Calibration and Alignment Framework for the Belle II detector

T Bilka and J Kandra et al. on behalf of the Belle II Collaboration
Charles University, Ovocny trh 3, Prague 1, 116 36, Czech Republic
E-mail: `jakub.kandra@karlov.mff.cuni.cz, bilka@ipnp.troja.mff.cuni.cz`

Abstract. The Belle II experiment [1] is approaching its first physics run in 2018. Its full capability to operate at the precision frontier will need not only excellent performance of the SuperKEKB accelerator and the detector, but also advanced calibration methods combined with data quality monitoring. To deliver data in a form suitable for analysis as soon as possible, an automated Calibration Framework (CAF) has been developed. The CAF integrates various calibration algorithms and their input collection methods for event-level data. It allows execution of the calibration workflow using different backends from local machines to a computing cluster, resolution of dependencies among algorithms, management of the produced calibration constants, and database access across possible iterations. One of the main algorithms fully integrated in the framework uses Millepede II [2] to solve a large minimization problem emerging in the track-based alignment and calibration of the pixel and strip detector, the central drift chamber, and the muon system. Advanced fitting tools are used to properly describe the detector material and magnetic field and include measurements of different sub-detectors into a single global fit performed by Millepede. This talk will present the design of the calibration framework, the integration of the Millepede calibration, and its current performance.

1. Belle II detector

Belle II is an experiment under construction at the High Energy Accelerator Research Organization in Tsukuba, Ibaraki prefecture, Japan. As an upgrade of the Belle experiment, it is expected to accumulate a 50 times larger data sample ($\sim$ 50 ab^{-1}). Belle II will continue and extend the physics program of Belle, marking the beginning of the precision era in CP violation measurements in B-meson decays, and search for new physics. The SuperKEKB accelerator is a next generation high luminosity B-Factory designed to reach an instantaneous luminosity of $8 \times 10^{35} cm^{-2} s^{-1}$. The collider can provide a clean environment for production of B$\bar{\text{B}}$ pairs, $\tau^+\tau^-$ pairs and $c\bar{c}$ states. The Belle II detector utilizes a tracking system, particle identification detectors (PID), an electromagnetic calorimeter (ECL) and a K_L-muon (KLM) system (Fig. 1).

2. Calibration and alignment framework

The calibration and alignment framework (CAF) [3] provides automated calibration procedures of subdetectors of Belle II and is part of the Belle II analysis and simulation framework (basf2)[4]. The basf2 framework uses a modular architecture, C++ libraries and modules for event processing, together with extensive python bindings which allow for easy access to all framework features from python, the steering language of basf2. The basf2 framework controls all aspects of the detector simulation, reconstrution, analysis, and also data aquistion and data

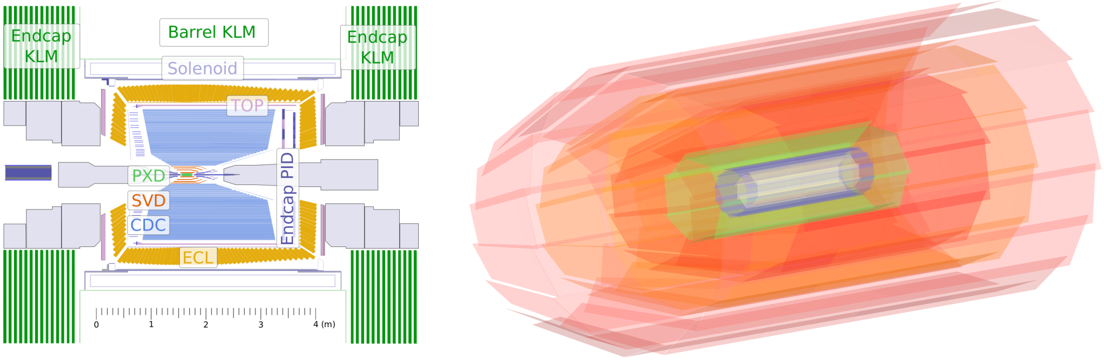

Figure 1. *Schematic cross-section of the Belle II detector (left):* The PXD (light green), the SVD (orange) and the CDC (blue) compose the tracking system. The TOP (pink) and the ARICH (violet) detector are particle identification detectors. The crystals of the ECL (yellow) surround the PID detectors. The outermost detectors are the barrel and endcap KLM (dark green).
The PXD and SVD in detail (right): Pixel sensors form first (gray) and second (blue) layer. Third (green), fourth (red), fifth (orange) and sixth (pink) layers compose strip detector.

processing. The CAF is a natural extension to the framework profiting from the mature state of basf2, extending its capabilities to calibration of the detectors using collected data, with the goal to deliver data with a quality suitable for physics analysis. The CAF standardizes and simplifies calibration development by exploiting the python side of basf2, where development is much more flexible to define data processing workflows, while the CPU-intensive calculations use compiled C++. The CAF should motivate developers to implement often existing algorithms within basf2 by providing an easy-to-use interface. This interface is similar to a well established access to data objects in basf2 and should thus be familiar to most basf2 developers, usually physicists.

The calibration procedures can be processed collectively in a predefined sequence. The procedures might run on input data with different run number and take into account variable time dependence of the calibration constants which are stored in a database. To reduce wall-clock time, independent procedures can be called in parallel. The calibration algorithm dependencies are resolved by CAF to determine the order and distribution of individual jobs, running data collection whose output is then passed to the specified algorithm. It is possible to manage and merge intermediate data objects (histograms, trees, or custom mergeable objects). The CAF should monitor the data quality to automate the workflow of the calibration process as much as possible. Data processing and final calibration constants should be monitored as well. The validity range of the constants is usually determined by CAF from input data, but can be adjusted by individual algorithms. The input data will be organized by run and experiment number, but the granularity can be arbitrary, even event-dependent.

The input data for calibration and alignment procedures comprise either simulated events offline or raw data, which could also be directly selected using the high level software trigger in the online configuration.

3. Design of CAF

The calibration and alignment framework is composed from two virtual C++ classes providing interfaces to calibration developers: *collector module* and *calibration algorithm*. A Python framework, with which the calibration user interacts via a Python interface [3], controls the execution of collectors and algorithms.

1) **Collector module:**
   Collector module builds data objects for the calibration algorithms. It has two parts: preparation and collection. In the first part, a user defines and initializes objects (trees, histograms, etc.), which are filled in the second part of collector module.
   For example, *MillepedeCollector* is used to align the Belle II tracking detectors (PXD, SVD, KLM, CDC). This module re-fits the selected reconstructed tracks with General Broken Lines in the collection stage and outputs track data including all measurements with their uncertainities and the full track covariance matrix, later needed by the Millepede II algorithm (for explanation see **Alignment tools**).
2) **Calibration algorithm:**
   Calibration algorithms is a base C++ class which developers inherit from to implement their own algorithm. It analyses collected data and determines calibration and/or alignment constants. The output status is one of:
   a) *Success*: The algorithm was successfully finished. Calibration or alignment constants are saved to current local database.
   b) *Iterate*: The algorithm was successfully finished, but iteration of collector and calibration is requested. The calibration or alignment constants are saved in current local database.
   c) *Not enough data*: The algorithm did not have enough data to successfully finish. No constants are saved.
   d) *Failure*: The algorithm failed for any other reason. No constants are saved.
3) **Python framework and user interface:**
   To provide the user with workflow flexibility, the Python interface makes it possible to separate the collection and calibration stages and to include or exclude other related standard modules (simulation, reconstruction, etc.) and user-written modules. The Python interface permits parallel processing of data from multiple runs. Fast turn-around can be achieved by using a computing cluster as processing backend. During calibration development, it might be more convenient to run separated algorithms locally for testing and debugging. The backend, splitting jobs locally among processor cores, is easily changeable for a job submission system allowing for a very large number of parallel processing jobs.

4. Alignment tools

One important function of CAF in the future Belle II experiment will be the position alignment of the detector sensors. As the pixel detector is mounted directly to the beam pipe, it is mechanically independent from the rest of the detector. It is expected that the relative position of both PXD and SVD will need to be continually monitored and aligned to deliver high quality data. This might gain additional importance with the data reduction schemes for the pixel detector. Another task is monitoring of the primary beam spot, whose position is a crucial input of the time dependent CPV analysis. To deal with possible correlations among the sub detectors, we use a global alignment method using the Millepede II (MP-II) tool [2], integrated in basf2 and CAF. MP-II is used for track-based alignment of the vertex detector (PXD + SVD = VXD), the beam spot position and the K_L and muon system (KLM). An interface to the CDC already exists and is being tested with first cosmic data. The integration of several completely different sub detectors profits from the generic approach of the MP-II, General Broken Lines refit[5] and GENFIT toolkit[6], described below.

i) **Millepede II algorithm**
   The Millepede II algorithm is a tool for determination of a large number of (for example)

alignment constants based on simultaneous (linearised) minimisation of measurement residuals with respect to all track and alignment parameters. As no approximations are used, except linearization, all correlations are kept in the solution. For nonlinearities, the procedure can be iterated.

ii) **General Broken Lines**
The General Broken Lines (GBL) method provides fast track refit with proper treatment of multiple scattering. The method introduces additional fit parameters: kink angles at predefined scattering points. A track seed is propagated in detector material to parametrize multiple scattering and populated with non-measurement scattering points. The constructed global covariance matrix of the track, containing correlations among detector layers induced by the multiple scattering, computed by GBL, is a crucial input of MP-II. In CAF it can be stored in binary files for MP-II or in an intermediate state in a ROOT tree.

ii) **GENFIT**
The GENFIT toolkit for generic track fitting in high energy physics experiments was heavily extended for the needs of the Belle II experiment, including integration of the General Broken Lines fitting method. GENFIT allows to seamlessly integrate measurements from different types of detectors and include them in the fit. It also contains all tools needed for extrapolation in the detector and handling its material distribution, important input to the General Broken Lines.

The alignment tools use initial positions of sensors, wires or plates and reconstructed track information. The calculated corrections to alignment constants of those elements are stored in a database and used for subsequent data collection in CAF. Due to the complexity of the experiment, it is desirable to check the validity of the used calibration procedures by ad-hoc studies or by developing an alternative method. An example is the drift chamber (CDC) where code inspired by the existing approach at Belle was used for the first acquired cosmic data. The MP-II technique might here be independently compared to a different procedure, for which CAF seems to be an ideal playground.

5. Types of calibration and alignment

The same or dedicated procedures might be employed at different levels of the calibration workflow. The introduced aproach can for example be used for a differential alignment of PXD, where the position and orientation of individual sensors are kept fixed, while the PXD is allowed to move as a whole or e.g. by layer. For such cases, a few hundred tracks are sufficient to determine accurate alignment parameters. On the other hand, for the finest calibration at sensor or wire level, many large track samples with different topologies, including magnet-off cosmics, are needed. In this sense we divide the calibration to several types, listed below.

I) **Online** calibration uses a small set of quickly reconstructed data. The online reconstruction uses a tailored set of quick algorithms for partial event reconstruction. The calibration of beam spot position is an example of a continuously monitored quantity important also for the accelerator, for which dedicated fast turn-around algorithms might be needed.

II) **Fast** calibration and alignment will determine constants required to produce analysis quality data. It will trade accuracy for speed. Calibration and alignment will be done during data conversion. Turn-around time should be less than one day from collection of the input raw data.

III) **Offline or reprocessing** calibration and alignment will run as many calibration algorithms as required to produce best-quality constants. Large datasets will be used including cosmic ray and field-off data.

6. Alignment and calibration examples

A) Vertex detector (VXD, PXD and SVD together)

- In simulated data (with only gaussian hit smearing to account for sensor resolution and only MC track finding), sensors are randomly misplaced by up to ∼200μm (u, v, w) and misaligned by up to ∼3.5 mrad (alpha, beta, gamma).
- After alignment, which considers all sensors to be rigid bodies with 6 parameters, residuals are ± 4.6 μm (du, dv, dw) and ± 0.1 mrad (dalpha, dbeta dgamma).
- Residual misalignment is almost independent on the degree of random misalignment. Dedicated studies are being performed for alignment of a systematically deformed detector.

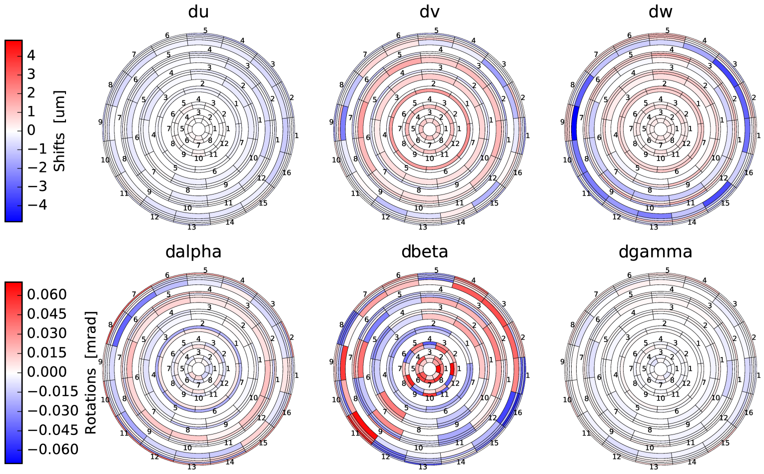

Figure 2. *Alignment of mis-aligned VXD in simulation:* The largest alignment corrections can be seen in wedge SVD sensors (innermost in last 3 SVD layers). The beta angle shows the largest residual misalignments as it is sensitive to the shorter size of the silicon sensors, making resolution in this angle worst.

B) Central drift chamber (CDC)

The CDC employs its own well established procedures for calibration of many different aspects of the drift chamber, including wire time offsets, x-t relation and resolution parametrization. It also has its own alignment procedure which was applied at layer level, allowing for layer shifts and rotations to be determined using the initial cosmics data. Example results showing the improvement in helix parameter estimation for cosmic tracks can be seen in fig 3.

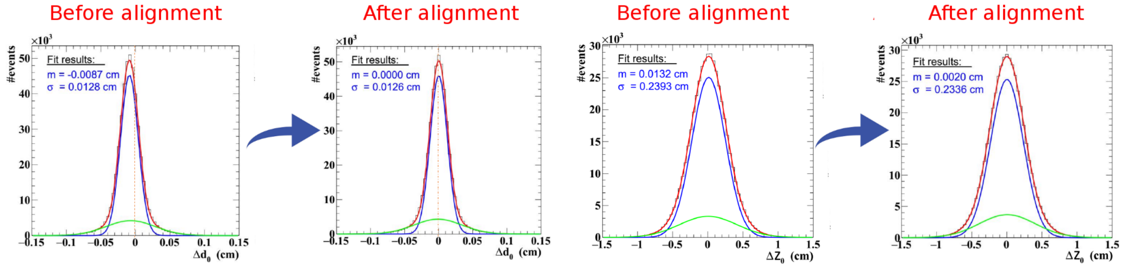

Figure 3. *Calibration of CDC using real data*: Distributions for Δd_0 (left double plots) and ΔZ_0 (right double plots) before and after using alignment procedure.

C) K_L and muon detector

KLM, divided into barrel (BKLM) and endcap (EKLM) parts employs a slightly different alignment approach. However, for both detectors the approach is very similar to the silicon detector, where the individual segments or modules of the detectors are considered as rigid bodies. In fig. 4, an example from a study of weak alignment modes using different simulated track samples can be seen for BKLM.

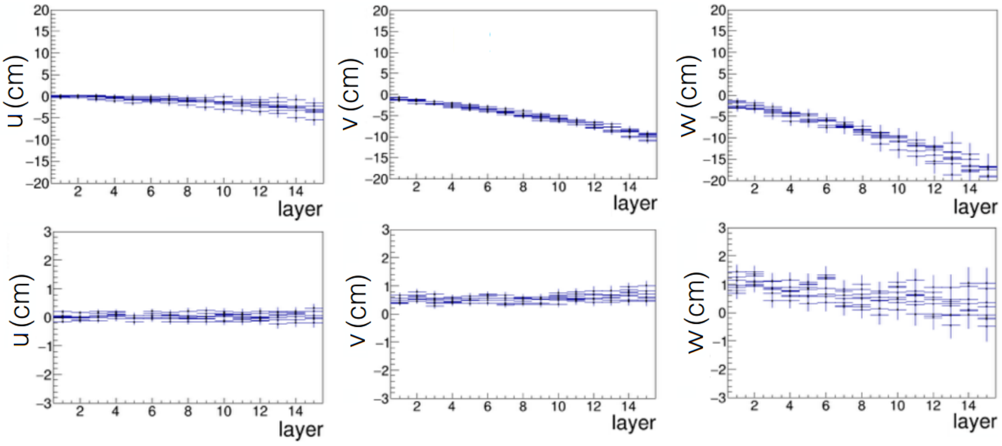

Figure 4. *BKLM alignment using simulation*: Comparison results of alignment using single muons (top line) and vertex-and-beam-constrained pairs of muons (bottom line). With the second sample, the vertex constraint and usage of the whole muon pair as a single object significantly reduces some weak modes in the alignment constants.

7. Summary

Calibration and alignment of the Belle II detector is a complex task which requires significant effort from many experienced physicists. The CAF framework presented here aims to allow experts to concentrate their effort on algorithm development and minimize time spent on coding and ensuring scalability. Large scale application of the Millepede II algorithm for several Belle II sub-detectors is one example, which will profit from CAF features. With the initial cosmic ray data, and with first physics data in 2018 without the full vertex detector, the algorithms are expected to evolve and in many cases be fully integrated within CAF. In 2018 they need to reach the performance expected for full detector operation and this paper presented some of the continuous work required to reach that goal.

Acknowledgments

This project was supported by Charles University, project GA UK no. 404316.

References

[1] *Z. Dolezal, S. Uno et al.:* Technical Design Report, arxiv: 1011.0352v1
[2] *V. Blobel, C. Kleinwort:* A new method for the high-precision alignment of track detectors, arXiv:hep-ex/0208021
[3] *D. Dossett:* Status of Alignment and Calibration Framework at Belle II Experiment, goo.gl/SEYfRB [5th August 2017]
[4] *D. Y. Kim* Software and Physics Simulation at Belle II arXiv:1510.09008 [physics.ins-det]
[5] *C. Kleinwort:* General Broken Lines as advanced track fitting method, arXiv:1201.4320 [physics.ins-det]
[6] *J. Rauch, T. Schluter:* GENFIT a Generic Track-Fitting Toolkit, J. Phys.: Conf. Ser. 608 012042 (2015)