

Manual: Internal tracker alignment for the g-2 experiment

Gleb Lukicov ^a

^a*Department of Physics and Astronomy, University College London, London, WC1E 6BT, United Kingdom*

August 18, 2019

Abstract

This document outlines the need for the internal alignment of the tracking detectors, and the methodology of the alignment. The software methods used in the alignment are summarised to allow for future alignment constants to be established for Run-3 and beyond. Internal alignment of the trackers in Run-1 and Run-2 has been established, with the constants defined as the interval of validity (IoV) in the reconstruction database.

Contents

1	Introduction	2
2	Methodology of the alignment of the trackers	3
3	Alignment results	4
4	Software Methods	4
4.1	Run iterative internal alignment on a single run with residual verification	5
4.2	Run internal alignment monitoring	6
4.3	Run internal alignment to establish constants for a run interval . . .	6
5	Alignment constants and the reconstruction database	6

1 Introduction

In order for the tracking detector to reduce the systematic uncertainty on the a_μ measurement and improve the sensitivity to a muon EDM, a precise calibration of the tracking detector is required. One of such calibrations is the internal alignment, which considers the positions of the tracking modules within a station.

Track-based internal alignment of the two stations of the tracking detector was implemented using data from Run-1 and Run-2. A Monte Carlo simulation was also developed to understand the detector geometry, and how the detector geometry affects how well the alignment can be determined, as well as to test the alignment procedure itself. A single figure-of-merit can really emphasise the need for a well aligned detector, as shown in Fig. 1, which highlights that even a relatively small scale of misalignment can have a large impact on the detector performance w.r.t to the beam extrapolation. Additionally, alignment is essential for setting a new limit on the muon EDM. Broadly speaking, an internal misalignment of an element of a tracking detector manifests as a presence of a large residual between a measurement (e.g. DCA of a hit to a wire) and a DCA of a fitted track to that wire. This large residual arises from the fact that the assumed detector position, which was used in the fitting of the track, is not the actual position of that detector. The aim of the alignment procedure is to establish the corrections to the assumed detector position, and hence, minimise the residuals.

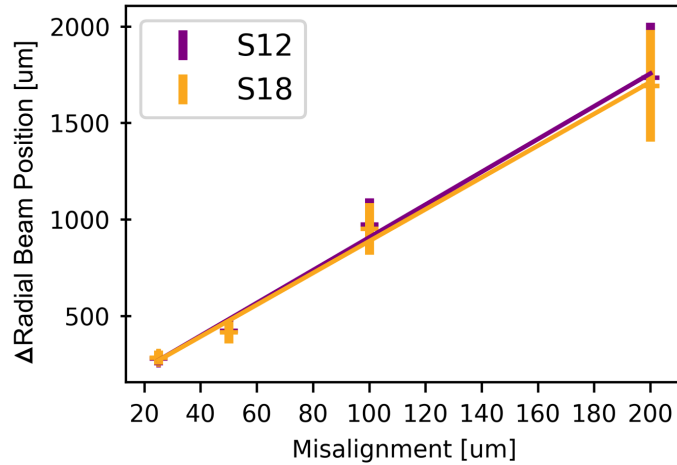


Fig. 1. Detector response to a uniform radial misalignment at different scales w.r.t to the difference in the extrapolated radial position.

2 Methodology of the alignment of the trackers

The chosen alignment framework was **Millepede-II** [1], which utilises a simultaneous fit of many parameters describing the detector geometry and the input data. Under this method correlations between different alignment elements are taken into account, to produce unbiased corrections to the position of the detector. The definition of the coordinate system used is shown in Fig 2.

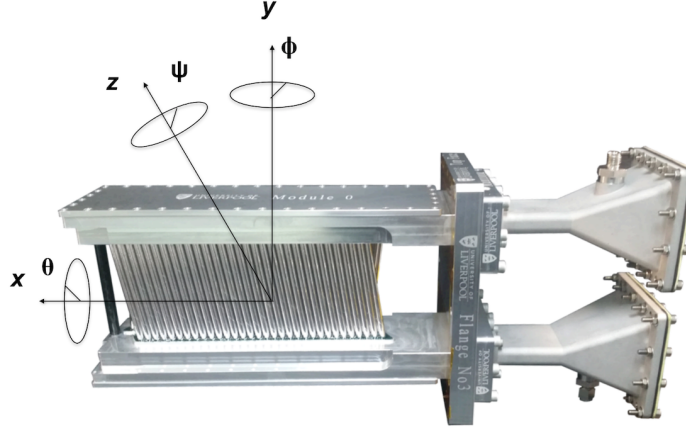


Fig. 2. Convention for misalignments (translations and rotation) in a module coordinate system. The radial and vertical shifts are along x and y , respectively.

The alignment task can be summarised as a Least Squares Fit (LSF) problem with a very large number of parameters. These parameters can be divided into two classes: global and local parameters. Global parameters (i.e. the alignment parameters) affect all tracks (e.g. radial position of a detector). Local parameters are associated with individual fitted tracks. For example, a straight line fit in 2D will have 2 local parameters: a slope and an intercept. **Millepede-II** solves the linear LSF problem with a simultaneous fit of all global and local parameters minimising a linearised function of a sum of residuals. An objective function, F , is then minimised with respect to a variation of global and local parameters

$$F(\mathbf{a}, \mathbf{b}) = \frac{\partial \chi^2(\mathbf{a}, \mathbf{b})}{\partial(\mathbf{a}, \mathbf{b})} = \sum_j^{tracks} \sum_i^{hits} \frac{1}{(\sigma_{i,j}^{det})^2} \frac{(r_{i,j}(\mathbf{a}_0, \mathbf{b}_{0,j}) + \frac{\partial r_{i,j}}{\partial \mathbf{a}} \delta \mathbf{a} + \frac{\partial r_{i,j}}{\partial \mathbf{b}_j} \delta \mathbf{b}_j)^2}{\partial(\mathbf{a}, \mathbf{b})} = 0, \quad (1)$$

where \mathbf{a}_0 and \mathbf{b}_0 are the initially assumed geometry and track parameters, respectively, r is the track residual (as shown in Fig. 3), and σ^{det} is the detector resolution. The corrections to the global parameters, $\delta \mathbf{a}$, allow for the minimisation of F , and are results of the alignment procedure. These corrections are then added to the assumed geometry of the detector to improve its performance.

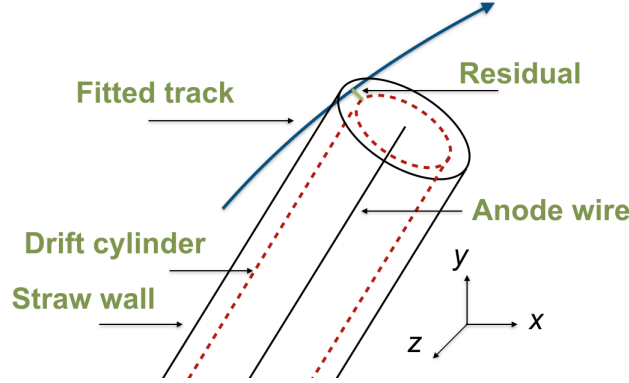


Fig. 3. Visualisation of the drift cylinder, which is defined by the measured drift time in the straw.

3 Alignment results

In simulation (MDC1), alignment stability was reached, and alignment convergence within $2\mu\text{m}$ and $10\mu\text{m}$ radially and vertically, respectively, was established as shown in **DocDB:17551**. Stable alignment results and improved residuals in data, using run 15922, were reached as shown in **DocDB:17722**.

Stability of the alignment constants across 3 IoV of Run-1 and Run-2 are shown in shown in **DocDB:AAAA**.

More detailed description of the alignment methods is contained in **DocDB:15106**. While the uncertainty on the beam extrapolation due to the misalignment was estimated in **DocDB:16685**.

4 Software Methods

The alignment software is part of the `gm2tracker` project: `gm2tracker/align/`. An alignment software module has been developed as part of the *gm2 art* software framework. This module comes last in the tracking chain, taking the final track-data product (`TrackArtRecords_tracks`) from either data or simulation. The geometry is fully described at the tracking stage and is passed to the alignment module, which also calculates the residuals of the selected tracks, as well as computes the local and global derivatives. A C++ infrastructure (`Mille`) is provided [1] to write this information into a binary file which is passed to `PEDE` (a Fortran executable), which performs a simultaneous fit via a matrix inversion described above. The alignment module is also responsible for writing a constraints file (specifying the redundant degrees of freedom), and a steering file (specifying the mathematical methods used by `PEDE`). The final outputs of the `PEDE` algorithm are labels and the corresponding

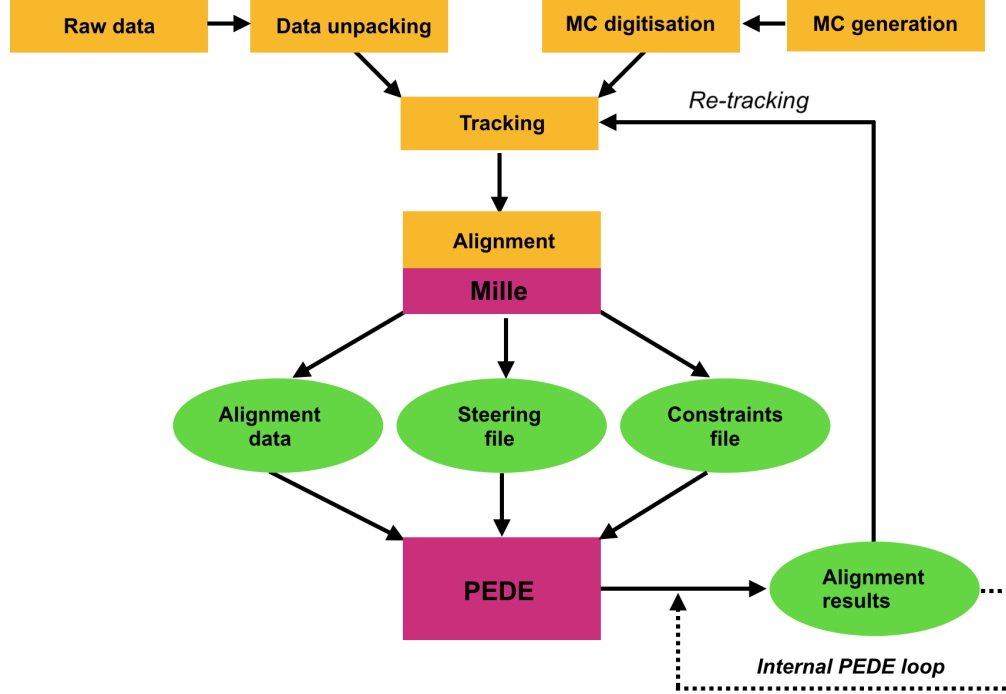


Fig. 4. The schematic of the Mille routine (shown in purple) with the *gm2 art* software framework (shown in orange). The final result of alignment is then used in tracking to improve the residuals.

fitted values of the global parameters and their errors, if applicable. This algorithm is shown in Fig. 4.

The steering file is

```

1 * g-2 Tracker Alignment: PEDE Steering File
2
3 ConstraintFile.txt ! Constraints text file
4 Cfiles ! Following bin files are Cfiles
5 Data.bin ! Binary data file
6 method inversion 10 0.01

```

4.1 Run iterative internal alignment on a single run with residual verification

Commands and plots

4.2 Run internal alignment monitoring

Commands and plots

4.3 Run internal alignment to establish constants for a run interval

Commands and plots

5 Alignment constants and the reconstruction database

The alignment constants will now be used in the *gm2geom* from the next release: *v9_2800* or *v9_2701*. Once DB methods are established, the alignment constants will migrate to the IoV table.

References

- [1] V. Blobel *Software Alignment for Tracking Detectors*, Nucl. Instrum. Methods A, **556**, 5 (2006).

Acknowledgements

I would like to thank the entire tracker group for their immense support in this project! In particular, I would like to express gratitude to Mark, Becky, James, Joe, Brendan, Saskia, Tammy and Nick for their advice.