Charles University in Prague

Faculty of Mathematics and Physics

MASTER THESIS

Tadeáš Bilka

Simulation and analysis of tests of Belle II vertex detector sensors

Institute of Particle and Nuclear Physics

Supervisor of the master thesis: RNDr. Peter Kvasnička

Study programme: Physics

Specialization: Particle and Nuclear Physics

Prague 2014

I would like to thank my supervisor, Peter Kvanička and consultant, Peter Kodyš for their support during the whole time of my master studies. I would also like to use this opportunity to thank Sergey Yaschenko, my former supervisor during the DESY Summer Student Program, where I was introduced to the problem of detector alignment. For introduction into Millepede and General Broken Lines, as well as for non-stop support of these tools and many interesting and valuable discussions, I would like to express my gratitude to Claus Kleinwort. Finally, this work would not be possible without support from my father, to whom I express the biggest thanks for all the support during my studies.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague, July 30, 2014

Název práce: Simulace a analýza testů sensorů vrcholového detektoru Belle II

Autor: Tadeáš Bilka

Katedra / Ústav: Ústav částicové a jaderné fyziky

Vedoucí diplomové práce: RNDr. Peter Kvasnička, Ústav částicové a jaderné fyziky

Abstrakt: Předložená diplomová práce popisuje vývoj a analýzu simulace, rekonstruckce a alignmentu pro testování sensorů vrcholového detektoru Belle II a proceduru pro alignment samotného vertexového detektoru. Prvním cílem práce bylo vytvoření a validace nástrojů pro analýzu testů sensorů přímo v prostředí společného softwarového rámce experiment Belle II. Druhým cílem byla příprava procedury pro alignment kompletního vertexového detektoru, která je nedílnou součástí procesu kalibrace. První část práce se věnuje experiment Belle II, vrcholovému detektoru, experimentálnímu testu jeho sensorů a popisu společného softwarového prostředí. Zvolená aligmentovací procedura, Millepede II, a fitování pomocí General Broken Lines jsou teroreticky objasněny. Druhá část práce pak shrnuje software, který byl pro práci vyvinut nebo použit a výsledky, které byly s jeho pomocí dosaženy, s důrazem na alignment.

Klíčová slova: Belle II, vrcholový detektor, alignment, fitování, testování detektorů

Title: Simulation and analysis of tests of Belle II vertex detector sensors

Author: Tadeáš Bilka

Department / Institute: Institute of Particle and Nuclear Physics

Supervisor of the master thesis: Peter Kvasnička, Institute of Particle and Nuclear Physics

Abstract: This master thesis describes development and analysis of simulation, reconstruction and alignment for tests of Belle II vertex detector sensors as well as alignment procedure for the vertex detector itself. The first goal is to develop and test software tools which would allow analysis of sensor tests within common Belle II software framework. The second goal is to prepare the alignment chain for the vertex detector, being essential component of the detector calibration. First part of the thesis describes the Belle II experiment and its vertex detector, as well as the experimental beam test of the sensors and the common software framework. The theory behind the used alignment procedure utilizing Millepede II for alignment and General Broken Lines for track fitting is explained. Second part of the thesis then summarizes software tools developed or used and the results reached with these tools with emphasis on the alignment.

Keywords: Belle II, vertex detector, alignment, fitting, beam test

Contents

Introduction

During past decades, high energy particle physics provided an impressive insight into the most fundamental aspects of nature. The standard model of elementary particles and their interactions evolved into a theory with large predictive power and during second half of 20th century, most of its predictions were confirmed in experiments. The discovery of its last missing piece, the Higgs boson in 2012 confirmed by two LHC experiments, ATLAS and CMS [1] is however not a satisfactory ending of the story. Many hints coming mainly from astronomical observations indicate the standard model is incomplete. One of the fundamental questions that remain unanswered is why the universe is made almost entirely of matter, while its counter-part, anti-matter is rather a rare substance. It was believed that matter and anti-matter should obey a combined, charge conjugation and parity symmetry, but then CP-violation was discovered in neutral kaons' systems in 1964. Such symmetry violation at the level of fundamental particles allows for possible explanation of matter – anti-matter asymmetry due to copious CP-violating interactions taking place short in the Big Bang. Even small effect could cause large overhaul of matter, however CP symmetry-breaking interactions discovered so far are not enough to explain matter abundance.

Since the discovery of CP-violation, intensive research led to many new findings, of which many took place at the Belle experiment at KEKB collider in Japan. This collider belongs to the so called B-Factories, which are designed to produce large amount of B mesons and their anti-particles. This is achieved by colliding beams of electrons and positrons at the center of mass energy equal to the rest mass of the $\Upsilon(4S)$ resonance. The mass of this resonance is just above threshold for decay into B^+B^- or $B^0\overline{B^0}$ decay which together make over 96% of the decay width. Belle belongs to experiments at the precision frontier, as the collision energy is fixed and thus high statistics and low systematic errors become essential to reach the desired physics performance. Belle was recording collision data for 10 years and after shutdown in 2010, it was decided to upgrade the accelerator and detector for much higher luminosity. This effort is represented by the Belle II collaboration.

The main upgrade of the detector is a completely new vertex detector (VXD), which incorporates two layers of pixel sensors with DEPFET technology. The pixel detector is surrounded by four layers of double-sided strip sensors. The VXD is an essential part of the detector responsible for precise vertex reconstruction. Before the complete device is delivered, it must be properly tested. This includes investigation of properties of the sensors under beam of high energy particles in so called beam tests, which study hardware operation but also can serve as validation of software tools for data acquisition and reconstruction. For reconstruction not only in the beam test, but mainly in the complete experiment, precision alignment of the detector is essential. This can be achieved by proper modelling of particle propagation and using measurements of charged particle trajectories in an alignment procedure.

The motivation for this thesis is twofold. The first one is need for a simulation reconstruction package for experimental beam test at DESY in 2014. The second is urgent need for an alignment procedure for the Belle II vertex detector, as well as for the beam test. The main goal was therefore implementation of the alignment. To achieve this goal a new track fitting procedure needed to be developed. The beam test served as a perfect simple test case to validate all the developed tools before being used for the Belle II detector.

This thesis therefore deals with two closely related topics connected with the vertex detector. Simulation and reconstruction of a beam test within standard software environment for Belle II is developed. In addition, an interface to General Broken Lines track fitting and alignment procedure utilizing Millepede II for the vertex detector as well as for the beam test is developed and results are shown. The First chapter describes the Belle II experiment and the vertex detector, the second one then gives an overview of the DESY VXD beam test in 2014. Chapter 4 explains the alignment procedure and related topics, while the GBL track fitting method, closely connected to the alignment, is discussed in Chapter 5.

The result of the work, developed software is described in Chapter 6. The remaining chapters give an overview and analysis of simulation and experimental results reached with the developed tools in the beam test and the complete Belle II vertex detector.

1 Belle II Experiment

The Belle experiment successfully operated from 1999 to 2010 and the data sample collected exceeds 1 ab^{-1}. The KEKB accelerator holds world record in instantaneous and total integrated luminosity and discoveries at Belle were recognized in the 2008 Nobel Prize award to Makoto Kobayashi and Toshihide Maskawa [2] for prediction of third family of quarks and its relation to CP-violation. Investigation of CP-violation in B meson decays by precise measurement of the decay channels was the main goal of Belle which was successfully reached, starting in 2001 by observation of time dependent CP-asymmetry in the decays process $B^0 \rightarrow J/\psi\, K_S^0$ and continuing with confirmation of both, direct and indirect CP-violation in several other decay channels. Belle also significantly contributed to precise measurements of complex phases of CKM unitarity triangle and confirmation of the standard model with Kobayashi-Maskawa mechanism. The spectrum of physics topics was much wider, including heavy quarkonium spectroscopy or tau lepton physics. The analysis of the huge data sample continues till present. [3], [4]

1.1 Belle II at SuperKEKB

To continue in deeper investigation of observed phenomena and to join the search for new physics behind the standard model, an upgrade of the accelerator and the detector was approved. The upgraded accelerator, SuperKEKB, will deliver 40 times larger luminosity than KEKB and the expected integrated luminosity is planned to reach $50\ \mathrm{ab}^{-1}$ in 2020. This increase in amount of data will pose new challenges to the detector and computing. [3]

To reach the extreme luminosity, SuperKEKB adopts the nano-beam scheme, where the vertical β function of the beam is significantly reduced by squeezing the beams just before the collision by accelerator optics. The upgrade of the accelerator also includes increase in beam currents and slight change in their energy. The high luminosity will lead to substantial increase in beam background, making also the upgrade of the detector necessary. [5]

The concept of the detector remains the same as in Belle: it is a cylinder which is forward/backward asymmetric due to the fact that collision products are boosted in the laboratory system. The decaying B mesons can thus travel further before the decay which improves resolution of the decay vertex position, being crucial for time dependent CP-violation measurements. The main improvement is a new pixel detector near the interaction point and extension of the strip detector to a larger radius. The detector is going to be equipped with a completely new particle identification device and new end-cap part of the calorimeter and the muon spectrometer. Momentum measurement relies on 1.5T magnetic field provided by a superconducting coil. The individual sub-detectors and upgraded parts are shown along the original Belle layout in Fig. 1.1.

Because of much higher background and resulting occupancy compared to Belle, the strip detector is not suitable anymore as first detection layer of the vertex detector. Therefore, the Pixel Detector (PXD) with two layers of sensors is surrounding the beam pipe. The strip vertex detector (SVD) follows, having four layers of double sided strip silicon sensors. PXD and SVD form the vertex detector (VXD), responsible for precise vertex reconstruction. All sensors of VXD are based on silicon semiconductor technology and, to not disturb particle trajectories due to multiple scattering, the design of the vertex detector must be made very lightweight, with support structures and electronics reduced as much as possible in the acceptance region of the detector. The PXD starts just 14 mm from the interaction point with two layers of DEPFET sensors surrounded by 4 layers of the strip vertex detector (SVD). These six layers of silicon sensors of the vertex detector of Belle II are discussed in the following sections in detail.

The vertex detector is surrounded by the Central Drift Chamber (CDC) which is filled by mixture of gasses (He + C_2H_6). It consist of about 42000 sense wires measuring signal of ionizing particles crossing CDC volume. The wires are organized into layers and superlayers, with larger density of sense wires near VXD in so called small cell chamber. The task of CDC is to measure momenta of particles from curvature in magnetic field and help in particle identification from average energy loss in the gas. It is also an important part of the trigger, providing signal for charged particles.

CDC is followed by the particle identification device (PID). It is based on Cherenkov radiation measurement and separated into barrel and end-cap part. The main purpose of PID is separation of kaons and pions using momentum measurement from CDC and VXD. The barrel part consists of time-of-propagation counters in quartz bars while the end-cap utilizes proximity-focusing aerogel ring image Cherenkov detector.

Energy of the particles is measured in the Electromagnetic Calorimeter (ECL). It isespecially important for energy measurement of photons from decays of neutral pions. ECL consists of from CsI(Tl) crystals in the barrel part and pure CsI crystal in the end-cap. The crystal are oriented so, that they point towards the interaction point. The light output of the crystals is converted to electronic signal in photomultipliers. For the end-cap, the lower light output of pure used CsI crystals with shorter scintillation time is compensated by using vacuum photo pentodes. ECL plays an important role in the triggering system.

Outside the superconducting coil, the KL and Muon spectrometer (KLM) is installed. The barrel part uses resistive plate chambers between iron plates, which serve as return circuit for the magnetic field flux. To deal with increased background, the end-cap is utilizing scintillators instead resistive plate chambers. The light output of the scintillators is guided to silicon photodiodes, able to operate in strong magnetic field of the detector.

The increase in experiment luminosity will pose a significant challenge to the trigger system, responsible for primary selection of events for readout. Because vast amount of data comes from PXD and SVD, readout of these detector is initiated by signal in outer detectors. CDC provides charged track information, ECL provides energy measurement for charged and neutral particles. For precise timing, the PID information is used and KLM offers information about muon tracks. The main purpose of the trigger is to select physically interesting events, like Upsilon decay or production of tau leptons. Several processes are selected for luminosity monitoring and calibration. The trigger will operate at maximum average rate of 30 kHz. [5]

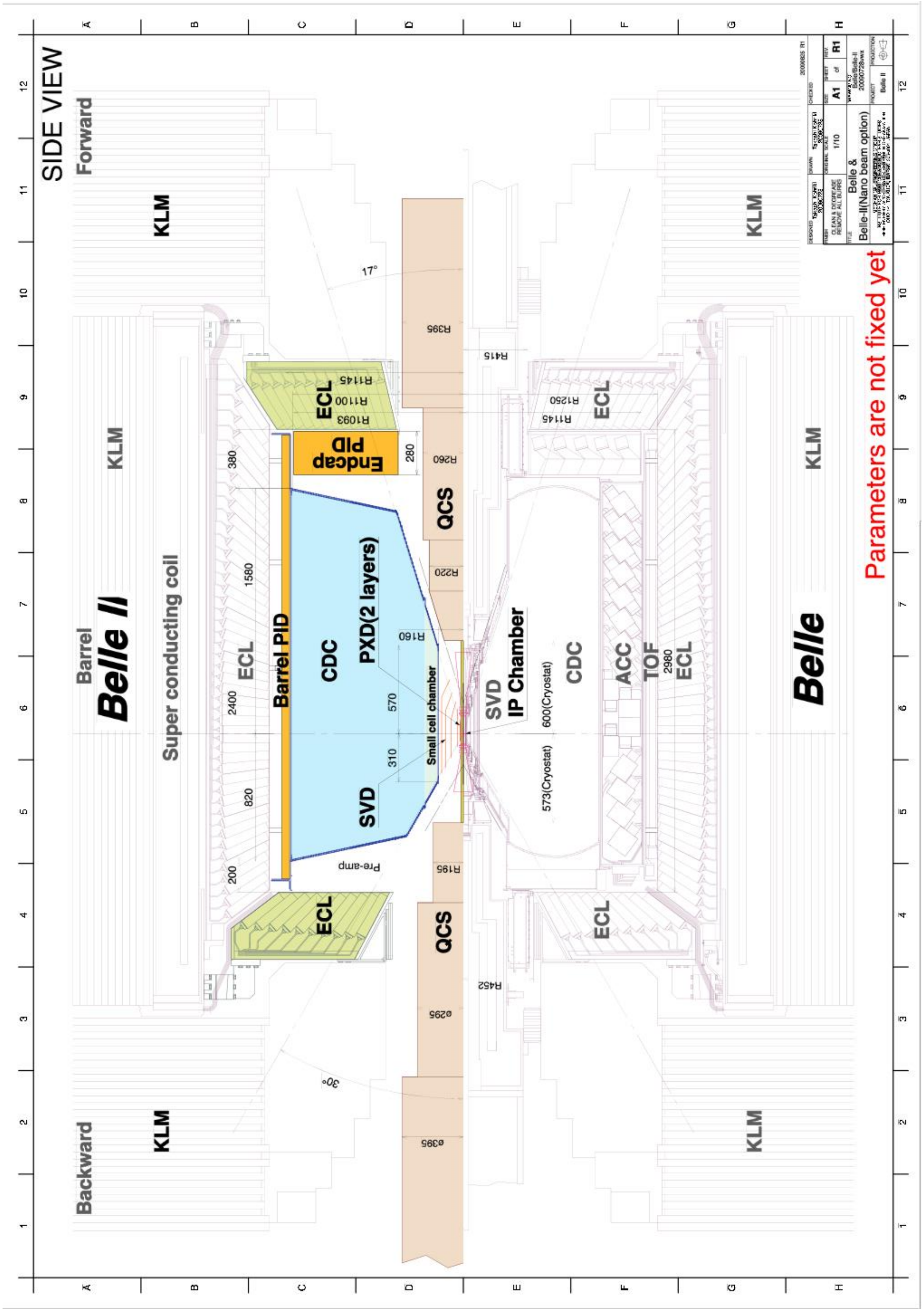


Figure 1.1 The design of the Belle II detector (top) compared to Belle (bottom). New or upgraded parts are colored. [5]

1.2 Pixel Detector

The pixel detector for Belle II must face several challenges. The radius of the beam pipe is only 10 mm and the first layer of sensor is just 14 mm from the interaction point. The extreme luminosity means significant increase of background mainly from QED processes. The proximity of PXD to the collision point allows for very precise vertex reconstruction of B mesons' decays, but because to reach such goal, the material budget of PXD has to be very minimalistic. On the other hand, the huge background such near the interaction point require very radiation-hard technology. All the requirements are fulfilled by the DEPFET (DEPleted Field Effect Transistor) technology, which has been chosen for the pixel detector of Belle II. [5]

The DEPFET technology was invented in 1987 by Josef Kemmer and Gerhard Lutz and its current development and large scale fabrication is maintained by the MPI-Semiconductor Laboratory. It combines particle detection and amplification within each single pixel in a compact structure. Low capacitance and high signal-to-noise ratio allows for fabrication of very thin devices. The area of DEPFET applications covers optical photon sensors, X-ray imagers and particle trackers. [6]

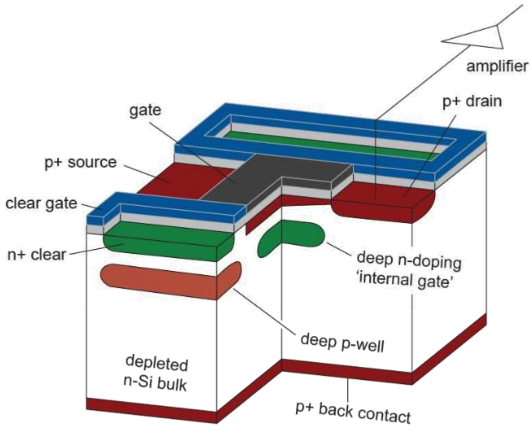


Figure 1.2 Cross section of a DEPFET pixel. The source and drain belong to the field effect transistor with gate shown in black. The internal gate is just below (green).

A cross section of single DEPFET pixel is show in Fig 1.2. It is composed from p-channel field effect transistor on n-type bulk. The bulk is fully depleted by applying high voltage to the bottom p-type contact. The doping profile of the bulk forms a minimum of the potential for the electrons is a small region under the transistor gate channel, called internal gate. A charged particle traversing the bulk leaves electron-hole pairs being immediately separated by the internal electric field. While holes drift to the p-contact, electrons concentrate in the internal gate. When readout is forced externally, the gate of the transistor opens and flowing current is modulated by the charge collected in the internal gate. The charge stored in the internal gate is not influenced by the readout and together with electrons from the thermal noise has to be removed. This is done by increasing voltage on the clear contact, normally shielded from the electrons during charge operation the p-type doping under the contact. During clear, electrons are removed from the internal gate and the pixel is ready to continue charge collection.

The larges DEPFET matrices in PXD have 1600 rows and the full device has be read-out within 20µs. The readout is done in parallel on four rows at the same time. During this operation, SWITCHER chips at the edge of the sensor turn on pixels' transistors in each row and the current is collected at the end of the sensor in drain current digitizer (DCD) chips. These convert the signal to an 8-bit number and send it to the data handling hybrid (DHH) chips, providing signal processing, zero-suppression, buffering and compressing.

Also, because PXD will have about 8 million pixels, it is not possible to readout complete PXD at full trigger rate. Therefore, the background has to be reduced before readout by selecting regions of interest using extrapolation from SVD.

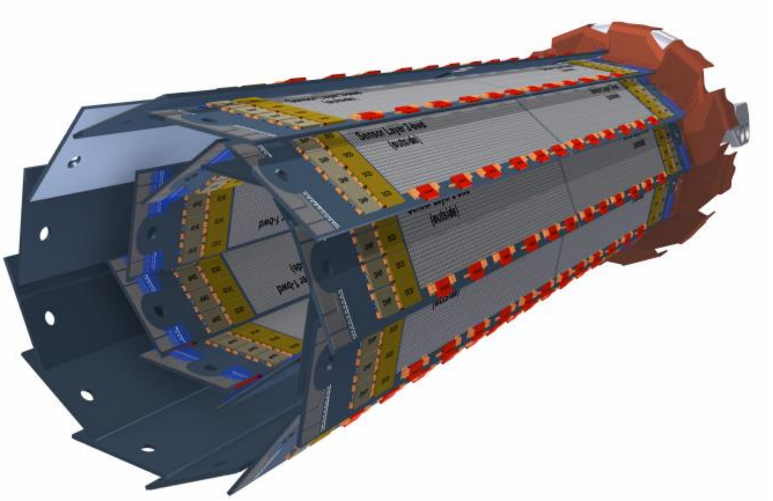


Figure 1.3 Drawing of the pixel detector sensors. The active area is shown in shadow, SWITCHER chips at the edges (red) and DCD (yellow) and DHH chips at the ends of the sensors.

PXD is composed of two sensor layers organized in a windmill structure, see Fig. 1.3. They are at radii of 14 and 22 mm and the support structure is mounted directly to the beam pipe, making PXD mechanically independent from the rest of the detector. There are 8 sensors in the first layer with pixel pitch 50x50um and 12 sensors in the second layer with 50x75um pixels. PXD covers the full angular acceptance $(17^\circ < \theta < 150^\circ)$ using modules with a width of 15 mm. The length of the sensitive area is 90 mm in the first and 123 mm in the second layer. All sensors have sensitive area thinned to 75 µm. Thanks to internal pre-amplification of DEPFET this does not reduce particle detection capabilities, but minimizes the material budget and multiple scattering occurring in PXD. Only non-thinned area of the sensor is its support frame which holds the SWITCHER chips at 2 mm wide sensor edges, while the DCD and DHH chips are located outside the acceptance region at the ends of the sensors as well as the support structure.

1.3 Strip Vertex Detector

The strip vertex detector (SVD) consist of 187 double sided silicon strip sensors made of n-type bulk with high resistivity. The thickness of the sensors is 320 μm and their area is close to current typical limits of production facilities for fabrication of monolithic silicon wafers. Large area is needed for material budget reduction. [5]

SVD cooperates with PXD mainly in measuring B decay vertices, but SVD allows to obtain vertex information also in other decay channels, where the decay happens after PXD. Determination of the vertex then requires SVD to connect tracking information from CDC and signal in PXD via extrapolation and to provide regions of interest for readout and data reduction in PXD.

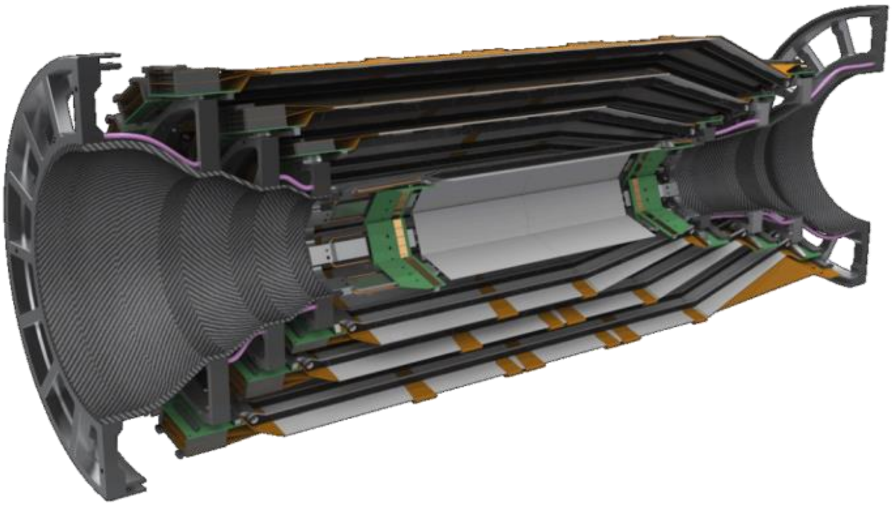


Figure 1.4.: Drawing showing a section cut of SVD and its support structure. The small rectangular sensors are in the middle near PXD (not shown). The slanted sensor in the forward region are visible on the right side.

The sensors of SVD are organized into four layers and 49 ladders at radii 38, 80, 115 and 140 mm, covering the full acceptance region of Belle II. Two kinds of sensor shapes are used. Rectangular sensor sensors are used in the barrel part, with smaller sensor at the inner layer. The parameters of the sensors can be found in Chapter 7 devoted to the beam test. In the forward region, sensors with trapezoidal shape which are slanted to reduce the material budget, see Fig. 1.4. They have a variable pitch size on the p-side, from 75 to 50 μm, and pitch 240 μm at the n-side. All sensors are connected APV25 front-end chips where each reads out 128 strips.

2 DESY VXD Beam Test

Before a real HEP detector is constructed, a long phase of design, testing and commissioning of its sub-components is necessary. Usually each experiment is unique and deals with different challenges. Because or large financial resources invested, it is of high importance to validate all components before final delivery. Silicon detectors can be tested in laboratories for electrical characteristics and, using laser beam, basic response of the detector can be studied. In reality, the detector will however measure trajectories of charged particles. In principle, cosmic rays can be used for such purposes, but the low intensity makes cosmic rays not feasible in most situations. Therefore the detectors are tested at particle accelerators with beam of high energy charged particles. Beam tests also allow to check the overall hardware infrastructure and software tools, and to gain experience with operation with the device under test.

Several laboratories around the world provide infrastructure for such beam tests. At CERN, for example, a 120 GeV/c pion beam being produced as secondary particles within heavy targets from protons of SPS accelerator. In Belle II, lower momentum of particles is available. The DESY laboratory in Hamburg, Germany, offers infrastructure which allows to almost reconstruct conditions in VXD. During 4 weeks from January to February 2014, a beam test was held at DESY, which for the first time investigated PXD and SVD cooperation also in magnetic field.

Idea of the beam test was to test one section of the vertex detector, 2 layers of PXD and 4 layers of SVD under different beam energies and also within magnetic field. The goal of this beam test was a complete system integration, from lowest to highest level, from hardware to software. Simultaneous cooperation of all sub-systems was achieved for the first time. Because full scale Belle II PXD matrix was delivered just on time, only one device was finally available for the beam test, which arrived having only basic electrical tests done in the lab. The goal of this beam test was achieved and the many completed milestones include:

- First test of simultaneous operation of PXD and SVD.
- Test of the cooling system and slow control systems.
- Complete read-out and data acquisition chain scaled for the beam test including all key features including Belle II online data reduction using both, hardware based and online high level tracking. The system use SVD to select regions of interest in PXD to be read out.
- Real time processing and reconstruction using basf2.
- Track finding based on combination of cellular automaton and neural network.
- High level software features such as alignment.

The beam test was conducted on beam line 21 at DESY II accelerator. Beam of this electron/positron synchrotron goes through a carbon fiber and bremsstrahlung photons are converted in an exchangeable target, see Fig. 2.1 (left). A magnet then separates electrons and positons and using the collimator, desired momentum of the beam is selected and sent to the experiments. Momentum of the beam goes up to 6GeV/c and its rate depends on the selected momentum and used target, see Fig. 2.1 (right). The beam has energy spread about 5% and divergence of 2 mrad in the experimental area.

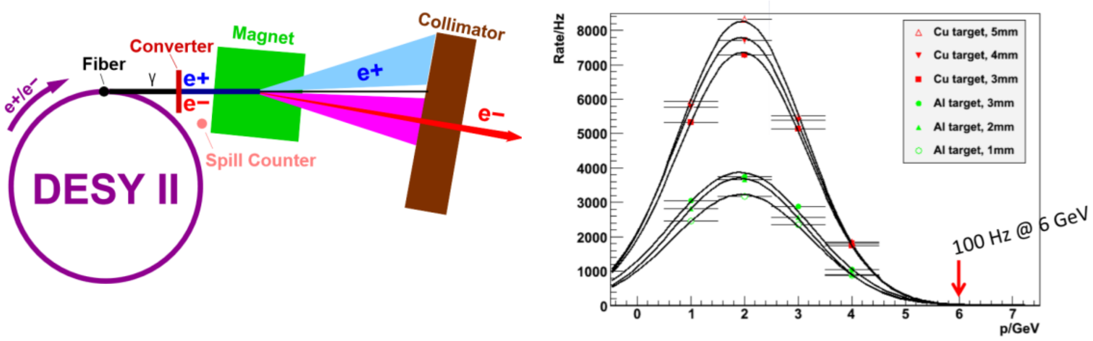


Figure 2.1 Left: The DESY II electrons produce photons in carbon fiber, which are converted in the target and split by the magnet. The collimator then selects electrons/positrons and momentum. Right: Average rate of particles in the beam for different momenta and selected targets. [7]

To investigate operation of the sensors in magnetic field, the PCMAG superconducting coil was installed in the area, see Fig. 2.2 (right). This magnet offers magnetic field with strength up to 1T. As the beam goes through magnet wall it was made quite lightweight, see Fig. 2.2 (left) and its total radiation length is only about 20% X_0. PCMAG offers field which is almost constant in the central part of its experimental chamber. The magnet is installed at a moveable stage which allows for precise rotation around vertical axis and 3D translation.

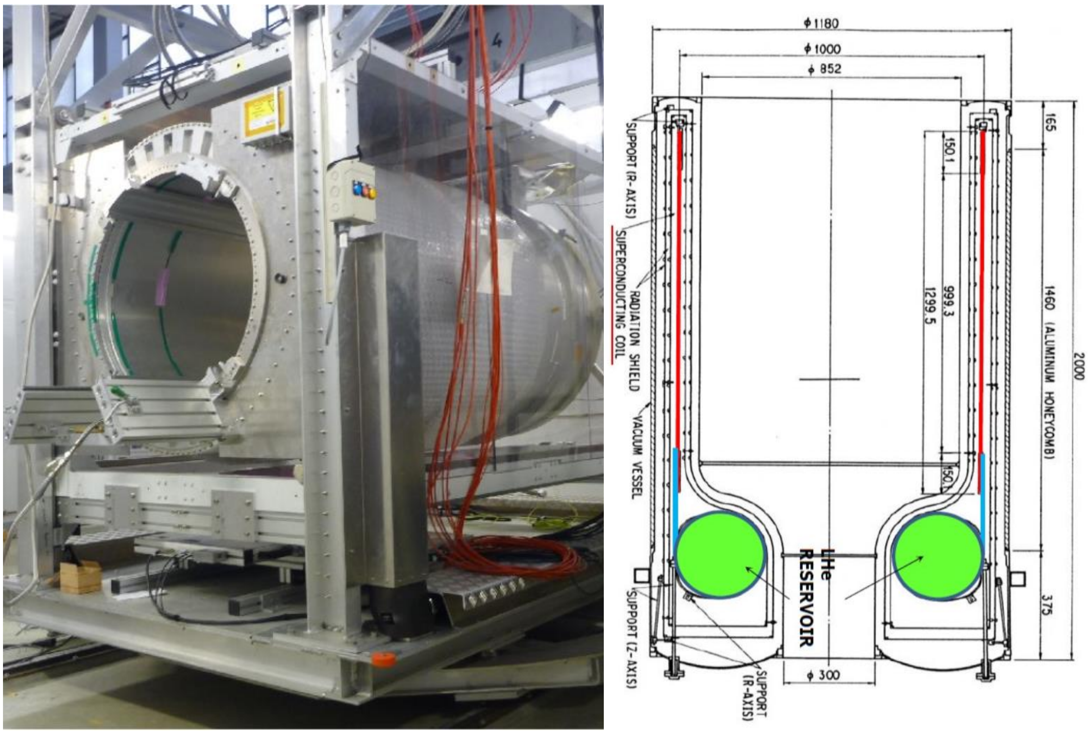


Figure 2.2 Photo of PCMAG superconducting magnet at DESY (left) and schematic drawing of magnet structure with superconducting coil depicted in red (right). [8]

First layer of tested sensors is occupied by DEPFET pixel detector with design very close to final Belle II PXD, except the sensor is thinner (50um) than final version for Belle II (75 um thickness). It is the first time, full scale large matrix is tested in the beam. Main parameters of the sensor are listed in Table 2.1. The sensor is fully equipped by SWITCHER, DCD and DHH chips (Fig. 2.3 left) and installed at electronic hybrid board placed into metal support frame. As mentioned before, out of two intended PXD sensors, only one was finally used in the beam test. This sensor is installed into position of layer two and layer one is left empty. A copper cooling block with circulating CO2, visible in Fig. 2.3 (right), is attached to the board with thermal contacts to insensitive sensor area with DHH and DCD chips.

Parameter	Value	Unit
Thickness	50	µm
Area	36×9.6	mm×mm
Matrix pixels [v×u]	480×192	
Pixel pitch [v×u]	75×50	µm × µm

Table 2.1 Parameters of PXD sensor under test.

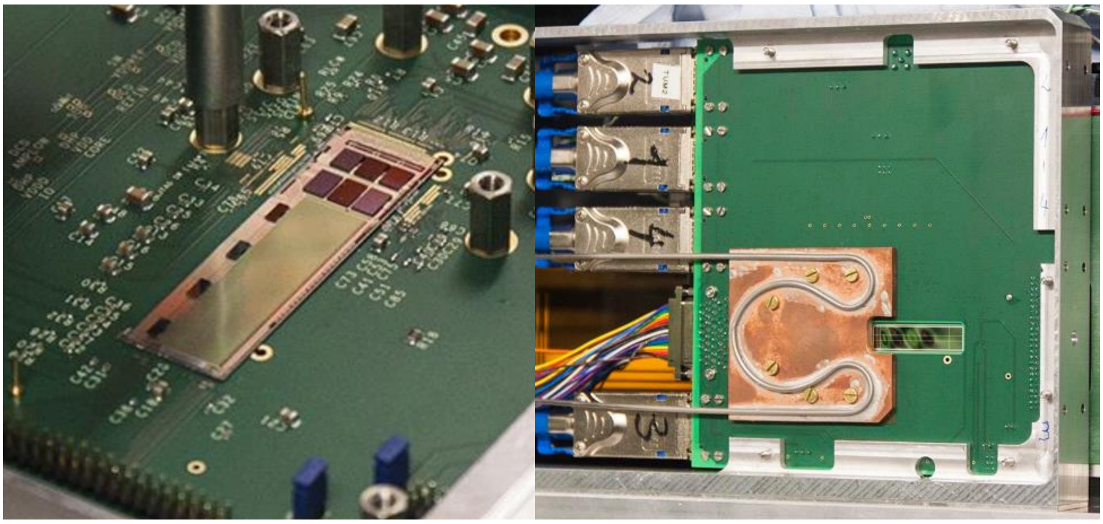


Figure 2.3 Left: PXD sensor installed on electronic board. Right: A view including support frame and cooling block with CO2 pipes, showing the sensor sensitive area from backside.

After PXD, four layers of double sided strip, Belle II SVD design sensors follow with parameters listed in Table 2.2. First sensor is smaller and installed on conventional hybrid board. Only strips of its central part on longer (v) side were connected and thus the sensor does not record data on left and right edge. Remaining three layers are from wide sensors installed on Origami hybrids. All modules are cooled with CO2.

All VXD sensors are enclosed in a dry dark box from polyethylene with N_2 flow. Additional six EUDET telescope sensors are installed before and after the VXD sensors. The telescopes are made of MAPS matrices with 18.4um pixel pitch used regularly for beam tests and offer very high tracking resolution and detection efficiency. They are divided in half to forward and backward telescope arm. All sensors are of the same type and their main parameters are listed in Table 2.3.

Sensor	Type	Side	Pitch [µm]	Strips	Area [mm × mm]	Thickness [µm]
L3	Small rectangular	p (u)	50um	768	122.88×38.4	320
		n (v)	160um	768*		
L4, L5, L6	Large rectangular	p (u)	75um	768	122.88×57.6	
		n (v)	240um	512		

* Only central 512 strips connected

Table 2.2 Parameters of SVD sensors layer 3 to 6, installed in the beam test.

Parameter	Value	Unit
Thickness	50	µm
Area	21.6×10.8	mm×mm
Matrix pixels [v×u]	1152×576	
Pixel pitch [v×u]	18.4×18.4	µm × µm

Table 2.3 Parameters of telescope pixel sensors

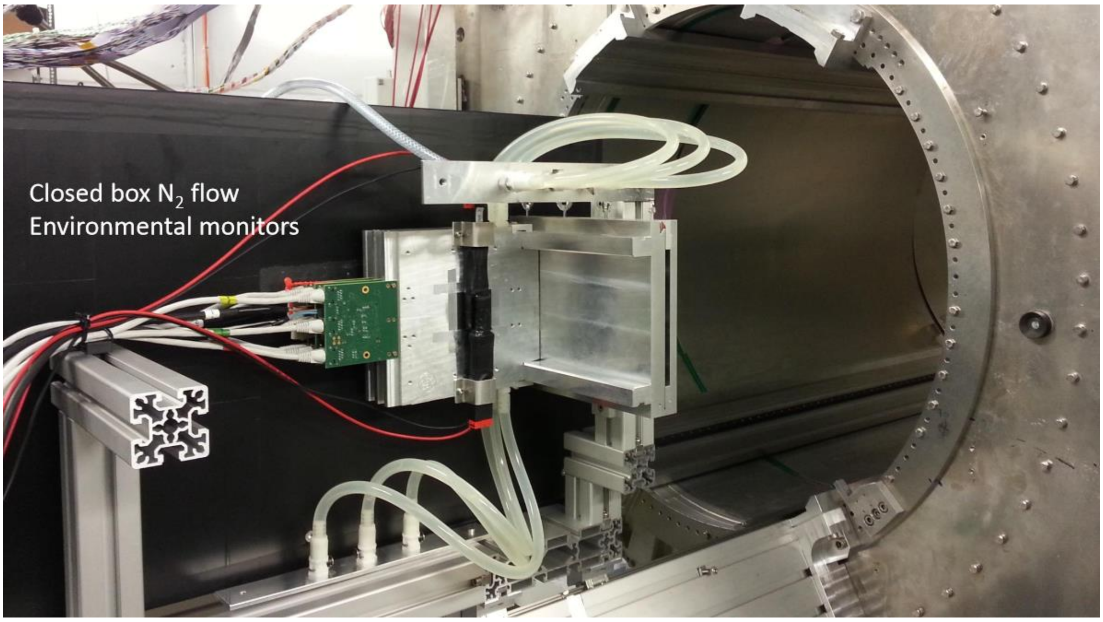
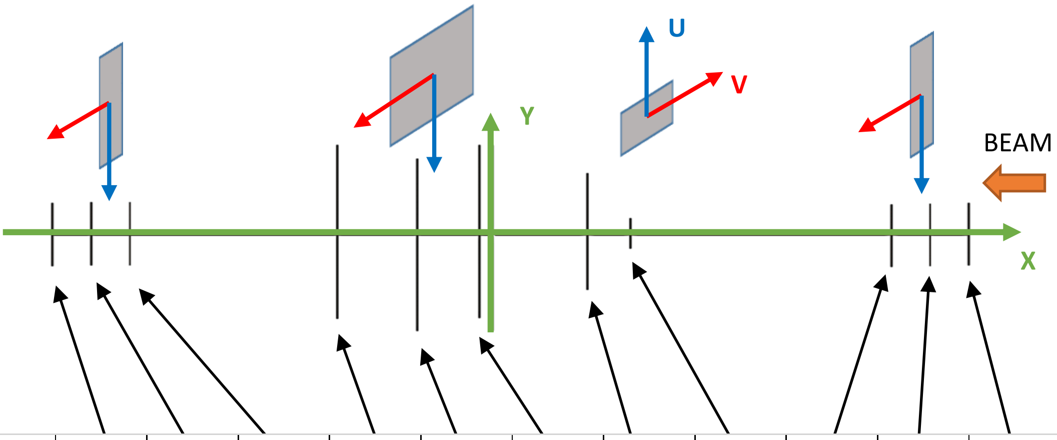


Figure 2.4: The complete setup with VXD inside dry box (black) and six EUDET telescopes (only three in forward arm visible) before roll-in into the PCMAG.

The complete setup was used in two different arrangements of the telescopes. Initial runs were conducted with larger distances between telescopes. After PXD installation, the telescopes were moved as close as possible to the VXD dry box. Only this geometry is investigated. Nominal positions of all sensors in the geometry are shown in Fig. 2.5. Center off sensors lie on x-axis, with exception of SVD L5, which is moved by 4.8mm down. A simplified drawing, showing positions of the sensors and their numbering scheme, is shown in Fig. 2.5. The distances between VXD sensors correspond to corresponding radius of the layers in Belle II VXD.



ID	T6	T5	T4	L6	L5	L4	L3	L2	T3	T2	T1
Type	Backward Telescopes			SVD				PXD	Forward Telescopes		
X [cm]	-17,05	-15,55	-14,05	-6,02	-2,92	-0,52	3,68	5,35	15,45	16,95	18,45
VXD ID	7.3.6	7.3.5	7.3.4	6.1.6	5.1.5	4.1.4	3.1.3	2.1.2	7.2.3	7.2.2	7.2.1

Figure 2.5.: Schematic drawing of the DESY beam test setup showing the detector planes in the global coordinate system, their position and orientation of their local axes, u (blue) and v (red). The attached table summarizes sensor identification and numerical values for their positions. All sensors have center at the x-axis, except SVD L5, being shifted down by 4.8 mm.

3 Belle II Software Framework

HEP experiments typically produce wealth amount of experimental data, which has to be processed by advanced and often CPU and memory intensive techniques. During Belle II operation, the extraordinary luminosity will pose a significant challenge in raw data collection and processing, event selection and reconstruction, data storage and reprocessing, and physics analysis. Prior to start of the experiment, a full Monte Carlo simulation of the detector, to investigate its best design to reach desired physics performance, is an absolute must. It also allows to test software tools long enough before real data arrival. At the very start of experiment preparation, often existing tools are adapted for such simulations. However once the detector starts to record collisions, a new round of Monte Carlo simulation begins, as important second input of physics analysis of real data. The easiest way to keep consistency between recorded and simulated data is to use the same tools and chains of processing. Also data handling and distribution, storage and access to operation conditions would benefit from using common unified tools.

For Belle II, it was decided to develop a completely new software framework for such purposes. It is called **basf2** (Belle 2 analysis and simulation framework). It benefits from experience from Belle as well as from other contemporary HEP software, developed intensively mainly to support large LHC experiments. The framework should cover all topics of Belle II experiment computing – related tasks. Starting with detector geometry and MC simulation, detector response simulation, raw data processing, real time applications, online and offline reconstruction, calibration, data quality and environment monitoring or physics analysis. Because of large computing and storage requirements, it is not possible, mainly for financial and safety reasons, to handle all computing tasks at single site, like KEK. Therefore the framework utilizes advanced method of parallelized and distributed computing to spread hardware requirements and backup of the data across collaborating institutes around the world.

Development of the framework is supported by wealth of tools such as automated centralized build system on all supported platforms with web interface for automatic reporting on code issues, issue tracking system or automated generation of code documentation. Automated tests and high level validation watch changes in code behavior and possible problems can be detected soon.

3.1 Framework Overview

The basf2 framework is written in C++ programming language and utilizes its modern C++11 specification. For user control over the framework, Python language is used in so called steering scripts. Configuration data is stored as XML and/or in conventional database. Third – party software, called externals, brings many standardized tools into the framework, like boost and CLHEP libraries, ROOT framework [9], Geant4 toolkit [10] or SCons build system. The framework is organized into packages, which encapsulate specific sort of task (geometry, simulation…) or are specific to a given subdetector (PXD, CDC…). [11]

The key concept of basf2 is modularity. Most of framework features are contained in reusable independent **modules**. These modules are dynamically loaded on request and have access to common framework tools. Parameters of the modules are used in the steering script to configure their behavior. Modules are executed in order specified in a so called **path** (Fig. 3.1 (a)). A single path is a linear sequence of modules, but return values of the modules can be used to switch to a different path and perform conditional processing chain in certain situations, see Fig. 3.1 (b).

The modules exchange data using a common storage in memory, called **DataStore**, capable to store or retrieve any C++ object for which a ROOT dictionary is generated. These objects can be in addition arbitrary connected by **relations** in the DataStore. Data coming from DataStore to disc (or vice versa) are handled by specialized modules. Output files contain serialized objects and are compatible with any ROOT based tool which allows for easy inspection and data manipulation using standardized methods and use of advanced I/O features.

Modules are allowed to interact between each other only using the DataStore. This keeps the structure of the framework transparent and eliminates bad design decisions through feature development. Some features however should be shared among modules. Besides core features, individual developers can produce code reusable in other modules, too. Such code is organized into **libraries**, compiled within the framework. The modules should be usually lightweight and often a module makes a user interface to some library.

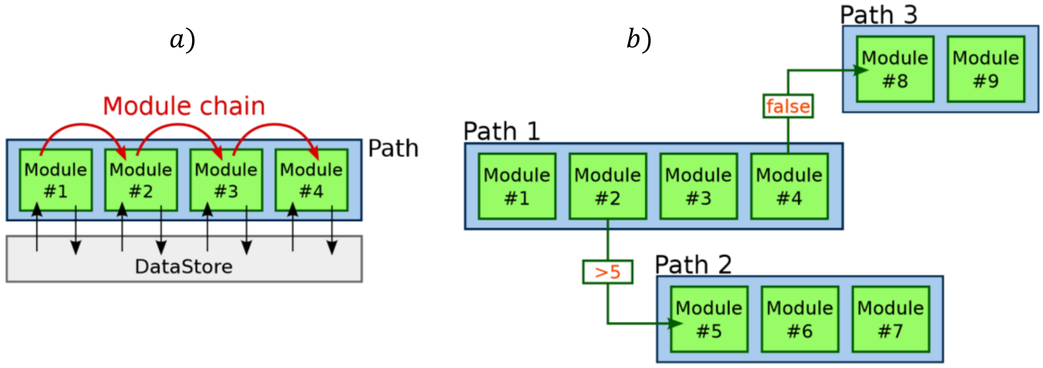


Figure 3.1 Modules are executed in a chain specified in the path and access a common DataStore (a). More paths can be created and based on module return value, a conditional processing chain can be executed (b). [11]

One of core features of the framework is contained in the geometry package. At any stage, a single geometry is defined and used during framework execution. This is essential, as many different tool access the geometry information at different stages from simulation to reconstruction. The geometry is represented as hierarchical structure of parameters stored in XML files. Other operational parameters are can be stored this way too. The framework unifies access to parameter storage using **GearBox** library and corresponding module. Parameters are accessible in a tree structure via XPath expressions in a user friendly interface, which allows e.g. automatic unit conversion. Parameters stored in XML are loaded by the GearBox module. The **Geometry** module then uses Geometry handling library which calls classes specified in the XML. These classes, called **GeoCreators**, handle corresponding subset of the XML tree and construct C++ objects which represent the hierarchy of Geant4 volumes, see Fig. 3.2. It also registers selected volumes as sensitive and makes a fallback from simulation to corresponding class which handles energy deposit from Geant4.

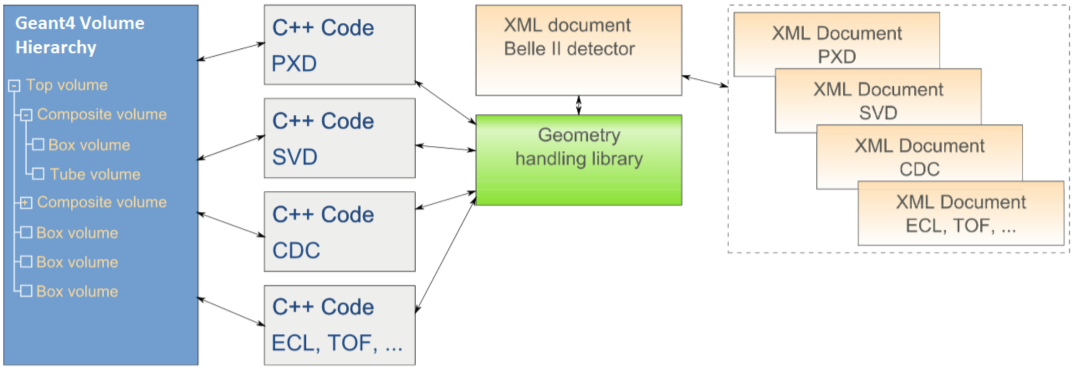


Figure 3.2.: Geometry construction from XML. The structure of XML documents processed by GearBox is accessed by the Geometry handling library which calls corresponding GeoCreator classes. These then create the hierarchy of volumes representing geometry of the detector. [11]

The simulation package encapsulates full MC simulation within FullSim module. Prior to simulation, source of particles has to be chosen. This can be either a simple ParticleGun module or advanced physics generators. Generated particle is propagated using Geant4 toolkit connected to the common geometry. Particles propagating through sensitive volumes make subsequent calls to handling classes defined during geometry construction. These classes then store information from each step of the propagation and its corresponding energy deposit into DataStore.

The simulation is typically followed by a digitization for each subdetector, where response of the detector to energy deposited by the particle is simulated. For VXD, the response in individual strips or pixels is clusterized in another modules. Modules related to VXD simulation are discussed in Chapter 6. Once response of the detector is known, track finding modules look for patterns of particle trajectories, fitted in tracking and extrapolation modules. The Display module allows to visualize complete event as recorded by the detector. Plenty of advanced modules can be found in analysis package. These were just examples, the framework currently contains more than 120 modules for many different tasks.

As mentioned, the framework is controlled by Python steering files. These files are in fact very powerful tool, because basf2 exposes a Python interface for its classes. It is possible to access the DataStore and use advanced Python features, e.g. for drawing. One can even write a basf2 module within the script and execute it in the path along C++ modules. This makes the framework extensible even by non $-$ C++ developers, although there exist some limitations of what can be done from the steering scripts.

3.2 GENFIT 2

Large HEP detectors are typically composed of different subdetectors which produce measurements of various quantities and dimensionality. A track reconstruction tool must be aware of material distribution inside the detector with detailed description of the magnetic field. Tools for particle propagation and extrapolation are necessary for fitting algorithms to work. GENFIT is a toolkit [12] for generic track reconstruction which coordinates all these different tasks in an experiment – independent way. For Belle II, a major upgrade of the toolkit, which is now called GENFIT 2, has been developed.

GENFIT 2 has modular design and makes extensive use of object oriented C++. Its main components can be divided into three groups:

- *Hit reconstruction* deals with different dimensionality of measurements and for non – planar measurements constructs virtual planes, in which the measurement is expressed.
- *Track representation* is a set of tools for particle propagation and extrapolation, connected through interfaces to detector geometry and magnetic field.
- *Track fitting algorithms* take reconstructed hits and using the track representation propagate between hits and fit the track. Available algorithms. Kalman Filter and Deterministic Annealing Filter (DAF) are in regular use in GENFIT 2. Extension to fitting with General Broken Lines is part of work done for thesis and is discussed in Chapter 6.

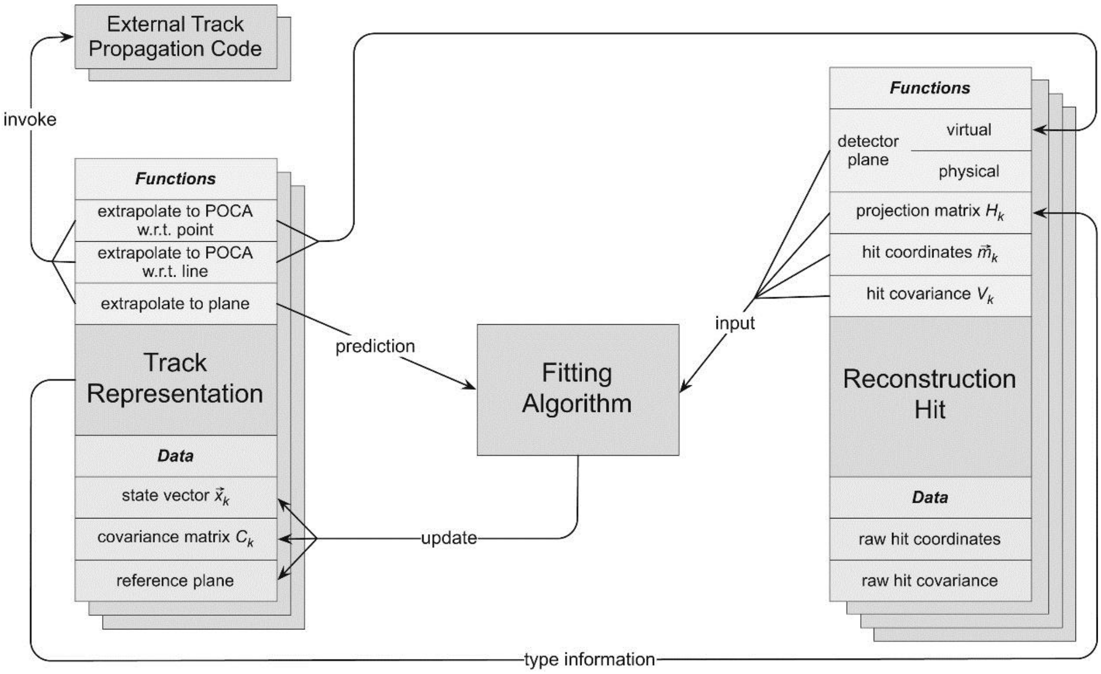


Figure 3.3: Interplay of hit reconstruction using track representation to construct virtual planes. The Fitting algorithm then propagates among planes using the representation, which state is updated by results of the fit.

The interplay of GENFIT components is illustrated in Fig. 3.3. Initially measurements are stored as raw coordinates and covariance. For planar hits, the sensor position and orientation is enough to reconstruct its spatial position. But for, e.g. a wire hit in CDC, the extrapolation tools are used to construct virtual planes to which track parameters from track representation are projected. The fitting algorithm then uses extrapolation to propagate between (virtual) planes and updates the state and covariance of track parameters with results of the fit.

One of key features of GENFIT is a complete separation of the fitting algorithm from specifics of various kinds of measurements. After each propagation, the resulting state is expressed at some plane. GENFIT therefore uses local coordinates for the state vector representing the trajectory at given plane, in magnetic field $\overrightarrow{x_k} = (q/p, u', v', u, v)^T$. For planar measurement, it is the plane of the sensor, Fig. 3.4 (a). For wire measurement in CDC, for example, some preceding state of the track is taken and a virtual plane is constructed, such that it is perpendicular to the predicted track position and wire direction; the track parameters are then expressed at the virtual plane, Fig. 3.4 (b). In this way, raw measurements are expressed in the same coordinate system (of the virtual plane) as prediction the track representation. The dimensionality of the measurement is defined by a so called H-matrix and the residual can be expressed from reconstructed measurement (adopting notation from Fig. 2.3):

$$\overrightarrow{residual} = \overrightarrow{m_k} - H_k \overrightarrow{x_k}, \tag{2.1}$$

where the dimension is 1 for wire and strip measurement, 2 for a pixel measurement etc.

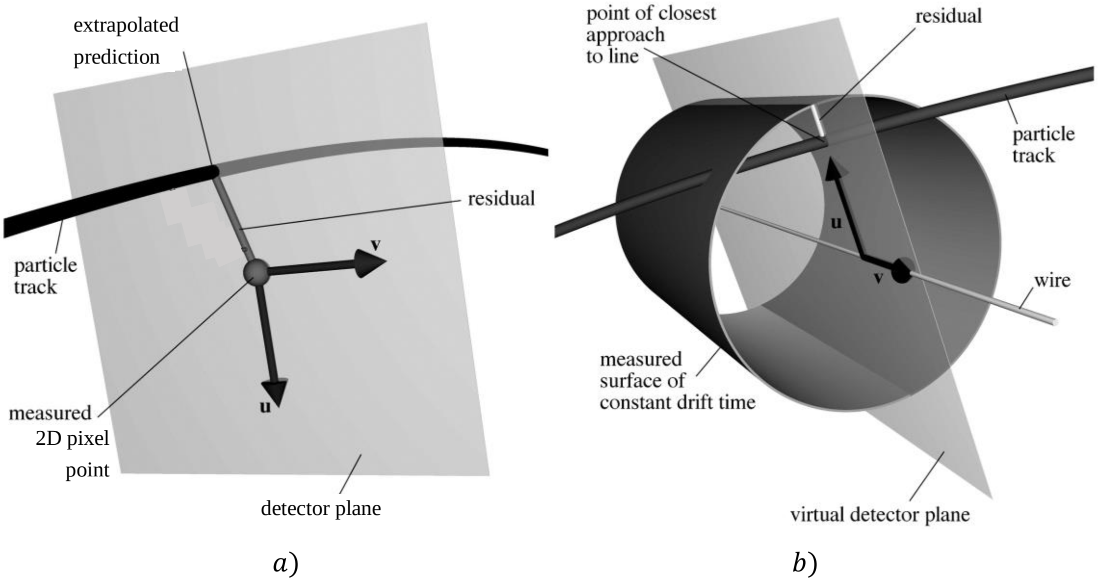


Figure 3.4 A track propagated to pixel sensor and residual expressed in local plane system (a). For wire measurement a virtual plane is constructed to express track and measurement in the same coordinate system to calculate residual (b). [12]

For the default representation, Runge – Kutta approximation methods are used to propagate state with dimension 7. After each extrapolation, the state is projected to the final plane.

4 Track Based Alignment

This chapter concentrates on alignment for the Belle II vertex detector and VXD beam test. Introduction and definition of the alignment task is presented. The Millepede algorithm, selected as the default alignment procedure for VXD is explained and the program Millepede II is described. The alignment parameterization and a brief introduction to weak modes is discussed. [13]

4.1 Introduction

Parameters of reconstructed tracks are major input of physics analysis. They are determined from local measurements on individual sensors transformed into a common global coordinate system. This transformation requires information on detector geometry. If the actual geometry differs from the supposed one (is misaligned in other words) in a sufficiently precise detector, this would result a measurable change in predicted parameters of the tracks. This change can have negative impact of physics performance of the detector. As an example, physics of CP violating effects relies on precise vertex reconstruction of B mesons. Misalignment can significantly influence the vertex resolution and produce a bias in vertex reconstruction. Therefore precise alignment is of very high importance. [14]

An initial alignment can be done by precise construction of the detector and additional e.g., laser measurements. Precision of such an initial alignment is however usually much worse than intrinsic resolutions of individual detector elements. Sufficient accuracy can only be reached by a track based alignment. In this approach, reconstructed tracks of charged particles and knowledge of particle propagation are input to the alignment procedure.

It is important to collect diversity of tracks from different sources and under different conditions, because parameters of single tracks are often correlated and thus can introduce a bias in the determined alignment. Such sources include:

- cosmic ray tracks recorded with and without magnetic field,
- tracks from interaction point, or
- particles coming from beam interactions with the beam pipe.

Another advanced methods, e.g. including mass and/or vertex constrained tracks can also have positive impact on the accuracy of alignment and removal of potential bias. In this thesis, only alignment with single tracks (cosmic ray, interaction point, and beam) is developed. This is completely sufficient for the beam test. For the alignment of complete VXD, the most significant in alignment improvement is the cosmic ray sample, which has quite different topology from tracks coming from IP.

The aim of the alignment procedure is the determination of the alignment parameters $\boldsymbol{a}$. The knowledge of particle propagation is incorporated in a track model used to fit the measurements and determine predicted track parameters $\boldsymbol{q}$. The resulting prediction depends on fitted track parameters and on the supposed geometry represented by the alignment parameters.

Individual uncorrelated measurements m_{ij} on a track j can be compared to the prediction p_{ij} of the track model. Their difference divided by the uncertainty of the measurement σ_{ij},

$$z_{ij}(\boldsymbol{a}, \boldsymbol{q}_j) = \frac{m_{ij} - p_{ij}(\boldsymbol{a}, \boldsymbol{q}_j)}{\sigma_{ij}} = \frac{r_{ij}}{\sigma_{ij}}, \tag{4.1}$$

is called normalized residual (pull). Residual of a single measurement is denoted as r_{ij}. The χ^2 function is defined as a sum of squares of pulls over the whole sample of tracks and their measurements:

$$\chi^2(\boldsymbol{a}, \boldsymbol{q}_1, \boldsymbol{q}_2, \dots) = \sum_{j}^{\text{tracks}} \sum_{i}^{\text{meas}} z_{ij}^2(\boldsymbol{a}, \boldsymbol{q}_j). \tag{4.2}$$

In the χ^2 minimization, the alignment (and track) parameters are varied to reach an optimal solution, which corresponds to an aligned geometry.

In order to minimize (4.2), one has to operate in a space of parameters with very large dimension. The global minimum is found by taking derivative w.r.t. each single alignment and track parameter

$$\frac{d\chi^2}{d(\boldsymbol{a}, \boldsymbol{q}_1, \boldsymbol{q}_2, \dots)} = 0. \tag{4.3}$$

This expression is usually linearized around some initial values of alignment and track parameters and written as a matrix equation. Such matrix equation however is of dimension $n = n_{\text{par}} + n_{\text{loc}} \times n_{\text{track}}$, where n_{par} is number of alignment parameters to be determined, n_{loc} is number of local parameters per track and n_{track} is number of tracks included in the minimization procedure. A brute − force approach of direct inversion is not feasible for typical applications in complex detectors where the number of alignment parameters exceeds thousands and millions of tracks have to be used. [13]

The exact approach how the minimization is performed makes the main difference between alignment algorithms. Algorithms regularly used in HEP experiments include:

- HIP algorithm. The χ^2 function is minimized with fixed track parameters for each sensor individually. This procedure is iterated until a convergence is reached. In this approach, correlations are ignored and uncertainties of estimated alignment parameters are not accessible. Used, e.g., in ATLAS experiment.

- Kalman Filter algorithm. The alignment parameters are updated each time a new track is added. This algorithm was implemented in the CMS experiment.

- Millepede algorithm. Minimizes χ^2 with respect to alignment and track parameters simultaneously. Details are discussed in the following sections.

4.2 Millepede Algorithm

The Millepede algorithm is a method for solution of linearized least squares problems (4.3) with very large number of parameters if only the parameters $\boldsymbol{a}$ have to be determined. The algorithm however does simultaneous minimization also w.r.t. to track paramaters, conserving all correlations and without any approximation. [15]

In the Millepede formalism, two kinds of free parameters in the optimization are recognized.

- *Global parameters*, which correspond to alignment parameters. These parameters contribute to all measurements, but usually only small subset of them is connected by local parameters.
- *Local parameters* correspond to parameters of individual tracks. These parameters influence only the local fit (of the track model). Millepede cannot determine local parameters, but their influence is fully contained in the determined global parameters.

Millepede algorithm allows the determination of global parameters in a single step in simultaneous fit with local parameters. This method does not introduce any approximation, except a linearization. For small parameter corrections, expression (4.2) is linearized to first order around some initial alignment and track parameters.

$$\begin{aligned}\chi^2(\Delta\boldsymbol{a},\Delta\boldsymbol{q}_1,\dots) = \sum_j^{\text{tracks}} \sum_i^{\text{meas}} \frac{1}{\sigma_{ij}^2}\Bigg(& m_{ij} - p_{ij}(\boldsymbol{a}_0,\boldsymbol{q}_{0j}) + \frac{\partial r_{ij}}{\partial \boldsymbol{a}}\Delta\boldsymbol{a} \\ & + \frac{\partial r_{ij}}{\partial \boldsymbol{q}_j}\Delta\boldsymbol{q}_j\Bigg)^2 .\end{aligned} \tag{4.4}$$

Local parameters of individual tracks (local fit objects) do not connect different tracks because they are independent (or should be, which could be violated by using correlated data as input). In complex detectors, a single track often connects only small subset of detector parts making a link between local and global parameters.

Taking the derivatives, as in (4.3), the equation can be reorganized into a block matrix equation with bordered block diagonal form, schematically as

$$\begin{bmatrix} \boldsymbol{\Sigma}\, \boldsymbol{C_{1j}} & \dots & \boldsymbol{G_j} & \dots \\ \vdots & \ddots & 0 & 0 \\ \boldsymbol{G}_j^T & 0 & \boldsymbol{\Gamma}_j & 0 \\ \vdots & 0 & 0 & \ddots \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{a} \\ \vdots \\ \Delta\boldsymbol{q}_j \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}\, \boldsymbol{g_{1j}} \\ \vdots \\ \boldsymbol{\beta_j} \\ \vdots \end{bmatrix} \tag{4.5}$$

The sum sign in matrix on l.h.s. of (4.5) indicates a sum over all tracks (local fits). The matrix involves three kinds of contributions. First is a sum of terms, which depend only on derivatives w.r.t. global parameters,

$$(\boldsymbol{C}_{\mathbf{1}j})_{kl} = \sum_{i}^{meas} \frac{1}{\sigma_{ij}^2}\left(\frac{\partial r_{ij}}{\partial a_k}\right)\left(\frac{\partial r_{ij}}{\partial a_l}\right). \tag{4.6}$$

Elements at the borders combine local and global parameters,

$$(\boldsymbol{G}_j)_{kl} = \sum_{i}^{meas} \frac{1}{\sigma_{ij}^2}\left(\frac{\partial r_{ij}}{\partial a_k}\right)\left(\frac{\partial r_{ij}}{\partial q_{j,l}}\right). \tag{4.7}$$

And finally, gamma matrices in the block diagonal part combine only derivatives w.r.t. track parameters:

$$(\boldsymbol{\Gamma}_j)_{kl} = \sum_{i}^{meas} \frac{1}{\sigma_{ij}^2}\left(\frac{\partial r_{ij}}{\partial q_{j,k}}\right)\left(\frac{\partial r_{ij}}{\partial q_{j,l}}\right). \tag{4.8}$$

The vector of r.h.s. of (4.5) contains a sum of residuals related to the global or local parameters multiplied by the corresponding global or local derivatives:

$$(\boldsymbol{g}_{1j})_k = \sum_{i}^{meas} \left(\frac{\partial r_{ij}}{\partial a_k}\right)\frac{r_{ij}}{\sigma_{ij}^2}, \qquad (\boldsymbol{\beta}_j)_k = \sum_{i}^{meas} \left(\frac{\partial r_{ij}}{\partial q_{j,k}}\right)\frac{r_{ij}}{\sigma_{ij}^2}. \tag{4.9}$$

The special structure of this matrix allows for prior solution of individual local fits and reduction of size of the matrix (4.5) to size of matrix (4.6), proportional to the number of alignment parameters. The idea of matrix reduction can by illustrated on a block matrix equation

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{4.10}$$

which, expressed as 2 single equations, can be solved by getting y from the 2nd equation, if D is invertible:

$$y = D^{-1}(b - Cx). \tag{4.11}$$

Substitution back into the 1st equation results in a reduced problem of determination of x,

$$(A - BD^{-1}C).x = a - BD^{-1}b, \tag{4.12}$$

where the solution can be obtained by inverting matrix on l.h.s., called Shur complement. [16]

Successive application of the matrix reduction on 3.5) will result in removal of the diagonal and border part occupied by Γ and G matrices and updated upper left part of the matrix and vector on r.h.s. of Eq. (4.5). The algorithm accumulates results of local fits $\boldsymbol{\Gamma}_j^{-1}\boldsymbol{\beta}_j$ and constructs matrix and vector

$$\boldsymbol{C} = \Sigma\, \boldsymbol{C}_{1j} - \Sigma\, \boldsymbol{G}_j \boldsymbol{\Gamma}_j^{-1} \boldsymbol{G}_j^T \qquad \boldsymbol{g} = \Sigma\, \boldsymbol{g}_{1j} - \Sigma\, \boldsymbol{G}_j \boldsymbol{\Gamma}_j^{-1} \boldsymbol{\beta}_j, \tag{4.13}$$

where sums run over all tracks j. Because no approximation is introduced, for purely linear problems, the solution can be reached in a single step. The corrections to the initial global parameters are finally determined by inversion of the $\boldsymbol{C}$ matrix:

$$\Delta \boldsymbol{a} = \boldsymbol{C}^{-1}\boldsymbol{g}. \tag{4.14}$$

For non-linear problems or large correction to initial values, iterations may be required with updated linearization point in (4.4).

4.3 Millepede II

Millepede II is a second version of implementation of the Millepede algorithm, developed by V. Blobel. It is now maintained by Helmholtz Alliance "Physics at the Terascale" and available at [17]. The program was successfully used in the H1 experiment [18] and is in regular use, for example, in alignment of the CMS tracker [19]. Second version allows one to include hundreds of thousands of parameters in the alignment procedure. In addition to the Millepede algorithm itself, the program offers wealth of additional tools and options required for successful solution of the alignment task.

The program is separated into two parts. The Mille subroutine is used for accumulation of data from measurements into binary files. Binary Mille files are then used by the Pede application, which performs the actual matrix reduction and final inversion (4.14). Pede is a Fortran application controlled by commands specified in a text file with a list of input Mille binary files.

Millepede II requires to provide for each single measurement the following data, which can be identified in expression (4.4) [16]:

- Residual $m_{ij} - p_{ij}(\boldsymbol{a}_0, \boldsymbol{d}_{0j})$ of the measurement and initial prediction of the track model.
- Standard deviation of the measurement σ_{ij}
- Local derivatives of the residuals w.r.t. local parameters of the track $\frac{\partial r_{ij}}{\partial \boldsymbol{q}_j}$ as array.
- Global derivatives of the residuals w.r.t. global parameters $\frac{\partial r_{ij}}{\partial \boldsymbol{a}}$, where each of the derivative in the array is identified by an unique integer label of the corresponding parameter passed as a vector

For the final inversion, several solution methods are available:

- *Inversion* is the standard method for small number of global parameters (~ 1000). Memory requirements scale with the second and CPU time with the third power of number of parameters. All parameter errors and correlations are available.
- *Diagonalization* is about 10 times slower than inversion. All correlations are available together with eigenvalues and eigenvectors of the global matrix (4.14). This allows for study of weak modes, which correspond to eigenvectors with vanishing eigenvalues.
- *Minimization of residuals (MINRES)* method is an iterative method feasible for large sparse matrices. Sparse matrices are result of limited number of global

parameters connected by local parameters. For example, tracks from interaction point do not connect opposite half-shelves of the tracker. The solution can be improved by preconditioning by following methods:

- *Cholesky decomposition* method with variable band size. This method does not take into account all correlations and is therefore only approximate but suitable for preconditioning.
- *HIP algorithm* method makes use of the HIP approach, where correlation between sensors are ignored and only a block diagonal matrix has to be inverted for preconditioning of MINRES.

An optimal and reliable solution to (4.3) can only be found if measurement errors are Gaussian distributed. Outliers can distort the solution, because they have large impact on χ^2 and should be removed from the data. For this purpose, Millepede utilizes internal iterations for down-weighting of measurements and rejection of bad data. More strict cuts are applied on χ^2 divided by number of degrees of freedom (χ^2/NDF) values of tracks in successive iterations. Impact of outliers on the overall χ^2 can be also reduced by weighting of normalized residuals. In this approach, impact of large outliers gets reduced. It is possible to specify the number of down-weighting iterations and limits for track χ^2/NDF values for each iteration. In the first two iterations, the Huber function is used, for which large pulls have constant influence on χ^2. In following iterations, the Cauchy function makes larger pulls to have even decreasing impact.

The solution can be required to satisfy specified linear constraints. These constraints can be used to remove global undefined degrees of freedom, three rotations and three translations, as well as to implement structural constraints, which correlate alignment parameters of structures and their sub-components. In this way, the mechanical structure can be reflected in the solution. The constraints are implemented as Lagrange multipliers. A set of linear constraints for global parameter corrections is expressed as a matrix equation

$$\boldsymbol{A}\Delta\boldsymbol{a} - \boldsymbol{m} = 0. \tag{4.15}$$

New set of parameters $\boldsymbol{\lambda}$ corresponding to Lagrange multipliers is introduced for each constraint and extended equation (3.12) is solved for global parameters which satisfy the constraints:

$$\begin{pmatrix} \boldsymbol{C} & \boldsymbol{A}^T \\ \boldsymbol{A} & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\boldsymbol{a} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{m} \end{pmatrix}. \tag{4.16}$$

Individual parameters can be controlled in text files. It is possible to fix certain parameters, provide starting values and/or specify an error estimation for the parameter. The output of Millepede II running is also stored in text files. [13]

4.4 Alignment Parameterization

[karimaki] [stoye-thesis] [claus-tpc-alignment] [claus-private]

In this section, the alignment parameters are defined, the effect of changing alignment parameters is explained, and the global derivatives are derived.

The sensors of VXD are treated as planar rigid bodies, with local coordinate system (u, v, w), where w is perpendicular to the sensor and u and v have direction of sensitive coordinates of the sensor, with origin at its center, see Fig. 4.1. A measurement $\boldsymbol{m}_l$ in the local sensor system can be transformed into the global coordinates $\boldsymbol{m}_g$,

$$\boldsymbol{m}_g = \boldsymbol{R}^T \boldsymbol{m}_l + \boldsymbol{r}_0, \tag{4.17}$$

where $\boldsymbol{R}$ is a matrix of rotation between the local and global sensor systems and $\boldsymbol{r}_0$ is position of the sensor center in the global system.

In the alignment, corrections to the sensor rotation $\Delta\boldsymbol{R}$ and translations in its local system $\Delta\boldsymbol{r}$ are determined and the corrected formula reads

$$\boldsymbol{m}_g{}' = \boldsymbol{R}^T \Delta\boldsymbol{R}(\boldsymbol{m}_l + \Delta\boldsymbol{r}) + \boldsymbol{r}_0. \tag{4.18}$$

The rotation is parameterized by Euler angles and is performed around the sensor center such that no shift of the sensor origin is induced by the rotation. The Euler angles are shown in Fig. 4.1.

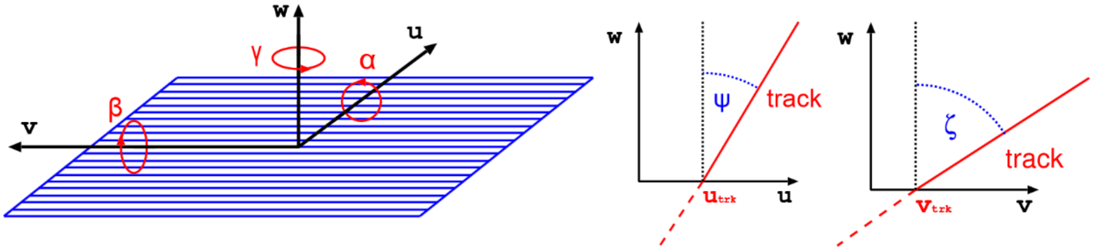


Figure 4.1: Illustration of the local system (u, v, w) and Euler rotation angles (α, β, γ) for planar sensor (left) and definition of the track slopes $u' = \tan\psi$ and $v' = \tan\zeta$ in local sensor system (right).

The six individual rigid body alignment parameters (3 translations + 3 rotations) for a sensor are

$$\boldsymbol{a} = (\Delta u, \Delta v, \Delta w, \Delta\alpha, \Delta\beta, \Delta\gamma)^T$$

and for small corrections, the corrected measurement in local system can be expressed, neglecting higher order terms, as

$$\boldsymbol{m}_l' = \begin{pmatrix} \mathbf{1} & \Delta\gamma & \Delta\beta \\ -\Delta\gamma & \mathbf{1} & \Delta\alpha \\ -\Delta\beta & -\Delta\alpha & \mathbf{1} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \begin{pmatrix} \Delta u \\ \Delta v \\ \Delta w \end{pmatrix}, \tag{4.19}$$

where the vector $(u, v, w)^T$ should contain the best estimate of the measured position on the sensor, being usually a combination of initially measured position and the fit. As this estimation must lie in the sensor plane, the third component is identically zero. Derivatives of the measurement therefore read

$$\frac{\partial \boldsymbol{m}_l'}{\partial \boldsymbol{a}} = \begin{pmatrix} 1 & 0 & 0 & 0 & -w=0 & v \\ 0 & 1 & 0 & w=0 & 0 & -u \\ 0 & 0 & 1 & -v & u & 0 \end{pmatrix}. \tag{4.20}$$

If the plane changes, the prediction has to be propagated to a new intersection point. For this purpose, the prediction $\boldsymbol{p}$, expressed in original plane system, is linearized around the initial intersection point and parameterized by arc-length s,

$$\boldsymbol{p}(s) = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} + s\boldsymbol{t}, \tag{4.21}$$

where $\boldsymbol{t} = (u' = \frac{du}{dw}, v' = \frac{dv}{dw}, 1)^T$ is the direction of the linearized track in vicinity of the sensor expressed by track slopes, or equivalently $\boldsymbol{t} = (\tan\psi, \tan\zeta, 1)^T$, using local track angles ψ and ζ, sketched in Fig. 4.1. The intersection of the linearized track with corrected measurement plane in coordinates of the original plane is determined by

$$0 = (\boldsymbol{m}_l' - \boldsymbol{p}(s)).\boldsymbol{n}, \tag{4.22}$$

where $\boldsymbol{n} = (0,0,1)^T$ is a normal to the original plane. Solving Eq. (4.20) one gets $s = [\boldsymbol{m}_l']_3$ being the 3rd coordinate of the corrected measurement in the system of the uncorrected one. The residuals as a result of small alignment deviation read

$$\boldsymbol{r} = \boldsymbol{m}_l' - \boldsymbol{p}(s) = \boldsymbol{m}_l' - \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} - [\boldsymbol{m}_l']_3 \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}, \tag{4.23}$$

and derivatives of the residuals w.r.t. to corrected measurement are:

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}_l{}'} = \begin{pmatrix} 1 & 0 & u' \\ 0 & 1 & v' \\ 0 & 0 & 0 \end{pmatrix}. \tag{4.24}$$

Finally, the desired derivatives of the residuals w.r.t. rigid body parameters of the sensor can be calculated using the chain rule:

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{a}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}_l{}'} \frac{\partial \boldsymbol{m}_l'}{\partial \boldsymbol{a}} = \begin{pmatrix} 1 & 0 & -u' & vu' & -uu' & v \\ 0 & 1 & -v' & vv' & -uv' & -u \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.25}$$

Empty last row reflects the fact that a planar sensor is not sensitive in the w coordinate. The first two rows are input of the alignment procedure and the derivatives are computed using the best available prediction of the track linearization at the plane.

A similar approach can be used to get derivatives w.r.t. corrections in the global coordinate system. Formula (3.20) for alignment parameters of the global system, track direction and normal to the sensor plane need to be expressed in the global coordinates,

$$(u, v, w = 0) \to (x, y, z).$$

The derivatives are then again obtained using the chain rule (3.25) with

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{m}_l{}'} = \delta_{ij} - \frac{t_i n_j}{\boldsymbol{t}.\boldsymbol{n}}. \tag{4.26}$$

If alignment parameters in the global system are added to the problem, they introduce six new degrees of freedom, which are undefined. It is possible to move all sensors in their local system by any distance and remove this shift by doing opposite movement of the global system without any change on χ^2. These degrees of freedom have to be removed by addition of six linear equality constraints, fixing the overall coherent movement of the sensors w.r.t. the global system.

Note that there are always at least six undefined degrees of freedom, which have to be fixed in the residual minimization. Residuals are insensitive to shifting or rotating all detectors coherently. These degrees of freedom are removed either by fixing some of the parameters, by adding absolute references to the data (e.g. vertex constraint) and/or by introducing constraints on the parameters. The global alignment parameters $\boldsymbol{a}_g$ can be translated into local sensor parameters via a transformation matrix:

$$\boldsymbol{a}_i = \boldsymbol{C}_i \boldsymbol{a}_g. \tag{4.27}$$

The equality constraint then requires that global alignment parameters are invariant under changes of alignment of all sensors:

$$0 = \sum_i \boldsymbol{C}_i^{-1} \boldsymbol{a}_i \tag{4.28}$$

Details on the transformation between systems and derivation of the constraints can be found in Ref. [13]. When constraining w.r.t. to the global system for a sensor placed at position $\boldsymbol{r}_0 = (x, y, z)$ with rotation matrix $\boldsymbol{R}^T$ (4.17), its contribution to the constraints is contained in a 6×6 matrix

$$\boldsymbol{C}_i^{-1} = \begin{pmatrix} \boldsymbol{R}^T & -\boldsymbol{T}\boldsymbol{R}^T \\ 0 & \boldsymbol{R}^T \end{pmatrix}, \tag{4.29}$$

where

$$\boldsymbol{T} = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}. \tag{4.30}$$

The disadvantage of using constraints is that the result is expressed in a different coordinate system (defined by the constraints) and for checking of the alignment procedure, an additional transformation has to be done for a direct comparison of computed and simulated misalignment. For real applications in HEP detectors, equality constraints are very important, however, being used at different levels of the hierarchy of mechanical structures to reflect the correlated nature of realistic misalignment.

Alignment procedure discussed in this thesis utilizes only corrections in local coordinate systems of the sensors, but the possibility of alignment in the global system (where all sensors share the same set of six additional rigid body parameters) has been implemented.

4.5 Weak Modes

The matrix $\boldsymbol{C}$ in Eq. (4.14) must be inverted to solve the system of equations for alignment parameters. However it may happen that this matrix is singular or near to singular. This singularity corresponds to weakly defined degrees of freedom, which left the overall χ^2 unaffected. If the matrix $\boldsymbol{C}$ is diagonalized to get its eigenvalues and eigenvectors, weakly defined degrees of freedom correspond to eigenvectors (linear combinations of alignment corrections) with vanishing eigenvalues. If such a linear combination is present in the data as misalignment, the alignment procedure cannot determine such deformation. On the other hand, such a combination can be artificially introduced into alignment parameters by the alignment procedure if these degrees of freedom are not properly controlled. Such combinations of parameters are called weak modes and pose a significant challenge to all alignment procedures involving residual minimization.

Six trivial weak modes, which are result of missing absolute reference and correspond to global rotation or translation of the setup, were discussed before. The challenge is in those weak modes that correspond to coherent deformations of detector internal structure. Because these weak modes affect physically relevant quantities by producing a bias into track parameters not visible in simple residual minimization, it is important to systematically investigate and control such weak deformations. As a demonstration, illustration of so called curl deformation, a typical weak mode for detectors with cylindrical symmetry, such as Belle II VXD is shown if Fig. 4.2. This weak mode can occur in samples of tracks from collisions in a magnetic field. The detector is made of concentric layers and if each layers is misaligned by an angle $\Delta\varphi \sim R$ proportional to its radius, this can be compensated by lowering transverse momentum of negatively charged tracks (red) and increasing transverse momentum of positively charged tracks (blue). The χ^2 is unchanged but a bias is introduced into parameters of the tracks.

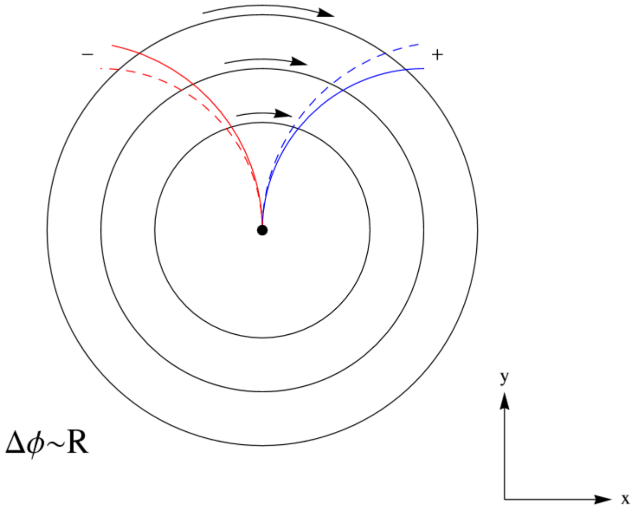


Figure 4.2 Curl deformation of a detector with cylindrical symmetry. Arrows indicate layer misalignment proportional to its radius. Dashed lines indicate true trajectories of charged particles and solid lines show how they are reconstructed with the deformation. Track with positive charge is shown in blue, negative in red. [20]

Several ways of controlling weak modes exist depending on their nature. The curl deformation, for example, would be visible in the asymmetry of oppositely charged track parameters. An external measurement can be added to constraint transverse momentum, e.g., using prediction from CDC. Another way to reduce a curl mode is to include cosmic ray tracks with magnetic field in the alignment procedure. For such tracks going near the interaction point, the curl deformation is not a weak mode, because the bias in the transverse momentum changes sign when the track crosses the center of the detector. With a single curvature parameter fitted to the track, the curl mode cannot be compensated by changing this parameter of cosmic tracks. [20]

This example only illustrates how to get rid of weak modes. The set of weak modes depends on the geometry and segmentation of the detector and on used tracks. One should, in general, use as much information as possible in the alignment procedure. This means a combination of different track samples, from different source and different topologies recorded under different conditions (magnet on/off). Also introduction of any external reference or constraint can help in removal of weak modes. This includes vertex-constrained tracks and combined objects from tracks of decaying particles, where their known mass is added as an external measurement [19].

5 Track Fitting

This chapter discusses an advanced track fitting model, General Broken Lines (GBL), a default fitting algorithm for alignment purposes of Belle II VXD. Effects on trajectories of charged particles relevant for this thesis are discussed. Finally, an overview of the new approach used to describe multiple scattering within GBL parameterization is given with results of a toy simulation for illustration.

5.1 Introduction

A trajectory of a charged particle propagating in vacuum is a straight line described by 4 parameters or a helix with 5 parameters in presence of a constant magnetic field. In HEP detectors, the particle trajectory is affected by interaction with the detector material. A proper parameterization has to account for energy loss, mainly from ionization, bending in the magnetic field and multiple scattering. Energy loss affects momentum of the particle, while multiple scattering is randomly changing track slopes and position. No simple parameterization can account for such effects, especially if the magnetic field is not homogeneous. Various methods were developed to address this issue.

A standard approach to track fitting in HEP is the Kalman filter [21]. This is an iterative method that updates its prediction each time new measurement is added and does not rely on any specific parameterization. It is used in many fields, also outside HEP. Kalman filter estimates the state vector of a dynamic system in series of measurement whose uncertainty is known. An initial value and covariance is propagated between measurements updating the predicted state by measurements weighted by their errors. The propagation and update of states proceeds in several iterations until convergence is reached. Propagation of the covariance usually includes taking into account traversed material. Uncertainties on track parameters are added as noise in the Kalman formalism. The Kalman filter is a default fitting algorithm for tracking in Belle II.

Effects of multiple scattering can be explicitly fitted as scattering angles. This increases the number of parameters of the track to $n_{\text{par}} = 5(4) + 2n_{\text{scat}}$, where n_{scat} is the number of thin scatterers added to particle trajectory. At each thin scatterer two parameters are fitted that describe the scattering angles. The scattering angles for a thin scatterer have zero expectation values and a variance. Introduced new degrees of freedom are removed by specifying the variance at each thin scatterer. This information must be retrieved from the description of the material traversed by the particle. In this way, the track slopes are constrained along the trajectory according to the material distribution.

5.2 General Broken Lines

General Broken Lines (GBL) is an advanced track model that utilizes the approach of additional track parameters to properly describe multiple scattering. Resulting special structure of system of linear equation allows for fast solution with full covariance matrix available. It has been implemented by C. Kleinwort based on the original broken lines by V. Blobel and is maintained by Helmholtz Alliance "Physics at the Terascale". It is available in Python, Fortran and C++ at [22]. GBL allows to easily output results of track fit for use in Millepede II alignment. The GBL track model is used for example in CMS tracker alignment together with Millepede II. [19], [23]

GBL does a refit of a particle trajectory, in such sense, that only relative changes to an initial reference trajectory are computed. Effects of multiple scattering are added to the trajectory as additional points next to the measurements, called scatterers. GBL is usually seeded by a simple 5(4) parameter reference trajectory, which is a result of a fit to all its hits coming often from pattern recognition. This fit must be done outside GBL. The reference trajectory is used to describe propagation of track parameter variations In the magnetic field and the various transformations between different local frames of the hits and to calculate the initial residuals.

The trajectory is constructed from points with measurement and/or scatterer. At each point, a local system $(u,\ v,\ w)$ is defined and the trajectory is locally parameterized by slopes and offsets $(q/p, \boldsymbol{u}' = \left(\frac{du}{dw}, \frac{dv}{dw}\right), \boldsymbol{u} = (u, v))$ where the signed inverse momentum q/p parametrizes track curvature in presence of magnetic field. The points on the trajectory are connected by Jacobians for parameter propagation from the previous point [21]:

$$\boldsymbol{J}_{i-1\to i} = \frac{\partial(q/p,\ \boldsymbol{u}',\ \boldsymbol{u})_i}{\partial(q/p,\ \boldsymbol{u}',\ \boldsymbol{u})_{i-1}}. \tag{5.1}$$

Small change in the offset, slope or curvature propagates like

$$\Delta\boldsymbol{u}_{i+1} = \frac{\partial\boldsymbol{u}_{i+1}}{\partial\boldsymbol{u}_i}\Delta\boldsymbol{u}_i + \frac{\partial\boldsymbol{u}_{i+1}}{\partial\boldsymbol{u}'_i}\Delta\boldsymbol{u}'_i + \frac{\partial\boldsymbol{u}_{i+1}}{\partial q/p}\Delta q/p, \tag{5.2}$$

where the individual partial derivatives can be obtained by decomposing the propagation Jacobians. Eq. (5.2) allows for three points with offsets $(\boldsymbol{u}_{i-1}, \boldsymbol{u}_i, \boldsymbol{u}_{i+1})$ to determine two slopes at the central point. Their difference is called kink:

$$\boldsymbol{k}_i = \boldsymbol{u}'_{i,+} - \boldsymbol{u}'_{i,-}. \tag{5.3}$$

If the reference trajectory is based on broken lines, there can be initial non-zero kinks $\boldsymbol{k}_{0,i}$ at the scatterer.

For a measurement without a scatterer, there is no kink and prediction is obtained by interpolating between enclosing scatterers to get interpolated offsets $\boldsymbol{u}_{\text{int},i}$. The measurement has covariance $\boldsymbol{V}_m$, which can be internally diagonalized in GBL.

A scatterer of zero thickness is a source of slope variance θ_0^2 gained by the particle trajectory in co-moving frame. Each scattering point represents a thin scatterer. If none of the axes of local frame of the point is perpendicular to the track direction, covariance matrix becomes non-diagonal:

$$V_k = \frac{\theta_0^2}{(1 - c_1^2 - c_2^2)^2} \begin{pmatrix} 1 - c_2^2 & c_1 c_2 \\ c_1 c_2 & 1 - c_1^2 \end{pmatrix}, \tag{5.4}$$

where $c_i = \boldsymbol{e}_{\text{track}}.\boldsymbol{e}_{u_i}$ is a scalar product of unit vectors in direction of the track and the two local coordinates u and v. GBL can internally diagonalize the scattering covariance.

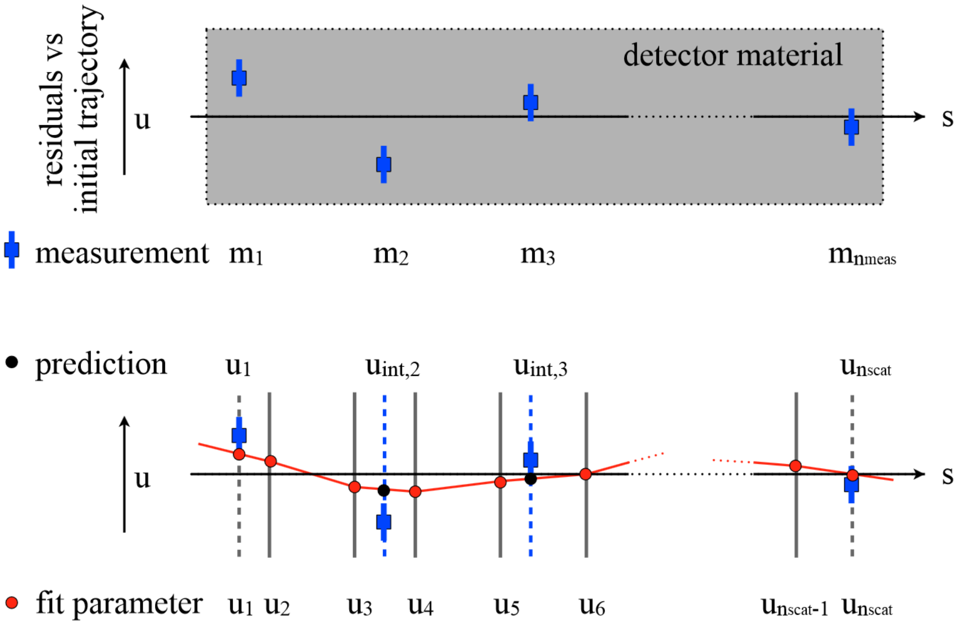


Figure 5.1 Top: Simple Broken Lines trajectory without magnetic field in two dimensions with two scatterers between each pair of measurements $m_1, \dots\ m_{n_{\text{meas}}}$ in detector material and the initial reference trajectory parameterized by arc-length s. Bottom: The corresponding GBL trajectory with fit parameters as offsets $u_1, \dots\ u_{n_{\text{scat}}}$ at scatterers. Predictions to measurements $u_1, u_{\text{int},2} \dots\ u_{int,n_{\text{meas}}}, u_{n_{\text{scat}}}$ are retrieved by interpolation between offsets of the preceding and following scatterer, 1st and last measurement is added to sequence of scatterers (no interpolation). [21]

Two-dimensional offsets at scatterers $\boldsymbol{u}_i$ together with common change of curvature $\Delta q/p$ compose the $2\text{n}_{\text{scat}} + 1$ fit parameters $x = (\Delta q/p,\ \boldsymbol{u}_1, \dots, \boldsymbol{u}_{\text{n}_{\text{scat}}})$ which define the broken trajectory. Note that these offsets and curvature change are relative to the reference. The GBL trajectory is illustrated in Fig. 5.1.

An external seed, prediction from other detector or other constraint can be added to one point at the trajectory. This seed constraints the prediction at that point to not change on average, with variance of the local track parameters $\boldsymbol{V}_s$.

The fit then has to minimize the following χ^2 expression to solve for the large vector of parameters $\boldsymbol{x}$:

$$\begin{aligned}
\chi^2(\boldsymbol{x}) \; = & \sum_{i=1}^{n_{\#\,\mathrm{meas}}} \left(\frac{\partial \boldsymbol{u}_{\mathrm{int},i}}{\partial \boldsymbol{x}}\boldsymbol{x} - \boldsymbol{m}_i\right)^T \boldsymbol{V}_{m,i}^{-1} \left(\frac{\partial \boldsymbol{u}_{\mathrm{int},i}}{\partial \boldsymbol{x}}\boldsymbol{x} - \boldsymbol{m}_i\right) && \text{from measurements} \\
+ & \sum_{i=2}^{n_{\#\,\mathrm{scat}}} \left(\frac{\partial \boldsymbol{k}_i}{\partial \boldsymbol{x}}\boldsymbol{x} + \boldsymbol{k}_{0,i}\right)^T \boldsymbol{V}_{k,i}^{-1} \left(\frac{\partial \boldsymbol{k}_i}{\partial \boldsymbol{x}}\boldsymbol{x} + \boldsymbol{k}_{0,i}\right) && \text{from scatterers} \\
+ & \left(\frac{\partial(\Delta \boldsymbol{q}/\boldsymbol{p}, \boldsymbol{u}', \boldsymbol{u})_{\mathrm{int},s}}{\partial \boldsymbol{x}}\boldsymbol{x}\right)^T \boldsymbol{V}_{s,i}^{-1} \left(\frac{\partial(\Delta \boldsymbol{q}/\boldsymbol{p}, \boldsymbol{u}', \boldsymbol{u})_{\mathrm{int},s}}{\partial \boldsymbol{x}}\boldsymbol{x}\right), && \text{from external seed}
\end{aligned} \tag{5.5}$$

where the derivatives w.r.t. fit parameters are build from the decomposed Jacobians and project the kinks and measurement predictions from the fit parameters. Written as matrix equation for the minimum of (5.5),

$$\boldsymbol{\Gamma x} = \boldsymbol{\beta}, \tag{5.6}$$

the matrix is composed of the many derivatives and has a special structure. Because the kinks and interpolated predictions for a point depend only on previous and next plane and for all points depend on the common curvature correction, $\boldsymbol{\Gamma}$ is bordered band matrix (border size $b = 1$ or $b = 0$ for no magnetic field and band width $m = 5$). The band part is decomposed using root-free Cholesky decomposition into diagonal and left unit triangular band matrix. This approach allows one to reduce computational complexity of inverting $\boldsymbol{\Gamma}$ $(n \times n)$ which is $O(n^3)$ to $O(n(m+b)^2)$ for obtaining the solution $\boldsymbol{x}$ and $O\big(n^2(m+b)\big)$ to get the full covariance $\boldsymbol{\Gamma}^{-1}$ for the solution.

If the corrections to initial trajectory are not small, linearization (5.2) may not be accurate and the procedure has to be iterated. Approach of GBL is mathematically equivalent to the Kalman filter, which needs to invert a 5x5 matrix each time a measurement is added, but it is different computationally. GBL was showed to perform even better than Kalman in a toy setup [23].

The implementation of GBL has a convenient interface, which allows to pass all required information while GBL builds the linear equation system internally and performs the solution. The required information for construction of a trajectory follows:

- For each point, user has to provide the Jacobian (5.1) which translates track changes from previous to current point
- A scatterer needs the initial kinks and precision (inverse covariance) matrix, which may be non-diagonal if the local frame does not coincide with the frame of the propagating track. This happens especially in case where there is also a measurement at point with scatterer. One should then use (5.3)
- A measurement requires the residual to prediction, precision matrix and a projection matrix from the local system at the point into the measurement system. The dimensionality of the measurement is not restricted and all 5 parameters of the local track parameters can be included into residual. If some parameter has to be disabled in the fit, e.g., for a strip measurement, this can be reflected in the precision matrix by setting appropriate element to zero (infinite error).

- Additional fit parameters $\boldsymbol{l}$ can fitted with a measurement. For each such parameter, a vector of local derivatives of measurement residual w.r.t. each additional fit parameter is needed. Multiple parameters can be added. The derivatives have to be provided as a matrix $\frac{\partial r_i}{\partial l_j}$.
- Global parameters with derivatives as matrix $\frac{\partial r_i}{\partial a_j}$ and labels can be stored in the trajectory. They are not fitted by GBL and only used by Millepede, if the trajectory is stored in the Mille binary file.
- External seed can be added to a specified point. The precision matrix for local track parameters has to be specified.

GBL assumes by default that the trajectory is described by 5 parameters in presence of magnetic field. If no magnetic field is present, fitting of the q/p parameter can be disabled by a flag. For inhomogeneous magnetic field and energy loss, the effects on propagation are contained in the Jacobians. Therefore, it is possible to include such effects into the fit in this way.

GBL allows one to reduce impact of bad measurements on the track fit by outlier down-weighting with the method of M-estimators, which requires internal iterations to be performed, giving less weight to measurements with large residuals.

A constructed trajectory can be directly written to the Mille binary file by GBL. This makes the step from fitting to alignment quite easy. One only has to provide the global derivatives and their labels. All remaining information is already used for the fit. The local derivatives are obtained from point to point Jacobians and possible additional local parameters are attached. The constructed matrix (5.6) directly corresponds to (4.8) and Millepede does in fact the same inversion as GBL utilizing the same optimization methods.

5.3 Effects on charged particle trajectories

Interactions of charged particle with medium result in change of momentum magnitude and direction. This section gives a brief overview of effects taken into account in track fitting performed in this thesis. The full Monte Carlo simulation in basf2 incorporates much wider range of effects, which are part of Geant4 propagation. For tracking, simplified treatment is used to include the main deterministic effects. For treatment in GBL, only noise coming from multiple scattering is included in the fit. The effects of average energy loss from ionization, excitation and bremsstrahlung are included through propagation Jacobians as derivatives of q/p parameter of the track.

Excitation of shell electrons and ionization of atoms are main contributions to energy loss of low-momentum particles. An average energy loss of a heavy particle due to ionization and excitation of electrons within atoms of traversed medium can be described by Bethe-Bloch formula [24]:

$$-\langle\frac{dE}{dx}\rangle = 0.307075\ \mathrm{MeV} \times z^2 \frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 T_{\max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2}\right], \tag{5.7}$$

where z is the charge of the projectile in units of e, βc its velocity and γ the relativistic factor. For the medium, Z is atomic number, A its atomic mass in $\mathrm{g.mol^{-1}}$, I is mean excitation energy in eV. m_e is the mass of the electron and $\delta(\beta\gamma)$ is relativistic density factor relevant only for very high $\beta\gamma$ of the projectile. $T_{\max}$ is maximum energy transferred between the projectile and electron in the medium in a single collision:

$$T_{\max} = \frac{2m_e c^2 \beta^2 \gamma^2}{1 + \frac{2\gamma m_e}{M} + \left(\frac{m_e}{M}\right)^2}. \tag{5.8}$$

It should be noted, that (5.7) only describes mean energy loss and fluctuations are rather large, caused by large momentum transfers to electrons, being knocked out of the nuclei. These secondary particles, called δ –electrons, can cause further ionization.

Mainly for electrons, as the lightest charged particles, the radiative losses, called bremsstrahlung, cause a significant energy loss at high momentum exceeding ionization losses. This is caused by interaction with nuclei with emission of photon. For electrons, the average energy loss due to bremsstrahlung can be written as

$$-\frac{dE}{dx} = \frac{E}{X_0}, \tag{5.9}$$

where X_0 is so called radiation length specific to given material. This results, on average, in exponential decrease of electron initial energy E_0:

$$E = E_0 \exp\left(-\frac{x}{X_0}\right), \tag{5.10}$$

where again this is only the average energy loss. [24]

Mostly due to Coulomb scattering from nuclei, the particle gets deflected many times when traversing material of even μm thickness. Each of this deflection is usually small and the net effect after travelled distance x in material with radiation length X_0 can be described by Gaussian distribution of the deflection angles with zero mean and the width given as

$$\theta_0 = \frac{13.6\ \mathrm{MeV}}{\beta c p} z \sqrt{\frac{x}{X_0}} \left[1 + 0.038 \times \ln\left(\frac{x}{X_0}\right)\right], \tag{5.11}$$

where βc is the velocity of the particle, z is its absolute charge in units of e and p its momentum. A real distribution is not exactly Gaussian and has wider tails, which correspond to Rutherford scattering at large angles. Formula (5.9) approximates the central 98% part of the distribution with accuracy better than 11% for $10^{-3} < \frac{x}{X_0} < 100$ [pdg]. The deflection angle θ with variance (5.11) is understood as projected into one plane, for the width of the distribution of space angles, it is $\theta_{\mathrm{space}}^{\mathrm{RMS}} = \theta_0\sqrt{2}$.

5.4 Treatment of Multiple Scattering

For particle traversing material of small thickness, the effect of multiple scattering can be approximated by an idealized thin scatterer. Thin scatterer produces no direct spatial displacement, only changes in slopes of the track. These changes are random and their variance $\Delta\theta^2 = \theta_0^2$ depends on type and momentum of the charged particle as well as on properties and thickness of the traversed material. In two dimensions with no magnetic field, with track represented by its slope and offset (θ, y) and parameterized by arc length s, the covariance gained in track parameters just after traversing a thin scatterer with the variance $\theta_{0,i}^2$ at s_i reads

$$V_i(s_i) = \begin{pmatrix} \Delta\theta^2 & \Delta\theta\Delta y \\ \Delta\theta\Delta y & \Delta y^2 \end{pmatrix} = \theta_{0,i}^2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{5.12}$$

A difference in the slope introduces correlation between track offset and arc length, as for small deflection angles ($\sin\Delta\theta \sim \Delta\theta$) it propagates like $\Delta y(s) = (s - s_i)\Delta\theta, \Delta\theta(s) = \Delta\theta$ as shown in Fig. 5.2 a). The effects of single thick scatterer as seen at arc-length s are contained in the propagated covariance matrix

$$V_i(s) = \theta_{0,i}^2 \begin{pmatrix} 1 & s - s_i \\ s - s_i & (s - s_i)^2 \end{pmatrix}. \tag{5.13}$$

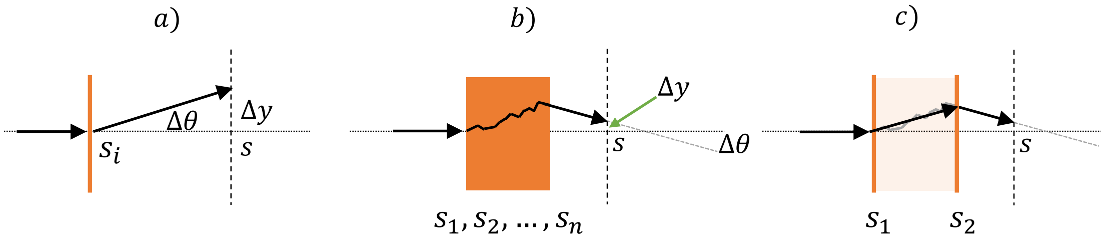


Fig. 5.2: Propagation of slope deflection from thin scatterer at s_i to measurement at s producing offset Δy and slope $\Delta\theta$ at the measurement plane (dashed vertical line) (a). Effect of thick scatterer made of (infinitely) many idealized thin scatterers $s_1, s_2, \dots, s_n$ producing offset and slope difference at measurement, which cannot be described by single thin scatterer placed before the measurement (b). Two thin scatterers can describe the effect of such a thick scatter, as shown in (c).

Such approximation might not be valid if the traversed material is not thin. In a realistic material, the particle undergoes many scattering events and the result is a combination of slope difference and spatial shift, see Fig. 5.2 b). A thick scatterer can be seen as a sum of many ideal thin scatterers (5.12) and the propagated variance is a sum of many contributions (5.13), where the sum of elements [2, 2] of matrix (5.13) needs a little manipulation. If we denote the sum of variances and the mean position of scatterers weighted by corresponding variance as

$$\theta^2 = \sum_i \theta_i^2, \quad \bar{s} = \frac{1}{\theta^2} \sum_i s_i \theta_i^2, \tag{5.14}$$

the contribution to the element [2, 2] reads

$$\sum_i (s - s_i)^2 \theta_i^2 = s^2\theta^2 - 2s\bar{s}\theta^2 + \sum_i s_i^2 \theta_i^2 \tag{5.15}$$

and the sum on r.h.s. can be expressed using the variance of weighted positions of the scatterers

$$\Delta s^2 = \frac{1}{\theta^2} \sum_i (s_i - \bar{s})^2 \theta_i^2, \tag{5.16}$$

$$\sum_i s_i^2 \theta_i^2 = \Delta s^2 \theta^2 + \bar{s}^2 \theta^2. \tag{5.17}$$

The effect of several thin scatterers as seen at arc-length s thus can be written as:

$$\sum_i V_i(s) = \theta^2 \begin{pmatrix} 1 & s - \bar{s} \\ s - \bar{s} & (s - \bar{s})^2 + \Delta s^2 \end{pmatrix}. \tag{5.18}$$

The thick scatterer is therefore described by triplet $(\theta^2, \bar{s}, \Delta s^2)$. For continuos distribution of infitely many thin scatterers, these quantities can be understood as a scaling factor, mean and variance of distribution of density of multiple scattering. In such case, the sums (5.14) and (5.16) change into integrals

$$\theta^2 = f\left(\int \rho(s)\mathrm{d}s\right), \quad \bar{s} = \frac{1}{\theta^2}\int s\rho(s)\mathrm{d}s, \quad \Delta s^2 = \frac{1}{\theta^2}\int (s - \bar{s})^2 \rho(s)\mathrm{d}s, \tag{5.19}$$

where $\rho(s)$ represents a lineary additive quantity, distance in units of radiation length, and f expresses the angular variance as a function of the sum of this quantity over the whole distribution of scatterers. The integration goes within any area, which has to be described as thick scatterer, usually between two consecutive measurements. Note that these expression are only accurate if the momentum loss is not significant in the integrated area. Momentum enters the multiple scattering formula (5.11) and it should be also integrated for materials with very large radiation length. In this thesis, the momentum is supposed to be constant between two measurements in calculation of variance of multiple scattering deflections. [25]

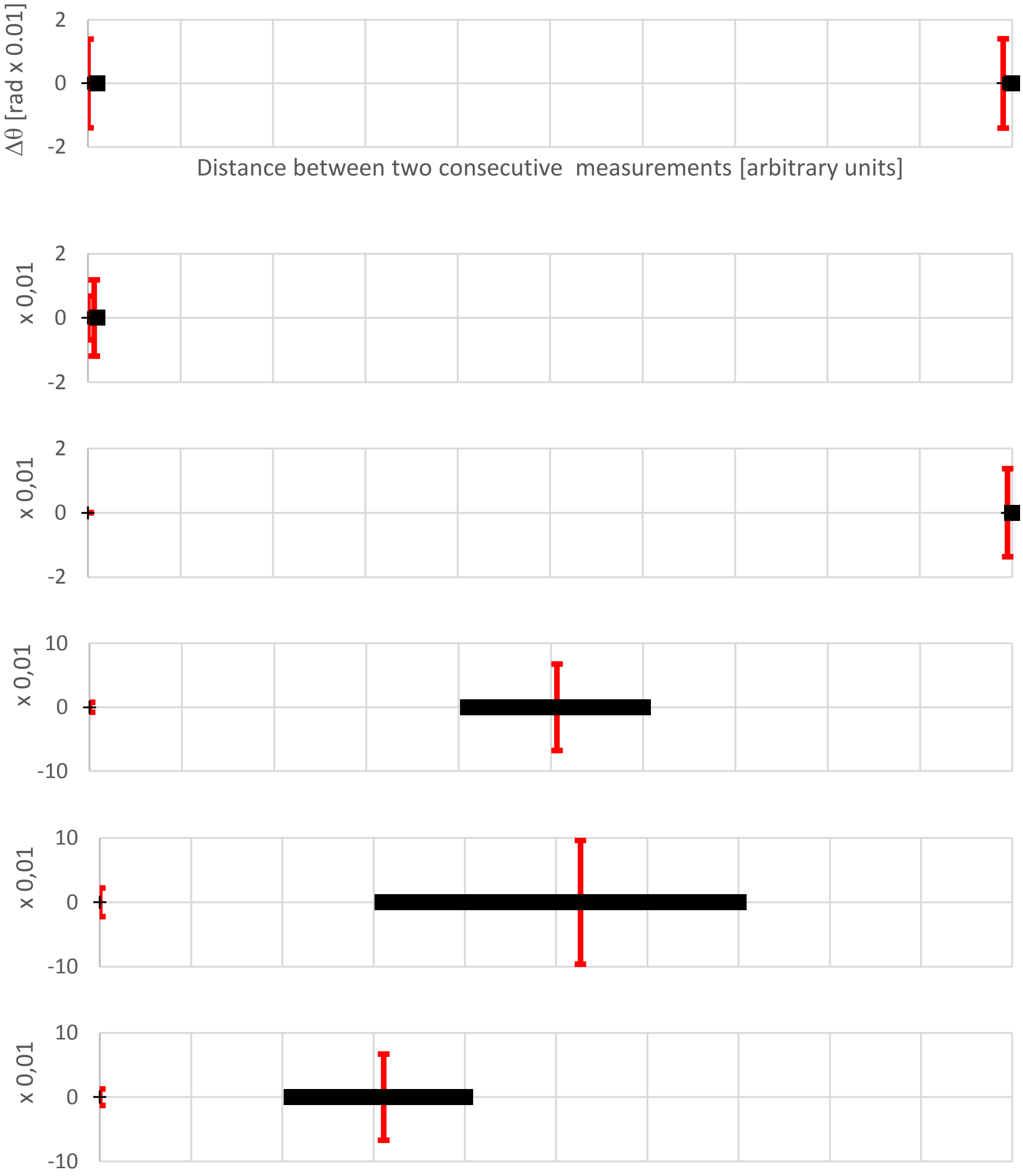
Any thick scatterer can be equivalently expressed as two thin scatterers as illustrated in Fig. 5.2 c). Two thin scatterers are described by two doublets (θ_1^2, s_1), (θ_2^2, s_2) which describe their variance and position along arc-length. One of these variables can be chosen, because in order to fit to a thick scatterers, only three equations must hold, (5.14) and (5.16) for $i = 1{,}2$. It will be convenient to express the effects as seen between two following measurements. Let the first scatterer be placed at the position of the first measurement being at zero arc length:

$$s_1 = 0. \tag{5.20}$$

Using (5.20) in 2nd definition in (5.14), substituting into 1st definition and inserting both into (5.16) to solve for s_2, allows determination of $\theta_1^2, \theta_2^2, s_2$ which are equivalent to a given thick scatterer $(\theta^2, \bar{s}, \Delta s^2)$:

$$s_2 = \frac{\Delta s^2 + \bar{s}^2}{\bar{s}}, \qquad \theta_1^2 = \frac{\theta^2 \Delta s^2}{\Delta s^2 + \bar{s}^2}, \qquad \theta_2^2 = \frac{\theta^2 \bar{s}^2}{\Delta s^2 + \bar{s}^2}. \tag{5.21}$$

In this way any thick scatterer can be simulated by two ideal thin scatterers which are used by GBL and emulate arbitrary distribution of scattering material between two measurements. Usage of (5.20) is not necessary but convenient for further implementation, derivation of the general case $s_1 \neq 0$ is straightforward.



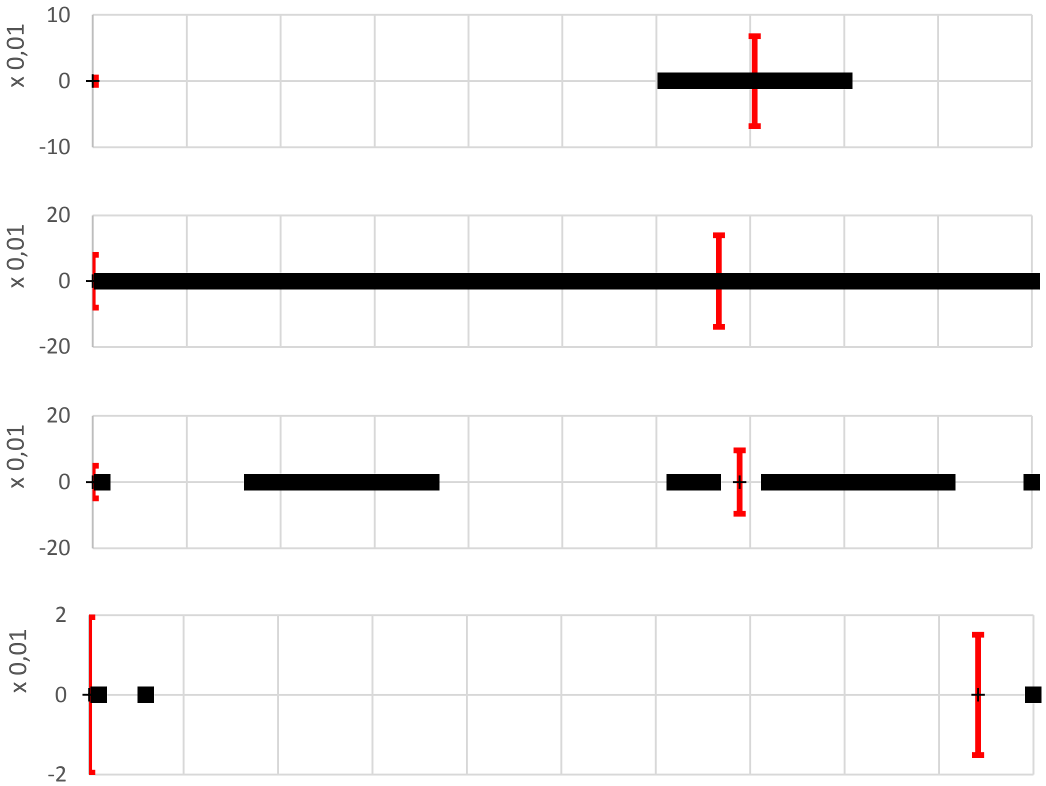


Figure 5.3: Representation of various material distribution by two equivalent scatterers. Plots (1) to (10) (from top to bottom) showing variance and position of two equivalent scatterers and their variance in a toy setup. Particle goes from left to right. Axes as in (1): vertical axis shows computed scattering variance for each thin scatterer as red error bars (only for relative comparison), placed at scatterer positions. Big black squares show material distributed in bins of length 1 with radiation length 1 per bin. Toy particle has unit charge, momentum 1GeV/c and mass 0.1GeV/c^2. Horizontal axis goes from 0 to 100 in a toy setup. First scatterer is always at the beginning. (1) to (3) show almost thin material at both sensors, first and second sensor. (4) to (7) show how a central thick scatterer, enlarged or moved, affects the position and variance of thin scatterers. (8) shows homogeneous material distribution between measurements (in total $100X_0$) and (9) some arbitrary sequence of materials. Case (10) illustrates a typical situation caused by sensor overlaps, where particle crosses additional (inactive) material near the 1st measurement.

For illustration how this representation of material distribution with thin scatterers works, see sequence of plots in Fig. 5.3. It is shown, for various cases of material distributions, where the two thin scatterers (1st always at detector plane, according to (5.20)) are placed and their computed variance. It is supposed that a measurement is done at the beginning and the end of the horizontal axis and the material in between is simulated by thin scatterers. For the first three extreme cases, where material is concentrated as thin layers near measurements, this corresponds to simple treatment used, e.g., in CMS, when a thin GBL scatterer is placed at each sensor plane. Other examples illustrate the situation when the material cannot be treated as thin.

In reality, particles have varying momenta, cross the material under various angles, and travel different distance between planes. As seen from Fig. 5.2, the distribution of scattering material can significantly affect the distribution of equivalent scatterers. A realistic example using full simulation of Belle II vertex detector is shown in Fig. 5.4. The detector is illuminated by muons and anti-muons coming from the interaction point. The red structure of thin scatterers between detector planes is caused by overlaps of the detectors, where more amount of material is traversed on average by particles coming from IP uniformly. An average material distribution in the vertex detector corresponds to case (1) of Fig. 5.3, where for overlap region, the thin scatterer near next measurement gets detached from the measurement plane, which corresponds to case (10).

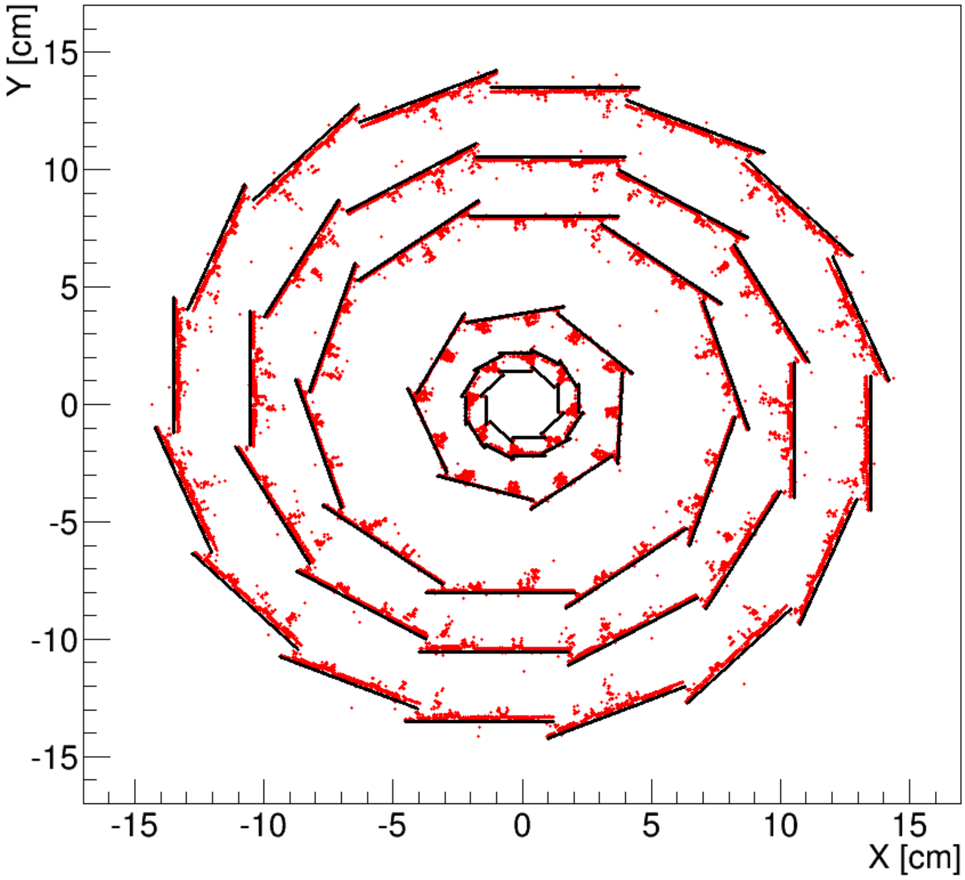


Figure 5.4: Belle II vertex detector in $R - \varphi$ projection. Thin scatterers at detector planes (black) and between (red). Picture is a result of full simulation with muons with momentum in uniform range from 0.05 to 3.0 GeV/c coming from IP uniformly in φ and θ. Only small cut of the detector within $|z| < 2cm$ is shown to clearly see the specific structure of distribution of intermediate scatterers. The black structure basically reflects the distribution of material inside the vertex detector, located at sensor planes.

6 Software Development

Although the main part of the work done for this thesis is software development, collecting a complete documentation here is inappropriate for three main reasons. First of all, a meaningful detailed documentation cannot be written without exposing the reader to almost complete structure and terminology of GENFIT and basf2, which is far beyond the scope of this thesis. Also, all discussed software is under heavy development to be delivered for the experiment on time, for which reason, any paper documentation becomes obsolete usually before being printed. The third reason is, that automated tools are used to generate user – friendly documentation from the code with special commands, being the most up – to – date and all software developed in this thesis is heavily documented in this way. Therefore, this document describes the key ideas behind the software used and developed. The implementation is usually discussed in lower detail with focus on key concepts demonstrated on examples which may be useful for the user.

6.1 GBL Extension for GENFIT 2

An extension which allows to use GBL within GENFIT 2 has been developed for this thesis. This extension is not specific to basf2 and should allow to use GBL also in other experiments without additional effort. Though, there are some limitations for measurements used in fit, which will be discussed later. It is available within GENFIT 2 at SourceForge:

http://sourceforge.net/projects/genfit/

The main features include:

- Automatic construction of a GBL trajectory from a GENFIT track with having an initial seed position and momentum.
- Update of the track with results of GBL fit and possibility of additional iterations of the broken trajectory.
- Two and one – dimensional pixel/strip/wire measurements can be freely combined in the trajectory. Only single measurement at a point is supported (see Implementation).
- Arbitrary orientation of measurement planes without any assumptions on the geometry of the detector.
- Automatic treatment of material distribution by simulating thick scatterers.
- It is possible to use thin scatterers at measurement planes optionally to save space on disc or computing time or for lightweight geometries.
- Fit with or without magnetic field is supported.
- Inhomogeneous magnetic field, as well as energy loss is taken into account through Jacobians provided by GENFIT.
- Measurements can provide local and global derivatives attached to the trajectory, which can be easily output for alignment with Millepede.

The extension to GENFIT 2 includes a version of C++ GBL implementation. To establish interaction between GBL and GENFIT 2, an interface was developed, forming an independent **GblFitter** and several helper classes. The implementation and basic usage are in more detail discussed in Appendix A.

6.2 Alignment Package

In basf2, the fitting with GBL and alignment with Millepede II is available in the alignment package which uses the GblFitter inside GENFIT. It consists of two modules, GBLfit and MillepedeIIalignment. The latter is only used to execute Millepede within basf2, compute geometry constraints and convert Millepede result into XML file.

The GBLfit module allows to use GblFitter to fit track candidates provided by a track finding module. It is almost a clone of GenFitter module which provides standard GENFIT fitters in basf2. Therefore the resulting fitted tracks and other related objects are stored in DataStore and allow further retrieval of fit data by other basf2 modules. This should allow to simply interchange these two modules in tracking. However some tools rely on features specific to Kalman fitters, like event display. Experimental upgrade is available (and used to generate figures in this thesis), but not yet officially.

The geometry including magnetic field has to be loaded in basf2 to perform fitting. The GBLfit module starts by constructing GENFIT tracks from track candidates. During this process, hit indices are converted to so called RecoHits, which, for each subdetector keep the raw measurement and information needed to transform into reconstructed hit. Following basf2 RecoHit classes are supported:

- SVDRecoHit for single strip, one dimensional measurement
- SVDRecoHit2D for two corresponding SVD strips found by the track finder combined into single two dimensional measurement. This is necessary to be done outside GENFIT, because for slanted SVD sensors, additional geometry information is necessary to compute two dimensional covariance properly
- PXDRecoHit for 2D PXD hits
- TelRecoHit for basf2 implementation of EUDET telescopes with 2D pixel hits
- CDCRecoHit for 1D wire measurements in CDC. Only measurement with highest weight is taken.

That is, any combination of these hits can be fitted.

Constructed track is then processed by an instance of GblFitter. All options of the fitter can be set using parameters of the module. A successfully fitted track is written to DataStore. Following conditions allow so make basic selection of tracks to be used in alignment procedure:

- maximal χ^2 at each point of the track, default is 50,
- minimal P-value of the track, or
- number of degrees of freedom of the track.

Accepted tracks are written Mille binary file with name specified in module parameters.. Currently, alignment for planar PXD, SVD and telescope sensors is supported. For this purpose, the RecoHit classes were extended to implement the

ICalibrationParametersDerivatives interface. The global derivatives for rigid body parameters in local sensor system are computed as in (5.XX).

The alignment package contains a set of examples which allow to perform basic misalignment studies for VXD. They demonstrate generation of samples with collision and cosmic ray tracks, reconstruction in misaligned geometry, running of Millepede II and using its results to correct the geometry.

The samples are generated in nominal geometry and stored. The reconstruction is run in another instance of basf2, which uses geometry different from that used for generation of the samples. In this way, misalignment is simulated in course of this thesis. More advanced methods which allow to only shift position of hits during reconstruction are being developed.

6.3 Testbeam Package

The Testbeam package encapsulates simulation and reconstruction of VXD beam tests completely within basf2. The main idea for this package is to reuse as many existing framework tools as possible. This approach allows to test significant part of the software in challenging conditions of the beam test, where operation of many components may not be ideal like in MC simulations. In addition, simulation and offline reconstruction of EUDET telescopes is added to the package, which forced development of several beam test specific modules, extending the list of supported detectors beyond that of Belle II.

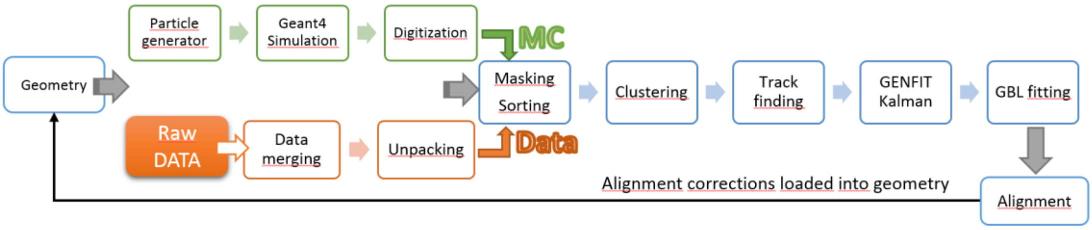


Figure 6.1.: Diagram showing the processing flow in the beam test package. The simulation branch (green) and real data branch (orange) merge before clustering and the remaining processing is the same for both kinds of input.

This section describes the individual parts of the framework used in beam test simulation and reconstruction. Tools newly developed or updated for the beam test are presented, from which the core feature is the beam test geometry, discussed first. Then also several features from the common software tools and developed for the beam test by other collaborators which is essential for the beam test simulation and reconstruction. The scheme of processing flow using the tools discussed in the following is shown in Fig. 6.1.

6.3.1 Geometry and Magnetic Field

The geometry is stored as hierarchy of parameters in XML. It does not rely on some specific arrangement of the components and volumes and detector can be placed freely. A

C++ class **VXDTBCreator** which converts the XML into Geant4 hierarchy of volumes is developed. An arbitrary structure of volumes can be defined and any of these sensors used: PXD, SVD (not slanted) and TEL for EUDET telescope.

The hierarchy of volumes is contained inside Volumes tags. Each volume must be entirely contained in its mother volume, with respect to which its position is defined by three translations and three rotations w.r.t. to default placement at center of mother sharing its coordinate system. Two shapes of volumes are supported: box and tube. In addition, boolean operations (subtraction, union, intersect) are allowed on volumes, which allow to create quite complex structures. A box volume can be set as active. In such case, type of the sensors must be defined and its operational parameters. There is a rule, that each volume must have a name tag or attribute, to keep the hierarchy readable. An example how to create simple geometry with one telescope sensor is shown in Appendix B.

A set of linked XML files for the DESY beam test geometry was created for this thesis based on information on collected materials and technical drawings of the setup collected from different groups. Individual building blocks of the geometry are located in separated XML and can be reused. A visualization of the geometry is shown in Fig. 6.3. Each material is displayed in different color and a view of the PXD sensor itself is shown.

The magnetic field of PCMAG was measured in [26] and radial field map is available as ASCII file. This file was converted into format of basf2 field map and included in the Testbeam package. However recent update of the implementation in basf2 is not compatible with this simple map. A straightforward modification of the geometry package can be done to use detailed map of the field, see Fig. 6.2. The influence of realistic field on tracking is studied in this thesis, but constant field is used if not states otherwise. For online reconstruction, only a constant field was used.

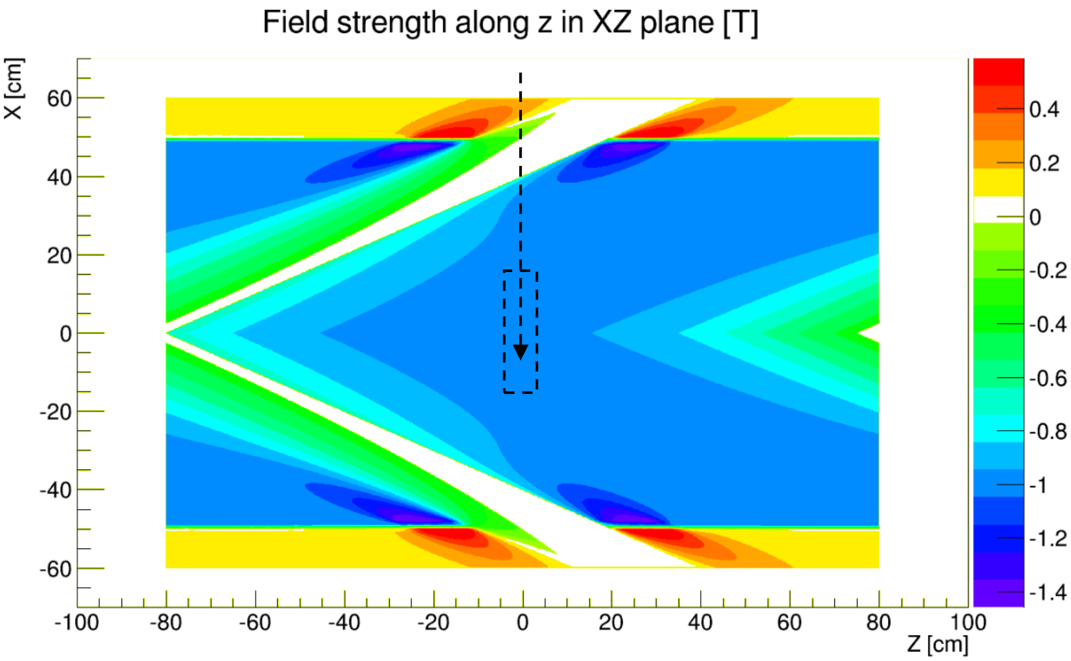


Figure 6.2: The map of Z-component of PCMAG magnetic field in XY plane, where the beam propagates (dashed arrow). The field is nearly constant in the center of the magnet, where the setup is placed (bashed box) but has very high gradient near the walls. The conic structure is most likely caused by special winding of the coil.

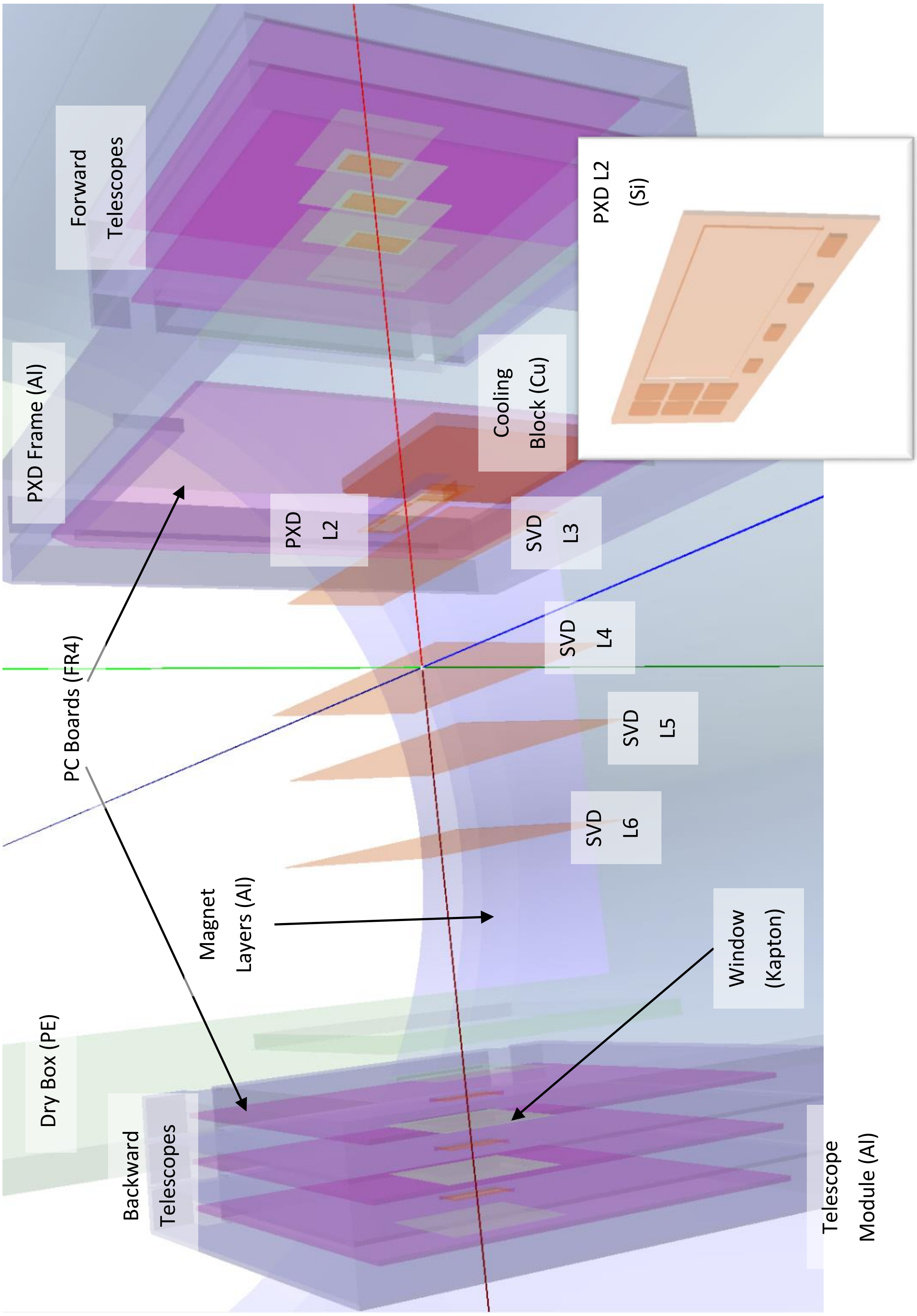


Figure 6.3: Visualisation of beam test geometry in ROOT OpenGL viewer zoomed into sensor area. Materials are distinguished by color: Al (blue/shadow), Si (orange), Cu (red), FR4 (violet), polyethylen and kapton (green). A view on PXD itself with SWITCHER, DCD and DHC chips clearly visible (inserted image).

6.3.2 Simulation and Digitization

The Monte Carlo simulation for the beam test is performed by means of standard basf2 modules performing Geant4 propagation and energy deposit. For a source of primary particles, a simple particle gun is used, which allows to define generation of vertex, momentum, number and kind of particles. Standard settings for beam test simulation are summarized in Table 6.1. These values were chosen according to independent study of B. Schwenker based on measurements with EUDET telescopes at DESY beam line 21.

Parameter	Range / value(s)	Unit
Momentum	2, 3, 4, 5 ± 5%	GeV/c
Theta	90 ± 0.005	degrees
Phi	180 ± 0.005	degrees
Vertex X-position	100	cm
Vertex YZ-position	0 ± 0.3	cm

Table 6.1.: Parameters for beam generation by the particle gun.

A finished simulation leaves the information on energy deposit from each Geant4 step as a simulated hit, **SimHit**. This object is stored in DataStore and records also additional information on the traversing particle, like current momentum. Because small 5um steps are used for propagation in active silicon, there is usually a lot of SimHits created per each hit. From these a single intersection point of particle track and the sensor is computed and another object, **TrueHit**, is created and stored. TrueHits represent the truth impact point of the track which can be used in MC studies without performing simulation of detector response.

For all discussed silicon sensors, the main part of simulation of detector response proceeds similarly using standard basf2 modules called Digitizers. The energy deposit from each step is converted to equivalent number of electron − hole pairs. These pairs are separated into smaller groups, clouds, and for each cloud and the motion to readout side of the sensor is simulated. This takes into account enlarging of the cloud due to diffusion of the charge during drift in electric field of the sensor and Lorentz shift induced by motion of the charge in magnetic field. At the readout plane, the cloud is seen as Gaussian distribution and for each strip or pixel, the corresponding amount of charge from each cloud is integrated. Digitizer output is stored as Digits in the DataStore, representing single pixel/strip signal.

For PXD, the readout plane is slightly below the surface of the sensor, where the internal gates are located. Lateral diffusion to the gates is simulated as Brownian motion of the clouds. The charge collected in each internal gate then corresponds the generated signal and converted to output of readout electronics

Each digitizer is able to add noise at specified level to the collected signal and particular digitizer utilize special features of the sensor, like strip crosstalk due to inter-strip capacitance for SVD. Effect of electronic noise are added and the ADC coven version can be performed.

The digitizer for EUDET telescopes for the Testbeam package has been developed by Peter Kvasnicka (Charles University in Prague) and is based on simplified version of the PXD digitizer module.

6.3.3 Raw Data Processing

Data from sensors are collected by DAQ and written to sequential ROOT files. At this stage we call it raw data which are input of the reconstruction. Before raw data can be processed, it must be unpacked and transformed into Digits. During this process, pixel and strip signals have to be mapped to corresponding pixel/strip of each sensor. For further processing, an initial sorting of the Digits may be necessary and for this purpose, sorting modules are used.

Because in reality sensors might have damaged or non-working parts, like hot pixels/strips which give signal all the time, without any real particle passing through. Digits from these strips/pixels have to be removed before hit reconstruction. For this reason, possibility of masking was added to the sorting modules for all sensors in this thesis. The mask defining non-working areas of the sensors is stored as XML file. For strip sensors, individual strips or their range can be specified. This file is processed into a look-up table. The sorting module then checks for each strip the look-up table and removes the masked Digits from further processing. For PXD, the approach is similar. The look-up table is used for individual pixel and row/column masking. In addition, ranges of rows/columns can be masked as well as rectangular areas. These are stored sorted by area and each pixel is checked if it belongs to some area specified in the map.

For inclusion of EUDET telescope data into offline processing, first the data from VXD and the telescopes has be merged into single ROOT file. For this purpose, a merger module has been developed by Peter Kvasnicka and included into new eutel package into basf2. This package integrates main parts of framework developed for the telescopes and allows to read their specific raw data files.

6.3.4 Hit Reconstruction

Usually for a particle traversing the sensor, more pixels/strips give signal. Digits corresponding to single hit have to be merged into clusters by clustering modules. The result is a single hit position estimation and error computed taking into account shape and charge distribution in the cluster. The clusters are formed by merging neighboring digits with signal over a specified threshold around peak of the cluster (seed). The position is then computed using center-of-gravity algorithm for clusters with two digits or analogue head-to-tail algorithm for larger clusters. Standard basf2 modules are used for this purpose.

6.3.5 Tracking and Alignment

For MC simulation, the default track finding module of basf2 (TrackFinderMCTruth) is cloned for the beam test package (TrackFinderMCVXDTB) where support for the telescopes was added. This module uses the truth information from MC simulation which contains relations between hits and particles that produced them. It can work with either clusters or TrueHits, skipping the digitization and clustering effects.

For real data, track finding is a real challenge. This is also the only feature of basf2 which had to be significantly upgraded to work for the beam test. For this purpose, the VXD track finder (VXDTF) module for Belle II was extended by its author, Jakob Lettenbichler, to support the beam test setup including the telescopes. The track finder is based on cellular automaton which reduces the complexity of possible hit combinations by filtering steps with increasing complexity [article]. The cellular automaton is composed of cells representing combination of hits. These combinations are checked for compatibility, e.g. by angle made by hits at three consecutive planes. In the final state, cells contain the track candidates to which quality indicators are assigned using simple circle / straight line fit to the candidate hits. At this stage the track candidates still can overlap and this is resolved by a neural, Hopfield network, selecting the best set of non-overlapping track candidates. The track finder uses so called sector maps generated in MC simulation to optimize cutoffs applied at different processing levels and determine compatible regions (like adjacent sensors). These sector maps must be generated before VXDTF can be used for track finding. The Testbeam package contains a set of sector maps for different primary momenta and magnetic field on/off. The track finder is able to perform track finding even if momenta of sector maps and real beam do not exactly correspond, but the efficiency or purity of the track candidates might not be optimal. Output of VXDTF are GENFIT track candidate objects, containing list of cluster indices and momentum and position seed which defines an initial reference trajectory for track fit. These can be then converted into tracks providing corresponding RecoHit classes.

For track fitting, the default basf2 module (GenFitter) was cloned and added to the package (GenFitterVXDTB), which supports the telescopes. In this thesis, mainly the newly developed GBLfit module, which also supports the telescopes, is used for this purpose. All these modules start with conversion of track candidates into tracks by creating corresponding RecoHits for each hit. GenFitter module then allows to use Kalman filter for fitting in the beam test, while GBLfit utilizes the general broken lines refit of reference trajectory computed by the track finder.

Output of GBLfit module in form of Mille binary files is then used for alignment with Millepede II using the newly developed alignment package. For the beam test, set of helper python scripts has been written for easier manipulation with Millepede results and loading of the alignment parameters back into geometry in a fresh run of basf2.

7 Beam Test Results

In this chapter results of the DESY beam test based on the developed software tools discussed in previous chapters are presented with emphasis on the alignment. Before alignment was possible, the complete chain of beam test software tools had to be ready. The complete procedure is validated using MC simulations in the beam test setup. As a demonstration of correctness of the fitting procedure by GBL is shown in Fig. 7.1, where the digitization and clusterization is skipped in the reconstruction and the measurements are Gaussian smeared with error equal to $\mathtt{PITCH}/\sqrt{12}$. This is used to get rid of additional reconstruction effects which affect the errors and thus Chi2, p-values and pulls.

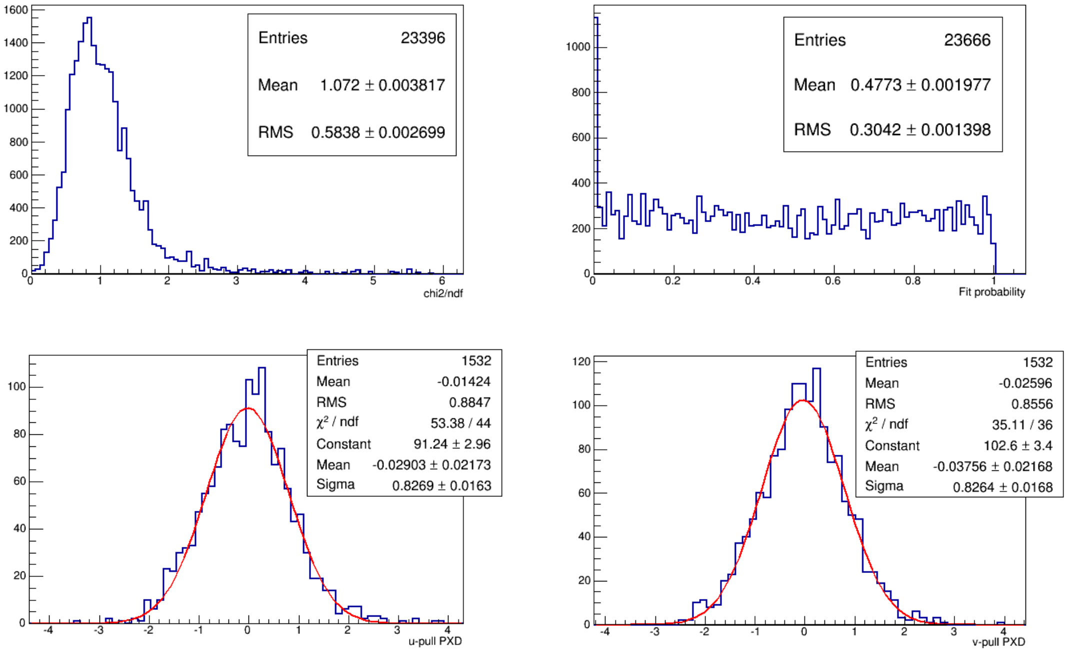


Figure 7.1: Chi2/NDF and p-value distribution for MC simulation of the beam test with Gaussian error smearing (no digitization and clustering) and two selected pull distributions in PXD layer 2. The red curve shows Gaussian fit to the pulls.

The author developed and maintained the geometry and sensor parameters during the beam test. Several scripts for processing were created for easier use and the complete processing chain was established in basf2. Prior to the alignment, masking of hot pixels/strips was necessary, for which the masking modules were developed and initial analysis to determine masking for SVD sensors was done during the beam test. The output files were later reprocessed and the damaged digits removed before clustering. The analysis in this thesis is based on these files, prepared by Peter Kodys, which have also masked PXD sensor and greatly reduced size.

During the beam test, the alignment procedure has been applied to SVD sensors. The alignment could be done directly starting with nominal geometry of SVD within Millepede. Example of difference of residuals for nominal and aligned geometry is shown in Figure 7.1. Before PXD was added to the alignment, it was manually pre-aligned by inspection of the correlation plots. It was necessary because it showed, the center of the PXD sensor is shifted by almost 5mm horizontally from the expected nominal position. The original idea was to use the EUDET telescopes for precise alignment. Telescope data were merged offline after the beam test. Unfortunately, due to synchronization problems, practically all telescope data were not available for track analysis.

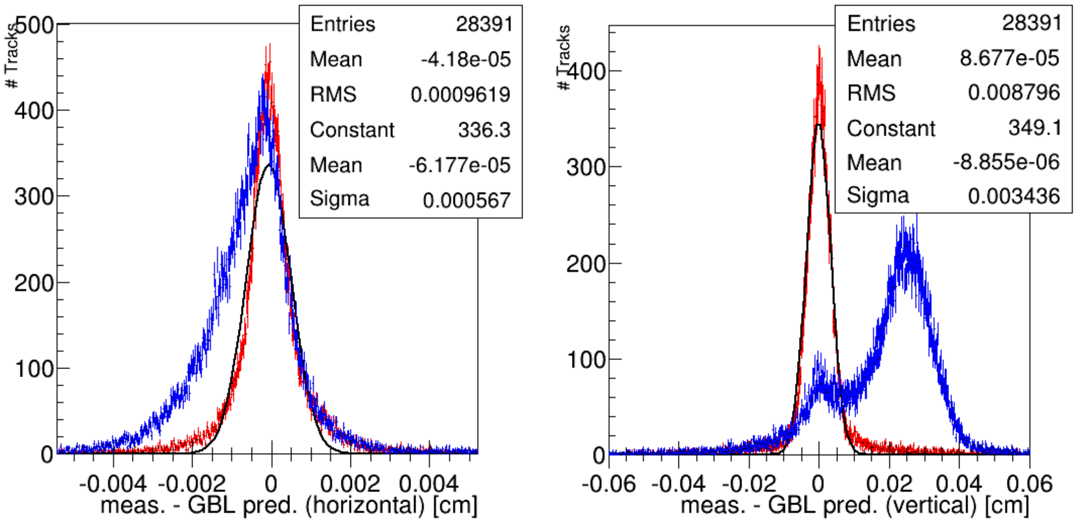


Figure 7.1: GBL residuals in V(left) and U (right) directions of SVD L4 within nominal (blue) and aligned (red) geometry. The parameters of the Gaussian curve (black) to the aligned case are given.

However, some telescope tracks crossing the whole setup without magnetic field were recovered. This was achieved by setting very loose cut-offs in the track finder. The obtained sample of track candidates is heavily contaminated by false tracks, where a track in the telescopes out of synchronization is merged with some hits (from other real track) in VXD by the track finder. The sample is fitted by GBL, where all hit errors are multiplied by factor of ten and used for Millepede alignment. The alignment procedure is iteratively repeated. In each step, an upper limit for hit pull is lowered and only tracks where all hits that passed this constraint are used for alignment. Because Millepede can internally remove outliers from the data, it can be used even for contaminated data, if the "bad track fraction" is less than one third (hard-coded limit). The successive cut-offs for pulls remove all data with large residuals and thus most of the false tracks. The scale of the hit errors is then reverted and final alignment is performed. The final limit for hit residual in each coordinate is five standard deviations.

The amount of telescope tracks retrieved by this approach is very low. More than 1000 tracks has not been collected from any single analyzed run. Outlier rejection in Millepede further reduces this number. Nevertheless the telescopes have very high resolution and thus even with such low statistics, the determined alignment for combined telescope and VXD runs is perfectly valid for reconstruction in VXD only, as shown below. The alignment obtained by this approach is used as basement for VXD only alignment. The outer SVD layers L3 and L6 (shifts only) are fixed at positions determined from the alignment with telescopes in nearest run. Then only inner layers, PXD and rotation angle for SVD L6 are aligned. For run 470, which is used as reference in following analysis, the determined parameters and their errors (dominated by low statistics) for all sensor are listed in Table 7.1.

As a demonstration, run 470 (no magnet) is now discussed. A MC simulation is performed to produce digits and then the same reconstruction chain as for real data is used. The MC simulation uses the clusters to estimate position error. In Fig. 7.3 (left), the p-values for MC (black) show an excess for measurements with high probability (near one). This is a clear indication that the position errors are overestimated. On the other hand, real data have more uniform p-value distribution, as also shown in Fig. 7.3 and there is almost no difference between telescope and VXD only alignment. This seems strange at first sight, but the most probable explanation is that the real data measurements have larger errors than expected in MC simulation. This in turn means real errors are closer to the “overestimated” cluster errors, computed by the clusterizers.

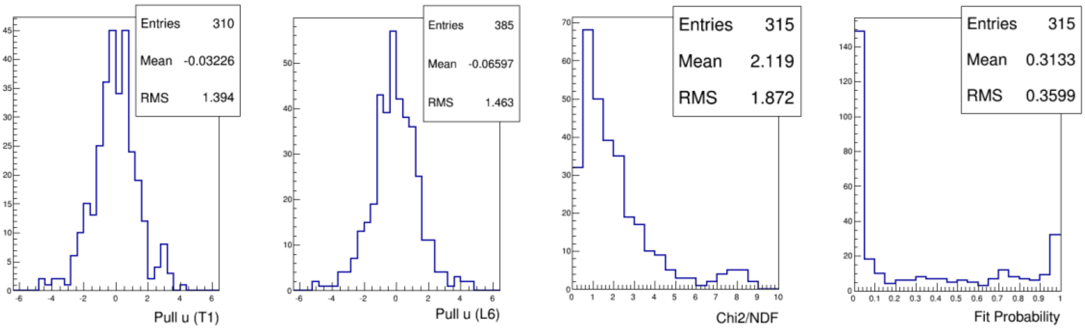


Figure 7.2: Example of pull, Chi2/NDF and p-value distributions in the telescope alignment for run 470. The five sigma cut is visible in the pull distributions.

	Δ*u* [μm]			Δ*v* [μm]			Δγ [mrad]		
L2 PXD	290	±	3	-4836	±	4	-14	±	0.2
L3 SVD	-1016	±	2	-659	±	2	-10	±	2
L4 SVD	-1021	±	2	-25	±	3	-17	±	2
L5 SVD	-965	±	3	189	±	10	-14	±	2
L6 SVD	-823	±	2	-257	±	3	-18	±	3
T1	0		fixed	0		fixed	0		fixed
T2	241	±	1	262	±	1	6	±	0.4
T3	407	±	1	282	±	1	3	±	0.5
T4	-173	±	1	-307	±	1	-24	±	3
T5	-69	±	2	-428	±	1	-23	±	3
T6	0		fixed	0		fixed	-19	±	4

Table 7.1: Telescope alignment for run 470. The final Chi2/NDF is 1.96 after Millepede alignment which also provided the errors. The outer telescopes are fixed.

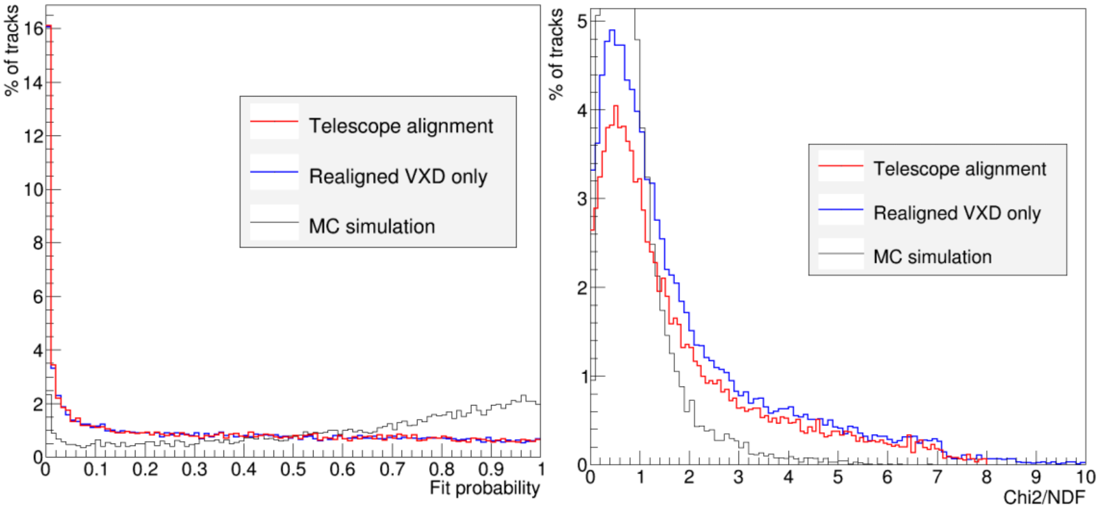


Figure 7.3: Distribution of p-values and Chi2/NDF of tracks in run 470 (blue/red) and full MC simulation (black).

The VXD only alignment does really not introduce any significant improvement, despite statistics is more than two orders higher compared to the telescope alignment. For example the residual distributions for the PXD sensor is shown in Fig. 7.4. Comparison also to MC simulation shows the alignment precision is close to its limits. This is better visible if mean values of all VXD layers are compared in Fig. 7.5. Even the MC simulation in ideal geometry has similar range of fluctuations as computed alignment for experiment data. The precision in horizontal, v-coordinate is clearly worse than for the vertical. This is caused by higher pitch and thus worse resolutions in this coordinate for all VXD sensors. From Fig. 7.5, the precision of alignment in the measurement coordinates can be estimated as about 1 μm for vertical and 3 μm for horizontal coordinate.

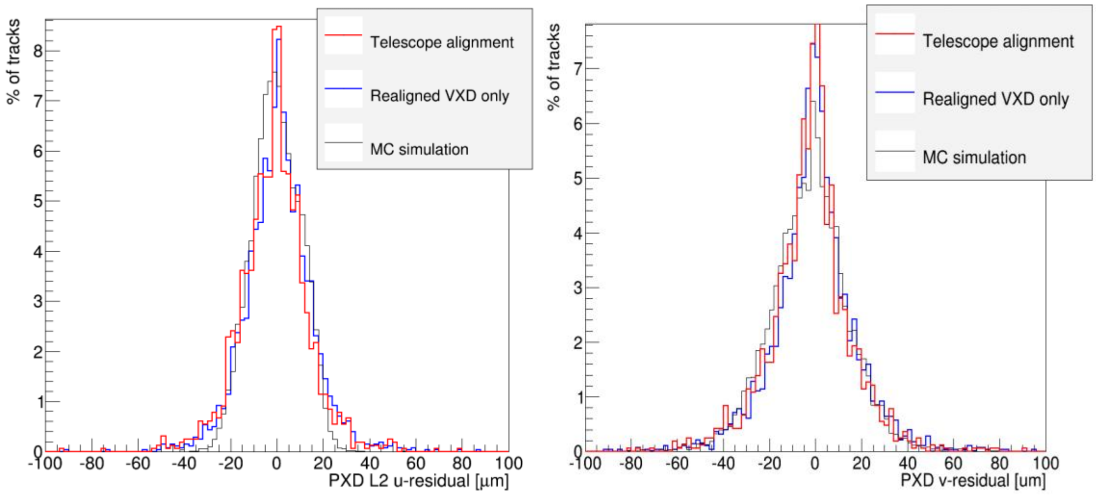


Figure 7.4 Residuals in run 470 for PXD in U (left) and V (right) for initial telescope alignment (red) and re-aligned geometry using only VXD (blue). The distribution is compared to MC simulation (black).

	Δ*u* [μm]			**Δ*v* [μm]**			**Δγ [mrad]**		
L2 PXD	287	±	1	-4839	±	1	-13.3	±	0.2
L3 SVD	-1016		Fixed	-659		fixed	-10		fixed
L4 SVD	-1021.5	±	0.2	-24.8	±	0.5	-18.0	±	0.2
L5 SVD	-966.2	±	0.2	192	±	2	-15.1	±	0.3
L6 SVD	-823		fixed	-257	±	fixed	-19.9	±	0.4

Table 7.2: Alignment parameters determined for VXD only alignment in run 470.

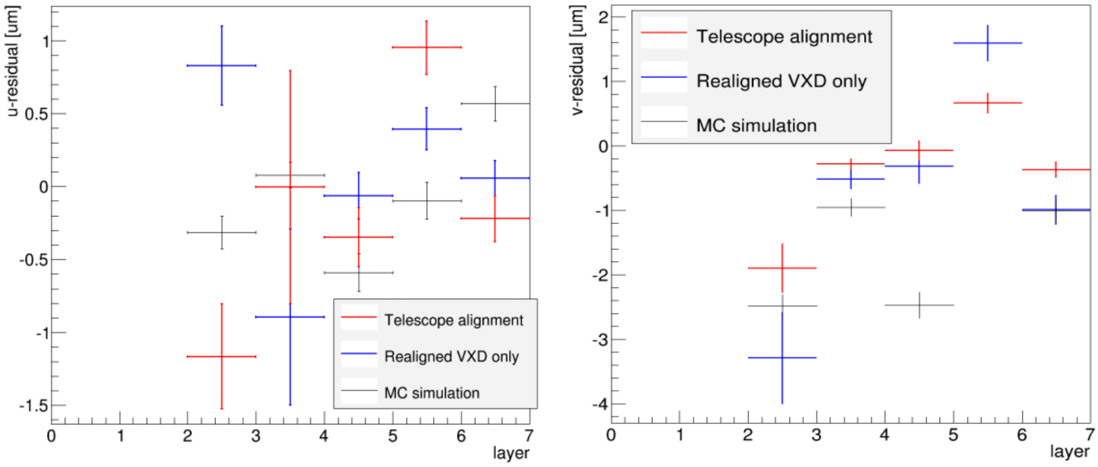


Figure 7.5: Mean value of residuals per VXD layer for MC simulation in ideal geometry (black) and run 470 using either initial telescope alignment (red) or re-aligned geometry using VXD tracks only (blue). Left: vertical, u-residuals. Right: horizontal, v-residuals. Error bars are statistical errors for the residual mean value (not RMS).

So far only alignment without magnetic field was discussed. In magnetic field, an additional degree of freedom has to be fixed in the bending plane to prevent deformations caused the track curvature. Another solution is to combine runs with different energy and with particles of opposite charge with runs without magnet, provided the geometry is not influenced by the magnetic field. Only electron runs are available for analysis and only three run with magnetic field survived initial data selection. Four runs were selected for this analysis. Two magnetic runs with energy 3 and 5 GeV are preceded and followed by runs with magnet turned off. A common alignment is determined and checked whether it fits to all these runs as demonstrated in Fig. 7.6, where the mean value of the residual over each 5000 tracks is computed in the vertical (bending) coordinate. A large shift about 40 μm is observed for the PXD sensor when magnet is turned on. In the horizontal coordinate, such shift is not observed and shifts in other sensors are much lower but still visible within alignment resolution. Fortunately there seems to be no significant systematic evolution of the residuals within the runs. On the other hand, a combined alignment of runs with and without magnet is not feasible in this situation. Even for the two magnet runs, which just follow each other, there is a visible change in PXD position. Therefore for magnetic field, an additional coordinate L5-u is fixed and the alignment is performed only on single runs. Example mean values of residuals for run 509 is shown in Figure 7.7.

The precision of alignment in sensor planes can be estimated by plotting the v-residual versus u-prediction and vice versa. The relation between the residuals and alignment parameters can be seen in Eq. (4.25), for example

$$\frac{\partial r_u}{\partial \gamma} = v \rightarrow \Delta r_u = v \Delta \gamma,$$

which will for sensor misaligned in γ produce a straight correlation line with slope corresponding to the misalignment angle (if small). All sensor show no or very small correlation with only rare and small improvements from initial telescope to VXD only alignment. None of these control plots shows significant rotation and all look similar to example shown in Fig. 7.8 for SVD layer 5.

	Δ*u* [μm]			**Δ*v* [μm]**			**Δγ [mrad]**		
L2 PXD	247	±	1	-4830	±	1	-17.3	±	0.1
L3 SVD	-1016		Fixed	-659		fixed	-10		fixed
L4 SVD	-1021.0	±	0.1	-15.8	±	0.2	-8.7	±	0.1
L5 SVD	-957	±	Fixed	131	±	1	-0.8	±	0.1
L6 SVD	-823		fixed	-257	±	fixed	10.6	±	0.2

Table 7.3: Alignment parameters determined for VXD only alignment in run 509 in magnetic field.

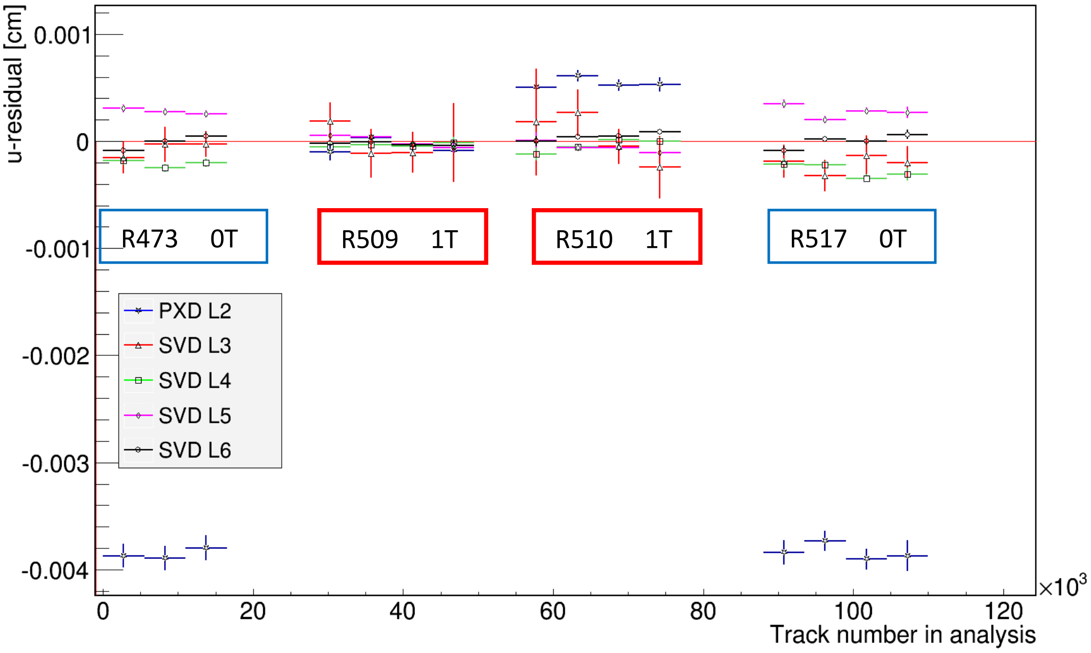


Figure 7.6: Evolution of residuals over analyzed runs using common alignment obtained from all these runs. PXD, shown in blue, is shifted by about 40um when magnetic field is turned on.

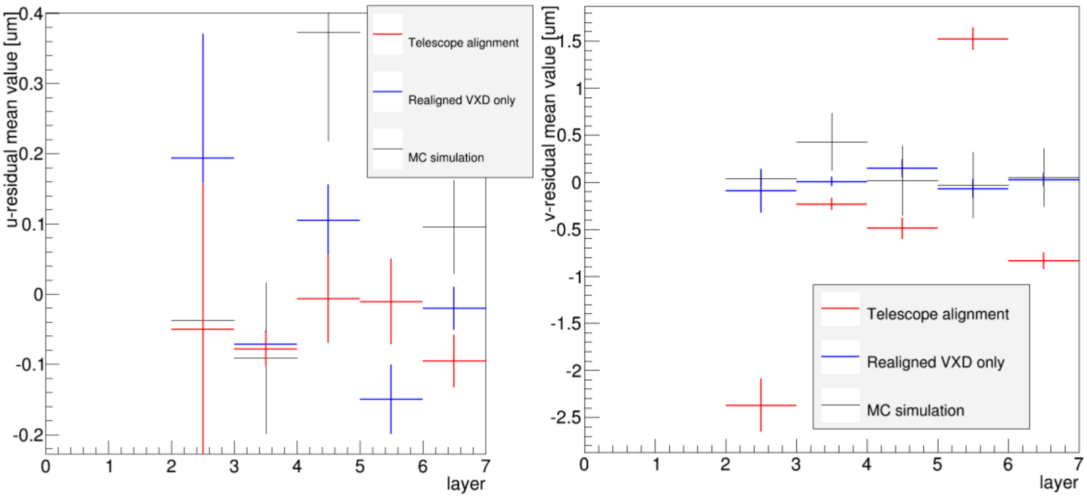


Figure 7.7: Mean value of residuals in run 509 with magnetic field using initial telescope alignment (red), re-alignment with VXD only (blue) and comparison to equivalent MC simulation in ideal geometry (black). Error bars are errors of mean values (not RMS).

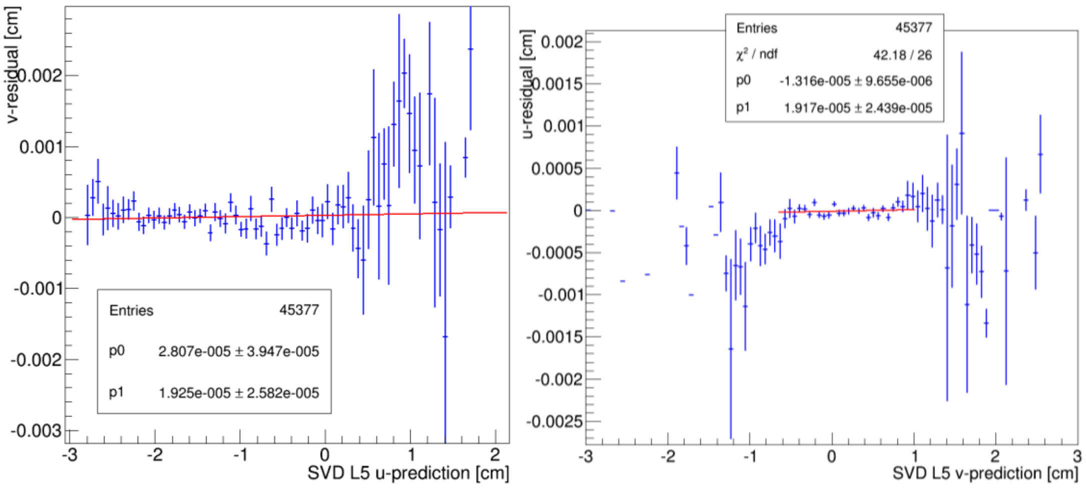


Figure 7.8: Example of control plots to check remaining rotation in sensor plane, showing perfect agreement with aligned geometry for SVD L5.

With the final alignment, a deeper analysis of the beam test tracks can started, for which the reached precision should be entirely sufficient. For example, the momentum distribution of reconstructed electrons and positrons for two different nominal beam energies in magnetic field are shown in Fig. 7.9. The precise alignment is also essential to estimate detector resolutions after additional corrections. The tracking resolution can be estimated from Fig. 7.10, where RMS values of the residual distribution are shown for each layer, showing also agreement with MC simulation.

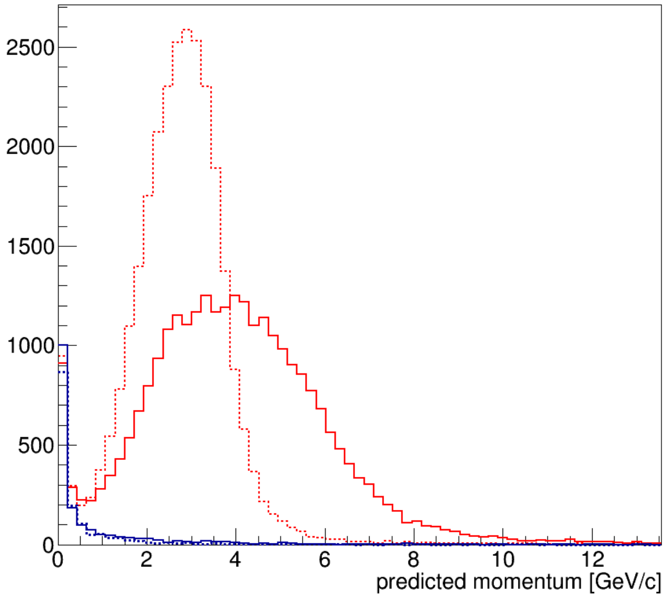


Figure 7.9: Momentum at first hit predicted by GBL. Reconstructed momentum for electrons (red) and positrons (blue) is shown for run 509 (solid lines) with 5GeV/c electrons and for run 510 (dashed lines) with 3GeV/c primary electrons.

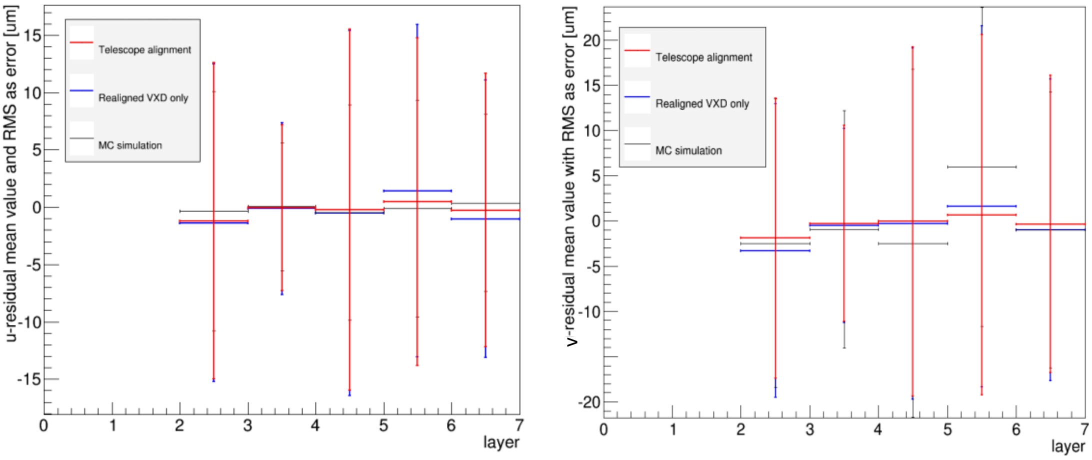


Figure 7.10: Run 470: the mean value and RMS (shown as error bar) of residual distribution for each VXD sensor in U (left) and V (right). The RMS values are consistent for all cases, two versions of alignment and the MC simulation.

8 Results of Belle II VXD Alignment

The fitting and alignment procedure developed is applied to the complete Belle II vertex detector simulation. All simulation results in this chapter skip the digitization and clusterization for VXD and only use Gaussian hit position smearing. The developed fitting tools allows to include CDC hits in the GBL trajectory, see Fig. 8.1, which however are not part of the alignment procedure. The basic distribution evaluating statistical properties of the fit within VXD only are shown in Fig. 8.2, where for tracks in VXD and CDC, the results are shown in Fig. 8.3. In both cases, the mean value of Chi/NDF is close to unity and the p-values are close to uniform.

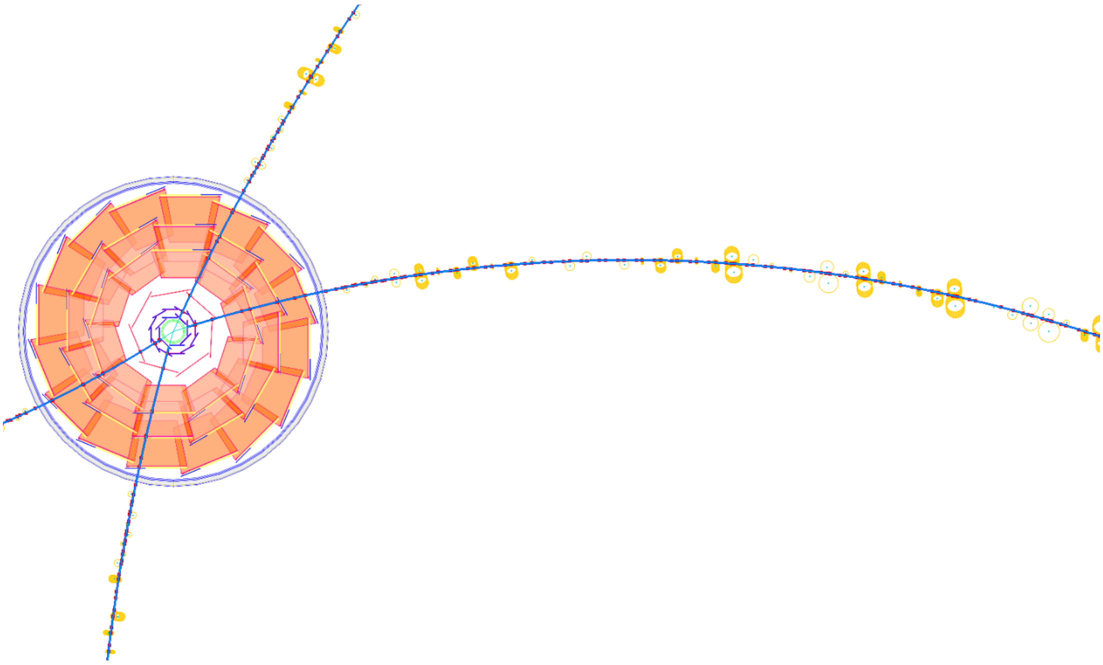


Figure 8.1: Four muon tracks fitted by GBL starting from interaction point a continuing from VXD to CDC in basf2 event display. This simulation used thick scatterers, which are visible as “double” points clearly in the CDC volume (CDC itself not shown). The yellow circles show surfaces of constant drift time to a wire.

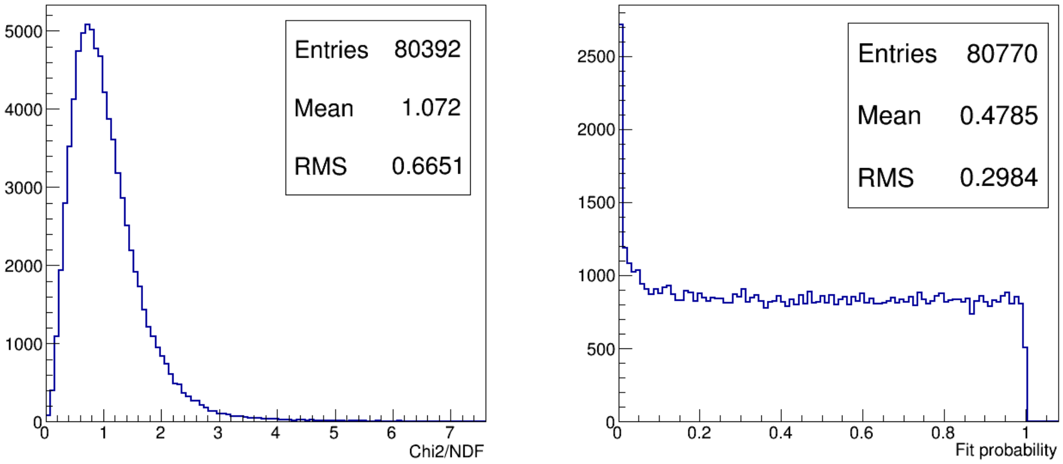


Figure 8.2: Left: The Chi2 divided by degrees of freedom for MC simulation using Gaussian error smearing (no digitization and clusterization) in full Belle VXD. Right: p-value distribution for the same set of tracks.

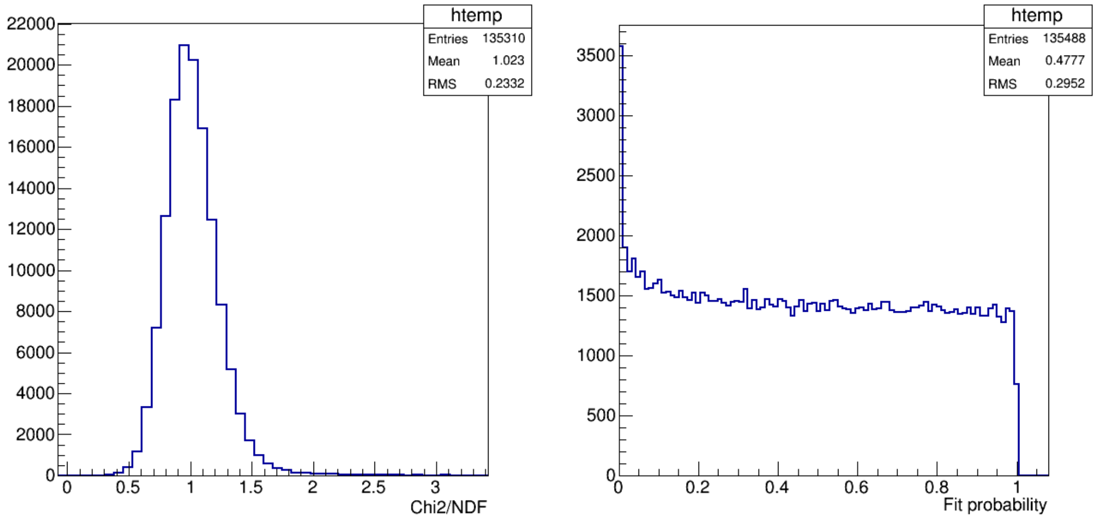


Figure 8.3. The distribution of tracks' Chi2/NDF and p-values for tracks in VXD and CDC. The average number of degrees of freedom for a track is more than 100 if using CDC, which is reflected in almost Gaussian distribution of Chi2/NDF.

For investigation of the alignment, several data samples of tracks were simulated. These samples combine different conditions and particle sources available for the developed alignment procedure:

- 80k muon pairs with default particle gun setting (uniform angular coverage and momentum generation from 0.05 to 2 GeV/c) in magnetic field,
- 168k muon pairs from interaction point with momentum 2 to 4 GeV/c in magnetic field,
- 45k cosmic ray muons constrained to go near interaction point without magnetic field,
- 31k cosmic ray tracks without constraints illuminating the detector more uniformly without magnetic field,
- 55k cosmic ray tracks within magnetic field

- 50k muon pairs coming from interaction point and continuing from VXD to CDC with momentum in range of 3 to 4 GeV/c, and
- 8k tracks for a “control sample” at fixed momentum of 2 GeV/c with fixed vertex at detector origin. This sample is not used for alignment. It is used to check the computed alignment and mainly vertex resolution.

The sample of CDC tracks works as an external reference and thus it can replace the constraints and sensor fixing. A small systematic study is done in the ideal geometry, where tracks are fitted and passed to the alignment procedure. The corrections should be obviously zero. In fact the alignment procedure will slightly misalign the ideal geometry and the goal is to reduce any systematic bias introduced by the alignment procedure. The sample of track was extended until the bias is reduced. Combining all the samples, the results for parameter corrections are show in Fig 8.4. Apparently the most problematic parameter is the w-shift in outer SVD layers, where the correction are largest.

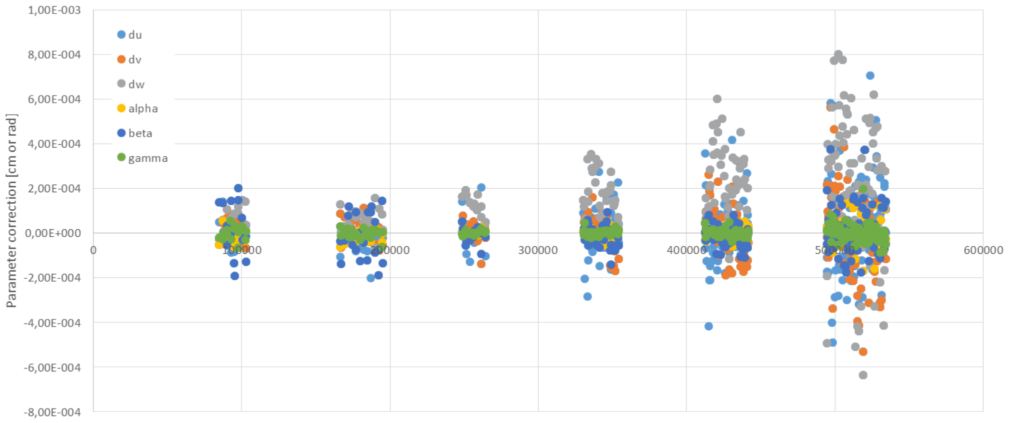


Figure 8.4: Corrections to alignment parameters computed by Millepede in ideal geometry. The horizontal axis corresponds to internal number of the parameter, separated in groups for each layer. The vertical axis show value of the computed correction, where the units correspond to cm for shifts and rad for angles. The parameter types are distinguished by color, showing the most problematic parameter being w-correction in outer SVD layers.

With such tracks sample and having checked that large systematic bias is not present in the determined alignment, it was possible to proceed to misalignment simulation. All the samples are fitted again in geometry, where the sensors are randomly misaligned in range of:

- 50 µm in U and V,
- 20 µm in W, and
- 1 mrad for all angles.

The new samples are aligned with Millepede and this procedure is repeated once again. A comparison of ideal, misaligned and aligned geometry after the second iterations is presented. Obviously the alignment is still far from perfect as can be seen from Fig. 8.5. The improvement in resolution is however large, as demonstrated in residual distributions in Fig. 8.6. The control sample allows to investigate vertex resolution in the aligned geometry, as shown in 8.7. The vertex resolution is significantly improved when compared to the misaligned geometry but a small bias is clearly visible in the aligned geometry.

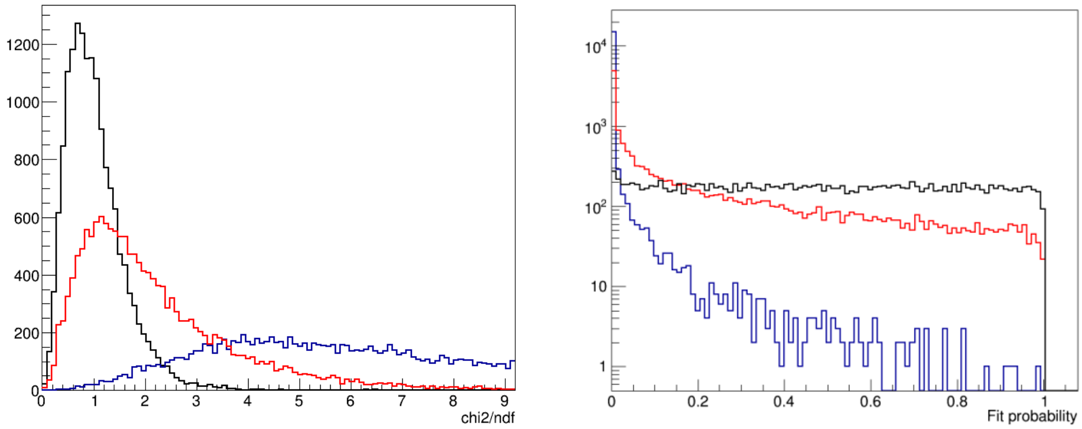


Figure 8.5: Chi2/NDF and p-value distributions for control sample tracks in ideal (black), misaligned (blue) and aligned (red) geometry.

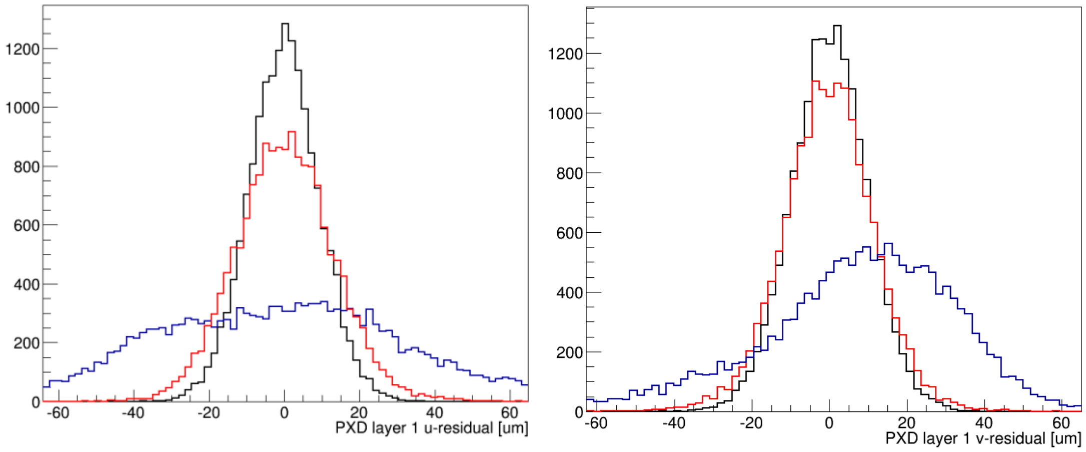


Figure 8.6: Residual distributions in first PXD layer in U (left) and V (right) for control sample tracks in ideal (black), misaligned (blue) and aligned (red) geometry.

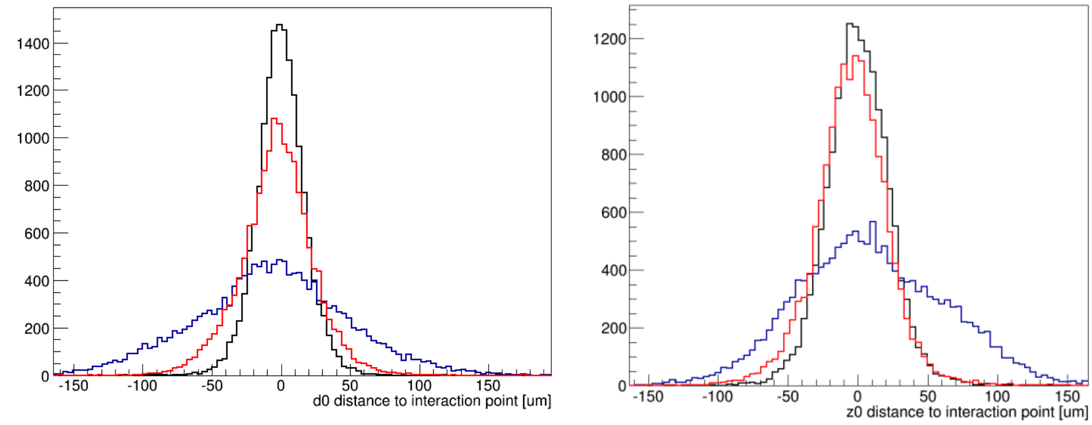


Figure 8.6: Vertex resolution in helix parameters at point of closest approach to beam line for control sample tracks in ideal (black), misaligned (blue) and aligned (red) geometry. Left: signed radial distance from interaction point d_0. Right: z_0 distance from interaction point in z-direction.

Discussion

The presented alignment procedure utilizes approach used also in other experiments, particularly the usage of Millepede II and General Broken Lines together. In this thesis, these tools are tightly integrated into the basf2 framework and especially the fitting with GBL utilizes several novel approaches not used so far, like advanced material description with thick scatterers. The development was done in close cooperation with Claus Kleinwort (DESY), a Millepede/GBL expert. It widely benefits from the approach of GENFIT toolkit which separates the fitting algorithms from detector specifics thanks to which the software is easily extensible.

The alignment is a task complicated by many facts. Usually position of large number of detectors has to be computed, while often degrees of freedom which can be hardly determined are present. It is also necessary to combine tracks recorded under different conditions for best results. In this thesis the alignment procedure was first applied to the beam test as a simple test case and showed satisfying results, with precision better than 3 um and 1 mrad for most parameters. Unfortunately combination of various track samples was not possible because the sensor position are not enough stable when magnetic field is turned on/off.

The alignment procedure is then applied to the complete vertex detector, with 1266 free alignment parameters to be determined (six per each sensor). The input track sample was extended by cosmic ray tracks with/without magnetic field and also with tracks in CDC. No significant systematic bias is observed for parameter correction determined in ideal geometry, but a misalignment study clearly shows the alignment procedure is still far from perfect. The problems are concentrated in particular areas of the detector, especially in outer SVD layers. Nevertheless the goal of this thesis was to offer a working alignment procedure. This goal was achieved and further investigations beyond the scope of this thesis are needed to “learn” how to align the Belle II vertex detector properly.

There is also plenty of space for improvements of the alignment procedure itself. It would benefit from mass and/or vertex constrained tracks from particle decays as well as from separation of the geometry into hierarchical structure.

The Testbeam package allows for full MC simulation and reconstruction of the DESY beam test and can be easily used in future beam tests. Unfortunately telescope data could not be fully used for precision alignment. Nevertheless the software is ready for telescopes simulation, reconstruction and alignment.

Conclusions

The complete geometry of the DESY beam test was developed and maintained during the experiment by the author. It is part of the Testbeam package of the basf2 framework where the author contributed also in other areas, from extending MC tracking tools, writing masking modules, preparation of magnetic field and general maintenance to development of the fitting and alignment modules. The package, utilizing common tools from the framework, is capable of full MC simulation as well as online reconstruction of beam test data. The EUDET telescopes are fully supported offline.

While the developed Testbeam package uses existing framework tools as much as possible, the GBL interface and the alignment procedure has been newly developed by the author from the scratch, using standardized tools of HEP experiments, Millepede II and GBL. The GBL interface is made as experiment-independent and is able to take into account arbitrary material distribution, detector geometry or inhomogeneous magnetic field. The alignment procedure is successfully applied to the DESY beam test, fully working with/without magnetic field and supporting the EUDET telescopes. Results of the beam test including the alignment are published [27].

The developed alignment tools are used to perform initial study of vertex detector alignment. A set of simulated track samples utilizing data with and without magnetic field, cosmic ray and collision tracks or tracks continuing to CDC is generated. A small systematic study showed that the alignment procedure is in principle working, without introducing significant bias to determined parameters. As next step the misalignment was simulated and using the generated samples of tracks, the alignment procedure is used in two iterations to correct the misaligned geometry. Obtained results show significant improvement in detector resolution, which is compared to the case of ideal geometry. This comparison indicates the alignment procedure still can be improved, utilizing more information from tracks, larger and wider sample of tracks or introducing hierarchy constraints in the alignment procedure.

The developed tools should allow for easy addition of new features. All developed software is part of official basf2 distribution and the GBL interface is available as Open Source in the GENFIT 2 toolkit.

References

[1] ATLAS Collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC,* arxiv:1207.7214 [hep-ex], Aug 2012

[2] The Nobel Foundation. *The Nobel Prize in Physics 2008*. Nobelprize.org [online]. http://www.nobelprize.org/nobel_prizes/physics/laureates/2008 (cit. 2014/06/30)

[3] DOLEŽAL, Zdeněk. Super KEKB and Belle II: Status of the KEK Super B Factory. arxiv: 0910.0388v1 [hep-ex]. 2 Oct 2009

[4] T. Aushev et al. Physics at Super B Factory, arxiv: 1002.5012v1 [hep-ex], 26 Feb 2010

[5] Z. DOLEŽAL, S. UNO et al. Belle II Technical Design Report. arxiv: 1011.0352v1 [physics.ins-det], 1 Nov 2010

[6] Max-Planck-Institut fur extraterrestrische Physik und Max-Planck-Institut fur Physik. *Research Activities* [online]. Munchen: MPI Halbleiterlabor, Juni 2007 (cit. 2014/06/30)

[7] DESY web page. *Test Beams at DESY* [online], http://testbeam.desy.de/ (cit. 2014/06/30)

[8] Diener, R., DESY. *PCMAG Solenoid Upgrade and DESY Testbeam Area T24/1* [online], AIDA Annual Meeting, Apr 10, 2013, https://indico.cern.ch/event/209429/session/12/contribution/122/material/slides/0.pdf

[9] BRUN, R. et al. ROOT, A Data Analysis Framework [online], http://root.cern.ch (cit 2014/06/30)

[10] GEANT4 Collaboration. *Geant4: A toolkit for the simulation of the passage of particles through matter* [online], http://www.geant4.org (cit. 2014/06/30)

[11] MOLL, A. The Software Framework of the Belle II Experiment, 2011 J. Phys.: Conf. Ser. 331 032024, 2010

[12] C. Höppner, S. Neubert, B. Ketzer, S. Paul, A novel generic framework for track fitting in complex detector systems, Nucl. Instr. and Methods A, 620 (2010), 518-525

[13] M. Stoye, Calibration and Alignment of the CMS Silicon Tracking Detector, Dissertation, Universität Hamburg, 2007

[14] DESY, Belle II: Tracker Alignment & Calibration [online], http://belle2.desy.de/e101530/e111982/ (cit. 2014/05/24)

[15] V. Blobel, C. Kleinwort, F. Meier, Fast alignment of a complex tracking detector using advanced track models, Computer Physics Communications, 182 (2011) 1760-1763

[16] V. Blobel, Millepede II – Draft Manual, University of Hamburg, 2007, available online: http://www.desy.de/~kleinwrt/MP2/doc/html/draftman_page.html (cit. 2014/07/01)

[17] V. Blobel, Millepede II, https://www.wiki.terascale.de/index.php/Millepede_II (obtained 2 September 2013)

[18] C. Kleinwort, *H1 Alignment Experiences*, Hamburg, available online: http://webh06.cern.ch/lhc-detector-alignment-workshop/Proceedings/h1rep.pdf

[19] CMS Collaboration, Alignment of the CMS tracker with LHC and cosmic ray data, arXiv:1403.2286 [physics.ins-det], 2014

[20] J. Alison, *The Road to Discovery: Detector Alignment, Electron Identification, Particle Misidentification, WW Physics, and the Discovery of the Higgs Boson*, Dissertation, 2012

[21] C. Kleinwort, General Broken Lines as advanced track model, Draft Manual, DESY, 2011

[22] Physics at the Terascale, *General Broken Lines* [online], https://www.wiki.terascale.de/index.php/GeneralBrokenLines (cit 2014/07/01)

[23] C. Kleinwort, *General Broken Lines as advanced track fitting method*, NIM A, 673 (2012), 107-110

[24] University of California, *Particle Data Group* [online], http://pdg.lbl.gov (cit. 2014/06/30)

[25] Claus Kleinwort, private communication

[26] Ch. Grefe, *Magnetic Field Map for Large TPC Prototype*, Diplomarbeit, Hamburg, 2008

[27] T. Bilka, G. Casarosa, R. Fruhwirth et al. *Demonstrator of the Belle II Online Tracking and Pixel Data Reduction on the High Level TriggerSystem*, arxiv:1406.4955v1 [physics.ins-det], 19 Jun 2014

[28] Karimaki V. et al.: *Sensor alignment by tracks. Computing in High Energy and Nuclear Physics*, 24-28 March 2003, La Jolla, California; arxiv: physics/0306034

[29] C. Kleinwort, Track fitting and alignment for a TPC with General Broken Lines, DESY, 2014, available online: http://www-flc.desy.de/lcnotes/notes/LC-TOOL-2014-008.pdf

Appendix A

GBL Interface Implementation and Usage

This section describes the implementation and basic usage of the developed GBL interface for GENFIT2.

The fit needs an initial track seed parameters for initial extrapolation. The initial seed is propagated to 1st measurement. If the hits in the track might not be sorted properly, an initial extrapolation which sorts the hits is possible. Sorted track can be propagated among measurements to create a reference trajectory. At each measurement, a plane (possibly virtual) is constructed to which a previous state is propagated.

From the extrapolation steps, which contain information on distance traversed in each particular material with its radiation length, the parameters of a thick scatter between each two measurements are obtained. The integrations (4.17) are performed, taking into account that the material radiation length in each step is constant. Equivalent scatterers are computed using (4.21). If only thin scatterers are simulated, all the scattering variance is attached to a measurement plane:

$$s_2 = 0, \qquad \theta_1^2 = \theta^2, \qquad \theta_2^2 = 0,$$

and a scatterer is not placed between measurements. Similarly a scatterer is not inserted if the computed covariance is too low to be inverted. A scatterer is placed as ThinScatterer object to a point in the track with its variance. If thick scatterers are simulated, the state from measurement is propagated to calculated position of the scatterer at s_2. The local coordinate system is constructed now from the co-moving system of the track. This is done by taking the plane from extrapolation. A new point with a thin scatterer is added to the track, which does not have a measurement and is a pure scatterer.

The plane of pure scatterer is furthermore not changed and therefore after update from a fit, it might no more coincide with co-moving frame of the track. Similarly the local system at a point with a measurement does not coincide with track. For this reason, formula (4.4) is always used to provide full scattering covariance matrix projected to the (virtual) plane to be diagonalized within GBL.

At each point with a GblFitterInfo is attached. It stores a Jacobian and reference state from extrapolation and scans for measurements at the point and constructs first raw measurement found. From constructed measurements at plane, only that with highest weight is used. This limitation comes from the fact, that there cannot be more measurements at a point in GBL trajectory. For corresponding strip measurements, one has to initially combine them if possible in two-dimensional hits before processing. For wire measurements, the left – right ambiguity must be resolved before the fit by setting weight=1 for the measurement which has to be fitted.

Once all fitter info are constructed, they are collect from the track, a GBL trajectory is constructed and fitted. The results of the fit are then again passed to the fitter info to update the states, covariance, etc. This is the 1st external iteration. The trajectory can be however written to Mille binary even before, once the reference state is propagated. More external iterations can be performed. Because new prediction might result in different virtual planes (e.g. in wire chamber), predictions need to be extrapolated to these new planes before next iteration. For this purpose, it is possible to enforce re-extrapolation of predicted states (now containing kinks) and recalculation of Jacobians after an external iteration. Updated Jacobians might improve fit results also for planar measurements, if large corrections were computed by GBL fit.

To establish connection between the fit and alignment derivatives, an **ICalibrationParametersDerivatives** interface is included in the extension. During collection of points, the current state of the track is passed to a function declared in this interface which should calculate the derivatives. Labels for the global derivatives also have to be provided by actual implementation. Each raw measurement, for which additional, global or local, derivatives should be attached to a GBL point, has to implement this interface. For example, to declare an alignable planar measurement:

```
class AlignablePlanar : public genfit::PlanarMeasurement,
public genfit::ICalibrationParametersDerivatives
{
  // ... implementation of standard planar measurement
  // Define these functions to return matrix of derivatives and vector of labels
  TMatrixD derivatives(StateOnPlane* state);
  vector<int> labels();
};
```

In implementation, for alignment in local plane coordinates for silicon sensors, one can use the 2×6 nonzero matrix (3.25). The labels then have to identify each plane and corresponding rigid body alignment parameter.

For the end-user, GblFitter can be used as any other fitter:

```
using namespace genfit;
using namespace std;
using namespace gbl;

Track* track;
// … track construction
// … or retrieval from datastore
GblFitter fitter;
// Process for all track representations
fitter.processTrack(track);
```

Several optional parameters can be used to control the fitting. It has be set to the fitter before track processing:

```
fitter.setOptions(internalIterations, enableScatterers, enableIntermediateScatterer,
externalIterations, recalcJacobians);
```

Follows an explanation of the various options:

- **internalIterations** is a string setting the internal down-weighting for the GBL fit, usually an empty string meaning no down-weighting is sufficient, see the GBL manual.
- **enableScatterers** enables placing thin scatterers in the trajectory. If set to false, no multiple scattering effects will be added
- **enableIntermediateScatterer** switches between simulation of thick scatterers between measurements (true) and only thin scatterers at measurement planes (false). See the implementation bellow.

- **externalIterations** is integer defining number of times, the trajectory is fitted by GBL and updated with results. Usually one iteration is sufficient.
- **recalcJacobians** defines number of time Jacobians will recalculated and states re-extrapolated with results of the fit after external iteration. If set to zero, Jacobians are not recalculated. If set to i, the update is done after i^st^ external iteration.

A processed track then can be asked for results of the fit. The most common need is access to fitted prediction. However there can be two predictions at a point which differ in slopes for a point with scatterer. At the same time, there is no biased/unbiased prediction at a point like for Kalman fitters. For this reason and to be consistent in the interfaces, the prediction before a kink is called unbiased and after the kink, the prediction called biased. This is only reinterpretation of otherwise unused parameter. It does not mean the prediction before the kink is unbiased in any way. The two fitted states and their covariance therefore can be retrieved from a processed track as follows:

```
int id = 1; // index of point on track (includes scatterers)
AbsTrackRep* rep = track->getCardinalRep(); // default track representation
TVectorD stateBeforeKink = track->getFittedState(id, rep, false).getState();
TVectorD stateAfterKink  = track->getFittedState(id, rep, true).getState();

TMatrixDSym covarianceAfterKink  = track->getFittedState(id, rep, true).getCov();
```

Note the states can only differ in 2nd and 3rd column of the five – dimensional state of the track. The track functions to get the fitted states are shortcuts to methods of special helper classes, **GblFitterInfo**, which for each point store fitter information. Therefore the position residual can be retrieved from any of these predictions:

```
TVectorD residual =
               track->getPoint(id)->getFitterInfo(rep)->getResidual().getState();
```

Covariance matrix retrieved for the residual is a result of the GBL fit and is diagonal. A point should be checked if it has a measurement, otherwise getResidual(…) will retrieve zero 2D vector and 2x2 covariance.

The results of the fit can be retrieved from GblFitStatus attached to a processed track:

```
double ndf  = track->getFitStatus(rep)->getNdf();
double chi2 = track->getFitStatus(rep)->getChi2();
```

A processed track can be now easily be written to Mille binary file. The GBL trajectory is constructed from points collected along the track by the fitter

```
// Create new mille binary
MilleBinary file("tracks.mille");
// "tracks.mille" is recreated

// ... process tracks ...

bool curvature = true; // Fit curvature or not ?
// Construct GBL trajectory from a track fitted by GblFitter
// (See GBL manual)
vector<GblPoint> points(fitter.collectGblPoints(track, rep));
GblTrajectory traj(points, curvature);
// Output to Mille binary
traj.milleOut(file);
```

The constructed GBL trajectory can also be again fitted separately by using the standard GBL implementation:

```
// degrees of freedom, chi squared
int ndf; double chi2, lostWeight;
traj.fit(chi2, ndf, lostWeight);
```

Appendix B

Geometry XML example

This example demonstrates the tree structure of geometry XML, how volumes are defined, volume subtraction and how to make create an active sensor. Note that this example only shows the part of XML specific to VXDTBCreator. For example the magnetic field is defined higher in the hierarchy and maintained by different creator class. A complete, but much more complex example can be found in the data folder in the package. The resulting geometry created by the example is show in Fig. B.1.

```xml
<DetectorComponent name="simple_beam_test">
  <Creator library="testbeam">VXDTBCreator</Creator>
  <Content>
    <!-- This sets the material used if none is specified for a volume -->
    <DefaultMaterial>Air</DefaultMaterial>
    <Alignment>
      <!—Additional transformation w.r.t. to placement bellow -->
      <Align component="1.1.1">
        <du unit="cm">0</du><dv unit="cm">0</dv><dw unit="cm">0</dw>
        <alpha unit="rad">0</alpha><beta unit="rad">0</beta><gamma unit="rad">0</gamma>
      </Align>
    </Alignment>

    <Volumes>
      <Volume name="open_box">
        <Material>Air</Material>
        <!-- Box shape is default with 3 params.
    Tube has parameters: InnerRadius, OuterRadius, Hz (half length in z)-->
        <Shape>Box</Shape>
        <!-- Box is default shape, you do not have to state that -->
        <HalfX unit="mm">100</HalfX>
        <HalfY unit="mm">100</HalfY>
        <HalfZ unit="um">100</HalfZ>
        <!-- Boolean operation: subtract a box to make hole for sensor -->
        <Subtract>
          <Shape>hole_in_box</Shape>
          <HalfX unit="mm">99</HalfX>
          <HalfY unit="mm">99</HalfY>
          <HalfZ unit="mm">100</HalfZ>
          <DZ unit="mm">1</DZ>
        </Subtract>
        <!-- Now add an active sensor into the box -->
        <Volume name="test_sensor">
          <!-- Place sensor to z=10cm and rotate around its z-axis -->
          <DZ unit="cm">10</DZ>
          <Phi unit="deg">90</Phi>
          <!-- Identify sensor as alignable component-->
          <AlignComponent>1.1.1</AlignComponent>
          <Material>Si</Material>
          <HalfX unit="mm">10</HalfX>
          <HalfY unit="mm">20</HalfY>
          <HalfZ unit="um">25</HalfZ>
          <!-- Make the volume (only box) active by settings its VXD id -->
          <Active>
            <SensorID>1</SensorID>
            <Ladder>1</Ladder>
            <Layer>1</Layer>
            <DetectorType>TEL</DetectorType>
            <stepSize unit="um">5</stepSize>
            <!-- define the number of pixels in rphi and z direction -->
            <pixelsR>1152</pixelsR>
            <pixelsZ>576</pixelsZ>
          </Active>
        </Volume>
      </Volume>
    </Volumes>
  </Content>
</DetectorComponent>
```

Listing A.1 Demonstration of some capabilities of the geometry creator on simple XML defining a geometry with aluminum box with one side open and a telescope sensor placed inside and rotated.

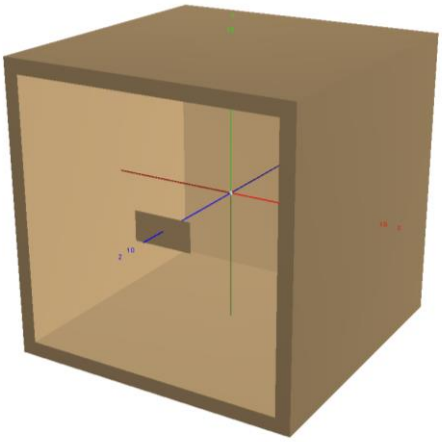


Figure B.1 The geometry in ROOT viewer created by Listing A.1 converted by the VXDTBCreator.