Alignment in FairRoot

M. Al-Turany[1], D. Bertini[1], R. Karabowicz[1], D. Kresan[1], and F. Uhlig[1]
[1]GSI, Darmstadt, Germany

Alignment means to create the geometry used in simulation, reconstruction, and data analysis as close as possible to the real one in the experiment. The agreement between these two geometries is needed to achieve for example the required experimental resolution.

The problem is to know the real position of the detectors and detector subcomponents as accurate as possible which cannot be taken from a technical drawing. During their assembly the detectors and detector subcomponents cannot be placed at the expected position, since the positions of subcomponents may change for example due to mechanical deformations.

The alignment procedure needs as a starting point a geometry which is as close as possible to the real geometry in the experiment. This geometry can be taken from the technical drawings, but in most cases it comes from an optical survey. The alignment procedure then improves the accuracy of the positions of the different volumes by using signals generated in the active parts of the detector. Depending on the experiment and the detector system these signals can be produced by a laser calibration system, cosmic particles or particles from collision events.

Alignment Procedures

One approximate alignment procedure is to fit data from single tracks assuming fixed alignment parameters. The deviations between the measured hit position and the fitted one (residuals) are then used afterwards to extract the alignment parameters.

The problem with this method is that the result and hence the extracted alignment parameters are biased, since one uses wrong hit positions for the track fit. In repeated fits using the alignment parameters extracted in the previous iteration one can reduce the residuals, but it is not clear if the procedure converges [1].

A more efficient and faster method is an overall least squares fit, with all the global parameters (alignment parameters) and local parameters (track parameters for one track), perhaps from thousands or millions of events, determined simultaneously. For this task the Millepede program [2] was developed. The global parameters are the different degrees of freedom for each volume to be aligned. Assuming a 3 dimensional cartesian coordinate system this can be up to 6 different degrees of freedom. 3 for shifts in the x-, y-, and z-direction and another 3 for the rotations around the 3 axis of the coordinate system. Not all of these parameters have to be free in the minimization process.

Using Millepede in FairRoot

The program Millepede actually consist of two program parts. One is the writer component called mille, which writes the required input data for the minimizer (pede) in the correct binary format. This split, and the fact that there is also a c++ implementation of the writer component make the usage of Millepede in the FairRoot framework rather easy.

A reconstruction or analysis run within the FairRoot framework [3] is always build out of one or more tasks which are executed in a given order. To use Millepede within this scheme one has only to create a task which collects for each event all the information needed by the minimizer component and uses the writer component to create the input data in the correct binary format.

The data to be written are the derivatives of the local and global parameters, the residual and the error for each measured hit together with labels to define which global parameters are correlated with this hit. This data is very much dependent on the track model used by the experiment and the global parameters and hence the implementation of the writer task is very much experiment specific.

The FairRoot framework contains a toy model experiment (Tutorial4) which is meant to explain how to write an experiment specific class to create the binary data file for the minimization part of Millepede. The ”experiment” has 40 silicon detector planes, which can measure the x- and y-coordinates of particles traversing the detector volume. In the simulation run for each event one particle from different vertices and with different incident angles is tracked through the experimental setup. In the reconstruction run as the first step the hit positions are calculated taking into account parametrized shifts of the detector planes (misalignment) and an uncertainty due to detector response. For each event the hits are now fitted to extract the track parameters. The used track model for this fit is a straight line. In the last step the derivatives of the local and global parameters and the residuals for each hit are written to the binary data file for the minimizer component of Millepede.

The calculated alignment parameters (shifts of the detector planes in x-, and y-direction) using the minimizer component of Millepede are in very good agreement with the simulated misalignment.

References

[1] http://www.desy.de/~blobel/

[2] https://www.wiki.terascale.de/index.php/Millepede_II

[3] http://fairroot.gsi.de/