ALIGNMENT AND CALIBRATION FRAMEWORK FOR THE BELLE II DETECTOR

Tadeas Bilka, Jakub Kandra et al. on behalf of the Belle II collaboration
Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic

Advanced Computing and Analysis Techniques in Physics Research in Seattle, 21st August 2017

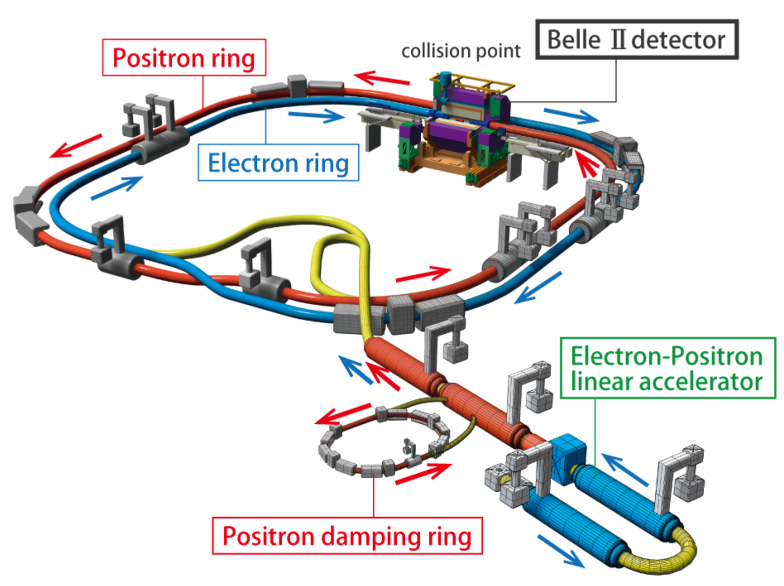
Belle II experiment

Belle II and SuperKEKB:

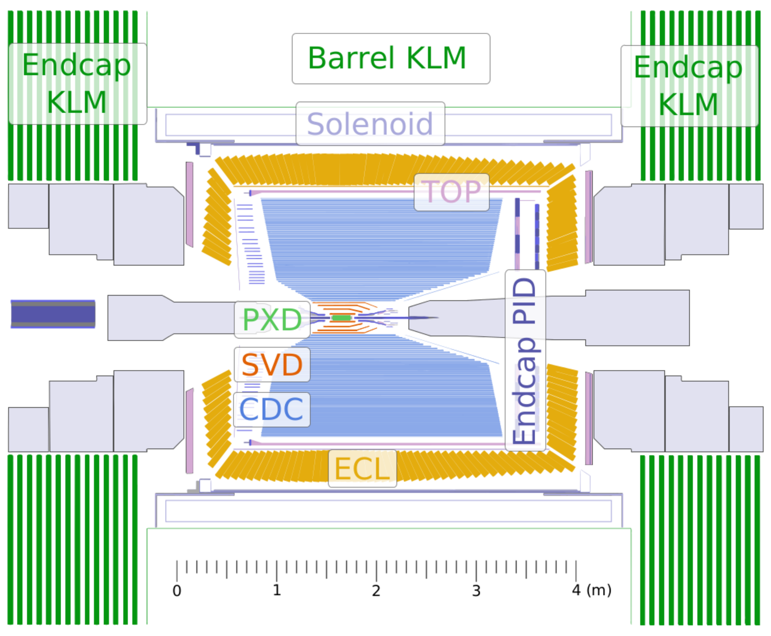
- KEK, Tsukuba, Japan
- Next generation B-factory
- Search for New Physics
- Study of CP violation in B-meson decay
- e^-e^+ asymmetric collider (SuperKEKB)
  - Beam energy of e^- is 7 GeV
  - Beam energy of e^+ is 4 GeV

Belle II detector:

- DEPFET pixel (**PXD**) sensors
- Double-sided strip (**SVD**) sensors
- Central drift chamber (**CDC**)
- Time of propagation (**TOP**) counter
- Aerogel ring-imaging Cherenkov (**ARICH**) detector
- Electromagnetic calorimeter (**ECL**)
- K_L and muon (**KLM**) detector



Schematic view of SuperKEKB collider



Cross-section of Belle II detector

Design of CAF

The calibration and alignment framework is implemented to Belle analysis and simulation framework 2 (basf2), which the user controls via a Python interface. It is composed by two virtual C++ classes: *collector module* and *calibration algorithms* and Python interface is used[1].

1) **Collector module:**
Collector module builds data objects for the calibration algorithms. It has two parts: preparation and collection. In the first part user define and initialize 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 is used the Millepede II algorithm (for explanation see **Alignment tools**).

2) **Calibration algorithms:**
Calibration algorithms is a base C++ class which developers inherit from to implement their own C++ Algorithm class. It analyse collected data and determine calibration and 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) **User interface in Python:**
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.

Alignment tools

The important function of CAF is the position alignment of the detector sensors. The modern alignment tools are used for silicon sensors (PXD and SVD) only, and they are able to clearly and precisely calculate position of wires (CDC), or muon-detector strips (KLM). The track-based alignment for tracking system, K_L and muon system use Millepede II (MP-II) algorithm and General Broken Lines (GBL) method to precisely determine calibration and alignment constants.

i) **Millepede II algorithm**
The Millepede II algorithm is tool for precise determination of alignment constants and it is based on simultaneous (linearised) minimisation of residuals with respect to all tracks and alignment parameters. No approximations is used. The linearised minimisation expects iterations of alignment and calibration algorithms, but all parameters corrections are kept from every step of iterations.

ii) **General Broken Lines**
The General Broken Lines method provide fast track refitting with 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. A track is populated with non-measurement scattering points.

The alignment tools use initial positions of sensors, wires or plates and reconstructed tracks information. The calculated correction to alignment constants and multiple scattering parameters can be stored in local database, binary files, or used in continued alignment analysis.

Types of calibration and alignment

I) **Online** calibration and alignment uses a small set of quickly reconstructed data. The online reconstruction uses a tailored set of quick algorithms for partial event reconstruction.

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

III) **Offline or reprocessing** calibration and alignment will run as many calibration algorithms as required to produce best-quality constant. Large set of data will be used.

Reference

[1] *D. Dossett*: Status of Alignment and Calibration Framework at Belle II Experiment, goo.gl/SEYfRB [5^{th} August 2017]

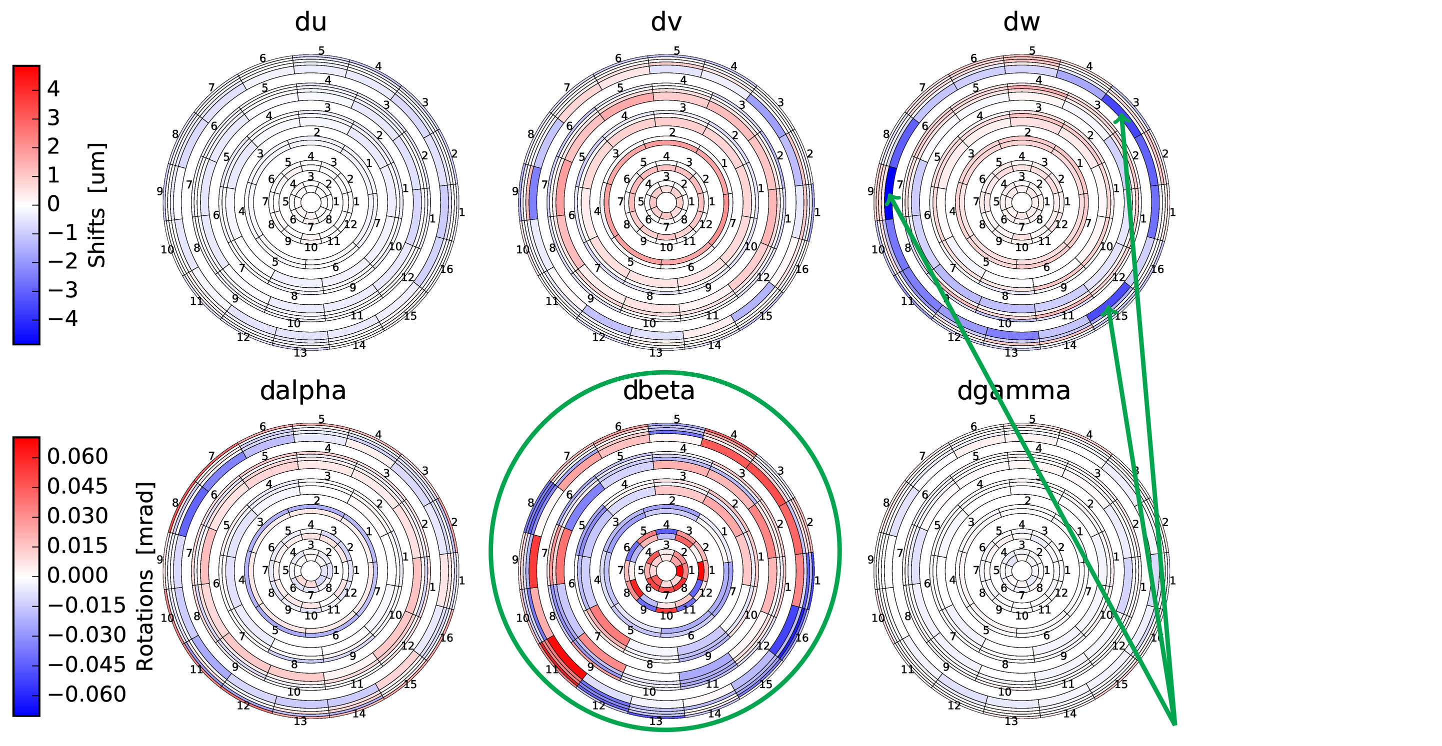
Calibration and alignment framework

Calibration and alignment framework (CAF) is built using automated calibration procedures of subdetectors of Belle II. CAF standardizes and simplifies calibration development. The calibration procedures can be processed collectively in predefined sequence. The procedures should run on input data with different run number. The output calibration constant must be written to a database. To reduce wall-clock time, independent procedures can be called in parallel processes.
The framework motivates to develop similar universal algorithms. The calibration algorithm dependencies should be managed and checked. It is possible to manage and merge intermediate data objects (histograms, trees, other objects). It should monitor data to automate start of new calibration process. Processing of procedures and final calibration constants should be monitored during use of CAF. Validity range of calibration should be managed. The input data will be organized dependent to run number.
The input data to calibration and alignment procedures comprise either raw or simulated events.

Alignment and calibration examples

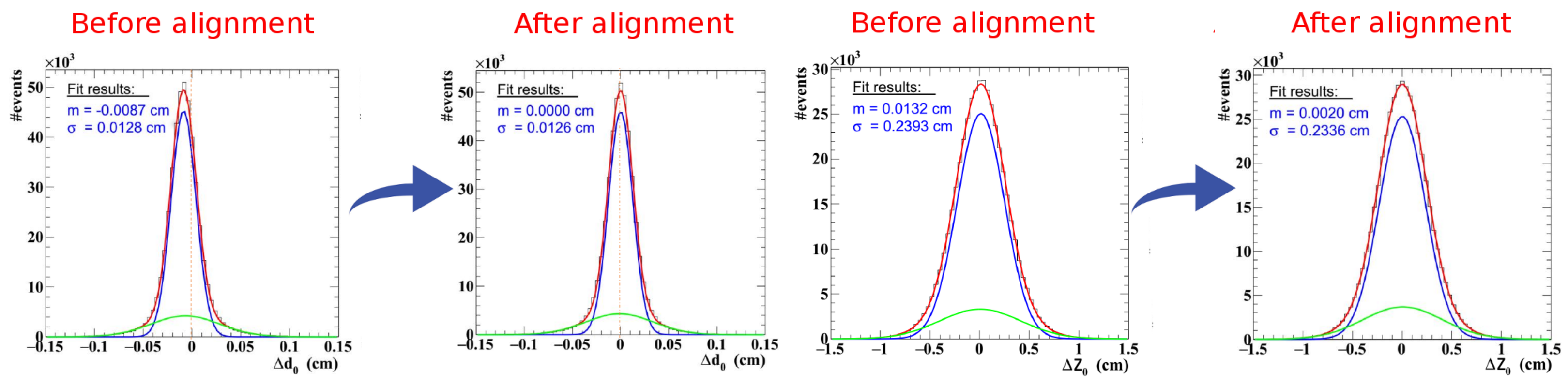
A) **Vertex detector (PXD and SVD together)**

- In simulated data, sensors are randomly misplaced by up to ~200μm (u, v, w) and misrotated by up to ~3.5 mrad (alpha, beta, gamma).
- After alignment, residuals are $\pm$ 4.6 μm (du, dv, dw) and $\pm$ 0.1 mrad (dalpha, dbeta dgamma).
- Residual misalignments are same as for perfectly aligned VXD and are independent of degree of misalignment.



Alignment of mis-aligned VXD in simulation: Alignment poorly fixes wedge-sensor position and long-axis rotations (beta angle).

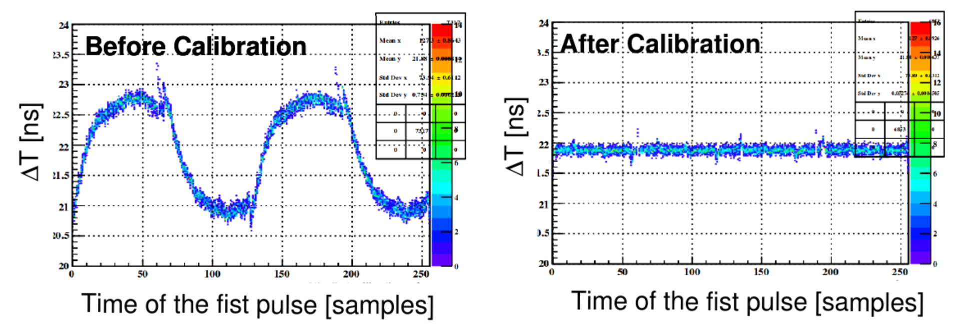
B) **Central drift chamber (CDC)**



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) **Time of propagation (TOP) counter**

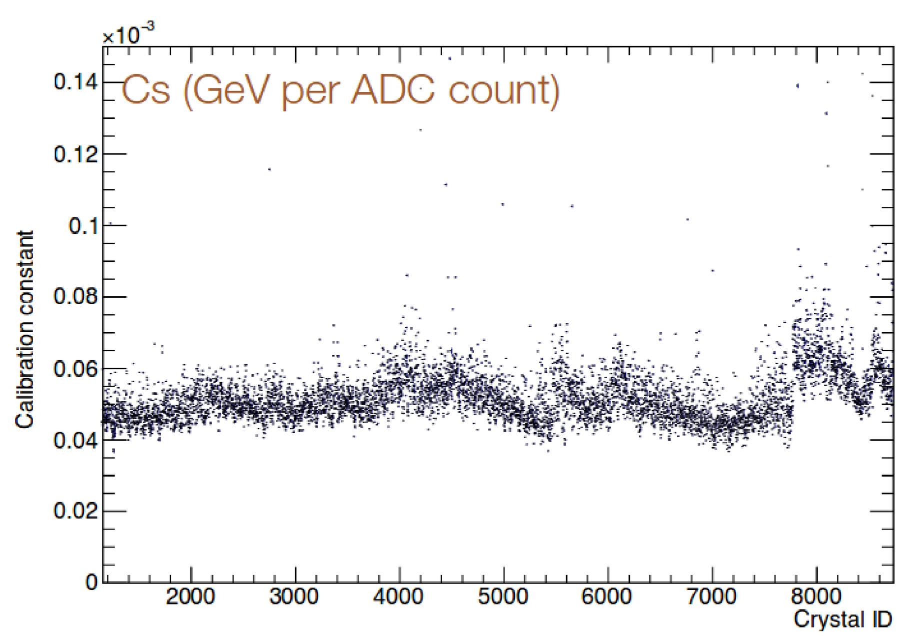
- Time base correction calibrates different time constant for capacitors.
- Calibration uses injected double pulse with known delay.
- Measurement of delay as function of sample number.



Calibration of TOP counters using real data: Time base correction before (left) and after (right) calibration

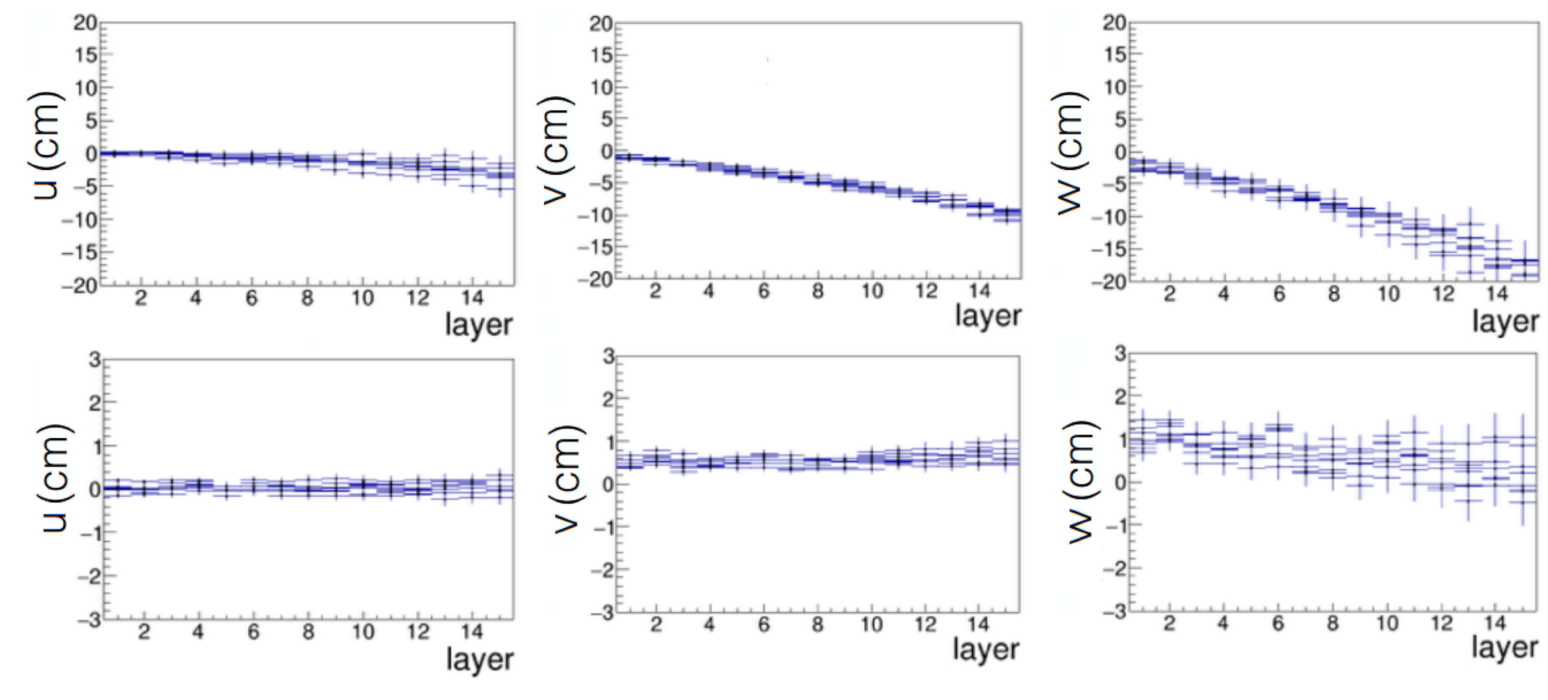
D) **Electromagnetic calorimeter**

- Individual crystal energy is defined as $E[GeV] = A[ADC] \times C_e \times C_s$, where A is measured amplitude in analog-to-digital units, C_e is electronic calibration constant and C_s is single crystal energy calibration constant.
- Single crystal energy calibration constant corrects for variation in light yield. Calibration constants can be calculated from several different datasets (etc. cosmic rays, $\gamma\gamma$, $\mu^-\mu^+$, ...)



Calibration of ECL crystals using real data: Values of calibration constants as function of crystal ID.

E) **K_L and muon detector**



BKLM alignment using simulation: Comparison results of alignment using single muons (top line) and vertex constrain pairs of muons (bottom line)