRT alignment parameters local x (left) and local rot-y (right) as a function of iteration for the MC sample (without any initial misalignment).

cannot be distinguished from that obtained with the nominal geometry. Note the width of the residual distribution is only marginally affected by the misalignment because multiple scattering effects are entirely dominant.

In Fig.11 some observables as a function of iterations are shown. The number of fitted tracks stays more or less unchanged while the number of selected tracks increases after the first iteration. The χ^2 of the alignment change shown in the left-most figure is a measure for convergence. It will be discussed in more detail in section 6.6.

6.5 Global TRT-SCT misalignment in data

The relative misalignment between the TRT and SCT barrel detectors can be expected to be large compared to *internal* misalignments in these detectors. The results from a survey measurement performed in SR1 are shown in table 3. The survey measures the relative alignment

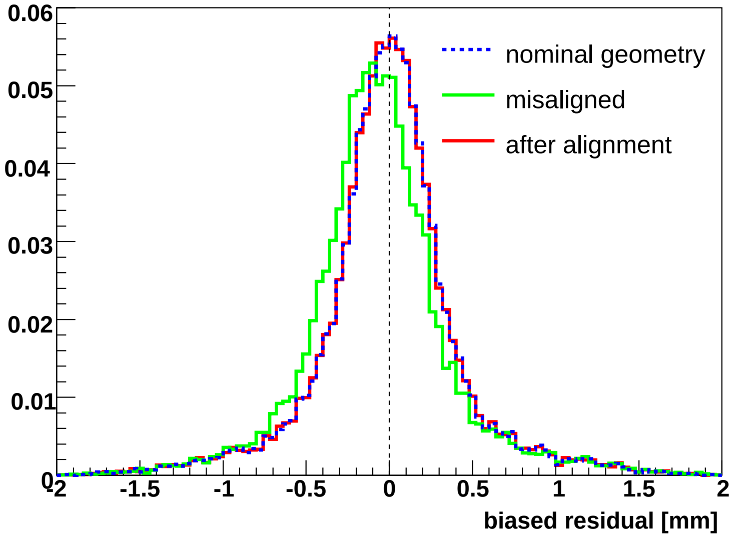

Figure 10. Residual distribution for one of the top sector TRT modules, with the nominal geometry (dashed, blue), with the misaligned geometry (green) and after alignment (red) in Monte Carlo.

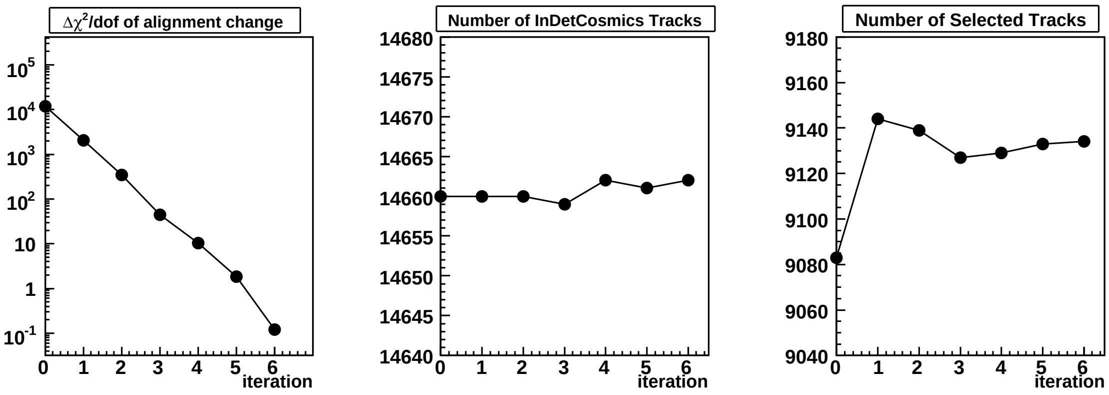

Figure 11. Module TRT alignment parameters local x (left) and local rot-y (right) as a function of iteration for the MC sample (without any initial misalignment).

of each SCT layer to the TRT axis. In order to compare to the alignment extracted from the data we have calculated an average of the survey measurements, shown in the bottom row in the table. We do not know the uncertainty on this average. Assuming that the uncertainty in each survey measurement is $10\,\mu$m (which is the precision with which the numbers were recorded) and that all measurements are uncorrelated, the uncertainty on the average is $10\,\mu$m. Judging from the spread of the measurements an uncertainty of $20\,\mu$m on the mean is probably more realistic. Table 4 shows the rotations and translations at $z = 0$ derived from the measurements in table 3.

Using reconstructed tracks the relative misalignment between TRT and SCT can be extracted from the cosmics data. In principle the alignment procedure described in the previous

	side A (z = 775)		side C (z = −775)	
	Δx [mm]	Δy [mm]	Δx [mm]	Δy [mm]
layer 1	-0.27	0.20	-0.54	0.45
layer 2	-0.32	0.21	-0.55	0.43
layer 3	-0.28	0.24	-0.55	0.45
layer 4	-0.29	0.19	-0.54	0.47
average	-0.29	0.21	-0.545	0.45

Table 3. Survey results: deviations of each of the four SCT cylinders with respect to the TRT.

η_x^{survey} [mm]	η_y^{survey} [mm]	$\epsilon_x^{\text{survey}}$ [mrad]	$\epsilon_y^{\text{survey}}$ [mrad]
-0.42	+0.33	+0.154	+0.165

Table 4. Translation and rotation at the origin extracted from the survey measurements in table 3.

sections is all we need: The global alignment is simply the alignment that we obtain if we combine all TRT modules in a single alignable object. In practice, the global misalignment is so large that it actually affects track finding which leads to a poor convergence of the alignment algorithm.

Therefore, we have also implemented a separate alignment algorithm that just extracts the single relative translation and rotation from track segments that have been fitted separately in TRT and SCT. The SCT segments are propagated to the nearest hit and expressed in the straw layer surface. At this surface a residual is formed in the Cartesian coordinates x and t_x, which are the position and direction of the track in the plane orthogonal to the wire. (These are the only two coordinates to which we have sensitivity in the TRT.) A χ^2 is constructed and derivatives to the translation and rotation are calculated in a manner analogous to the method for module alignment. The resulting algorithm is called `TRTGlobalAlignAlg` and resides in the `TRT_AlignAlgs` package.

Because the TRT only measures track parameters in the xy plane, a misalignment along the z-axis cannot be measured. Furthermore, because the cosmic trajectories are nearly parallel in our setup, we cannot reliably extract both Δx and Δy, nor both Δrot-x and Δrot-y. Since the tracks are more vertical than horizontal we choose to align Δx, Δrot-y and Δrot-z.

The result of the global alignment procedure is shown in table 5 for two runs that are at the beginning and at the end of a period of reasonably stable data taking. We attribute the difference in the rotations between the two runs to underestimated statistical uncertainties, rather than a real movement of the detector. To illustrate how poorly statistical uncertainties are estimated figure 12 shows the pull distribution for the two residuals in the two coordinates that enter the global alignment procedure, namely the relative position and orientation in the local TRT sensitive plane of the first hit. As one can see these pull distributions have a stan-

run	η_x [mm]	ϵ_y [mrad]	ϵ_z [mrad]
3007	-0.290 ± 0.001	0.277 ± 0.003	0.254 ± 0.002
3099	-0.289 ± 0.001	0.293 ± 0.003	0.226 ± 0.002

Table 5. Result for the global alignment of TRT ans SCT, without any 'internal' alignment corrections, using 5000 events. Note that the statistical errors are underestimated by at least a factor 7.

dard deviation of more than 7, which is due to the fact that the contribution from multiple scattering to uncertainty in the track parameters is ignored. Taking a factor 7 into account in the interpretation of the uncertainties in table 5 leads to the conclusion that the results are in perfect agreement. To improve the sensitivity we could use more statistics, but we have not done so yet.

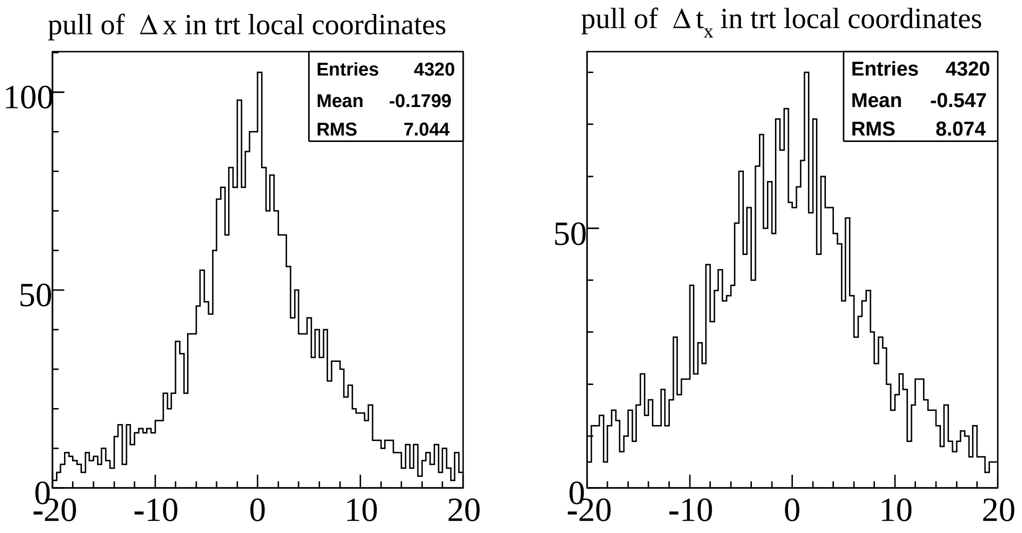

Figure 12. Pull for global track segment residuals in the global alignment procedure.

To compare the survey results to the alignment extracted from reconstructed tracks we need to take into account that we have only aligned for a subset of the available degrees of freedom: If the relative misalignment in the xy-plane is given by Δx and Δy the effective displacement seen in the x coordinate depends on the track angle and can be expressed as

$$\Delta x' \;=\; \Delta x - \Delta y / \tan(\phi_0) \quad , \tag{77}$$

where ϕ_0 is the azimuthal angle, the angle of the track with the x-axis in the xy-plane. Therefore, the observed translation and rotation are

$$\begin{aligned} \eta'_x &= \eta_x - \eta_y / \tan(\phi_0) \\ \epsilon'_y &= \epsilon_y + \epsilon_x / \tan(\phi_0) \end{aligned} \tag{78}$$

With the survey measurements in table 4 and $\langle \phi_0 \rangle \approx -1.92$ rad we obtain

$$\eta_x^{\mathrm{survey}\prime} \;=\; -0.300 \pm 0.008 \text{ mm} \qquad \epsilon_y^{\mathrm{survey}\prime} \;=\; 0.221 \pm 0.006 \text{ mrad} \tag{79}$$

where for the uncertainty we have assumed that the errors on the averages in table 3 are 0.02 mm and uncorrelated. The translation agrees perfectly with the result from the track based alignment in table 5 whereas the rotation is a bit on the low side.

6.6 Combined alignment of TRT and SCT in data

Unlike in the MC tests described in section 6.4, in real data the SCT is not perfectly aligned. Therefore, for the extraction of the TRT alignment in the data we have included the SCT modules in the alignment procedure. We have chosen a small set of the alignable degrees of freedom, namely only (local) Δx, Δy and Δrot-z in the SCT and Δx and Δrot-y in the TRT, all defined in the local module frame. We use the full statistics of run 3099 with about 24k events in the `InDetCosmicTrack` list of which about one third survive the track selection discussed in the beginning of this section. We apply alignment corrections only to modules with at least 25 hits, leading to a total of about 450 SCT modules and 12 TRT modules.

We run the alignment algorithm both in 'global' and in 'local' mode. Figure 13 shows the eigenvalue distribution of the matrix SAS^T discussed in section 2.5. When using the global method we suppress all eigenmodes with eigenvalue less than 40. The motivation of this threshold is discussed later in this section.

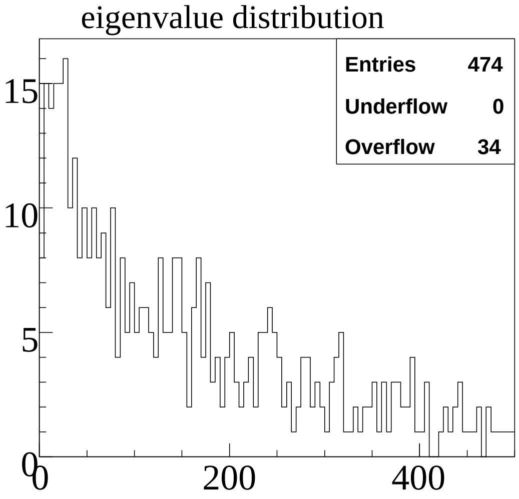

Figure 13. Eigenvalue distribution of the matrix SAS^T for the alignment of SCT and TRT in run 3099.

Figure 14 shows several observables as a function of the alignment iteration. The number of tracks in the `InDetCosmicTracks` track list decreases slightly in the first iterations. This could be due to a decrease in the fraction of tracks that is split. The number of selected tracks increases significantly after the first iteration, mostly due to a change in the track χ^2 distribution. The number of SCT hits and TRT hits per selected track is practically independent of the alignment. As an illustration of the improvement of the resolution, the last plot shows the RMS of the SCT residual distribution versus iteration. The global method converges to the 'ultimate' resolution in a single step, while the local method needs one iteration more.

Judging from the results in figure 14 the global and local method perform about equally well. We now look in more detail at the convergence. As a measure of the convergence we use the expected change in the total χ^2, defined in Eq.23. As explained in section 2, the value of $\Delta\chi^2$ is a measure of the significance of the change in the alignment parameters. Note that the full second derivative $d^2\chi^2/d\alpha^2$ is used in the calculation, including the module correlations. For the global method, $\Delta\chi^2$ can also be written as the sum of the contributions from the

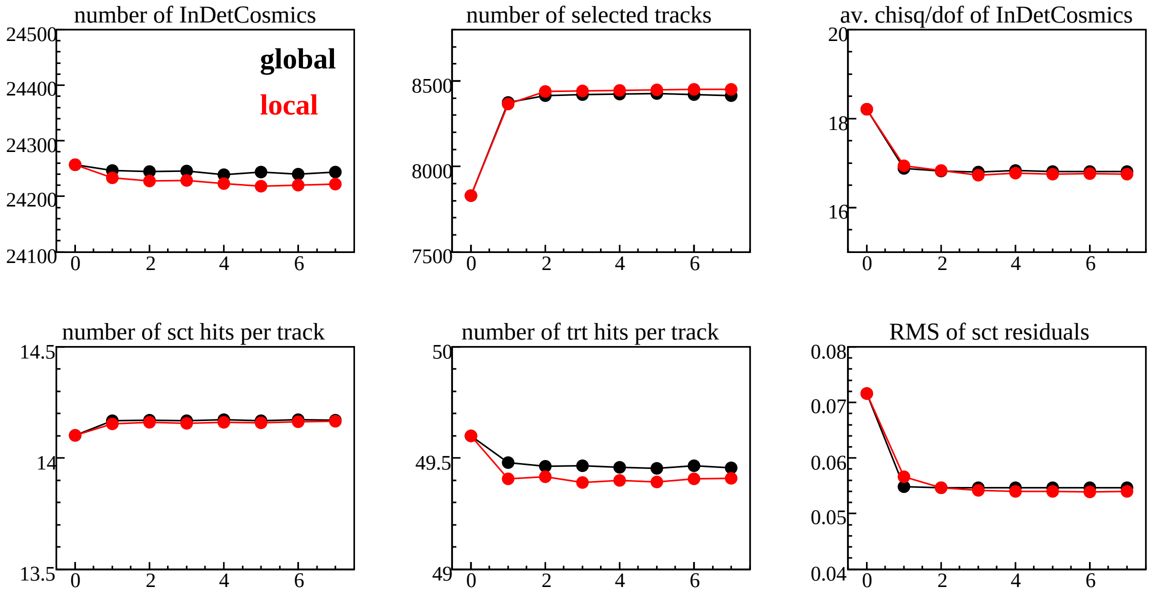

Figure 14. Several observables as a function of alignment iteration in the combined SCT-TRT alignment using both the global (black) and the local (red) method alignment method in run 3099. Note the suppressed zeros.

different eigenmodes, as in Eq.34.

Figure 15(a) shows (minus) $\Delta\chi^2$ as a function of iteration for both the global and local method. Ideally, if the tracks used in each iteration are the same, $\Delta\chi^2$ converges to zero. If the alignment problem is entirely linear, we expect the global method to converge in a single iteration, whereas the local method needs iterations to resolve the effect of module correlations. In practice, small fluctuations in the number of reconstructed tracks and in the hits assigned to the tracks lead to changes in the alignment constants for both methods even after several iterations. We observe in figure 15(a) that the global method indeed converges faster in the first iterations. However, to our surprise the global method does not converge to zero, but actually *diverges* slowly after a few iterations. We study this effect in more detail in section 6.7.

Note that, with a total of about 1400 alignment parameters, the asymptotic value of the $\Delta\chi^2$ per degree of freedom is about 1. In other words, the change in alignment between iterations is still of the order of 1 standard deviation even after several iterations. This is largely the effect of the changes in the input to the alignment between iterations, as mentioned above.

In the global method certain alignment modes are ignored because of the cut on the eigenvalue. To illustrate the effect of this cut on the χ^2 convergence, figure 15(a) also shows the $\Delta\chi^2$ that would have been seen if all modes were included in the calculation of α. Not surprisingly, the asymptotic value of $\Delta\chi^2$ is much larger now. It corresponds to the $\Delta\chi^2$ of the modes that are suppressed. Figure 15(b) shows the norm of the first derivative as a function of iteration. Although the scale of the norm is not entirely meaningful, the plot still illustrates the effect of the suppression of weak modes: in the global method the first derivative does not converge to zero.

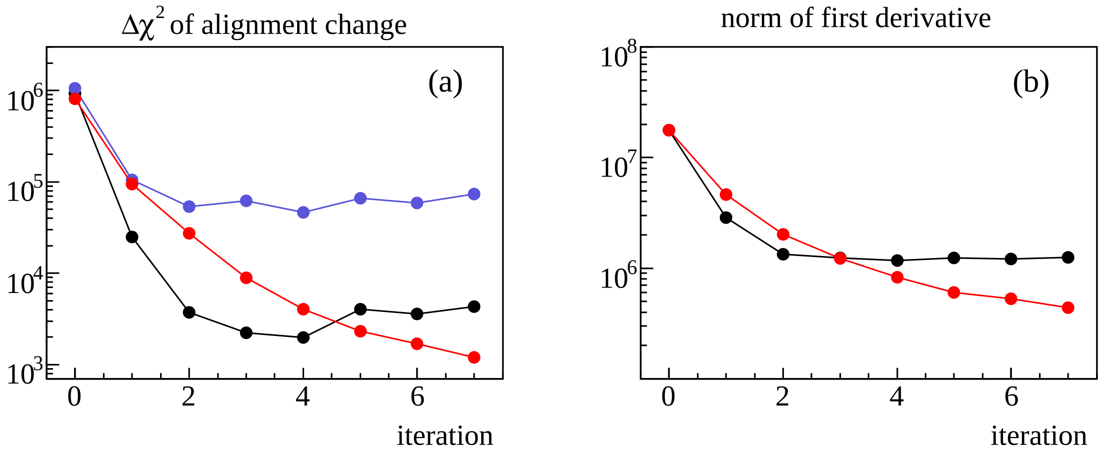

Figure 15. The change in the total χ^2 (a) and the norm of the first derivative (b) as a function of iteration for the global (black) and the local (red) alignment method in run 3099. The blue curve in the left figure is the change in the total χ^2 of the global method if also truncated modes are included in the calculation of the χ^2.

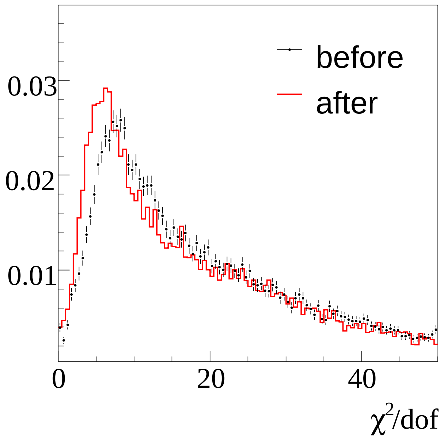

Figure 16. Distribution for χ^2 per degree of freedom for tracks on the `InDetCosmicTracks` list before and after alignment in run 3099.

Figure 16 shows the distribution of the χ^2 per degree of freedom for tracks in the `InDetCosmics_Tracks` list before and after alignment. It illustrates that the alignment actually only affects a small fraction of the tracks, simply because most tracks are totally dominated by multiple scattering. Note that in the alignment formalism the change of the total χ^2 of all tracks used in the alignment is given by Eq.23. In practice, the track sample differs from one iteration to the next, so the χ^2 change in the track sample is not in perfect agreement with the expectation from the change in the alignment constants.

Figure 17 shows the change in the TRT alignment parameters as a function of iteration. The alignment convergences reasonably fast. Figure 18 shows the residuals in the different TRT modules before and after the module alignment. (The global TRT-SCT alignment was

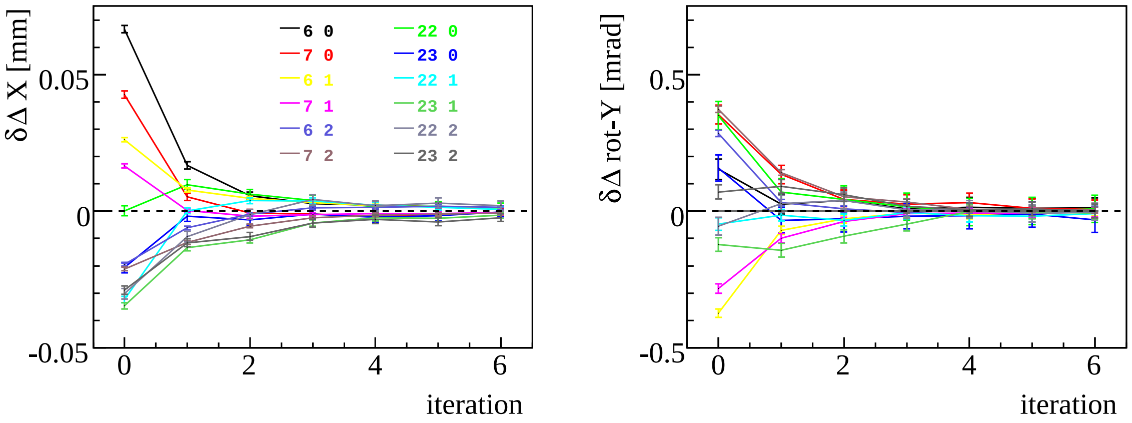

Figure 17. Change in local Δx and Δrot-y for all TRT modules as a function of iteration in the alignment of run 3099.

applied in both scenarios.) Because multiple scattering is not modeled in the track fit, the width of the residual distribution (approximately 400 μm) is much larger than the hit resolution of the TRT. Since the alignment corrections are small compared to the width of the residual distribution, the positive effect of the alignment is hardly visible in the residual distribution. This is in sharp contrast with the situation in the SCT: figure 19 shows the SCT residual distribution divided by layer and module side. The improvement is well visible because the module displacements are large in comparison with the hit resolution.

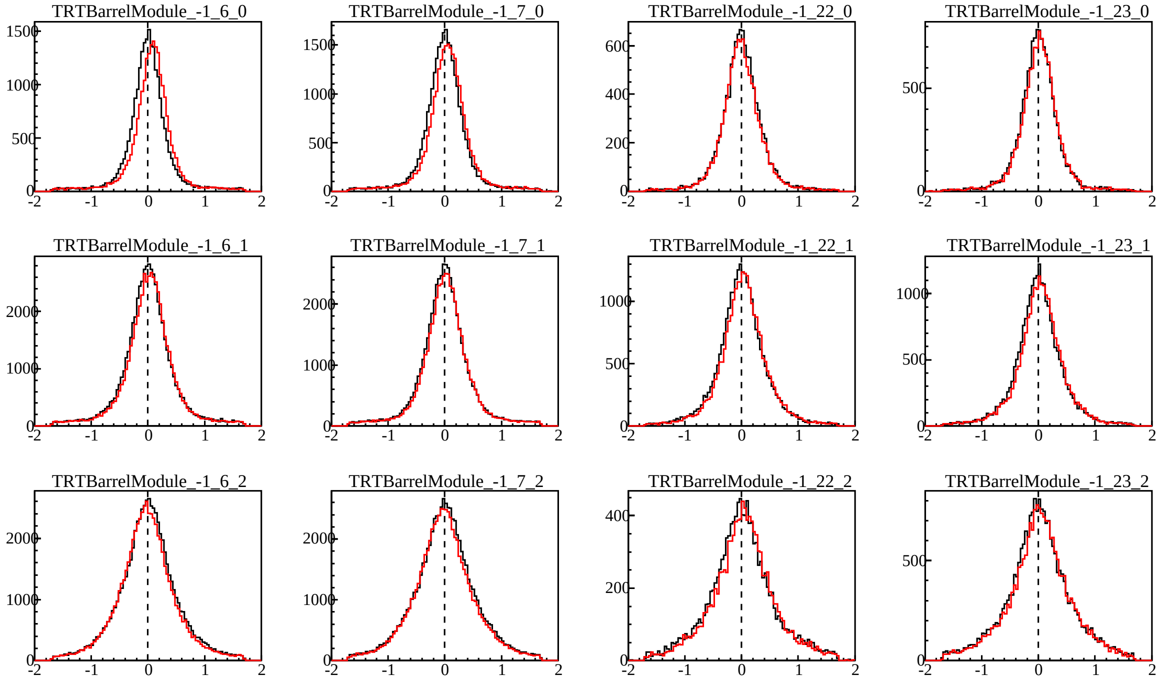

Figure 18. Residual distributions for TRT modules before and after alignment in run 3099.

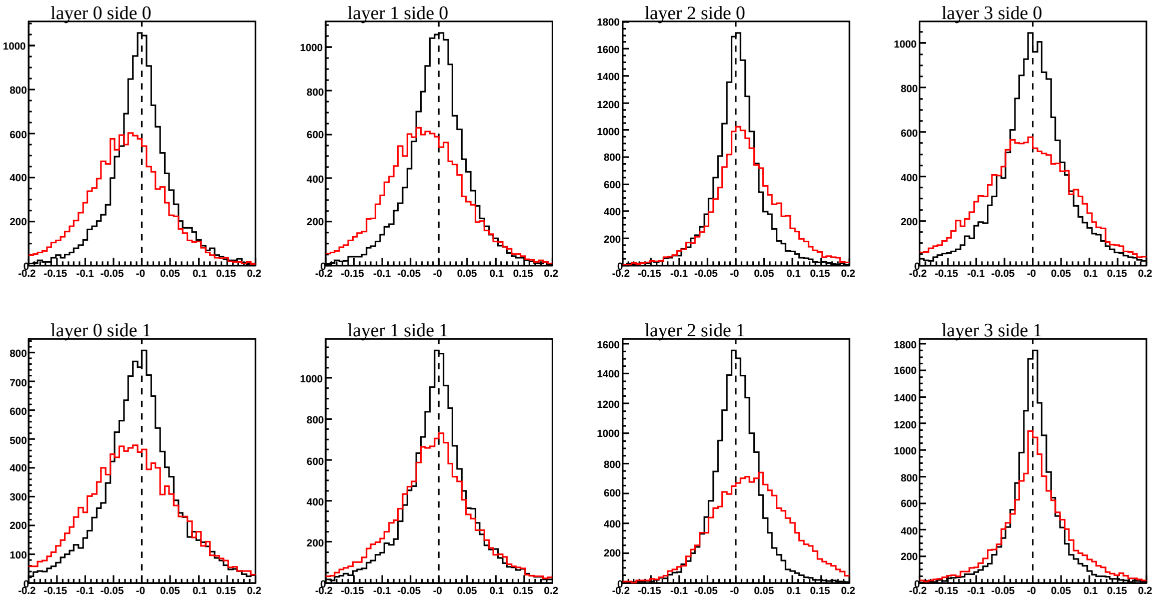

Figure 19. Residuals distributions for different SCT layers before (red) and after (black) alignment in run 3099, aligning for (local) Δx, Δy and Δrot-z.

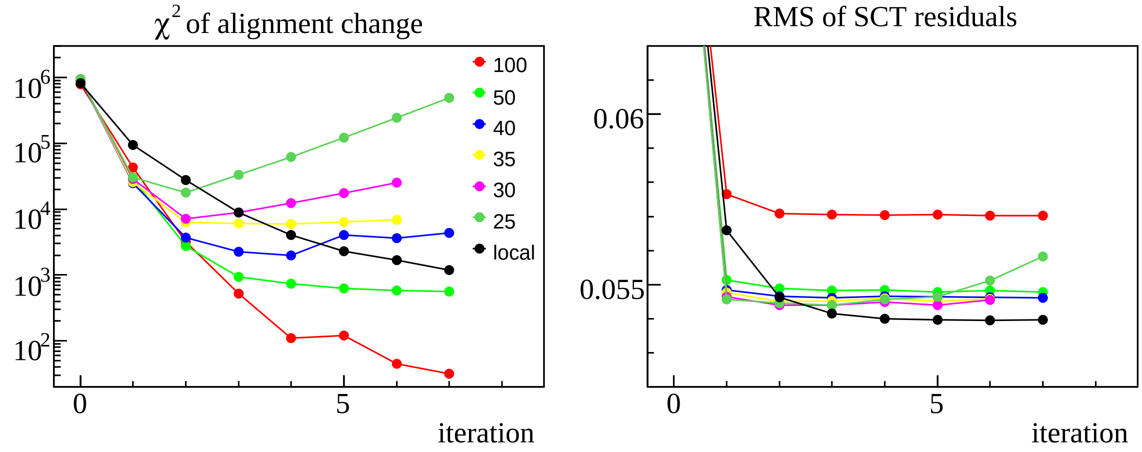

Figure 20. Alignment change χ (left) and RMS of SCT residual distribution (right) versus iteration for different cuts on the eigenvalue in the global method.

6.7 Convergence problems with the global method

While running the alignment using the global method we observed a problem when using a small cut in the eigenvalue spectrum: The alignment converges fast in the first few iterations but starts to diverge afterward. This divergence depends on the value of the eigenvalue cut, as illustrated in figure 20. With a χ^2 cut of 100 the convergence is perfect, but there is an adverse effect: with the available statistics in run 3099 the SCT alignment is poor because too many modes are removed.

Note in figure 20b that the SCT resolution obtained with the local method is actually

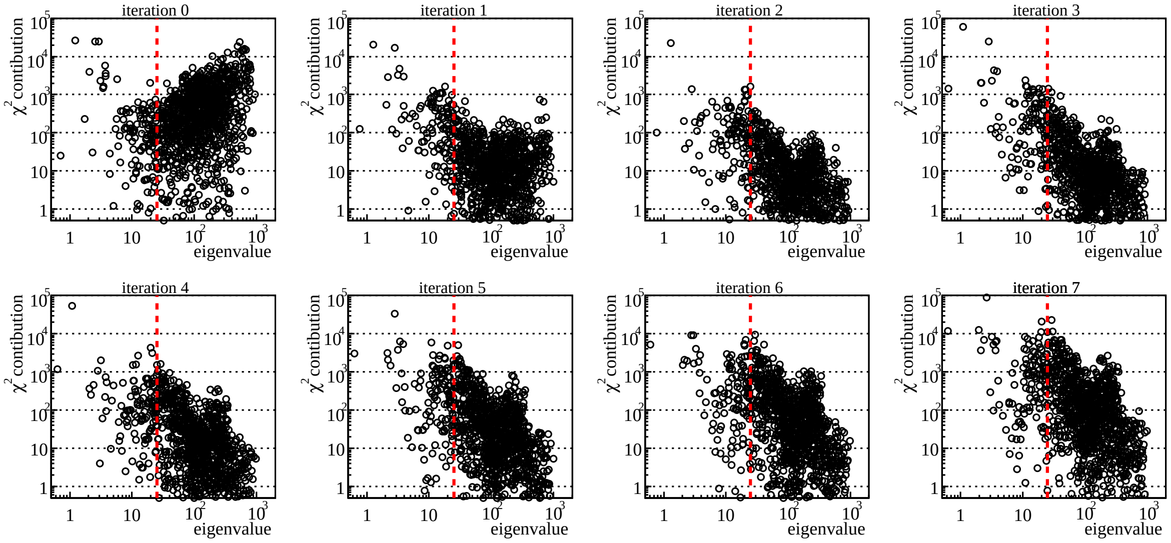

Figure 21. χ^2 contribution of modes versus their eigenvalue per iteration in the alignment with an eigenvalue cutoff of 25. (The vertical red line represents the value of the cut.)

better that that with the global method for any of the applied eigenvalue cuts. This is kind of expected and should not worry us. The eigenvalue cut is supposed to remove modes that we cannot align for with the available statistics. When we include those modes, like in the local method, we are actually aligning for statistical fluctuations, too. A real test of the quality of the alignment is a comparison of the residuals on a run that was not used for the alignment. We have not performed such a comparison.

Figure 21 shows the distribution of the χ^2 contribution of the alignment mode versus the eigenvalue of that mode for all iterations. It illustrates that the divergence is not restricted to certain modes but affects the entire eigenvalue spectrum. It is as if 'noise' from the poorly constrained modes enters into the well constrained modes.

A convergence problem can have only one origin: the 2nd derivative is not correct. This is surprising because we had convinced ourselves with the test reported in section 5 that the derivatives are in order. Furthermore, the local-method alignment converges perfectly fine indicating that at least the local (within-module) 2nd derivative is not entirely wrong.

Unfortunately, we have not been able to identify the cause of the divergence. The analysis in 5 suggests that there are no algebraic errors in the derivatives. Therefore, we suspect that there are rare numeric effects ('pathological events') that make the second derivative behave poorly. We have practically excluded that the problem is caused by an occasional problem in the track fit. The track fit information relevant to the second derivative consists of the derivative vector H and the covariance matrix C. A mistake in the derivative vector affects the alignment but it is hard to understand how it could make the alignment problem diverge. To rule out possible errors in the calculation of C, we have recalculated C explicitly in the alignment program, using Eq.7, leading to exactly the same divergence.

We have tried to 'moderate' the convergence by introducing a constraint to the nominal alignment, as discussed in section 2.6. We extract the uncertainties assigned to the nominal alignment by studying the distributions of the observed alignment constants after a converged alignment, shown in figure 22 to be discussed later on. In the SCT we find that suitable values are 100 μm in local-x, 500 μm in local-y and 2 mrad in local-rot-z. These uncertainties are much larger than the actual derived uncertainties on the alignment parameters. As a result, this constraint is totally irrelevant for the alignment and does not solve the problem.

We have also tried to reduce the number of degrees of freedom by aligning only for Δx in the SCT, *i.e.* just a single parameter per module. Though this reduced the pace of the divergence, it does not eliminate it.

Finally, we have considered the effect of small iteration-dependent changes in the 2nd derivative, affecting the diagonalization. If the eigenvectors change from iteration to iteration, one may expect that the eigen modes of the system are not entirely decoupled, which could explain how noise from weak modes can enter the well constrained modes. We have tested this hypothesis by running the alignment with a fixed 2nd derivative, obtained from the first iteration. The divergence remained.

Our conclusion is that we do not understand the origin of the divergence. However, the fact that the divergence sets in later if a larger eigenvalue cut is applied, makes us hopeful that the problem does not occur when much more statistics is used. We have not had time to perform a convergence test with significantly more statistics.

6.8 SCT module displacements

Figure 22 shows the distribution of the alignment parameters of SCT modules. The RMS of the distribution is about 50 μm in x, 500 μm in y and 1 mrad in rot-z. The uncertainties in the y direction (along the strips) are much larger than in x. Normalized to the estimated alignment uncertainty, the misalignments in y are less significant. Note that in these cosmics data the alignment uncertainties are not reliable because multiple scattering is ignored in the track fit.

It is possible to find some structure in the SCT module displacements. Figure 23 shows the distribution of the alignment parameters for individual SCT layers. Layer 2 is displaced with respect to the other layers, both in x and in rot-z. Within the limited statistics of run 3099 we have not found evidence of a correlation with the z coordinate (the ‘eta’ of the module).

6.9 TRT module displacements

Table 6 shows the displacement and rotation of TRT modules as extracted with the alignment procedure using the local method. The results for the global method are very similar, though with about 10 % higher statistical uncertainties: The difference is due to the contribution from module correlations which are ignored in the local method. The uncertainties are statistical only and, as explained earlier in this section, significantly underestimated.

The last column of the table shows the correlation coefficient between displacement and rotation. This correlation is determined by the illumination of the module by the tracks used for alignment: the coefficient is zero only if the center of gravity of the sample of tracks coincides with the rotation axis (which is approximately in the center of the module in the local frame).

Figure 24 gives a graphical representation of the translation and rotation. (Note that the values are scaled.) A clear pattern emerges. The rotation of the module depends on the layer.

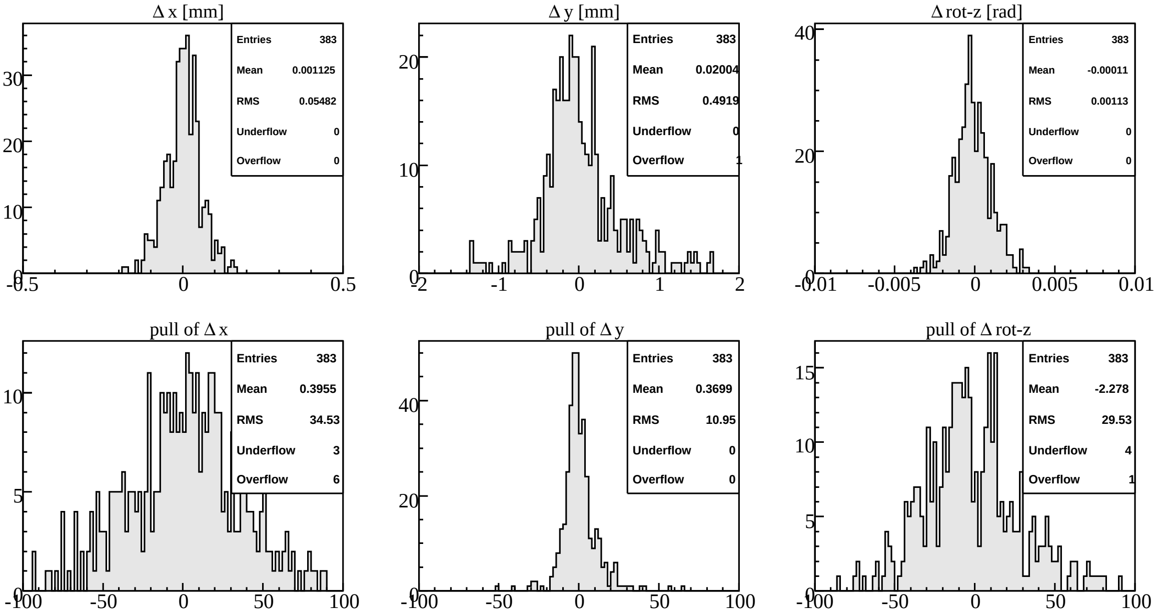

Figure 22. Distribution of alignment parameters for modules in the SCT for run 3099, absolute (top) and relative to the uncertainty (bottom).

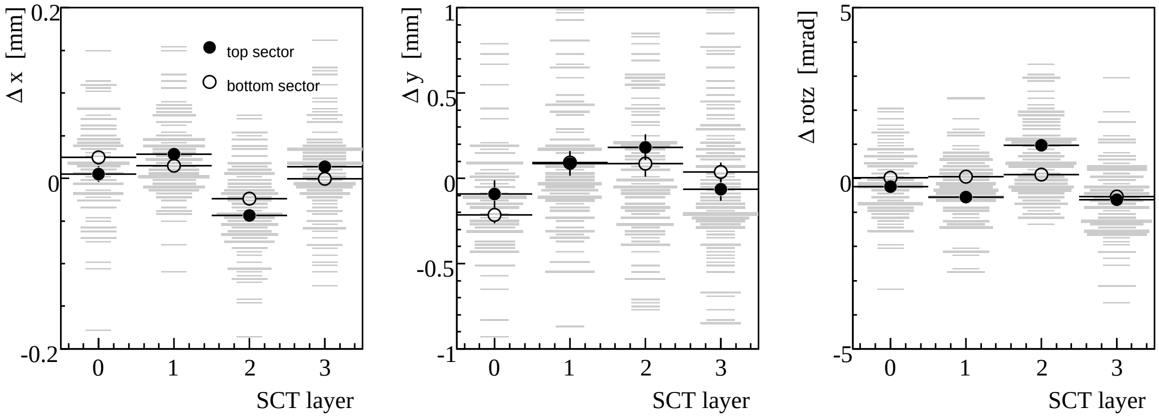

Figure 23. Distribution of alignment parameters for modules in the SCT versus SCT layer for run 3099. The points represent an unweighted average, separated in modules in the 'top' region and in the 'bottom' region.

It seems that modules type I and III (layers index 0 and 2) rotate in the same direction, whereas type II modules rotate in opposite direction, relative to the nominal alignment. A conclusive interpretation of these results cannot be obtained with cosmics data. We postulate that this pattern is either the result of an elliptical distortion of the TRT (or possibly the SCT) or that there is a problem in the implementation of the TRT geometry.

Δx [mm]	Δrot $- y$ [mrad]	corr.coef.		
6	0	0.0972 ± 0.0014	-0.049 ± 0.037	0.51
6	1	0.0398 ± 0.0008	0.538 ± 0.015	0.11
6	2	-0.0214 ± 0.0009	-0.389 ± 0.012	0.23
7	0	0.0469 ± 0.0013	-0.493 ± 0.034	0.61
7	1	0.0062 ± 0.0008	0.448 ± 0.016	0.16
7	2	-0.0497 ± 0.0007	-0.549 ± 0.011	0.18
22	0	0.0149 ± 0.0018	-0.478 ± 0.049	0.46
22	1	-0.0394 ± 0.0011	0.053 ± 0.022	0.07
22	2	-0.0471 ± 0.0018	0.018 ± 0.031	0.48
23	0	-0.0379 ± 0.0018	-0.057 ± 0.046	0.57
23	1	-0.0777 ± 0.0011	0.406 ± 0.024	0.20
23	2	-0.0846 ± 0.0014	-0.292 ± 0.025	0.55

Table 6. TRT alignment parameter in the local TRT module frame, as determined from the SCT-TRT combined alignment with the local method. The uncertainties are statistical only and understood to be severely underestimated. The last column shows the correlation coefficient for translation and rotation.

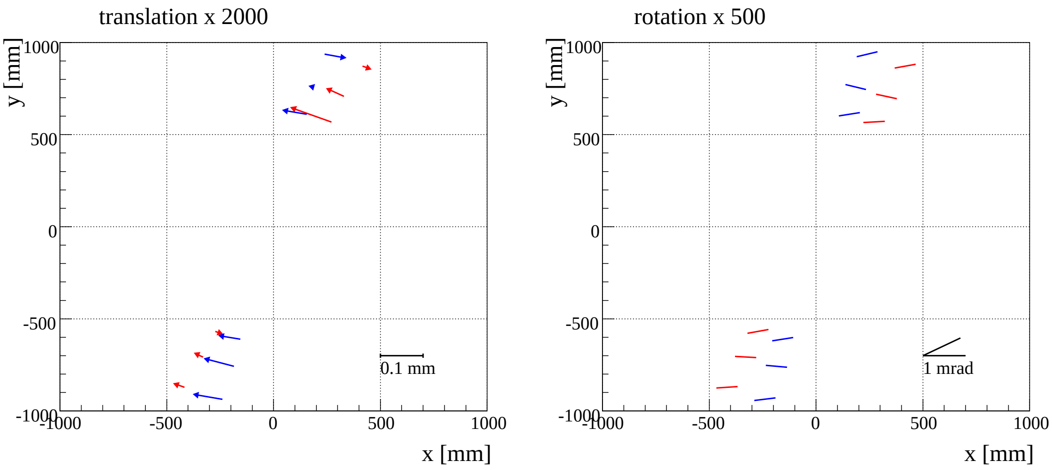

Figure 24. Measured translations (left) and rotations (right) of TRT modules in run 3099. The direction and size of the arrows in the left plot represent the direction and size of the translations. The rotation of the module in the right plot is proportional to the angle of the bar with the horizontal axis.

6.10 TRT Module deformations

The size of the SR1 cosmics data sample allows a limited study of TRT module deformation, *i.e.* deviation from the module rigid body approximation. Such deformations are expected because of construction imperfections, mechanical stress, small deviation from the nominal design of the supporting frame, etc. Whatever are the causes, they effectively produce a displacement of the hit position depending on the position inside the module. If their effects are bigger than the detector resolution, a correction must be applied. Survey data from module construction and installation, albeit available, do not provide strong constraints for alignment purposes. Furthermore, it cannot be excluded that further deformations have occurred after installation. Therefore the best way to investigate the presence of module deformations is using tracks reconstructed in real data. It will also allow to quantify their impact (if any) in the TRT performance.

As mentioned in section 6.9, residuals in the TRT are mostly sensitive to the Δx and Δrot-y degrees of freedom (in the TRT local frame). Labeling the variation in the alignment parameters associated with these degrees of freedom as $\delta\Delta x$ and $\delta\Delta$rot-y respectively, we expect, after the alignment described in section 6.9, the average $\delta\Delta x$ and $\delta\Delta$rot-y to be zero. However, module deformations will show up as a dependence of $\delta\Delta x$ and $\delta\Delta$rot-y on the position inside the module. We have looked in particular at the dependence on the position *along* the straw axis (the local y direction or global z direction). Different module deformation scenarios, will lead to different observable patterns in the alignment corrections, for example

- a module "twist", a rotation of the two module plates in opposite direction, leads to a rotation Δrot-y that depends linearly on the coordinate along the straw;
- a module "bow" or in the local x plane leads to a displacement Δx that depends approximately parabolically[7] on the coordinate along the straw. This also holds for gravity induced wire sag.

In general more complicated deformations and patterns cannot be excluded.

In interpreting the observed effects one should take into account that, in the current setup, this analysis cannot be performed independently of the alignment of the SCT: remaining alignment deviations in the SCT can appear as module deformations in the TRT. This is in particular the case because, in contrast to the TRT, the SCT does not form a rigid body along the global z axis. (This is also why, after a proper understanding of module deformations, the TRT provides a useful constraint in the SCT alignment.) Since we cannot truly separate module deformations from remaining SCT misalignment, we will assume that the SCT is perfectly aligned and talk about the "effective" deformation of a TRT module. As in the previous sections we compare the observed effects to their statistical uncertainties, knowing that those are significantly underestimated. To minimize statistical fluctuations, we have used almost all the available cosmics data for this study. Eventually about 70K tracks, collected in events from run 2935 to run 3100 and reconstructed with the alignment constants presented in section 6.6 passed our selection requirements (reported in section 6.3).

The local y coordinate was sliced with a granularity big enough to allow in each bin for a significant amount of hits. Despite that, non-negligible bin statistic fluctuations cannot be completely removed because the hit distribution along the wires is not uniform. This is partially due to the particular scintillator setup (fig. 5) whose geometric acceptance varies along the TRT module length. Figure 25 shows the hit multiplicity along the straw for the different layers in phi sectors 6 and 22 on a common scale. Higher statistical fluctuations at the edges of the modules and for modules of type I are hence unavoidable. (The first 9 layers in type I modules (layer 0) have no sensitivity in the region between -400 and 400 mm.) For the bottom sector it is particularly hard to identify deformation patterns of the type above, because only one side of the modules is instrumented. Hence in the following we will show results only for top sector modules (phi sectors 6 and 7).

We have already mentioned several times that the alignment errors are underestimated. To have an idea on how a "null hypothesis" would look like, we used a MC sample searching for

[7] Whether it is a parabola or really a catenary is probably irrelevant at this (and any future) stage!

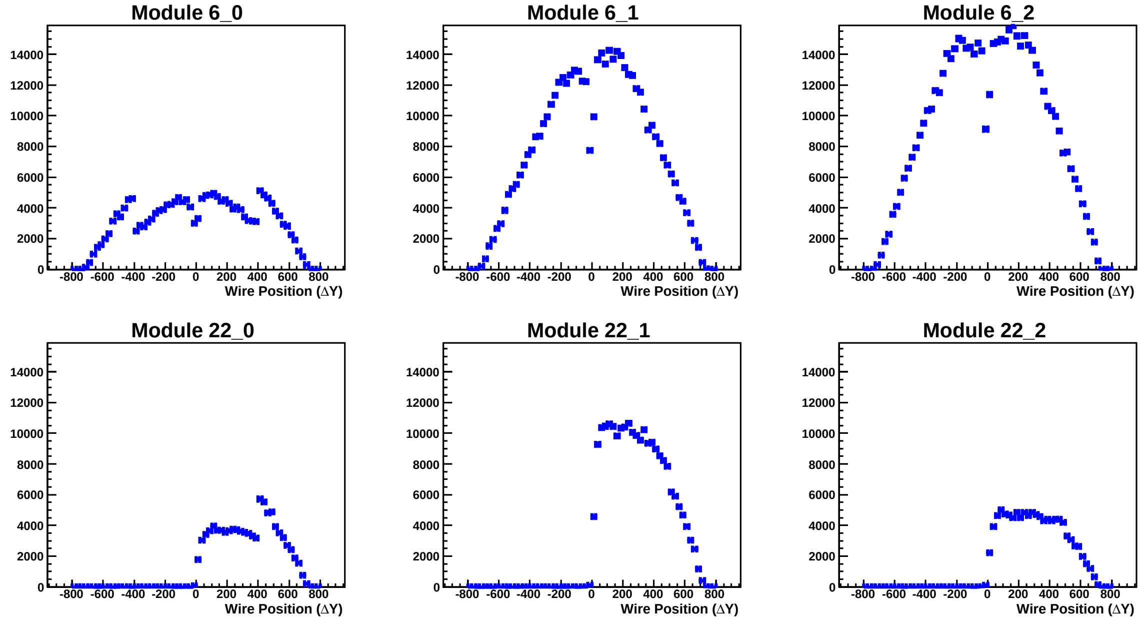

Figure 25. Number of hits distribution for modules on phi sector 6 (top row) and phi sector 22 (bottom row). The distributions reflect the scintillator acceptance along the modules. Modules on the bottom sector have only the positive half instrumented. The first 9 straw layers of type I modules (layer 0) have a dead region between -400 and 400 mm causing the number of hits to drop in that region. The sharp dip at around zero is due to the wire joint located at the module center.

module deformations. Of course we do not expect to find any, so this is a good exercise to test the sensitivity of our approach. In fact, whatever we observe can definitely be interpreted to be a pure statistical fluctuation. In addition we employed a factor three less statistics with respect to what we have in the data in order to have a safe reference for what we cannot claim to be a real deformation.

In figure 26 $\delta\Delta$rot-y as a function of the wire coordinate is shown in the case of a MC reconstructed with a nominal geometry (no induced misalignment). We can see that the distributions are not perfectly flat with more fluctuations for type I modules and at the module edges, as expected. Although the actual bin to bin fluctuations are much larger than the assigned uncertainty, no clear pattern is observed, as expected.

In figure 27 we report the same distributions as they appear in real data. We see clear deformation patterns in some of the modules, but in fact no two modules look alike. The straight line approximation, the simple model of a module twist, fits only poorly to the observed data, except for module 7 – 2. Module 6 – 2 looks as if its center is rotated with respect to the two end plates, a twist of a different kind.

To convince ourselves that these are not purely statistical effects, we divide the data sample in three non-overlapping sets. If the patterns are real, we should be able to identify them on all of these three chunks of data. In figure 28 the same plots as in figure 27 are shown for the three different subsamples. As can be seen the deformation patterns observed for modules 7 – 2 and 6 – 2 exists in all run periods.

In figure 29 we report instead the $\delta\Delta x$ again versus the local y coordinate. In this case it

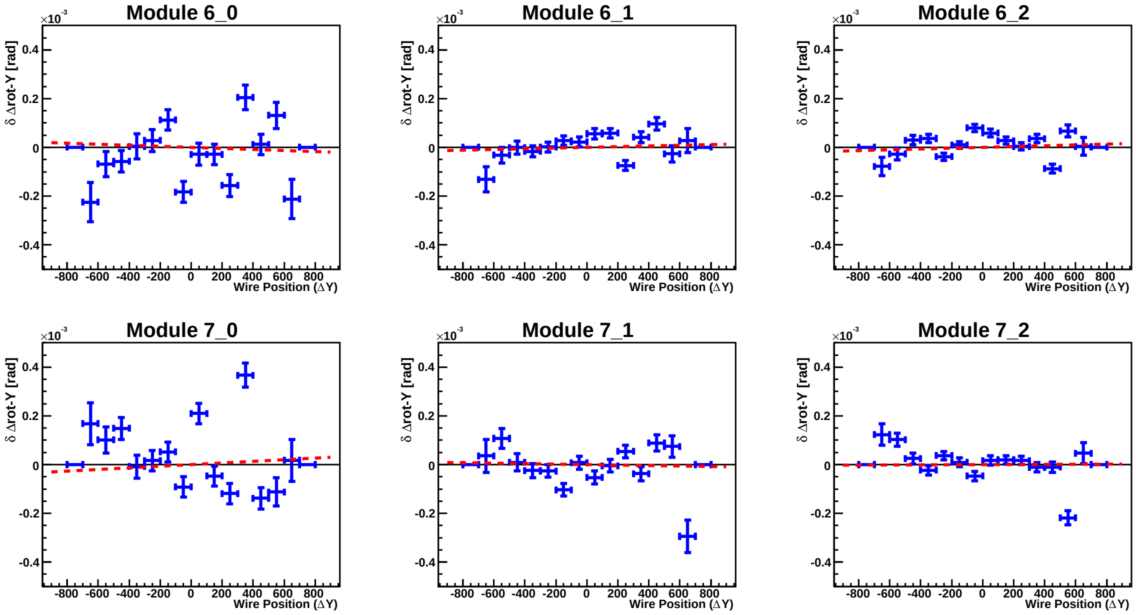

Figure 26. In this plot the $\delta\Delta$rot-y value as a function of the wire coordinate Δy is shown for the *simulated* cosmic sample. The dashed red line is the result of a fit to a straight line.

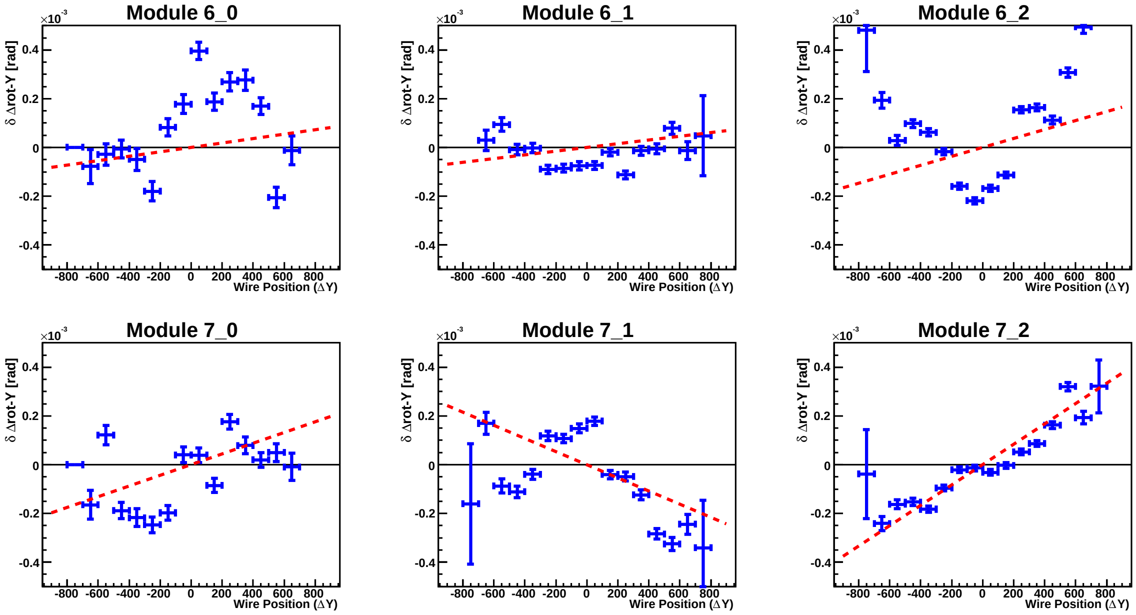

Figure 27. In this plot the $\delta\Delta$rot-y value as a function of the wire coordinate Δy is shown for *real* data. The dashed red curve is the result of a fit to a straight line.

seems even more difficult to provide an interpretation of the results. We might guess that the modules $7-2$ and $6-0$ are slightly bent in the xy plane, whereas modules $7-1$ and $6-2$ are rotated along the local z axis (the axis perpendicular to the straw layer planes). It is of course also not excluded that these effects are induced by other deformations projected on the

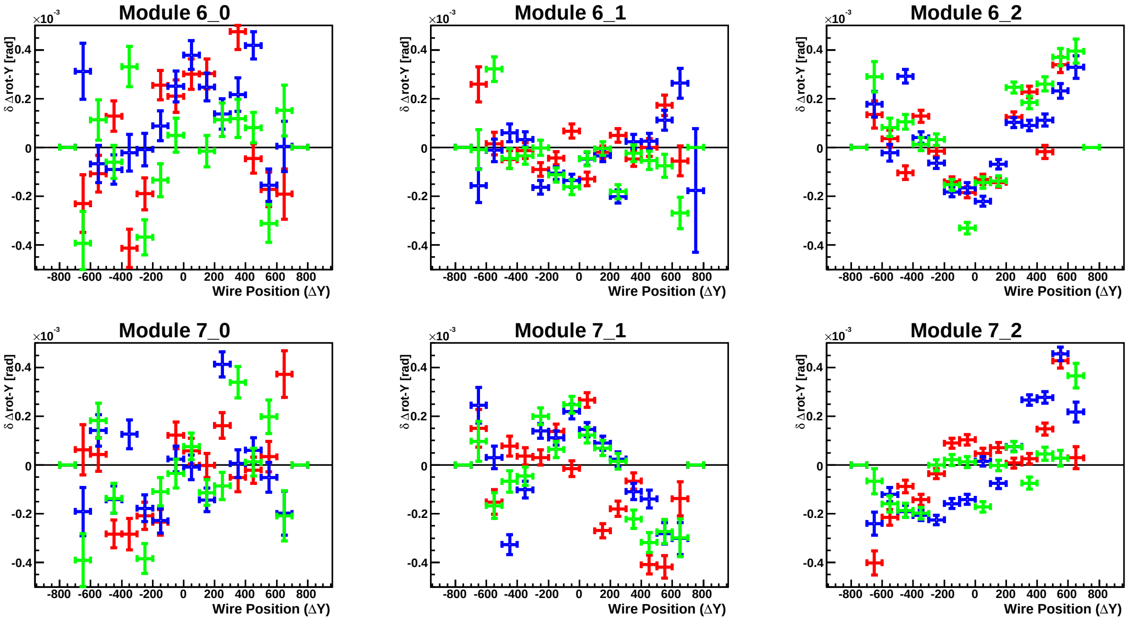

Figure 28. In this plot the $\delta\Delta$rot-Y value as a function of the wire coordinate ΔY is shown for three different (not-overlapping) sets of runs. Except for modules of type I (layer 0) where the high fluctuations prevent to draw any conclusion, for modules of type II and III (layer 1 and 2) the same patterns are observed for all three sets of runs. We interpreted this as a strong evidence for such patterns to be "real" and not due to statistical fluctuations.

xy plane.

Even without a full understanding of the results presented so far, we can conclude that TRT modules are not rigid bodies and that we are sensitive to deviations even with tracks of relatively poor quality. To allow for a correction for these deformations in the reconstruction, the Atlas TRT geometry implements translations at the level of an individual straw: for each barrel straw, 4 parameters are used to move the endpoint of the two wire halves in the straw layer plane. (This plane is approximately local x, but it is not exactly the same for each straw layer in a module.) In this projection a rotation is equivalent to a shear movement. Therefore, a module twist can be implemented by moving the endpoints of the wires

$$\delta x_i \;=\; \alpha y_i$$

where α is the twist angle and y_i is the local y-coordinate of the endpoint. This is currently how the twist due to end-plate rotation is applied in the alignment algorithm, the only module deformation that was actually implemented and aligned for until now. However, as the presented results indicate, this simple twist is only one out of many possible deformation scenarios. To reach an optimal performance these deformations need to be explored with a significantly larger track sample and for a larger set of modules. Hopefully, such a data sample can be obtained from cosmics events in the pit.

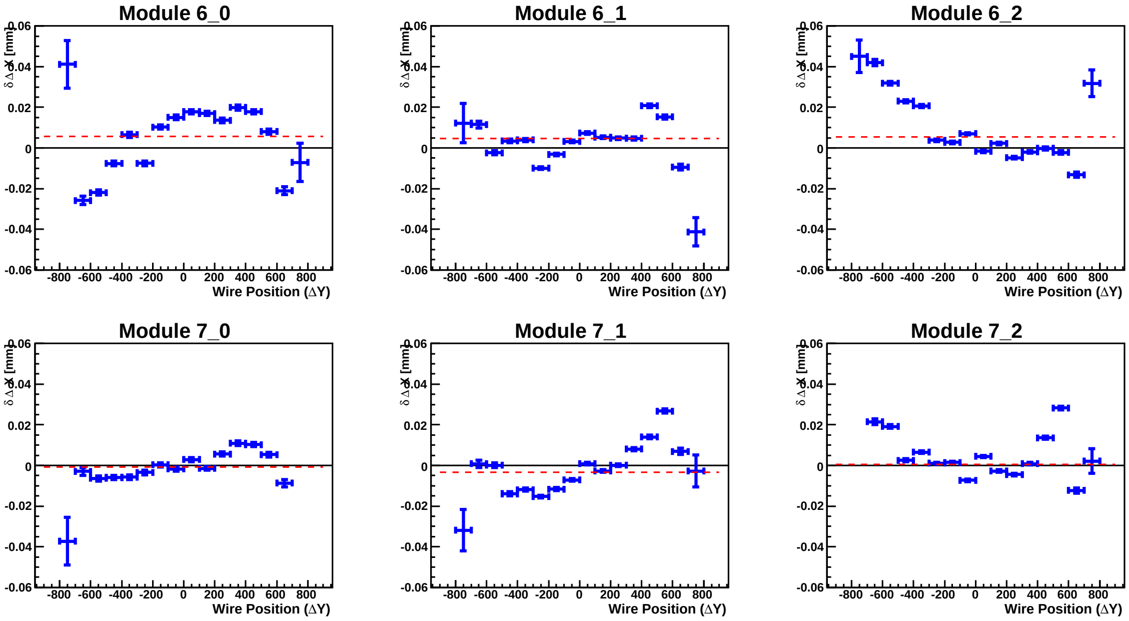

Figure 29. In this plot the $\delta\Delta X$ value as a function of the wire coordinate ΔY is shown for *real* data. Deviations from strait line are observed suggesting some sort of module bending along the straw layer plane (ΔX)

7 Conclusions and Outlook

In this note we have described a procedure for the alignment of the ATLAS Inner Detector, developed for the analysis of cosmics data taken in SR1. Our algorithm is based on the minimization of a χ^2 accumulated over a large sample of tracks. The mathematical formalism, derived in the first section of this report, relies on a global track fit and is equivalent to that presented in [4, 1]. We have shown, possibly for the first time, that the minimum χ^2 condition takes a simpler form than presented in these references: due to a cancellation that occurs if the track fit is at minimum χ^2, residuals of constraints that do not directly depend on the alignment parameters, do not explicitly enter the minimum χ^2 condition. Examples of such constraints are hits in a 'reference' systems, scattering angle constraints and vertex constraints. This significantly simplifies a practical implementation. We have also proposed a method to put the eigenvalues of the matrix that represents the derivative of the χ condition, on a common scale, which facilitates the selection of poorly constrained alignment degrees of freedom.

The algorithm has been implemented in the Atlas Athena software framework. It has been applied to the alignment of TRT and SCT modules with SR1 cosmics data, both in the 'global' (or closed form) and the 'local' (or iterative) χ^2 minimization approach. We have found that the alignment using the global method converges faster, but that, on the data sample used for the tests, a process occurs which leads to divergence already after a small number of iterations, even if weak modes are truncated. We have not fully identified the source of this divergence, but we have provided evidence that it is due to a too low threshold in the eigenvalue distribution, a value which is a compromise between the desired sensitivity to misalignments and the size of the data sample.

We have found a pattern in the misalignment of TRT modules suggesting that there is either a small mistake in the geometry description or an actual deformation of the TRT support structure. We have also performed an analysis of module deformations in the TRT. It was shown that such deformations exist and go beyond the simple twist model. At the time of writing only the alignment for a TRT barrel twist is implemented, but the algorithm can easily be extended to other parameterized deformations in the TRT or other subdetectors. (Pixel bow deformations are another important candidate.)

The module alignment algorithm has originally been written for the TRT barrel internal alignment, but the algorithm has been extended to the full TRT, SCT and Pixel. Recently, the algorithm has been applied to simulated multi-muon events in a detector with realistic misalignments and a magnetic field. These studied are in progress and not reported in this note.

Acknowledgments

For the development of the software and the preparation of this note, the help of our colleagues in the TRT, the Inner Detector reconstruction, offline commissioning and the alignment group has been extremely valuable. We would like to thank in particular Peter Hansen, whose earlier contribution to TRT alignment served was a basis for our work, and Daniel Froidevaux for his careful reading of this manuscript.

A Vertex constraints

In this appendix we briefly describe a method to include vertex and kinematic constraints in the alignment process. Our method does not require a refit of the track candidates. Instead, it exploits that the constraint from the vertex fit can be propagated to the residuals and their covariance through the track covariance matrix.

Our method relies on the existence of

- a global track fit that gives us access to the residual vector r, the derivative vector H and the covariance matrix C. We split the track parameter set x in the 'perigee' parameters q used by the vertex fit (which may or may not include the momentum) and the parameters θ that the vertex fit ignores, such as the scattering angles;

- a vertex fit that can provide us with updated track parameters $\tilde{q}^{(i)}$, a new covariance matrix $\tilde{C_q}^{(i)}$ for each track and eventually the correlation between the updated track parameters of different tracks. The vertex fit may include kinematic constraints, such as an invariant mass constraint;

For the calculation of the derivatives of the total χ^2 to the alignment parameters, we need

- the updated residual vector $\tilde{r}^{(i)}$ for each track

- the full covariance matrix for the residuals, which for a two track vertex schematically looks like

$$R = \begin{pmatrix} V^1 & 0 \\ 0 & V2 \end{pmatrix} - \begin{pmatrix} H^{(1)}\tilde{C}^{(1)}H^{(1)^T} & H^{(1)}\tilde{C}^{(1,2)}H^{(2)^T} \\ H^{(2)}\tilde{C}^{(2,1)}H^{(1)^T} & H^{(2)}\tilde{C}^{(2)}H^{(2)^T} \end{pmatrix} \tag{80}$$

To understand how the information from the vertex fit is propagated to the remaining track parameters we study a simple system with two parameters (a, b) and a covariance matrix

$$V = \begin{pmatrix} V_{aa} & V_{ab} \\ V_{ba} & V_{bb} \end{pmatrix} \tag{81}$$

where $V_{ba} = V_{ab}^T$. Now, assume that there exists a procedure that adds information to the system such that we obtain a better estimate of a. We denote the new value with $\tilde{a}$ and its covariance matrix with $\tilde{V}_{aa}$. We can 'propagate' the new information back to b, using

$$\begin{aligned} \tilde{b} &= b + V_{ab}V_{aa}^{-1}(\tilde{a} - a) \\ \tilde{V}_{bb} &= V_{bb} + V_{ba}V_{aa}^{-1}(\tilde{V}_{aa} - V_{aa})V_{aa}^{-1}V_{ab} \\ \tilde{V}_{ab} &= V_{ab} + (\tilde{V}_{aa} - V_{aa})V_{aa}^{-1}V_{ab} = \tilde{V}_{aa}V_{aa}^{-1}V_{ab} \end{aligned} \tag{82}$$

These expressions also hold when a and b are vectors.

We can use this result to propagate the information from the vertex fit into the full parameter set of the global track fit: The vector a represents the track parameters updated in the vertex fit and the vector b is the set of 'redundant' scattering angles. The covariance matrix of the full track parameter set is

$$\tilde{C} = \begin{pmatrix} \tilde{C_{qq}} & \tilde{C_{qq}}C_{qq}^{-1}C_{q\theta} \\ C_{q\theta}^T C_{qq}^{-1}\tilde{C_{qq}} & C_{\theta\theta} + C_{q\theta}^T C_{qq}^{-1}(\tilde{C_{qq}} - C_{qq}^{-1})C_{qq}^{-1}C_{q\theta} \end{pmatrix} \tag{83}$$

from which we can extract the covariance for the residuals. The change in the residuals follows from

$$\tilde{r} \;=\; r + H(\tilde{x} - x) \;=\; H \begin{pmatrix} 1 \\ C_{\theta q} C_{qq}^{-1} \end{pmatrix} (\tilde{q} - q) \tag{84}$$

The updated covariance matrix of the residuals of the single track is just given by

$$\tilde{R} \;=\; V - H\tilde{C}H^{T}. \tag{85}$$

These expressions are sufficient to calculate the first and second derivative of the total χ^2 to the alignment equations, *provided* that the correlation between residuals in different tracks is ignored in the second derivative. In practical applications it may actually be desired to ignore this contribution: the correlations will populate the entire second derivative matrix, such that the application of solution algorithms that rely on a sparse system are no longer applicable.

If one does not wish to ignore the correlations, they can still be calculated, provided that the vertex fit also gives access to the correlation between the vertexed tracks. To derive the expressions we first need to extend our simple (a, b) example. In addition to the pair (a, b) representing the parameters of track 1 we introduce another pair (c, d), representing track the parameters of track 2. At first these are uncorrelated, such that the full covariance matrix is given by

$$V(a, b, c, d) \;=\; \begin{pmatrix} V_{aa} & V_{ab} & 0 & 0 \\ V_{ba} & V_{bb} & 0 & 0 \\ 0 & 0 & V_{cc} & V_{cd} \\ 0 & 0 & V_{dc} & V_{dd} \end{pmatrix} \tag{86}$$

After the vertex fit we obtain a new pair $(\tilde{a}, \tilde{c})$ with covariance matrix

$$\tilde{V}(a, c) \;=\; \begin{pmatrix} \tilde{V}_{ac} & \tilde{V}_{ac} \\ \tilde{V}_{ca} & \tilde{V}_{cc} \end{pmatrix} \tag{87}$$

With the substitution $a \rightarrow (a, c)$ and $b \rightarrow (b, d)$ we can use Eq.82 to derive the full covariance matrix. The elements $\tilde{V}_{ab}$, $\tilde{V}_{bb}$, $\tilde{V}_{cd}$ and $\tilde{V}_{dd}$ follow trivially because they do not depend on $\tilde{V}_{ac}$. The new off-diagonal elements are

$$\begin{aligned} \tilde{V}_{ad} &= \tilde{V}_{ac} V_{cc}^{-1} V_{cd} \\ \tilde{V}_{cb} &= \tilde{V}_{ca} V_{aa}^{-1} V_{ab} \\ \tilde{V}_{bd} &= V_{ba} V_{aa}^{-1} \tilde{V}_{ac} V_{cc}^{-1} V_{cd} \end{aligned} \tag{88}$$

where it should again be understood that $V_{ij} = V_{ji}^{T}$ etc. Note that all these elements are proportional to $\tilde{V}_{ac}$: If the new information does not correlate a and c, the correlation between $(\tilde{a}, \tilde{b}$ and $(\tilde{c}, \tilde{d})$ should of course vanish.

B An alternative method for dealing with MCS in a track fit

It is sometimes argued that it is more convenient to 'eliminate' the scattering angles from the χ^2 minimization by including their contribution in the covariance matrix for the measurements

V. This elimination procedure is analogous to the elimination of the track parameters from the χ^2 for the alignment. In analogy to Eq.13 we derive for a total derivative to x,

$$\frac{\mathrm{d}}{\mathrm{d}x} = \frac{\partial}{\partial x} + \frac{\mathrm{d}\theta}{\mathrm{d}x}\frac{\partial}{\partial\theta} \quad \text{with} \quad \frac{\mathrm{d}\theta}{\mathrm{d}x} = -\frac{\partial^2\chi^2}{\partial x\partial\theta}\left(\frac{\partial^2\chi^2}{\partial\theta^2}\right)^{-1} \tag{89}$$

Starting from the χ^2 in Eq.25 the χ^2 derivatives in this expression become

$$\frac{\partial^2\chi^2}{\partial x\partial\theta} = H^T V^{-1} S \qquad \text{and} \qquad \frac{\partial^2\chi^2}{\partial\theta^2} = \Theta^{-1} + S^T V^{-1} S \tag{90}$$

where we defined $S = \partial r/\partial\theta$. Using the identity

$$V^{-1}S\left(\Theta^{-1} + S^T V^{-1} S\right)^{-1} \equiv \left(V + S\Theta S^T\right)^{-1} S\Theta \tag{91}$$

the derivative $\mathrm{d}\theta/\mathrm{d}x$ becomes

$$\frac{\partial\theta}{\partial x} = H^T\left(V + S\Theta S^T\right)^{-1} S\Theta \tag{92}$$

For the total first derivative of the χ^2 to x we then derive

$$\frac{1}{2}\frac{\mathrm{d}\chi^2}{\mathrm{d}x} = H^T V^{-1} r - H^T\left(V + S\Theta S^T\right)^{-1} S\Theta\left(S^T V^{-1} r + \Theta^{-1}(\theta - \hat{\theta})\right) \tag{93}$$

Using

$$V^{-1} - \left(V + S\Theta S^T\right)^{-1} S\Theta S^T V^{-1} = \left(V + S\Theta S^T\right)^{-1} \tag{94}$$

we obtain the simple expression

$$\frac{1}{2}\frac{\mathrm{d}\chi^2}{\mathrm{d}x} = H^T(V + S\Theta S^T)^{-1}\left(r + S(\theta - \hat{\theta})\right) \quad . \tag{95}$$

The expression for the second derivative follows in a similar way.

Note that Eq.95 still contains the residuals to the scattering angles explicitly. However, assuming the problem is sufficiently linear, the derivatives lead us to the correct solution independent of the initial values of x and θ. Therefore, we can start from an unscattered track with $\theta = 0$ and make all contributions from scattering angle residuals vanish. We then arrive at the result

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}x} = 2H^T W r \qquad \frac{\mathrm{d}^2\chi^2}{\mathrm{d}x^2} = 2H^T W H \tag{96}$$

with

$$W = \left(V + S\Theta S^T\right)^{-1} \quad . \tag{97}$$

In other words, with the substitution $V \rightarrow V + S\Theta S^T$ we have 'incorporated' multiple scattering contributions in the covariance matrix for the measurements. Two words of caution are required here:

- although we have eliminated the scattering angles, the derivatives to the scattering angles still need to be calculated. Therefore, from the algorithmic perspective the substitution method is not simpler than the simultaneous minimization to x and θ;

- the expressions follow from linearizing around an unscattered track. If the derivatives depend non-trivially on the actual values of x and θ, we do not arrive at the solution that minimizes Eq.25. Although we can remove non-linearities associated to x by iteration, we cannot remove those associated to θ because the requirement that $\theta = 0$ has been explicitly used;

C A formulation of the local method in terms of unbiased residuals

The local method can also be formulated in terms of so-called 'unbiased' residuals [2, 3]. Consider a subset 'A' of the measurements associated to the track that is small enough that the track is still overconstrained if the measurement coordinates are removed from the track fit. We call the track parameters that are the result of a χ^2 minimization without the measurements A, the *unbiased* track parameters and denote them by $\hat{x}$ and their covariance matrix by $\hat{C}$. We now define the χ^2 contribution of the removed measurements by

$$\chi^2_{\text{A, unbiased}} = \hat{r}^T \hat{R}^{-1} \hat{r} \tag{98}$$

where $\hat{r}$ is the unbiased residual vector and $\hat{R}$ is its covariance matrix

$$\hat{R} = V + H\hat{C}H^T. \tag{99}$$

Note that in this expression V and H only concern the subset of removed measurements.

Now consider the alignment of a *single* module, with alignment parameters α. If the subset A includes all measurements in this module, then the track parameter do not depend on α. Therefore, the dependence of the total χ^2 on α only comes through the unbiased χ^2 in Eq.98. Consequently, we have for the first and second derivative of the χ^2 to α

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}\alpha} = 2A^T \hat{R}^{-1} \hat{r} \qquad\qquad \frac{\mathrm{d}^2\chi^2}{\mathrm{d}\alpha^2} = 2A^T \hat{R}^{-1} A \tag{100}$$

One may now wonder how this related to the formulation in terms of biased residuals. In the least squares formalism the χ^2 contribution of the unbiased residuals is equal to the difference between the minimum χ^2 of the track that includes the hits A and the track that excludes the hits A,

$$\chi^2_{\text{track with hits A}}(x, C) = \chi^2_{\text{track without hits A}}(\hat{x}, \hat{C}) + \chi^2_{\text{A, unbiased}}. \tag{101}$$

Note that the track parameters used to evaluate the χ^2 contributions on the right and left side of this equation are not the same. The χ^2 contribution of the *biased* residuals of measurements A is given by

$$\chi^2_{\text{A, biased}} = r^T R^{-1} r \tag{102}$$

where r is the vector of biased residuals and $R = V - HCH^T$ its covariance. Note that this contribution is only defined if the matrix R is invertible, which is the case only if the subset of hits A is small enough as stated above. Within the least squares formalism one can derive that this contribution is equal to that for the unbiased residuals Eq.98, *e.g.*

$$\chi^2_{\text{A, biased}}(x, C) \;\equiv\; \chi^2_{\text{A, unbiased}}(\hat{x}, \hat{C}) \tag{103}$$

where we have made explicit that the left hand side and the right hand side involve different track parameters. Since the contributions in terms of biased and unbiased residuals are equal, the derivatives to α are equal. One can verify this explicitly by using the following relation between biased and unbiased residuals

$$r \;=\; V\hat{R}^{-1}\hat{r} \qquad\qquad R \;=\; V\hat{R}^{-1}V \tag{104}$$

which is a consequence of Eq.101 and has been derived in appendix A of [17] in the context of the Kalman filter. Substituting these expressions in Eq.100 leads to expressions Eq.19 and Eq.17.

The arguments above show that there is no need to 'unbias' residuals for the evaluation of the χ^2 derivatives for the local method. However, we should stress here that even if the derivatives are expressed in terms of unbiased residuals, the correlations between residuals from hits in the same module cannot be ignored. An exception can be made if a track contains at most two hits per module, as in the SCT. One can proof that a single unbiased residual is uncorrelated with all biased residuals on the track. (The proof follows from remark 7 in [6].) Therefore, one could formulate the chisquare contribution of the two hits in a module by unbiasing only one of the two hits. The expression for χ^2 derivatives for the two hits will be different, namely one follows the biased formalism and the other the unbiased. In the TRT the number of hits in one module is typically more than 2 such that this procdure cannot be applied.

In our formulation above we have assumed that all hits in a module are removed simultaneously to calculate unbiased residuals. One could also formulate the problem in terms of unbiased residuals that are calculated by removing only one hit at a time. In this case, the residuals are still correlated, but the calculation of the covariance matrix is not the $\hat{R}$ introduced above.

D Some details of the `TRTAlignAlg` implementation

The alignment algorithm described in this note has been implemented in the athena package `InnerDetector/InDetAlignAlgs/TRTAlignAlgs`. Being a first version, the implementation is not very sophisticated. In particular there is no clear separation of 'data' and 'tools'. The main C++ classes are:

`ModuleBase` Base class for 'modules', alignable objects in the inner-detector. The base class implements the calculation of residuals, derivatives of the residual to module translations and rotations and some basic monitoring. Specializations in the form of derived classes exist for TRT/SCT/Pixel and for barrel/endcap. These derived classes implement the interaction with the database and optionally provide specific implementations for derivatives to other alignment parameters (such as the module 'twist' in the TRT barrel) and additional histogramming. The `ModuleBase` class also keeps an (unowned) reference to the total χ^2 derivatives, implemented in the form of a `ModuleChisqData` object (see below). Modules are identified by an `Identifier` which in the simplest case corresponds to the identifier of a detector element (see below).

`ModuleChisqData` Stores the total first and second χ^2 derivatives of a the alignment parameters of a single module plus all non-zero module-module terms in the second derivative for modules with an identifier *smaller* than that of the referenced module. The size of the derivative vectors and matrices is determined by the corresponding `ModuleBase` object. Functionality exists to persist `ModuleChisqData` objects in binary files and also to add objects corresponding to the same module but different datasets;

`TRTAlignAlg` Top-level algorithm for module alignment. On each event is processes the tracks in a predefined track list, performing the following steps:

1. refit the track with `Trk::GlobalChi2Fitter` to extract the covariance matrix C and the derivative matrix H. Note that this includes the scattering angles;
2. loop over the list of associated hits and determine the module identifier (ID) for each hit. This identifier is calculated in the `TRTAlignAlg::moduleId` routine which takes as input the identifier associated to the `MeasurementBase`. If the hit is a module that is active in the alignment, the corresponding `ModuleBase` object is retrieved, or created if it does not yet exist. (`ModuleBase` objects are only created if there is at least one hit in the module.) A call to `ModuleBase::calculateResidual` is made to extract the residual and derivatives
3. calculate the full covariance matrix of the residuals by passing the list of selected hits to `TRTAlignAlg::calculateResidualCovMatrix`;
4. finally, add the collection of hits to the total first and second derivatives following the expressions in Eq.19 and Eq.17, respectively.

At the end of a job new alignment constants are calculated from the `TRTAlignAlg::finalize` method following either the local of global χ^2 method. Constants are written to text files using the tools `InDetAlignDbTool` and `TRTAlignDbTool`. The folders with alignment constants are also written to a POOL file.

A subset of particularly relevant methods in the classes above is:

`TRTAlignAlg::moduleId` calculates an identifier for a module associated to a particular hit. It takes the identifier of the detector element of the `MeasurementBase` as input and then find a corresponding module ID, taking into account the granularity parameters specified by `TRTAlignAlg` job-option parameters. The following granularity parameters exist:

`combineAllTRTBarrel` combines all modules in the TRT barrel

`combineAllTRTEndcap` combines all modules in each TRT endcap `siGranularityLevel` defines the level for granularity in SCT and Pixel: a value 1 (global) combines all modules in barrel and all in each endcap, 2 combines modules in a layer and 3 leads to separate alignment for each module.

Complete subsystems can be (de)activated using the parameters `alignTRT`, `alignSCT` and `alignPixel`. In the simplest scenario (SCT, Pixel at level 3) the identifier corresponds to the identifier of the detector-element associated to the hit. When several detector elements are combined in a module (as is always the case in the TRT), the identifier will in general correspond to that of the first detector element;

`TRTAlignAlg::findModule` retrieves the `ModuleBase` for a given module identifier using a simple `std::map` stored in `TRTAlignAlg`. If the module does not yet exist, it is created. The logic that decides if the module is TRT, SCT or Pixel and barrel or endcap is in this routine;

`ModuleBase::calculateTransformDerivative` calculates the derivative to the three delta translations and rotations. The parameters are defined in the so-called 'alignment-frame' (see below).

D.1 Reference frames in TRTAlignAlg

There are 4 important coordinate frames in TRTAlignAlg. The first two are the global and the measurement frames discussed in section 3.1. The remaining two are

database frame This is the frame in which the delta-transform stored in the database is applied. This frame follows from Eq.58 for the TRT and from Eq.60 for the SCT and Pixel;

alignment frame This is the frame in which the *delta* alignment constants (three translations and three rotations) are expressed. This frame can be arbitrarily chose, but there is a good reason to reason to make a specific choice. First, we can choose this frame such that the parameters are as independent as possible which simplified the interpretation of the observed shifts. For example, it makes sense to choose a frame in which the center-of-gravity of the module s th origin such that the rotation is more or less decoupled from the translation. Second, the interface to TRTAlignAlg allows us to deactivate parameters, for example because they correspond to weakly determined degrees-of-freedom: The meaning of these parameters is specific the the alignment frame.

In its current implementation the alignment frame for SCT and Pixel modules is the module frame obtained via `detelement->moduleTransform()`. For TRT barrel modules it is the frame of the most central detector element. ('central' is not entirely unambiguous. For details, see the code.) In the TRT endcap, where the alignable unit is currently a complete wheel, the alignment frame is just the translation to the center-of-gravity.

References

[1] P. Brückman, A. Hicheur and S. J. Haywood, *Global χ^2 approach to the Alignment of the ATLAS Silicon Tracking Detectors*, ATL-INDET-PUB-2005-002.

[2] R. Härtel, "Iterative local Chi2 alignment approach for the ATLAS SCT detector", diploma thesis TU Mǹchen, MPP-2005-174 (2005).

[3] T. Göttfert, "Iterative local Chi2 alignment algorithm for the ATLAS Pixel detector", diploma thesis Universität Würzburg,MPP-2006-118 (2006).

[4] V. Blobel, *Millepede: Linear Least Squares Fits with a Large Number of Parameters*, `http://www.desy.de/ blobel/wwwmille.html`. V. Blobel, "Software alignment for tracking detectors," Nucl. Instrum. Meth. A **566** (2006) 5.

[5] R. Kalman, Journal of Basic Engineering , 35 (1960).

[6] R. Frühwirth, Nucl. Instrum. Meth. A **A262**, 444 (1987).

[7] T. G. Cornelissen, *CTBTracking: track reconstrucion for the testbeam and cosmics*, ATL-INDET-INT-2006-001; T. G. Cornelissen, *Track fitting in the ATLAS experiment*, (PhD Thesis, 2006).

[8] M. Karagöz Ünel, P. Brückman de Renstrom, A. Hicheur and K .Bernadet, *Parallel computing studies for the alignment of the ATLAS Silicon tracker*, contribution to CHEP 2006.

[9] C. C. Paige and M. A. Saunders (1975), *Solution of sparse indefinite systems of linear equations*, SIAM J. Numerical Analysis 12, 617–629 (1975).

[10] P. Schleper, G. Steinbrc̈k, M. Stoye, *Software Alignment of the CMS Tracker using MILLEPEDE II*, CMS-NOTE-2006-011.

[11] A. Morley, *Solutions to Large Linear Systems for the Global X2 Alignment*, talk in the ATLAS ID week, September 2006, `http://indico.cern.ch/conferenceDisplay.py?confId=a0626#6`.

[12] I. S. Duff and J. K. Reid, Tech. Rep. AERE R10533, Her Majesty's Stationery Office, London (1982); S. Duff and J. K. Reid, ACM Transactions on Mathematical Software (TOMS), v.9 n.3, p.302-325, Sept. 1983; `http://hsl.rl.ac.uk/archive/hslarchive.html`

[13] I. S. Duff, ACM Transactions on Mathematical Software (TOMS), Volume 30 , Issue 2 (June 2004), `http://portal.acm.org/citation.cfm?id=992202`; `http://www.cse.clrc.ac.uk/Activity/HSL`

[14] O. Schenk and K. Gärtner, Journal of Future Generation Computer Systems, 20(3):475–487, 2004; O. Schenk and K. Gärtner, Elec. Trans. Numer. Anal., 23:158–179, 2006; `http://www.computational.unibas.ch/cs/scicomp/software/pardiso/`.

[15] R. Lehoucq, K. Maschhoff, D. Sorensen and C. Yang, http://www.caam.rice.edu/software/ARPACK/.

[16] P. Brückman de Renstrom and S. J. Haywood, *Least Squares Approach to the Alignment of the Generic High Precision Tracking System*, contribution to PHYSTAT 2005.

[17] W. D. Hulsbergen, *A study of track reconstruction and massive dielectron production in HERA-B* (PhD thesis, 2002), http://www-hera-b.desy.de/general/thesis/diss/diss_wouter_hulsbergen.pdf

[18] B. Demirköz *et al.*, *Atmospheric Muon based Trigger Setup for the Integration of the ATLAS Inner Detector*, Atlas internal note (in preparation).