IMPLEMENTATION OF A GLOBAL FIT METHOD FOR THE ALIGNMENT OF THE SILICON TRACKER IN ATLAS ATHENA FRAMEWORK

A.Hicheur*, RAL, Didcot, United Kingdom
S.Gonzalez Sevilla, IFIC, Valencia, Spain
The ATLAS Collaboration

Abstract

The ATLAS Inner Detector is composed of a pixel detector (PIX), a silicon strip detector (SCT) and a Transition radiation tracker (TRT). The goal of the algorithm is to align the silicon based detectors (PIX and SCT) using a global fit of the alignment constants. The total number of PIX and SCT silicon modules is about 5800[1], leading to many challenges. The current presentation will focus on the infrastructure of the processing part of the algorithm, leaving the technical issues related to the solving of a system with a large number of degrees of freedom to a separate presentation[1]. In this document, the main functionalities of the code will be presented and basic analysis of simulation will be shown.

INTRODUCTION

Aligning a high precision tracker using track-based information has become an important concern for any particle physics experiment over the last couple of decades. The different approaches that are envisaged can basically be divided into local and global fits. In local methods, the motion of a single module is determined iteratively by repeating a local χ^2 minimization and refitting tracks at each step. In global methods, a global minimization is performed: changes in alignment constants and track parameters are handled altogether, along with additional constraints that could be inserted into the minimization procedure. Both methods have been applied successfully in running experiments (e.g.: Babar for the local iterative method [2] and ALEPH for the global fit [3]). After reviewing briefly the main requirements for ATLAS physics and the principle of the global fit (explained in details elsewhere [4]), we will present the code implementation and some results of tests in simulation.

PHYSICS REQUIREMENTS

To pick up a significative example of the impact of alignment on physics performance [5], we just mention the case of the W boson mass:

$$m_W = (\frac{\pi.\alpha}{\sqrt{2}.G_F})^{\frac{1}{2}}.\frac{1}{sin\theta_W.\sqrt{1-\Delta R}} \qquad (1)$$

where α is the QED fine structure constant, G_F is the Fermi constant and θ_W the Weinberg angle. The term ΔR represents radiative corrections involving the top mass squared and the logarithm of the Higgs mass. A 20 MeV precision measurement of the W mass will allow to have a constraint on the Higgs mass to a precision of 30%. For this to be achieved, alignment would have to be understood to the 0($1\mu m$) level in the transverse plane.

PRINCIPLE OF THE GLOBAL FIT

Starting from a χ^2 built with residuals r and their covariance matrix V, $\chi^2 = \sum_{tracks} r^T V^{-1} r$, we perform a multiparameter minimization to find the minimum. The relevant parameters are the alignment parameters a, the track parameters π and the primary vertex v for each event.
To illustrate the procedure, we take the case where only consider a and π are considered. The minimization equations are

$$\frac{\partial\chi^2}{\partial a}|_m = 0 \quad \frac{\partial\chi^2}{\partial \pi}|_m = 0 \qquad (2)$$

To find an analytical solution for the minimum (a_m, π_m), one must perform an linear expansion of the residuals near the current parameters (a_0, π_0), leading to a parabolic approximation for the χ^2, in terms of $\delta\pi = \pi - \pi_0$ and $\delta a = a - a_0$. Using this approximation, equation 2 leads us to a linear system that can be solved to get the alignment parameters:

$$\delta a|_m = a_m - a_0 = -M^{-1}.V \qquad (3)$$

Where $M = \sum_{tracks} \frac{\partial r}{\partial a}^T W \frac{\partial r}{\partial a}$ is the alignment weight matrix, $V = \sum_{tracks} \frac{\partial r}{\partial a}^T W r_0$ is the alignment vector. $W = V^{-1} - V^{-1}\frac{\partial r}{\partial \pi}(\frac{\partial r}{\partial \pi}^T V^{-1}\frac{\partial r}{\partial \pi})^{-1}V^{-1}$ is simply the "gain" matrix for each track and contains both the measurement inverse covariance and tracking effects in the second term.
Adding the third variable (i.e, the vertex v) gives a more complicated formula for M and V but the principle remains the same.
It is important to mention here that by using a parabolic approximation for the χ^2, one assumes that we are not too far from the solution: large misalignments may require iterations.
Any constraint to the fit can be added as an additive term to the χ^2. As an example, we can mention measurements

* Adlene.Hicheur@cern.ch

[1] Distributed in 3+4 cylindrical PIX+SCT barrel layers and $2 \times (3+9)$ disc PIX+SCT endcaps

of module positions from hardware surveys. These measurements can be translated into constraints on alignment parameters, a_c with corresponding inverse covariance C. The term to be added to the χ^2 is $(a-a_c)^T C(a-a_c)$. Performing the minimization, it is easy to show that M and V are tranformed as $M \to M+C$ and $V \to V+C(a_0-a_c)$.

CODE IMPLEMENTATION

The alignment algorithm receives two types of input coming from upstream processing: geometry and tracking.

Geometry and module frame

The definition of the alignment frame is as follows: the local $\hat{x}$ and $\hat{y}$ axes are within the module plane, $\hat{x}$ being across the strips (short pixel side) for SCT (PIX) modules and the $\hat{z}$ axis is normal to the module plane. Since SCT barrel (resp. endcap) modules are double sided with one side having a stereo angle with respect to the axial (resp. radial) cylinder (resp. disc) directions, the module alignment frame is assigned to the non-stereo side, for convenience.

Track selection

The tracks used for the alignment are retrieved from a common interface providing the same information, whatever tracking technique is used for the reconstruction. We require that the number of detection layers missed by the tracks ("holes") does not exceed one. Furthermore a given track should not share hits with another one and we require that the track transverse momentum to be at least 2 GeV/c to avoid the huge multiple scattering effects for low momentum tracks. The momentum threshold cannot be raised too much to avoid the loss sensitivity to radial displacements of barrel modules.

Processing

The `GlobAlign` algorithm is developed in the ATLAS Athena framework and is therefore organized according to the common nomenclature. It inherits the initialize(), execute() and finalize() methods from the base class `Algorithm`.
In the *initialize* stage, the initialization of access to transient stores is done on top of the initialization of specific objects used during the processing: StoreGate toolkit [6] is used to retrieve transient data objects such as track lists (accessed on an event basis) and the DetectorStore client service provides access to the transient detector store from which we retrieve the Silicon modules objects.
In the *execute* stage, a list of modules is retrieved from the transient detector store in the first event only. The alignable geometry is selected at this stage. At each event, a track list is retrieved from the data objects transient store, after its creation by algorithms ran upstream. A loop over tracks is then performed to analyze the information from the hits, compute the residuals derivatives and fill the alignment objects. The alignment matrix and vector are filled at the end of the track processing if no primary vertex fit is performed or at the end of the event processing if the vertexing option is chosen. During the whole processing, hit and alignment index `std` maps are built to store the hit occupancy and the alignment matrix indices for all the selected modules.
In the *finalize* stage, the alignment system (matrix,vector) can be either solved directly or stored in binary format for further solving. In case of solving, the output alignment constants can be stored either in simple ASCII format or in the Conditions Database. For the storage in the Database, the constants are packed in the format of a single object per module, an `AlignableTransform`, comprising a translation and a rotation. When requested, the combination of the information from survey can be done prior to the solving/storing.
Figure 1 summarizes the alignment processing flow.

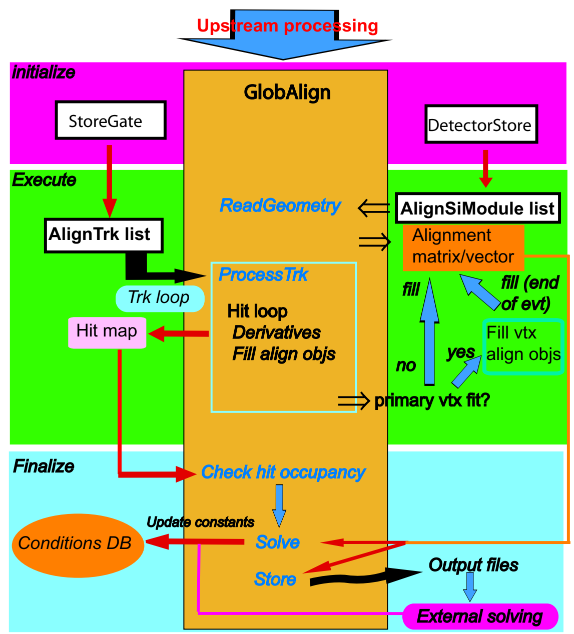

Figure 1: Alignment processing flow. For the sake of clarity, several helper classes and methods involved in the hard processing are not represented here.

Solving

Performing an internal alignment implies that the system has no absolute (spatial) reference which therefore leads to inherent degeneracies. As an example, the alignment procedure is not sensitive to a global translation of the whole tracker. Degeneracies can be removed totally or partially by some special features of the system (e.g: a rotation of the tracker about any transverse axis is not a degeneracy because the magnetic field imposes a direction) or by applying constraints (e.g: primary vertex fit constraint). The existence of degeneracies results in the alignment matrix

being singular. Instead of a direct inversion, it is therefore more appropriate to proceed to a diagonalization and an eigenmode analysis to remove the singularities (eigen vectors with zero or tiny eigen values) at this stage before completing the solving.
The alignment matrix is diagonalized using the `LAPACK` routine `dspev` which returns a set of eigen values and vectors. Up to a certain number of degrees of freedom (lower than 10000 typically), the system is still solvable with a single CPU. Above this threshold, the solving requires parallel computing to allocate the huge alignment matrix and diagonalize it[2].

TESTS ON SIMULATION

Test sample

To perform first validations of the alignment code, a 10000 multimuon events sample was used with the following requirements: an average multiplicity of 10 tracks per event, a flat distribution for the transverse momentum in the range [2,20] GeV/c, a flat pseudo-rapidity (η) distribution in the range [0,1] and primary vertex parameters such as ($\langle XY \rangle$ = 0, $\sigma_{XY} = 1$) mm, ($\langle Z \rangle$ = 6, $\sigma_Z = 1$) cm. The tracks produced by this sample cover most of the positive z half of the barrel region, firing 1030 PIX+SCT modules, which corresponds to 6180 degrees of freedom. The primary vertex was displaced in Z to allow a full coverage of the set of modules selected in the active η region.

Results with nominal geometry

Figure 2 shows the results for the distribution of the six alignment parameters. The distributions are all centered at zero and the widths of the PIX/SCT modules distributions are 22/23, 97/111, 128/144 μm for the $\hat{x}$, $\hat{y}$, $\hat{z}$ translations and 1.3/1.4, 1.7/1.7, 0.3/0.3 mr, for the corresponding rotations, respectively.

From the precision achieved along the most sensitive direction ($\hat{x}$), we infer that with a sample of O(10^7) tracks, we can meet the target requirements for physics. Another important feature is that the rotations stay under control (these degrees of freedom are usually hard to handle). We also observe that the coupling between modules does not introduce any serious bias in the distributions. This is due in big part to the primary vertex fit which corrects for the biases as shown in figure 3.
Figure 4 shows the pull distribution of the parameters from which we check that the error calculation is consistent.

First tests with misalignments

To simulate misalignments, the hits are moved around before the track pattern recognition is applied.
Two types of misalignments are simulated and combined:

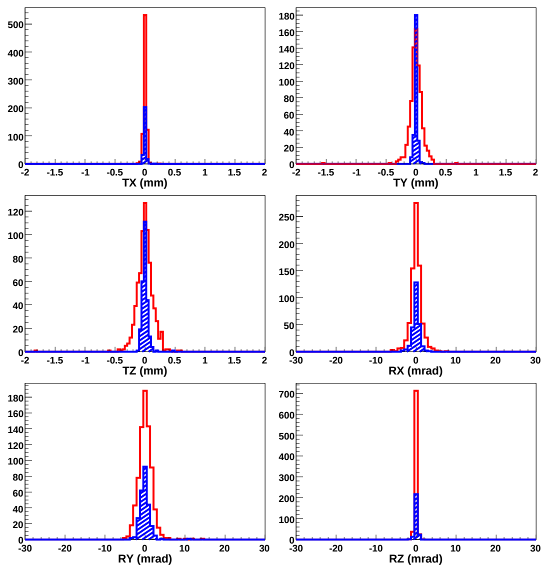

Figure 2: Distribution of the six alignment parameters obtained for the nominal geometry. The solid and hatched histograms represents the distributions for the SCT and Pixel modules, respectively.

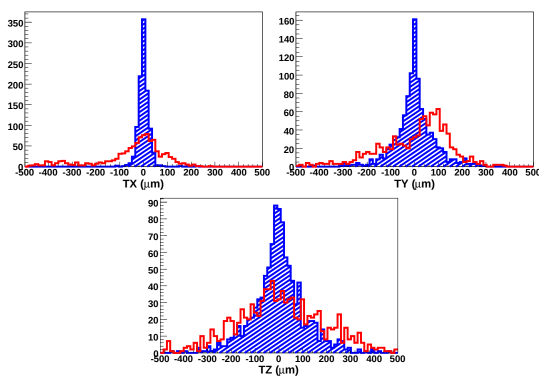

Figure 3: Distributions of the three translations. The contributions from SCT and Pixel modules are merged. The solid and hatched histograms represents the distributions for the with and without primary vertex fit cases, respectively.

random local misalignments and global layer to layer misalignments. For this study, we combine two sets:

- Set P1: Only Pixel modules are misaligned. We apply random misalignments of RMS equal to 30 μm in the local $\hat{x}$ and $\hat{y}$ directions and 20 μm in the $\hat{z}$ direction. On top of this we apply a collective misalignments for each layer, with an RMS of 20 μm in the three directions.

- Set S1: Only SCT modules are misaligned. we apply random misalignments of RMS equal to 100 μm in the local $\hat{x}$ and $\hat{y}$ directions and 500 μm in the $\hat{z}$ direction

The combination of both sets is called S1P1.

The distributions of the difference (applied - obtained) translations for the sets P1 and S1P1, after a first iteration,

[2] Allocating a big memory space is not the only problem. Above a critical size (15000), the usage of quadruple precision is required to avoid a complete loss of precision due to rounding errors.

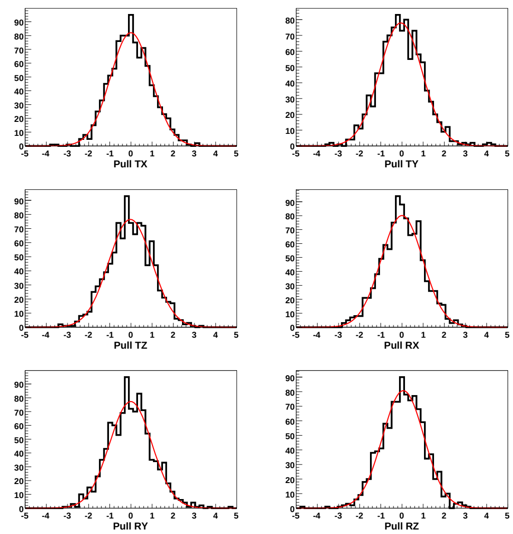

Figure 4: Distribution of the pulls of the alignment parameters obtained for the nominal geometry. The contributions from SCT and Pixel modules are merged.

are shown in figures 5 and 6, respectively.

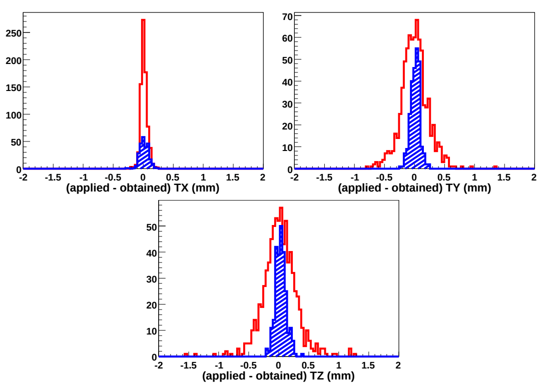

Figure 5: Distribution of the (applied - obtained) translation parameters for misalignment set P1. The solid and hatched histograms represents the distributions for the SCT and Pixel modules, respectively.

For P1, the distributions are broadened in comparison with the nominal case but are still centered and reasonably constrained: we expect that more statistics and a couple of iterations will improve the resolution. For S1P1, the distributions for the Pixel modules follow the same pattern while the spread for SCT modules is bigger as one would expect since the applied misalignments are much bigger for the SCT modules[3]. Here again, more statistics and further iterations are foreseen to provide more precision.

[3]The asymmetric tails that we observe are not only explained from the non-linear regime in which we are but it happens that the entries in the tails are also the ones with the biggest errors so that the pull distribution remains symetric.

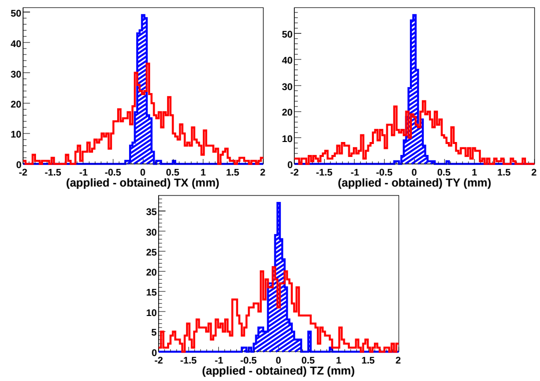

Figure 6: Distribution of the (applied - obtained) translation parameters for misalignment set S1P1. The solid and hatched histograms represents the distributions for the SCT and Pixel modules, respectively.

SUMMARY AND OUTLOOK

A software for the alignment of the ATLAS Silicon Tracker, based on a global χ^2 fit, has been implemented. The first tests have shown that this method can cope with a large number of degrees of freedom in principle, providing a sensible output. For the control of the singularities, the importance of the primary vertex fit has been demonstrated. However, more development and tests are foreseen to include additional constraints such as the survey measurements or the use of dimuon resonances. Extensive validation tests and development are planned with high statistics, recently simulated samples, and misaligned layouts.

ACKNOWLEDGEMENTS

We would like to thank our ATLAS collaborators in general and in particular S.Haywood, P.Bruckman, A.Bonissent, K.Bernardet, D.Fouchez and A.Tilquin for their first thoughts about applying the global fit to ATLAS Silicon Tracker and for doing feasibility studies with standalone prototype codes. We also thank Richard Hawkings and Grant Gorfine for providing all the necessary tools at the pre- and post-processing levels.

REFERENCES

[1] M.Karagoz Unel, Parallel computing studies for the alignment of the ATLAS Silicon tracker, these proceedings.

[2] D.Brown, A.Gritsan, G.Lynch, BaBar Analysis Document #486 (2003).

[3] A.Bonissent et al., *Alignment of the upgraded VDET at LEP2*, ALEPH 97-116

[4] ATLAS note, ATL-INDET-2005-004, P.Bruckman, A.Hicheur, S.Haywood.

[5] ATLAS note, ATL-INDET-2000-005 (and references therein), S.Haywood, S.Peeters, S.Snow, M.Wielers.

[6] P.Calafi ura et al., CHEP03, La Jolla, California, March 24-28 2003.