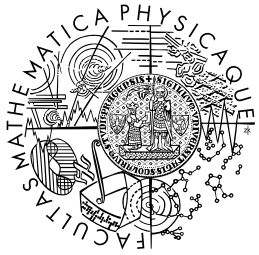

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Samuel Šitina

Calibration of the Belle II detector

Institute of Particle and Nuclear Physics

Supervisor of the bachelor thesis: Mgr. Radek Žlebčík, Ph.D.
Study programme: Physics
Study branch: General Physics

Prague 2022

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

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

In ..... date ....

Author's signature

I would like to thank my supervisor Radek Žlebčík for all the patience and all the valuable advice and comments. A big thanks also to my parents and the rest of my family who supported me at all times.

Title: Calibration of the Belle II detector

Author: Samuel Šitina

Department: Institute of Particle and Nuclear Physics

Supervisor: Mgr. Radek Žlebčík, Ph.D.

Abstract: This thesis proposes an improved method for calibration of the Belle II detector by introducing an algorithm that determines the optimal length of the calibration intervals. The calibrated quantity is the centre-of-mass energy of the e^+e^- collisions and is determined from the invariant mass of $e^+e^- \to \mu^+\mu^-$ events. For a given number of the calibration intervals, a method to find optimal positions of the interval boundaries is developed. Furthermore, a cross-validation technique is applied to choose the total number of intervals such that the data are not overfitted or underfitted. The performance of the algorithm is tested on the data from the Belle II experiment and the results together with further possible improvements are discussed.

Keywords: Calibration, CP violation, Belle II

Contents

Introduction

When it comes to our understanding of how the world of elementary particles works, the Standard Model as we know it today can provide brilliant, well-developed, and satisfying answers to good amount of questions physicists were able to ask so far [1]. The science, however, is never satisfied so easily and always digs deeper until some law is broken, some new particle is observed or some experiments do not match the theory as they should.

A good way to study our universe and its laws is by looking for some kind of symmetry. In particle physics this can mean studying charge conjugation (C), space mirroring (P - for parity), time-reversal symmetry (T), and others as well as their combinations [1]. Violation of some proposed symmetries can point us to more fundamental symmetries that need to be found.

In the second half of the 20th century, physicists started to be intrigued by this question related to symmetry: Is there a difference between the behavior of matter and antimatter? If not, why do we observe mostly matter rather than antimatter? And if yes, what is the difference? As mentioned in the paragraph above this can be described as a study of CP (a)symmetry [2].

The results - both experimental and theoretical - achieved since then were not insignificant and they let us build new experiments on solid ground. In 1964, by studying decays of neutral kaons, Val Fitch, James Cronin, and others discovered that the CP-violation indeed happens in the nature [3]. And even though the effect they observed was small, the conclusion that matter and antimatter actually follow different physical laws was inevitable . Fitch and Cronin were rewarded with the Nobel Prize in 1980 [4].

Theoretical works of Cabibbo (1963) and Kobayashi and Maskawa (1973) helped us understand how the CP-violation can appear within the Standard Model. Namely, they introduced what we nowadays call CKM matrix [5]. Even though it does not exhaust the cosmological problem of the unbalance of matter and antimatter, its importance has been undeniable.

The next milestone of studying CP-violation was building a collider so big that it could generate B mesons. Decay of B meson, or rather decay of pair of B meson and anti-B meson, provides a good ground for the study. With mass higher 5 GeV, B meson js one of the heaviest hadrons we know and its production in the amount that is significant for measurement of its decays is not at all trivial. The B-factories, as they are often called, can produce enough B meson pairs per second by colliding beams of electrons and positrons to see the effects Kobayashi and Maskawa described. There are also hadronic B-factories. These can have bigger production rate, but it also comes with complications because of a huge amount of "non-B events" [6].

The first generation of B-factories: KEKB with experiment Belle in Tsukuba, Japan [7], and PEP-II with experiment BaBar in California, United States [8], were successful - they confirmed the theory and because of this the gentlemen Kobayashi and Maskawa received a Nobel prize in 2008 [9]. Both Belle and BaBar observed the CP-violation in the system of B mesons [10, 11]. It is not possible to determine parameters of the CKM matrix only from Kaons, but together with these results it became possible.

The upgrade of Belle - Belle II - is located in the SuperKEKB in Tsukuba, Japan. It was launched in 2018 and as of today, the integrated luminosity is 430 fb^{-1} [12]. Despite not achieving the luminosity of BaBar (530 fb^{-1} [13]) its highly improved detector makes Belle II already competitive in some measurements. For example, at Belle II they were already able to measure the lifetime of D mesons with the best precision in the world [14].

To measure properties of B-meson and its decays with the highest precision, which is the main prospect of Belle II [15], it is important to use every single bit of information the detector provides. Namely, data analysis and the right calibration of the detector are vital parts of physical experiments with ambitions like Belle II, and only by not underestimating their importance something new can be brought to the world of science.

In this thesis, there will be provided an improved approach to the problem of the time dependence of calibrated properties. The important quantities (beam-constrained invariant mass and energy difference ΔE [2]) are both dependent on the centre-of-mass energy and it is therefore highly beneficial to know its value and its time-dependent deviation.

By now, the standard way of analyzing data from colliders and recognizing any systematic changes in centre-of-mass energies of collisions was to look at the data either in chunks covering time periods of the same length or by dividing the data at places that are reasonable for a particular case.

These methods of course have their benefits - one of them being their simplicity. In most cases, it is possible to spot major fluctuations by simply looking at histograms. However, the simplicity may easily become a disadvantage, if we want to know more. One of the problems of covering a time period of the same length is that it is not straightforward what the length should be. When they are too short, there might be a statistical inaccuracy caused by the data sample being too small. When they are too long there is a risk of not recognizing a sudden change in the centre-of-mass energy.

A better way to choose the time periods is not to be bound by the same length of the intervals, but rather directly take into account the character of the data we want to analyze. This thesis introduces an algorithm that chooses the ideal intervals in a way that the final description of the data is statistically more accurate.

The first part of this thesis is an introduction to the Belle II experiment specifying what quantities are measured and how are the measurements made.

The core of this thesis is a presentation of the algorithm and its results on data from Belle II. The algorithm itself is a combination of simple and well-known methods from statistical analysis, dynamic programming and machine learning. If one can apply these correctly to the particular problem of calibration, the data will provide information that is otherwise hidden to the human eye.

1. Experiment Belle II

The experiment Belle II was developed after the success of experiment Belle, which managed to meet all the expectations physicists hoped it would [16]. It studies B mesons which are in general quite hard to study as they are heavy thanks to the heavy b quark and therefore unstable like most of the particles - a mean lifetime of neutral B meson is $(1.519 \pm 0.004) \times 10^{-12}$ s [17]. Belle ran from 1998 to 2009 and then was shut down with prospects of building something even more powerful. Upgrade of the old technology consisted of replacing collider KEKB with SuperKEKB and building a more sensitive detector [18, 15]. In the following paragraphs, there will be explained what is SuperKEKB, how it works and why it was necessary to replace its predecessor.

1.1 SuperKEKB

SuperKEKB is a 7 GeV electron – 4 GeV positron double-ring collider [18]. KEKB was designed to perform collisions of positrons and electrons at a center-of-mass energy of 10.58 GeV [19]. The advancement of technology made it possible to build even better collider than KEKB, which was setting the records during its time [16]. In ten years of measurements at KEKB, the overall integrated luminosity reached the value of 1041 fb^{-1} which is ten times more than the initially required value [16].

SuperKEKB runs on $\Upsilon(4S)$ resonance, which helps to produce a very clean sample of $B^0\bar{B}^0$ pairs in a quantum correlated state [2]. In the Figure 1.1, there is a scheme of SuperKEKB showing the overall configuration of the 3-km-long accelerator.

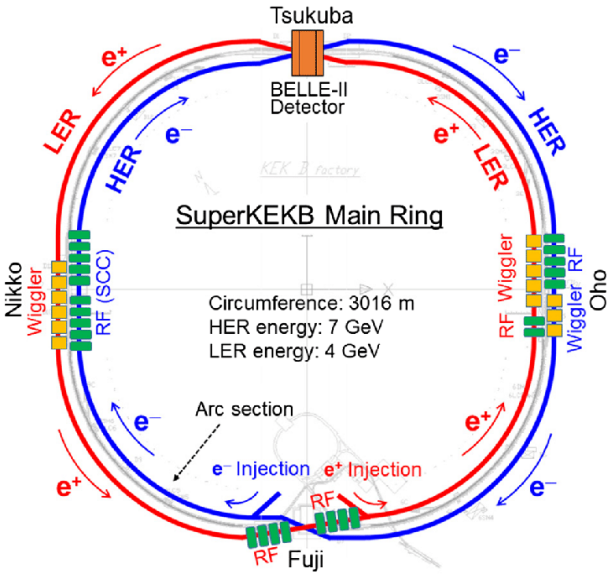

Figure 1.1: The schematic of SuperKEKB accelerator [20].

When it comes to upgrading colliders, there are two important parameters - energy of beams that are colliding and luminosity. The most of the data at Belle

II will be collected at the $\Upsilon(4S)$ resonance, which is just above the threshold for B-meson pair production [2]. The hadronic cross-section of e^-e^+ collisions can be seen in the Figure 1.2.

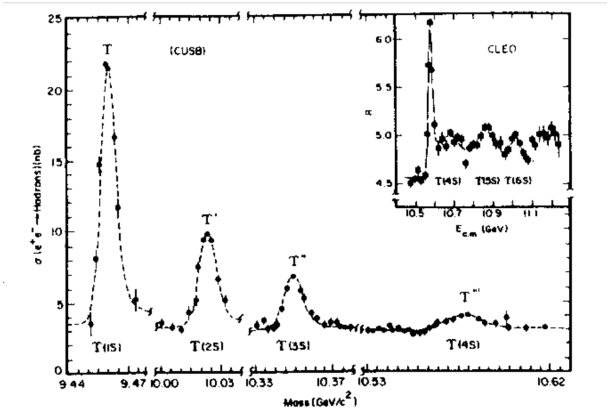

Figure 1.2: The hadronic cross-section of e^-e^+ collisions [21].

Having said that, it is clear that what needed to be upgraded was the luminosity. Firstly, let us quickly explain what is the concept of luminosity.

1.1.1 Concept of luminosity

Luminosity L is [22]:

> the proportionality factor between number of events per second $\mathrm{d}R/\mathrm{d}t$ and the cross section σ_p:

$$\frac{\mathrm{d}R}{\mathrm{d}t} = L \cdot \sigma_p \tag{1.1}$$

The number of events R is dimensionless, cross section is best described in units of m^2 or cm^2 and unit of time is a second. The most commonly used unit of luminosity is $\mathrm{cm}^{-2}\mathrm{s}^{-1}$.

Integrated luminosity is the integral of luminosity with respect to time:

$$L_{int} = \int_T L \,\mathrm{d}t \tag{1.2}$$

and is often expressed in the non-SI units fb^{-1} (femtobarn^{-1}), where $1\mathrm{b} = 100\ \mathrm{fm}^2$.

This concept is especially useful when working with events that can be classified as rare and the production of B mesons falls exactly in that category.

1.1.2 The thrive for precision

The precision of results is strongly tied with how many events are observed. The bigger the experimental data the smaller the statistical error. The most important changes resulting in the increase of the luminosity were a reduction of the beam size at the point of collision by a factor of 20 (from about 1 µm to 50 nm) and an increase in the currents by a factor of 2 compared to the KEKB [2]. This is known as a 'nano-beam' scheme, and was invented by P. Raimondi for the

Italian super B factory [2, 23] which was never built. The design luminosity of SuperKEKB is 8 $\times 10^{35}$ cm^{-2}s^{-1} which is around 40 times larger than the peak luminosity of its predecessor. Currently (June 2022), SuperKEKB holds a world record in luminosity - approximately twice the value of the record of KEKB from 2009 [24].

It is not only the amount but also the quality of the data that matters.

Most of the components of SuperKEKB come from KEKB [18]. The rest was either modified or replaced. For example, the beam energies were changed to 4 GeV in LER (low-energy ring) instead of the original 3.5 GeV and 7 GeV instead of 8 GEV in HER (high-energy ring) [18]. This was done to reduce losses due to so-called Touschek scattering in the lower energy beam [2].

The improvements also included increasing beam currents, upgrading the vacuum system, and many others. They are all described in [18].

1.2 Asymmetry of beam energy

SuperKEKB, like KEKB was, is energy-asymmetric [18]. That means that the colliding positron and electron do not have the same momentum in the laboratory frame of reference.

The motivation of this is practicality. In the case of symmetric beam energies, created pair of B mesons would not move and therefore they would both decay more or less in the same spot. The asymmetry ensures that they do indeed move and that therefore they do decay at different places and we are able to measure the difference. The velocity of $\Upsilon(4S)$ resonance is calculated in the following manner [25]:

$$\beta = \frac{\vec{P}}{E} = \frac{\vec{P_{\mathrm{HER}}} + \vec{P_{\mathrm{LER}}}}{E_{\mathrm{HER}} + E_{\mathrm{LER}}} \tag{1.3}$$

which after assuming head-on collisions gives us in natural units:

$$\beta = \frac{\vec{P}}{E} \approx \frac{E_{\mathrm{HER}} - E_{\mathrm{LER}}}{E_{\mathrm{HER}} + E_{\mathrm{LER}}} = \frac{7.007\ \mathrm{GeV} - 4\ \mathrm{GeV}}{7.007\ \mathrm{GeV} + 4\ \mathrm{GeV}} = 0.27 \tag{1.4}$$

Lorentz factor γ is then[25]:

$$\gamma = \frac{1}{\sqrt{1-\beta^2}} = 1.04 \tag{1.5}$$

and so called *boost factor*:

$$\beta\gamma = 0.28 \tag{1.6}$$

The value $\beta\gamma = 0.28$ is approximately two thirds of that in Belle [26]. Also, one can see that there is a measurable effect of the time dilation characterized by γ, but its magnitude is only 4%.

1.3 Belle II detector

To maximize the quality and quantity of the data, which will be gathered at Belle II throughout the years, it is crucial to pair a powerful collider with a sensitive

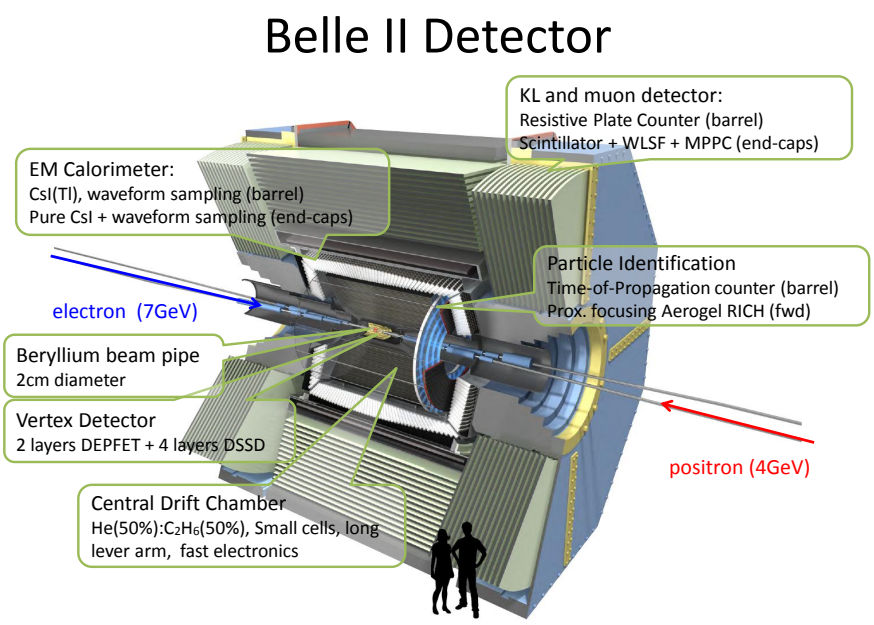

Figure 1.3: The schematic of Belle II detector [27].

detector. As in the upgrade of the collider, there was a solid foundation to build on thanks to the success of Belle [26]. In figure 1.3 there is a scheme of the Belle II detector and in the following paragraphs, there will be described its main components.

1.3.1 Interaction region

Because of the reasons stated in the previous section, the detector and the collider are asymmetric. Additionally, the two beams collide with a crossing angle of 83 mrad [26]. This helps to separate the beams more quickly which leads to a reduction of unwanted beam effects. One of the effects is often referred to as beam-induced background, which when studied carefully has more than one cause. The main causes are synchrotron radiation (SR) from HER upstream direction, backscattering of SR from HER downstream, Touschek scattering, and others. Knowledge of these effects helped to design the IP(interaction point) chamber with the ability to shade off some of them which led to clearer data. Unfortunately, these effects tend to be larger and therefore are harder to block with higher beam currents [26].

Also, thanks to the large crossing angle, it is now possible to put special quadrupole magnets closer to the IP to achieve more precise collisions of the beams [26].

1.3.2 Silicon Pixel and Strip Detector

The main role of the innermost parts of the Belle II detector is to measure the products of the decays of the pairs of neutral B mesons so it is possible to reconstruct them. The B mesons can not be detected directly, because they decay before reaching the detector. There are several challenges that needed to be over-

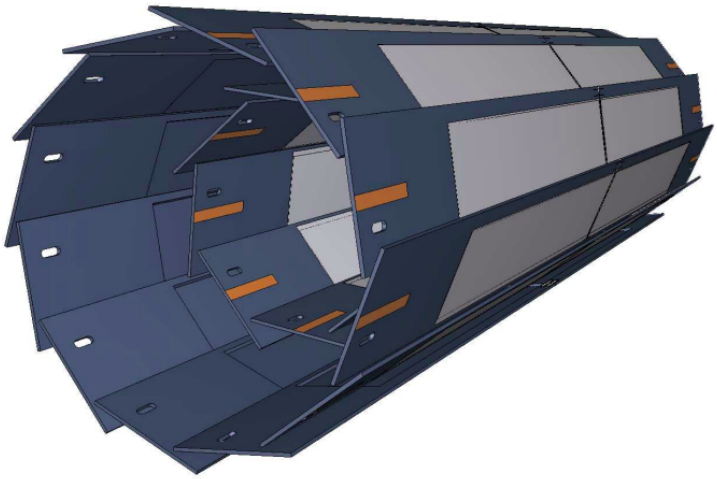

Figure 1.4: Schematic view of the geometrical arrangement of the sensors for the PXD. The light grey surfaces are the sensitive DEPFET pixels, which are thinned to 50 microns and cover the entire acceptance of the tracker system. The full length of the outer modules is 174 mm and the radius of the detector is 22 mm [26].

come to build such devices that could detect those particles without interacting too much with them and changing their energy and momentum. Compared to Belle, the new detector needs to cope for instance with higher event rates and larger radiation damage [28].

The two layers closest to the IP consist of the pixel detector (PXD). It works on a principle called DEPFET (fully depleted p-channel field-effect transistor) [29]. A more detailed description of DEPFET can be found in [28]. The scheme of PXD is shown in the Figure 1.4.

The next four layers are double-sided silicon strip detectors (SVD). These can besides helping to reconstruct B meson decays gather information about other decay channels including for example D meson decays.

Both PXD and SVD are close enough to the IP that there have to be cooling systems to keep the detectors operational for longer periods. This was also a great engineering challenge, as there is little space to work with and the materials that are used to support the cooling must not jeopardize the sensitivity of any part of the detector [26].

1.3.3 Central Drift Chamber (CDC)

This part of Belle II has three important roles. Firstly, it can reconstruct tracks of the particles with a non-zero charge. The fact that these tracks are curved thanks to a magnetic field makes it possible to precisely measure the momenta of these particles as well. Secondly, CDC can get information for identifying a particle by measuring the loss of energy in gas cells. With regards to the gas itself, there were attempts to find more suitable gas, but the results of the tests showed that the mixture used in Belle - 50% helium and 50% ethane - works the best [26]. The factors which decided whether gas was performing well are for example low radiation length, position resolution, energy loss resolution, the

low cross section for synchrotron radiation X-rays, and little radiation damage. Finally, it provides efficient and reliable trigger signals for charged particles.

1.3.4 Calorimeter (ECL)

The next layer of the detector consists of an electromagnetic calorimeter. The main purpose of the ECL is to detect photons (from decays of π^0s when $\pi^0 \to \gamma\gamma$ and from e) and it also plays a role in detecting kaons [2]. The energy range is from 20 MeV up to 4 GeV, which is great for distinguishing signal and background events only with mass distribution. On the other hand, high granularity is needed to distinguish two photons from high momentum π^0 [30].

The calorimeter contains 8736 CsI(Tl) crystals with total mass of 43 tons [26].

1.3.5 K_L^0 and muon detection (KLM)

Muon

The muon is a small elementary particle similar to an electron. It belongs to a group of particles that we call leptons and compared to electron it has the same charge but its mass is approximately 207 times greater. Larger mass lets muons travel longer distances through the detector because of smaller energy losses [31].

$\mathbf{K_L^0}$ (K-long)

K_L^0 is a weak eigenstate of neutral kaon K^0, which primarily decays into three pions. The "L" stands for long and points to the fact that it lives longer than the other kaon eigenstate. It can be detected in the KLM detector alone, the ECL alone, or both [26].

KLM detector

The KLM consists of active detector elements and 4.7-cm thick iron plates. A typical muon passes through all of the KLM layers, leaving a clean trail of hits to mark its passage. A typical K_L^0 will collide with an iron nucleus in the iron plates, resulting in a hadronic shower that can be detected [26].

1.3.6 Identification of muon and electron

In this thesis, it will be important to distinguish a muon from an electron with high certainty. Both of these are charged and therefore have similar properties. At Belle II, charged particle identification relies on likelihood-based selectors, where information from each PID (particle identification system) is analyzed independently [2].

Primary information for electron detection comes from ECL and KLM provides muon identification. The main feature for separating electrons from other charged particles (namely muons and pions) is the E/p value, where E is the energy measured in the ECL and p is the absolute track momentum [2]. For $p \geq 1$ GeV, there is sufficient distinction between electrons and other charged

particles, making it a useful parameter for electron identification. The separation between electrons and muons is good for muons with sufficient energy ($p > 0.3$ GeV - muons can reach the KLM).

1.4 Centre-of-mass energy

This thesis is focused on calibrating centre-of-mass energy. Centre-of-mass energy is the overall energy of the system in the centre-of-mass frame. In this case, there is an asymmetric positron-electron collision, so we need to calculate the invariant energy (or mass) of the system of positron and electron.

We know that in high-energy physics we are practically working with particles flying at the speed of light and so we need to use the relativistic formula for energy [32]:

$$E^2 = m^2 + p^2 \tag{1.7}$$

where p^2 is the squared size of 3-momentum and m is the rest mass. We are using *naturalunits*, where speed of light $c = 1$, reduced Planck constant $\bar{h} = 1$ so $E, pandm$ all have the same units [33]. Both the energy and the squared size of 4-momentum $P = (E, \vec{p})$ are conserved:

$$E_{\text{cms}} = P^2 = (E_1 + E_2)^2 - (\vec{p_1} + \vec{p_2})^2, \tag{1.8}$$

where E_1 and $\vec{p_1}$ are energy and momentum of colliding electron and positron. Then for invariant mass m_{inv}, where $m_{\text{inv}}^2 = P^2$ we get:

$$m_{\text{inv}} = \sqrt{(E_1 + E_2)^2 - p_1^2 - p_2^2 - 2\,|\vec{p_1}|\,|\vec{p_2}|\cos(\theta)} := \sqrt{s}, \tag{1.9}$$

where from the Equation (1.7) we get $|\vec{p_i}| = \sqrt{E_i^2 - m_i^2}$ and θ is the angle between the vectors of the electron and positron momenta.

After plugging in $\theta = \pi - 83$ mrad, $E_1 = E_{\text{HER}} = 7.007$ GeV, $E_2 = E_{\text{LER}} = 4$ GeV and $m_{1,2} = 0.511$ MeV we get $m_{inv} = 10.58$ GeV, which is approximately $\Upsilon(4S)$ resonance mass. In this particular case when $m_i \ll E_i \implies E_i^2 \approx p_i^2$ and $\theta \approx \pi$, the Equation (1.9) can be approximated by:

$$m_{inv} = \sqrt{4E_1E_2}. \tag{1.10}$$

This quantity can also be measured from the products of the collision. The instantaneous value should be always as close to $\Upsilon(4S)$ mass as possible so the maximum amount of B-mesons is produced. It is also needed for additional reconstruction of B mesons which will be described in the next paragraphs.

1.5 Beam-Constrained Observables

B mesons produced at the $\Upsilon(4S)$ resonance have well defined kinematics which is constrained by the mass of the $\Upsilon(4S)$ and by the properties of the beams [2]. To identify B mesons it is useful to define quantities which we call *beam-constrained*

mass m_{BC} and and energy difference ΔE which are calculated from B decay products [2]:

$$\Delta E = E^*_{\mathrm{beam}} - E^*_B \tag{1.11}$$

and

$$m_{BC} = \sqrt{E^{*2}_{beam} - p^{*2}_B} \tag{1.12}$$

where the * is notation for the rest frame, $E^*_{beam} = \sqrt{s}/2$ and $\vec{p^*_B}$ is the energy and momentum of B meson in the centre-of-mass frame. These are calculated as a sum of energies(momenta) of B meson decay products. For a correctly reconstructed B meson decay, the true values would be $\Delta E = 0$ and $m_{BC} = m_B$ [2].

The invariant mass from the Equation (1.9) is still needed to be measured and there are two main ways to do that.

1.6 Measuring the E_{cms}

At Belle II, there are two methods used for measuring E_{cms}. The first method studies decays $B \to D\pi$ and the second method is focused on $e^-e^+ \to \mu^-\mu^+$ events [34]. Each of these has its advantages and disadvantages so let us briefly mention the most important ones.

1.6.1 The method of hadronic decays of B mesons

The peak of the histogram when measuring E^*_B is quite narrow and since E^*_B is half of E_{cms}, the overall error is relatively small. Another reason for using this method is that it provides an insight into the energy spread not just mean value.

On the other hand, the cross section of $e^-e^+ \to B^0\bar{B}^0$ depends on the energy with the peak in 10579.4 MeV and for bigger or smaller values of energy cross section decreases. Because of this, E_{cms} has a tendency to have value closer to$\Upsilon(4S)/2$ and does not correspond to real $E_{\mathrm{cms}}/2$.

1.6.2 The method of $e^-e^+ \to \mu^-\mu^+$

First of all, since events $e^-e^+ \to \mu^-\mu^+$ are more common we have a bigger data sample to work with compared to the method of hadronic decays. That allows us to work with smaller calibration intervals with smaller statistical error. Also there is no energy-dependent bias as in the hadronic method.

On the contrary, the peak of the histogram from this method is much wider. What is more, measuring the momentum of a muon is dependent on the strength of the magnetic field, which appears to be quite hard to determine precisely. As we will see, that leads to a measured value of E_{cms} that differs from the real value.

In this thesis, we will be working solely on this method. The next section will show what is the typical distribution of di-muon invariant mass and what we can do to study its evolution.

1.7 Example of calculating centre-of-mass energy from $\mu^+\mu^-$ invariant mass fit

Let us now use the actual data from Belle II to show what it means in practice to compute centre-of-mass energy from detected muon pairs. The muon pairs are produced in the process $e^-e^+ \to \mu^-\mu^+$ which has a big cross section and so we detect lots of these events. When reconstructing the event, we need to use the Equation 1.9, but this time we plug in the measured values of the momenta and the energies E_1 and E_2 will be calculated using the Equation (1.7), where the rest mass of muon is equal to $m_\mu = 105.658$ MeV [35].

By looking at the Figure 1.5 it is possible to have a good insight into what the distribution of di-muon invariant mass looks like. The red curve is the result of an unbinned maximum likelihood fit which will be explained later.

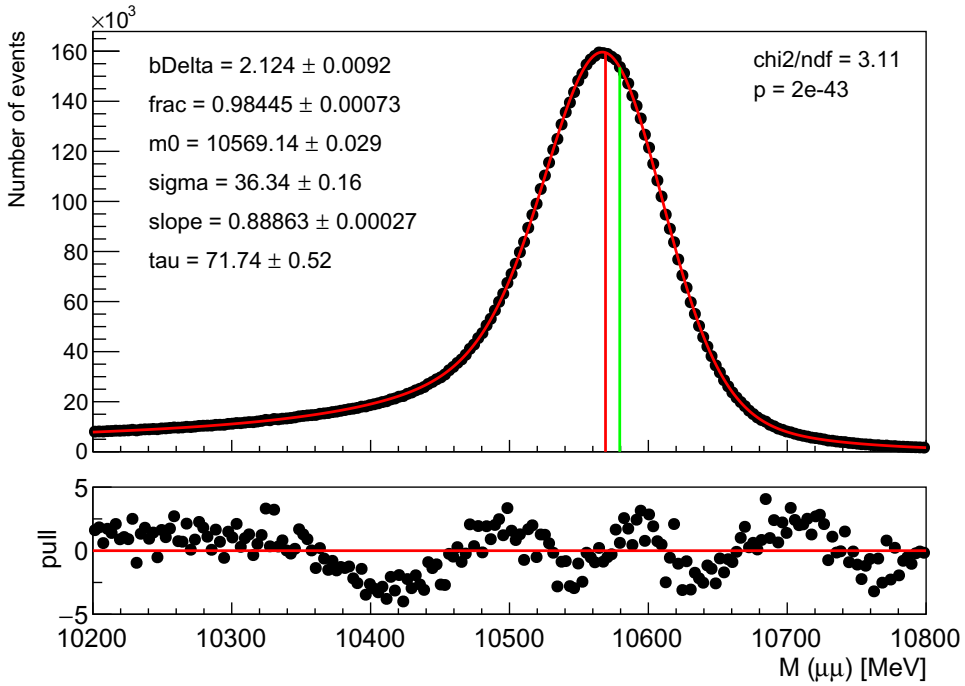

Figure 1.5: An example of muon-muon invariant mass distribution with fit.

It is clear that the distribution is not normal and that it was necessary to take this into account when performing the fit - specifically when choosing the fitted function. It is a convolution of a power function $(m_0 - m)^r$ with the resolution function of the detector. The resolution function is assumed to be a Gauss function with the exponential decrease on the tails. The power function $(m_0 - m)^r$ describes events in which there is also a photon which carries some energy away. The parameter m_0 represents the E_{cms} we get from the fit. That is the reason why it is more common to detect events with $M(\mu\mu)$ on the left side of the peak than on the right side, as we see in the graph above.

The green vertical line represents the nominal value of the centre-of-mass energy $m(\Upsilon(4S))$. The reasons why it differs from the measured value were already mentioned in the section 1.6.2.

This fit was calculated on the data from Belle II gathered in the first half of 2020. All the important details will be talked about in the chapter with the results. Under the histogram there is shown also a distribution of the pulls. The pulls represent the offset of the particular bin of the histogram from the fitted curve. When the fit is perfect, the pulls either all have value of zero or they are normally distributed around the red line.

What might catch the reader's attention is the character of the pulls in the Figure 1.5. In a sense, that is the fault of the fit, because it wasn't able to find such a curve that had the pulls distributed symmetrically at all points. However, a better way to look at it, is that it is an expected result given the fact that we know there is a time dependence of the invariant mass of the muon pairs. The maximum likelihood fit was designed for data that on a large scale follow a certain distribution. That cannot be exactly said about real data, particularly the data gathered during a large time period.

The simplest way of confirming the time evolution hypothesis is to plot the values of the fitted invariant mass with respect to time. Now we will simply divide the data into 10 chunks and look at how the value of the fitted invariant mass m_0 evolves.

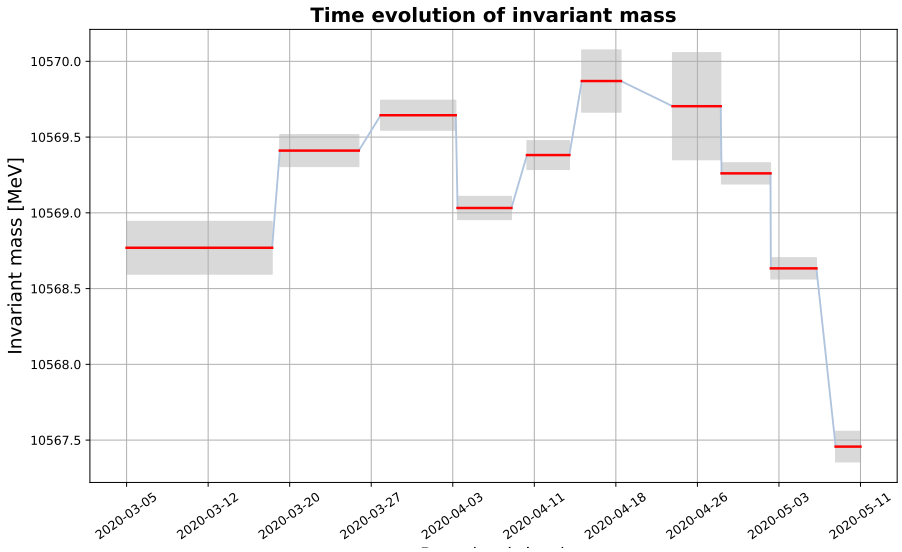

Figure 1.6: An example of time evolution of invariant mass of muon pairs calculated on 10 time intervals.

The red lines represent the fitted m_0 on a particular interval and the gray rectangle is the statistical error of m_0 given by the result of the fit.

As was already mentioned in the introduction, in this thesis, there will be introduced a better way to do this analysis. In the next chapter, we will take a quick look at the programming and the statistical methods used for developing it.

2. Programming and statistical methods

In this chapter, there will be introduced methods used while building the algorithm for analyzing the data from Belle II. The main parts include the principles of dynamic programming, K-fold cross-validation, and fitting functions using maximum likelihood.

2.1 Dynamic programming

The task of creating reliable and fast program includes having a good insight into the character of the problem the software is supposed to solve. This can mean for example understanding the input data and choosing the right data structure or simply recognizing a known pattern to make some calculations more effective.

One of the more common approaches to solving problems is looking for a subproblems to be solved first and only after that work on the bigger task. If the problems can be defined recursively, the problem might be solved by a carefully written function that calls itself. A great example of this is the famous problem of the Hanoi Towers [36].

Additionally, if the subproblems repeat throughout the algorithm, the immediate results can be stored in memory, and later, when the program needs the result of that computation, it does not have to waste time and repeat what it has already done, but it can reach into the memory and read the result from there instead. This usually leads to huge time savings, but what is more, some problems are not practically solvable without this technique.

2.1.1 Fitting a step function

A good portion of the algorithm that will be discussed in this thesis is focused on fitting a step function. A step function defined on an interval is characterized by k function values and $k-1$ break points, so $2k-1$ parameters in total. Any function can be approximated by a step function if k is large enough.

The task is to find such a combination of $k-1$ break points and k functional values that minimize the error function of the fit (or maximize the likelihood of the fit being correct). By error function, we mean a cumulative "distance" of the data points to the function value.

To find the parameters, we could try every combination of $k-1$ break points, calculate corresponding function values and from this determine the overall error of each of the fits. Then we would take as a result the combination where the error function is the smallest. However, one very quickly realizes that this is not realistic as the complexity of this is $O(n^{k-1})$, where n is the number of data inputs and k is the number of intervals of the step function.

By applying the principles of dynamic programming, we could cut down this time by simply calculating the matrix of the error function on each interval between two possible breaking points before starting the algorithm and we could pass this matrix itself as an input instead of the data.

Let us call the matrix $\mathbf{E}$ (stands for error). Then the element $\mathbf{E}_{ij}$ would be calculated as error of the fit when fitting constant value on interval (i,j), where i and j are representing i-th and j-th data input. Let us call the error function $\mathcal{E}$ where $\mathcal{E}_1(i,j) = \mathbf{E}_{ij}$ is the value of error function when there is no break point between i-th and j-th data point. The described matrix has dimensions $n \times n$, it is upper triangular and its form can be seen in the Equation 2.1.

$$\mathbf{E} = \begin{bmatrix} \mathcal{E}_1(1,1) & \mathcal{E}_1(1,2) & \cdots & \mathcal{E}(1,n)] \\ 0 & \mathcal{E}_1(2,2) & \cdots & \mathcal{E}_1(2,n) \\ 0 & 0 & \cdots & \mathcal{E}_1(3,n) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathcal{E}_1(n,n) \end{bmatrix} \tag{2.1}$$

This matrix is important to make the computation quick. There is, however, another principle that needs to be taken into account - recursion. In our case the recursion can be understood by following the following equation:

$$\mathcal{E}_k(1,n) = \min_{i \in \{1..n-1\}} \{\mathcal{E}_{k-1}(1,i) + \mathcal{E}_1(i+1,n)\}, \tag{2.2}$$

where k represents number of intervals we are looking for and function $\mathcal{E}_k(i,j)$ represents error of fit with $k-1$ breaking points on interval between i-th and j-th data point.

The base case of this recursion could be then expressed by looking for the smallest element of the sum of two vectors - one containing values of the error on intervals $(1,i)$ and the other on intervals $(i+1,n)$ - see (2.3).

$$\mathcal{E}_2(1,n) = \min_{i \in \{1..n-1\}} \{\mathcal{E}_1(1,i) + \mathcal{E}_1(i+1,n)\} \tag{2.3}$$

The position of the breaking point is then linked to the index of the minimum of the vector on the right side in equation 2.3.

The last thing that might still be unclear is what information should be stored in each iteration (except the matrix $\mathbf{E}$). Firstly, from each evaluation of the minimum of a vector we have to store the index of the minimum. When evaluating the outermost part of the recursion (eq. 2.2), we should besides vector $E_{k-1}(1,i)$ also have somewhere already stored the matrix of indices, where in each row there are indices of $k-2$ break points. It has $(n-k)$ rows because it does not make sense to work with case where there are two break points in the same place. In other words, we gradually decrease number of rows of the stored matrix while we increase the number of break points in each row. Storing the immediate results of $\mathcal{E}_{k-1}(1,i)$ is not necessary, but it can be useful when running this algorithm multiple times with increasing k.

2.2 K-fold cross-validation

In every situation of fitting data with a function, there is a risk of overfitting. The main cause of overfitting is usually a too large number of the parameters which characterize the fitted function. In the simple case of fitting polynomial function, increasing the degree of polynomial always leads to a lower error function. On

the other hand, in areas, where there is little or no data, the level of prediction from some point starts to decrease with more parameters.

By applying a suitable validation method, it is possible to optimize the number of the parameters of the fit and extract from the data the maximum possible information.

2.2.1 Leave-one-out

One of the simplest methods that decrease the chance of overfitting is called *leave-one-out*. This method consists of fitting the model on all the data except one input, which is later compared with the prediction of the model. After doing this for all the data - always leaving a different one aside - the prediction ability of the model can be easily calculated.

This method is not optimal for large datasets, especially when the process of fitting the model is time-consuming by itself.

2.2.2 Leave-p-out – K-fold cross-validation

The intuitive upgrade of the previous method is leaving out a whole bunch of data and doing the same. However, if we want to get rid of statistical fluctuations we would have to do even more model fitting, as there are $\binom{n}{p}$ (when leaving out p of n data) combinations we have to take into account. When dealing with large datasets, this is of course even worse than the leave-one-out method.

To get rid of such a problem, we can use statistics - by simply leaving out randomly chosen parts of the data. This is usually referred to as K-fold cross-validation.

2.2.3 K-fold cross-validation

In this approach, we divide the data into K groups and as was already mentioned, the division has to be random [37]. Then we fit (or train) the model on the remaining $(K-1)$ groups, which are usually together called *train-set*. The remaining group will be used for validation. The output of the validation should be some value that reliably quantifies the goodness of the model.

This process is repeated on all K groups so that each group plays the role of the *test-set* once. The illustration of this process is shown in the Figure 2.1.

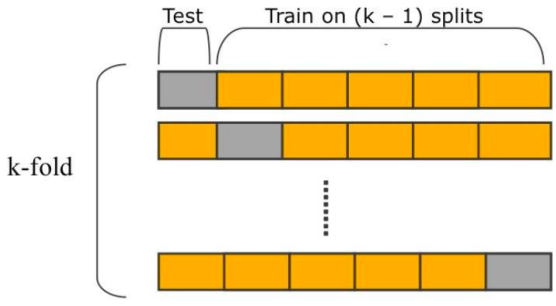

Figure 2.1: Illustration of a k-fold cross-validation [38].

The overall ability of a model to predict new results can be then calculated from the goodness of particular folds. K is usually set to be 10 but in general, it can vary. If K is equal to the size of the original dataset, this method is reduced to the *leave-one-out* case.

2.3 Data estimation

It was already hinted that fitting a function means finding the correct set of parameters that characterize the data.

2.3.1 Least squares estimation

In general, there are various methods of parameter estimation. One of the most used is the least squares estimation (LSE) and the main reason for using it is its simplicity. As the name suggest, in LSE one minimizes the function of square differences of fitted data and the fitted function [39]:

$$\sum_{i=1}^{n}(y_i - f_p(x_i))^2 \tag{2.4}$$

where n is the number of fitted data points (x_i, y_i) where $i = 1, 2...n$ and f_p is function that estimates the data with the set of parameters p.

If the noise is normally distributed with equal variances, the result of LSE is the result of maximum likelihood estimation [39]. Because of these conditions - LSE (in the form presented above) does not always provide the best estimation, but there is a possibility of modifying the function (2.4) so it can be used in more general cases. For instance in case that the variance σ_i^2 of measurement i depends on x_i, it can be taken into account that observations with larger σ_i^2 are less accurate [39]. Mathematically, we would then minimize function [39]:

$$\sum_{i=1}^{n}\frac{(y_i - f_p(x_i))^2}{\sigma_i^2} \tag{2.5}$$

This is often called weighted least squares estimation. Compared to other methods, most statisticians would not view LSE as a general method for parameter estimation, but rather as an approach that is primarily used with linear regression models [40]. However, reduction like this is not always necessary as there are also other usages such as nonlinear regression [39].

2.3.2 Maximum likelihood estimation

Another standard method of parameter estimation is the maximum likelihood estimation (MLE). This approach is more statistically rigorous and thanks to this also more consistent and efficient [40]. What is more, many methods of analysis require MLE as a starting point. This includes the chi-square test, G-square test, modeling random effects, and others [40]. In the following paragraphs, there will be explained the principles on which MLE stands.

Probability density function (PDF)

Similar to LSE, in MLE we also look for parameters of a function, but now the nature of the function is much more restricted. Namely, we want to find such a probability distribution so the observed data are the most probable - this is the principle of MLE [40]. Just like before, parametrization has to be chosen wisely.

Statistically speaking, data vector $y = (y_1, .., y_n)$ is a randomly selected set of values from an unknown population [40]. The population is characterized by some probability distribution and our goal is to approximate it by the data we observed.

Let $f(y_i|p)$ stand for the PDF that represents the probability of observing y_i with f defined by set of k parameters $p = (p_1, .., p_k)$. The correct p will then suggest which observations are more probable than others. To find p, it is useful to define another function similar to PDF which has a slightly different meaning.

Likelihood function

Let us define likelihood function by reversing roles of y and p in f:

$$f(y_i|p) \to L(p|y) \tag{2.6}$$

$L(p|y)$ is representing the likelihood of p being correct given we observed data vector y. The difference between f and L might seem subtle at first, but they have different domains and they are, in a sense, in opposition to each other - they describe mutually inverse problems [40].

The optimal parameter vector p is to be found in the k-dimensional parameters space [40]. The resulting vector is called *MLE estimate* and we will label it $p_{\text{MLE}} = (p_{1,\text{MLE}}, .., p_{k,\text{MLE}})$.

Likelihood equation

For several reasons it is convenient to work with logarithm of $L(p|y)$ [40]. The strongest argument for working with $\ell(p) = \log L(p|y)$ is probably this property of logarithm:
If

$$L(p|y) = \prod_{i=1}^{n} f(y_i, p), \tag{2.7}$$

then:

$$\log L(p|y) = \sum_{i=1}^{n} \log f(y_i, p). \tag{2.8}$$

Very small probabilities can easily cause trouble in floating precision and even though there are ways to store small numbers with little precision losses, logarithm makes sure we do not have to work with small numbers at all.

All following operations are possible thanks to the fact that logarithm is monotonous and therefore if there is a maximum of $L(p|y)$ in p_{MLE} then also has a maximum in p_{MLE} and vice versa.

From Calculus we know, that if a multi-variable function $\log L(p|y)$ has a maximum in a point $p_{\text{MLE}} = (p_{1,\text{MLE}}, .., p_{k,\text{MLE}})$, then:

$$\frac{\partial \log L(p|y)}{\partial p_{i,\text{MLE}}} = 0 \tag{2.9}$$

for all $i \in \{1, .., k\}$ and matrix $\frac{\partial^2 \log L(p|y)}{\partial p_{i,\mathrm{MLE}} \partial p_{j,\mathrm{MLE}}}$ is semi-negative. Solving the equations would then provide the correct set of MLE estimate parameters p_{MLE}.

In practice, it is rarely the case that such a solution can be calculated analytically [40]. This method was originally developed in the 1920s by R.A. Fisher [40] and at that time, the prospects of using it ended here. Luckily, with the power of modern computers, it is possible to find very precise numerical solutions in a reasonable time.

If the proposed PDF is chosen carefully and if the parameters have restricted values so that we do not search the whole k-dimensional spaces, but rather only some small area, by an iterative process we can converge to the correct solution [40]. However, even here the time can quickly pile up because in every iteration one has to calculate the loss function for every data point. In large data samples - like ours - this is a complication. One of the ways how to resolve this issue is to approximate the fitted function with a Chebyshev polynomial. However, this subject is not a part of the thesis.

3. The Algorithm

To pick up on chapter 2, in this chapter, there will be described the main structure of the program developed for this thesis.

3.1 Preprocessing of the data

The real data from measurements come out in standard ROOT format [41]. The first operation that has to be done is converting all the measured quantities to a structure that is relevant to us. As we will be working on E_{CMS} calibration, it is necessary to know the invariant mass in every detected event.

The detector can measure the momentum of each muon in three directions (x, y, z). We already know that muons are created in pairs so let us call vectors of their momentum $\vec{p_1}$ and $\vec{p_2}$. The centre of mass energy is calculated as in (1.9).

The rest mass of muon $m_\mu = 105.658 \times 10^{-3}$ GeV [35]. The invariant mass $M(\mu\mu)$ of each event is with the precise time of the event the only information we will need to demonstrate E_{CMS} calibration. The time of an event is also provided by the detector. The data should be sorted by time.

There is also an optional property of each event that can be set right in the beginning for convenience - a randomly generated integer between 0 (included) and K (K is a positive integer, excluded) for being able to quickly divide data for K-fold cross-validation.

3.2 The initial divisions

As there is a lot of data to work with and as the information we want to get to is to be seen only on a bigger scale, it is desirable to work with larger chunks. Fitting a step function can be then understood as merging the initial chunks together. The value of the invariant mass of a particular chunk is determined as a parameter m_0 of the fit example of which was provided in Chapter 1. There are several options on how to choose the size of the chunks and how to specify when they start and where they end on a time scale. We will discuss two of them.

Firstly, we could specify the number of chunks N we want and set the time-breaking points in a manner that all chunks will contain approximately the same amount of events. This approach is simple in implementation as well as in time complexity. Of course, for computational advantage we want the smallest N possible without sacrificing too much information. In the case N is too small we may easily lose some important detail about the "step" of E_{CMS}. To be more specific, the Belle II experiment is not operational at all times and the pauses between particular runs are expected to cause at least some small variation in E_{CMS}. Because of this, dividing the data at times independent from the times of runs is not what we want to do. In any case, using this method to get a first closer look at the character of the data might not be that useless.

The second option is to use the information about runs and pauses to our advantage. That information is already contained in the data.

3.2.1 Ideal intervals

From the discussion above, we know that the best solution to the problem of division is hidden in the data. Let us first propose the solution and then there will be arguments that support it.

Minimize the sum:

$$\sum_i \left(t_{\mathrm{Raw},i}^2 + \frac{C}{N_i} \right) \tag{3.1}$$

with respect to the boundaries of the intervals, where $t_{\mathrm{Raw},i}$ is length of time interval calculated as $t_{i2} - t_{i1}$ (end time minus beginning time of the i-th interval). N_i is the number of events in i-th an interval and C is a parameter that has to be specified but it is tied to the number of intervals we get, which on the other hand is not specified.

The reason why this works is quite intuitive. We can understand the first term as a penalty for the length of an interval and the second term plays the role of a penalty for too few events in an interval. If there were many little intervals, the first term would be small but the second would be huge. In the opposite case, when all the events fall into one big interval, the first term would be huge and the latter small. Minimization of the sum (after specifying C) should therefore give a more optimal division of the dataset.

3.3 Fitting the step-function

We already talked about the principles of fitting the step function in chapter 2.1.1. The main purpose was to demonstrate the power of dynamic programming. In this chapter, there will be a clarification of the details of the algorithm itself.

In chapter 2.3.2, we discussed why it might be beneficial to work with likelihood rather than some basic error function like LSE - it is simply more general. That of course has to manifest itself in fitting the kind of data we are working with. The error function $\mathcal{E}_1(i,j)$ will be replaced by the loss function:

$$\mathrm{loss}_1(i,j) = -\log L(p_{ij}, y_{ij}) \tag{3.2}$$

where y_{ij} is a chunk of data vector between i-th and j-th time breaking points (how we got these is discussed above) and p_{ij} are corresponding parameters maximizing the likelihood function. The elements of the matrix from the equation (2.1) are replaced by the term from the right side of equation (3.2). Working with chunks of data makes the matrix much smaller, which makes the computation faster. However, calculating a somewhat complex fit (finding p_{ij} where -log L is minimal) for every element above the diagonal will make it slow enough that we need to strongly restrict the initial number of the divisions.

Taking into account the K-fold cross-validation, where we set $K = 10$, we need to perform the fit for every element E_{ij}, where $i < j$, for every fold, therefore 10 times, and calculate the loss for train and test sets, therefore 20 times. We calculate these matrices beforehand so that adjusting the parameters of the step-function fitting algorithm is not slowed down by calculating the same thing over again.

When we look at the Equation (2.3), we see that the vectors on the right side are parts of the matrix E. The first one is a slice from the first row and the

second is a slice from the last column. For being able to fully understand what the Equation (2.2) represents, let us have a closer look at the case of $k = 3$. We get:

$$\text{loss}_3(1, n) = \min_j \{\text{loss}_2(1, j) + \text{loss}_1(j + 1, n)\} \tag{3.3}$$

where

$$\text{loss}_2(1, j) = \min_i \{\text{loss}_1(1, i) + \text{loss}_1(i + 1, j)\} \tag{3.4}$$

There are two details that are worth noticing. The first is the fact that whatever the value of k, we reach for the last column of E only once - in the outermost layer of recursion. Secondly, by increasing k there are smaller and smaller slices of E as it is possible to set restrictions on indices i and j without the loss of generality. By restrictions, we mean for example in equation 3.3 $j = 2$ does not make sense as we are looking for one breaking point between 1 and j. This means that the vectors we are summing are of size $n - k$. As it was mentioned in the previous chapter, we need to store the indices from the partial results.

The term $\text{loss}_2(1, j)$ can be understood as a vector. This vector contains the values of loss in a situations when there is one break point on the interval $(1,j)$. In general, for each j the minimum might be achieved by different position of the break point. That is why we also have to store the vector of these break points. The evaluation of (3.3) is finding such j for which the sum in the curly brackets is minimal and as the output we would get j as one break point and the j-th element of the stored vector of break points as the second one. For $k > 3$ we would append columns of break points until the evaluation of the outermost case.

The parameters we are looking for in the cross-validation is the optimal number of break points. It is intuitive that the more break points the better we can fit the training data (in each fold). Generally, the overall average training loss must be decreasing with the increasing k. That is not true for testing loss and therefore it is necessary to study the goodness of fit with respect to k.

The optimal value k_{opt} is usually approximately equal to k where test loss is the lowest. After finding k_{opt} we can run the algorithm once more now without dividing it to train and test set, as we have specified the main hyperparameter of the fit. For finding the positions of the break points we will need another matrix of loss which has to be calculated on the whole data sample before.

These are the principles of the improved approach. In the next chapter, there will be results obtained from the algorithm described above.

4. Results

4.1 Results from Belle II

The data we will be working with now come from the Belle II experiment and with several pauses they were gathered from the 5th of March 2020 to the 11th of May 2020. The event yields of $e^+e^- \to \mu^+\mu^-$ is shown in the Figure 4.1. Our task is to find the best amount of intervals and to find the step function that characterizes the evolution of the centre-of-mass energy which is determined by these di-muon events.

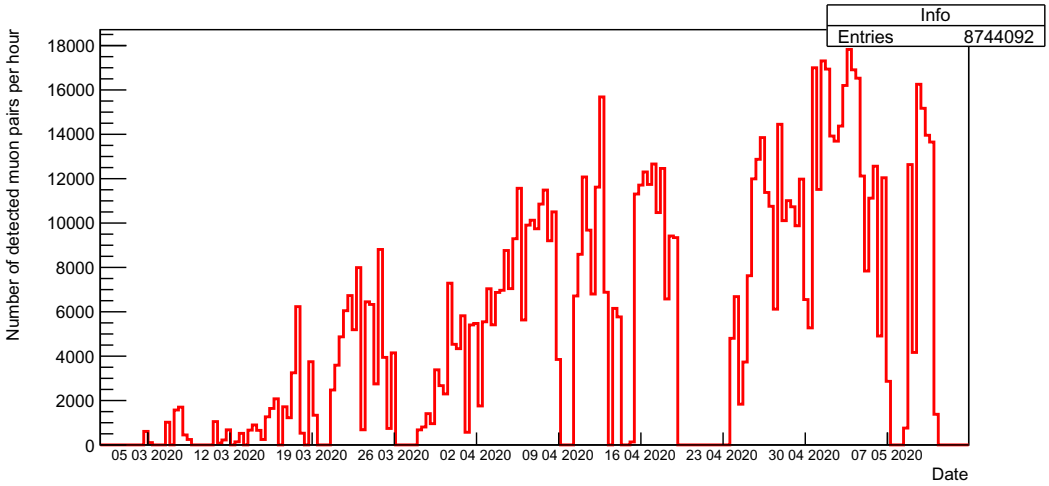

Figure 4.1: Event yields from 5th of March 2020 to 11th of May 2020

To get the best results, first, we need to divide the data as we described it in the section 3.2.1. After a little bit of experimentation, we chose C so that number of divisions is equal to 47. We are limited by computational time and 47 is a number of intervals sufficient for demonstration of the algorithm with a sufficient insight of what the data look like.

On each of the 47 intervals, we can perform the unbinned maximum likelihood fit mentioned in previous chapters and by plotting the E_{cms} together with their error we get the Figure 4.2. The red lines show the time period of the measurements and they have a value of m_0 from the invariant mass fit on each of the intervals. The grey rectangles represent the error of fitted E_{cms} in those periods. The blue lines show pauses in measurement and are there for better readability.

It is already possible to say something about the data before further analysis. We see that the value of E_{cms} is not exactly what we discussed in the first chapter. We claimed that to have the biggest event rate of B mesons we need the energy of $\Upsilon(4S)$ resonance. We even showed that the parameters of the SuperKEKB collider should ensure that electrons and positrons collide at as close to the mentioned energy of 10.58 GeV as possible. Our results, however, seem to have energy lower than that. This is caused by inaccuracies in the measurements of the magnetic field which are needed to calculate the momentum of muons. The momentum scale was therefore underestimated by approximately 0.1% which led

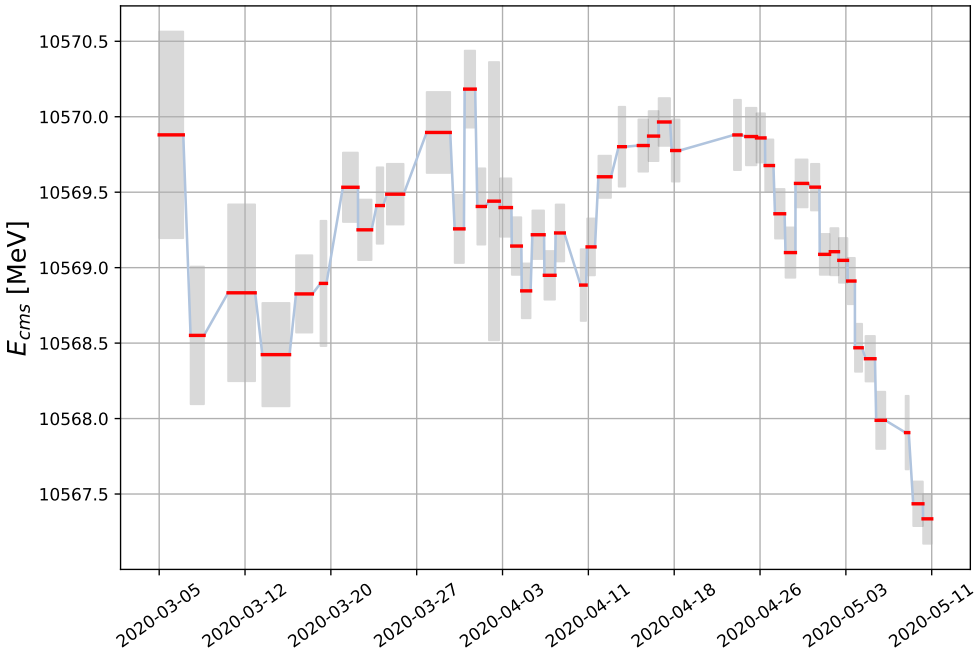

Figure 4.2: Time evolution of centre-of-mass energy determined using muon pairs after dividing data into 47 intervals.

to lower values of E_{cms} of di-muon events by approximately 11 MeV - instead of 10580 MeV we observe approximately 10569 MeV. All this is not that important to us because we are focused on the time evolution and the offset is mostly constant in time.

The next thing we can see is that the error of the first few intervals is greater than the error of the rest - this is caused by lower instantaneous luminosity in the first half of March. In the second half of April, we see a long pause after which started the centre-of-mass energy quite steeply to decrease. This is not good for the entire experiment as this might suggest that the E_{cms} fell below the optimal value and less B mesons were produced.

The next step is to calculate the matrices of train and test loss for all combinations of the intervals we got from the previous division. This, unfortunately, takes some time despite the fact that the number of intervals is quite small. Now is a good time to remind us that we need 10 sets of the matrices due to 10-fold cross-validation. Each of the matrices is upper-triangular so we need to perform $\frac{47*48}{2} = 1128$ fits per set as we do not fit the train and test sample separately, but fit the train dataset and then based on the fitted function we calculate the loss for both train and test set. We also cannot forget to once fit the data without splitting it to train and test set, so that we can evaluate when there was a jump in E_{cms} value after we find the optimal k. That makes a total of $1128 \times 11 = 12408$ fits.

If the fit takes 10 to 20 seconds we get approximately 30 to 60 hours of computational time. The time of a particular fit depends on various factors, but the most important one is the amount of data on a given interval, so bigger

intervals generally take longer to fit.

It is crucial for good results that each fit converges. One way to make sure of that is to look at the graphs of the fits and see whether the fit curve matches the histogram. The test by eye is very efficient and if the number of graphs was not so large it would probably be the go-to method. In our case of thousands of fits, we must find a different approach. Another option for checking whether fit converged or not is to use the χ^2 test. By calculating the χ^2 value for each of the fits and then looking only at those fits which score the worst in the χ^2 test, we get a reliable indicator of how good were the fits in general.

The sets of the matrices will be referred to by numbers from 1 to 11. The matrices 1 to 10 are the train and test matrices and matrix 11 corresponds to matrix containing loss, m_0 and the error of m_0 from dividing and fitting all the data.

The following graphs (figures 4.3 and 4.4) show the results of the fits with the worst χ^2/ndf, where ndf is the number of degrees of freedom - in our case it is the number of bins minus the parameters of the fitted function.

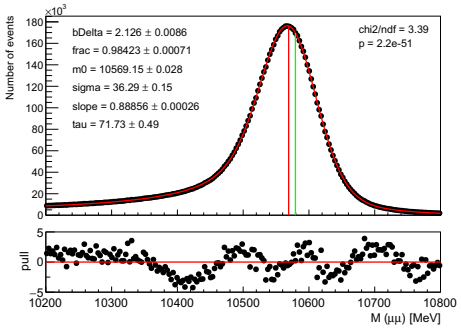

Figure 4.3: Fit from a position (2,47) of matrix 11

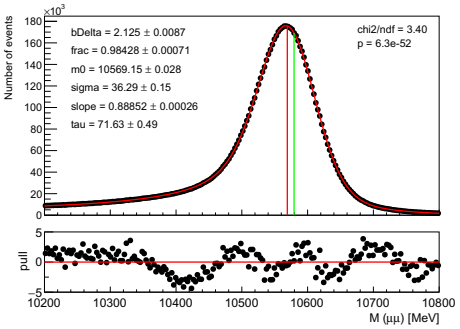

Figure 4.4: Fit from a position (3,47) of matrix 11

For comparison, here are fits of two much smaller time intervals - figures 4.5 and 4.6:

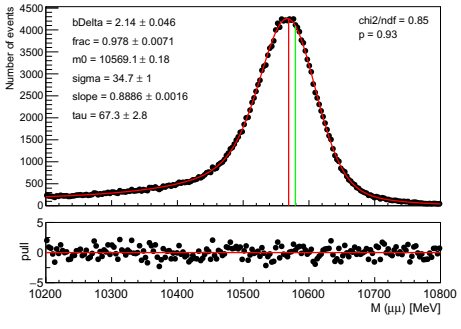

Figure 4.5: Fit from a position (35,35) of matrix 3

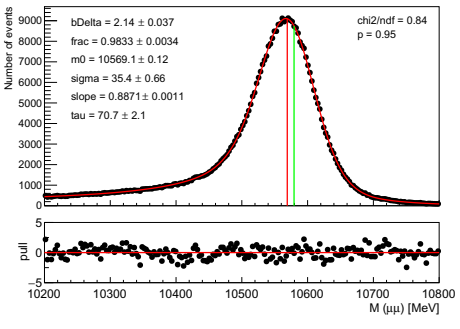

Figure 4.6: Fit from a position (19,20) of matrix 10

There is a visible difference in the distribution of the pulls, but because of a

smaller amount of events, there is also a bigger error in determining the E_{CMS} (in plots denoted as m_0) than in the graphs above.

After calculating the matrices and making sure that all the fits converged we can run the algorithm that finds the optimal number of intervals of the fitted step function. This algorithm was described in the previous chapter. However, there is still a question of how we will evaluate the overall loss so that we do not have to plot 10 sets of train and test loss as that would not be readable.

We divide the data into train and test datasets randomly by generating a random number between 0 and 9 and because of this the test and train sets may vary in size. This is why we have to "normalize" both train and test loss in each step of the 10-fold cross-validation by dividing the sum of losses by the corresponding number of events. We could plot the average of these "normalized" losses, but for statistical reasons we also multiply it by the overall number of events N - in our case $N = 8744092$. The thing is that by doing this we get another useful indicator that suggests the quality of the analysis.

We put these results into a graph as the loss (train and test) with respect to the number of calibration intervals. The graph can be seen in the Figure 4.7. One interval corresponds to no division at all. 47 intervals is the division used in the Figure 4.2.

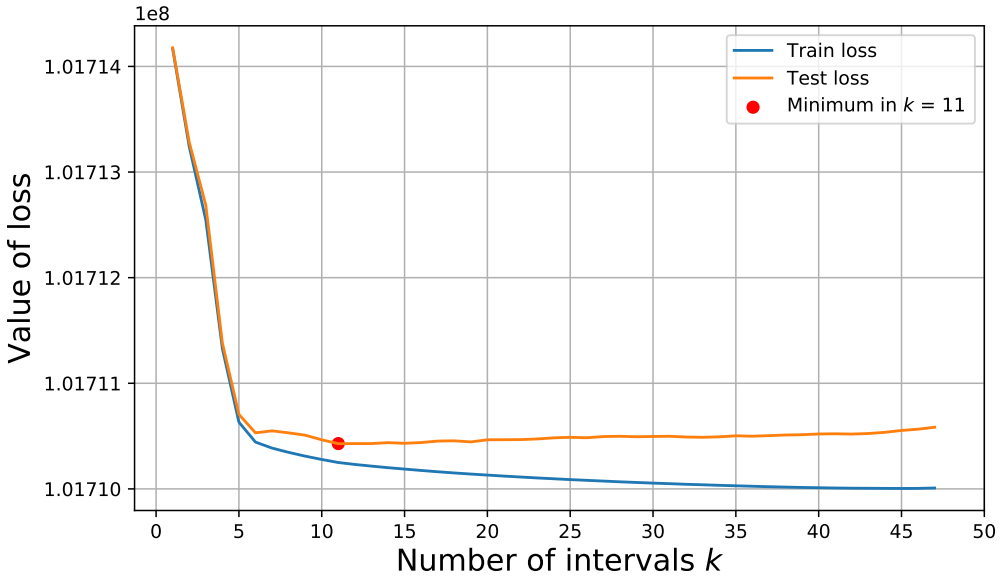

Figure 4.7: A graph with train and test loss with respect to number of calibration intervals.

As we said in the chapter about finding model parameters using K-fold cross-validation, train loss should be decreasing with the increasing number of intervals. The train loss in figure 4.7 has this property. This, of course, does not prove anything about the correctness of the algorithm, but checking a necessary condition for seeing reasonable results is always good for identifying a potential issues in the program.

The test loss is steeply decreasing at first, but then hits a plateau and slowly starts to increase. This tells us, that from the point where the test loss hits

minimum there is only small dependence on k. The test loss is the lowest when $k = 11$, which is highlighted by the red dot. There are also several local minima (for example in $k = 6$), but these are not important to us.

The final step is to take 11 as a parameter of our algorithm and fit the step function once again now without dividing it into train and test set. The result can be seen in the Figure 4.8.

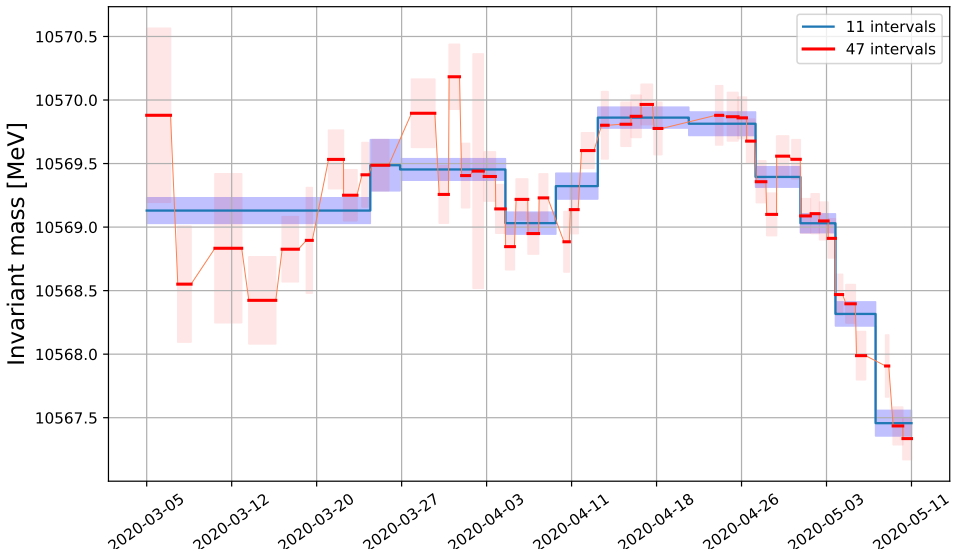

Figure 4.8: The result of fitting the step function with 11 intervals.

From the histogram of luminosity (figure 4.1) we know, that in the beginning, the luminosity was much lower than in the second half. That is why the latter intervals (especially those in May) have a bigger weight than those in March. The algorithm that looks for the best positions of interval breaks is sensitive to instantaneous luminosity as we can see in the graph. The first nine intervals were merged into one big interval. On the other hand, in the second half of the studied time period, the algorithm only merged a maximum amount of five intervals and in the end, it is only three.

The last thing that might have caught the reader's attention is the one small step between the second and third fitted interval. This is probably also caused by the difference in luminosities. If we look at the Figure 4.1 once again, there is a peak just before the 26th of March which is approximately the time of the second fitted interval. Also, if there was a difference in other parameters of the fits beside the small difference in m_0, that might be the reason why it was favourable to split the interval by a step. This question can be resolved if we inspect the 11 fits that preceded the Figure 4.8 more closely.

All 11 fits that were used to create figure 4.8 are shown below.

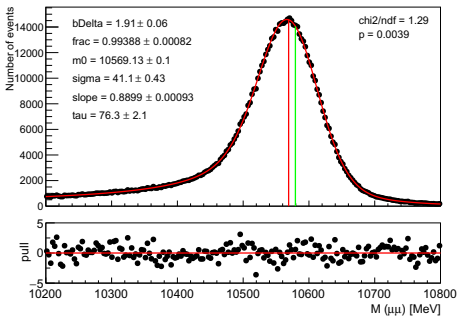

Figure 4.9: Fit of the first 9 intervals merged together.

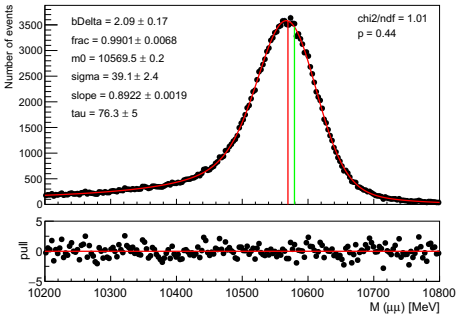

Figure 4.10: Fit of the 10th interval

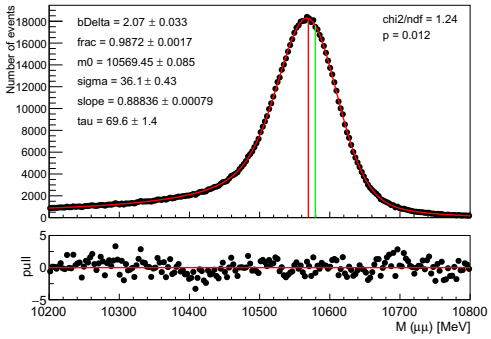

Figure 4.11: Fit of intervals 11 to 17 merged together.

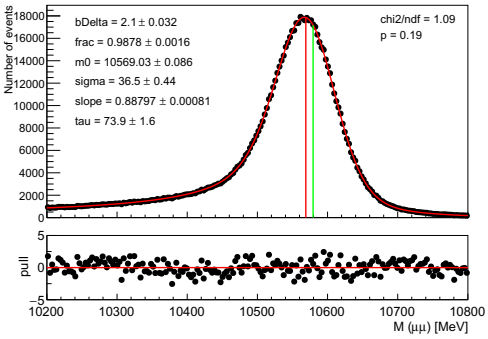

Figure 4.12: Fit of intervals 18 to 21 merged together

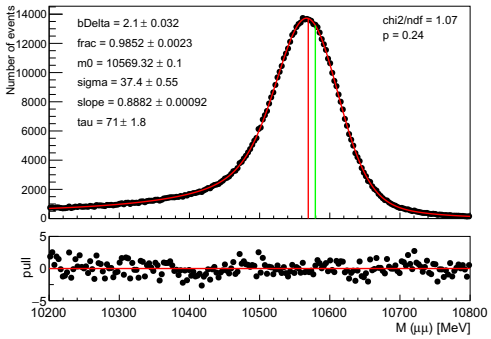

Figure 4.13: Fit of intervals 22 to 24 merged together.

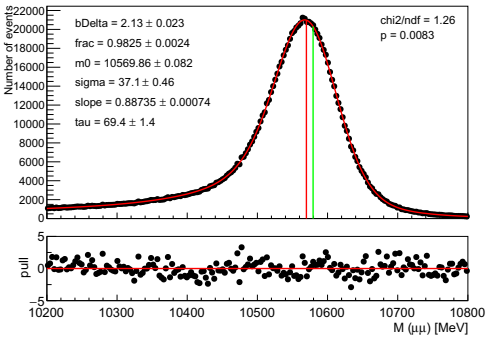

Figure 4.14: Fit of intervals 25 to 29 merged together.

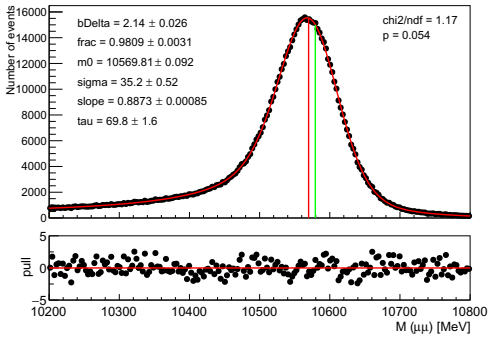

Figure 4.15: Fit of intervals 30 to 33 meged together.

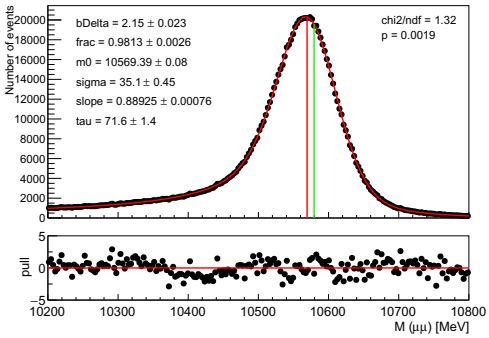

Figure 4.16: Fit of intervals 34 to 37 merged together.

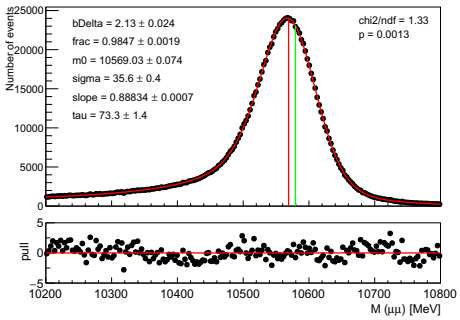

Figure 4.17: Fit if intervals 38 to 41 merged together.

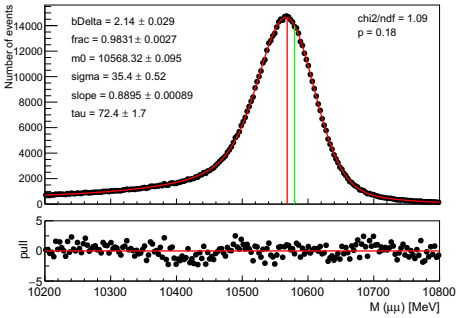

Figure 4.18: Fit of intervals 42 to 44 merged together.

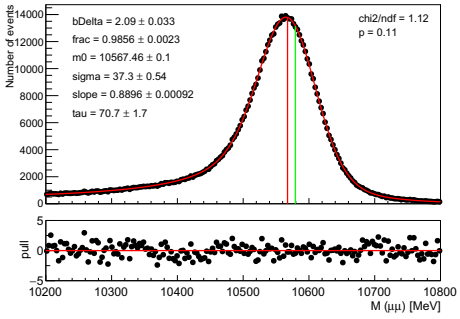

Figure 4.19: Fit of intervals 45 to 47 merged together.

Comparing the results in Figure 4.10 and 4.11 there is a difference in parameter called tau and sigma. Other parameters seem to be similar - especially m_0 which was already noticeable before.

These graphs also confirm what was said before about the fluctuations of the pulls. They show that the fitted function was chosen correctly for data which

do not have big changes in time. These intervals are not only shorter but the algorithm chose these intervals in particular so that the overall loss is the smallest. It is therefore only reasonable, that the fits are more accurate, which is also confirmed by the χ^2/ndf values shown in each graph.

The previous discussion about the graph with the train and test loss (Figure 4.7) led us to further analysis of the minimum of the test loss when $k = 11$. However, there is still a question of how much better will the description of the data be when k will be larger than 11. Setting $k = 13$ seems to be a reasonable choice as the test loss is larger only slightly. That case is shown in the Figure 4.20.

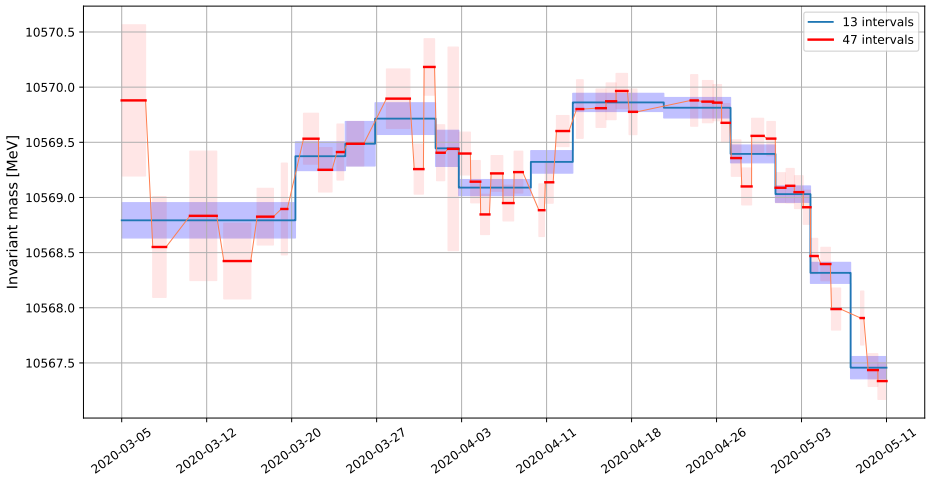

Figure 4.20: The result of fitting the step function with 13 intervals.

We see that the two more intervals smoothed the first half and also resolved the strange small step from the 2nd to the 3rd interval from the figure 4.8. The second half of the fit of the step function remained unchanged.

From this perspective, it is hard to say where should we stop when trying the different values of k, but let us not forget that this is why we used the 10-fold cross-validation in the first place. On the other hand, seeing the red lines which represent the division to 47 intervals might also be misleading. The Figure 4.20 gives us the impression that most of the data sample is fitted quite well because the blue line is overlapping with most of the red error rectangles. We have to remember that each of the red intervals could be further divided.

If we increase the initial number of intervals and calculate only the diagonal elements on the matrix of the whole dataset, we could generate a graph similar to the Figure 4.2 but denser. Let us also include the already calculated fit of 13 intervals from the Figure 4.20 so we can make a comparison. The result of this process is shown in the Figure 4.21.

The y-axis is a little bit stretched which makes it harder to read, but it is clear that the result is very similar to the former case. By almost doubling the number of intervals in the background there are only little changes to the final outlook. Contrary to the Figure 4.8 where we saw that one or two more intervals would certainly help, it would be hard to decide here where would we put the next breaking point.

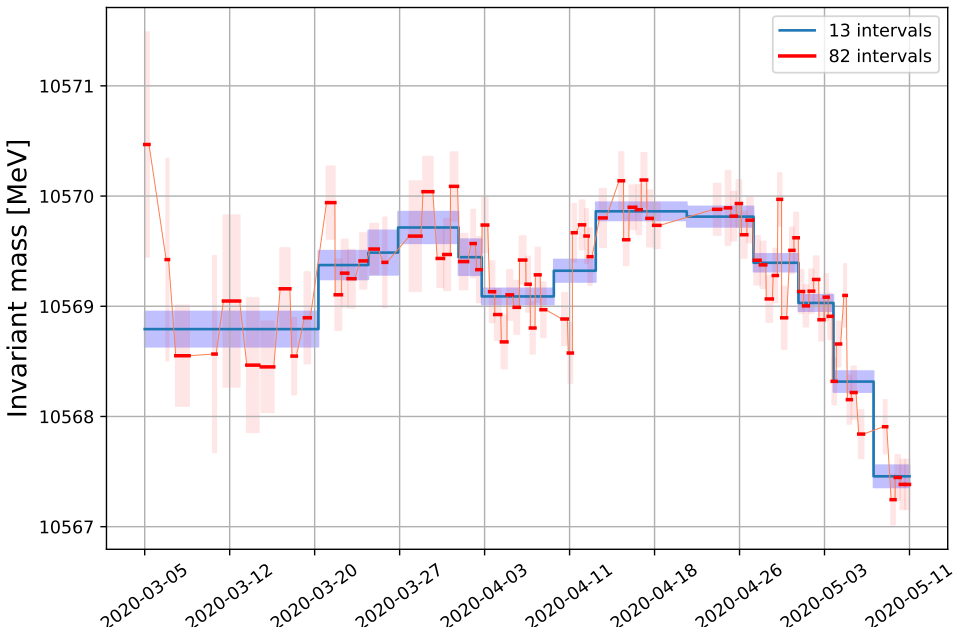

Figure 4.21: The result of fitting the step function with 13 intervals with 82 intervals in the background.

4.2 Discussion

It might seem it would be best if we simply repeated the entire process with a higher and higher number of intervals, but we will not do it, because we have already demonstrated the big potential of this method which was the main objective. There are, however, other reasons why inspecting higher k is not what we want to do.

First of all, we know that more intervals would characterize the data better, but we are looking for the optimal intervals. From the validation process we found that from $k > 11(13)$ we start to overfit. Because we are averaging the accumulated loss in the algorithm in which each of the folds was fitted independently there is an error when identifying the minimum k. Therefore, we should not be strictly saying that $k = 11$ is the optimum.

Secondly, it was also said before that smaller intervals carry a bigger statistical error as it was possible to see in the Figures 4.5 and 4.6. In general, one should be careful when increasing k.

In this thesis, we are slightly restricted by the available computational power, as we could not use much bigger matrices for testing the algorithm. On the other hand, in the case of big matrices, there is also an increase of the chance of the fits failing to converge, which might make the whole process longer and the implementation more difficult.

There is a question of how we could improve the presented approach after we had a chance to see its potential but also its limits. Firstly, we could find a more effective way to calculate the matrices either by passing values of some parameters between the fits or by improving the fitting process in general. However, the unbinned maximum likelihood fit is not a part of this thesis and there is a chance

that given what we know about the Belle II data there is only little we can do.

Next, it might be useful to determine the error of the minimum of k so we know which cases should be examined more closely and which are beyond the interval of tolerance. There is a possibility to take into account the average values of loss which were plotted and draw further conclusions from that.

Conclusion

In this thesis, the Belle II experiment is discussed. The Belle II among other phenomena studies CP asymmetry by measuring lifetimes of B-mesons. The main focus of this thesis is to present and demonstrate an algorithm for calibration of the centre-of-mass energy of the collisions in the experiment.

In the first chapter, there is a short description of the SuperKEKB accelerator and the Belle II detector. There are introduced the fundamental quantities that are directly and indirectly measured and there is emphasized the motivation for the improvement of the calibration.

Next, there are presented used programming methods which are on one hand widely known, but on the other hand, they were the foundation the algorithm for this thesis was built on. We define the term step-function and we show what is the optimal way of using it when fitting a data sample like ours. In the thesis, there is an emphasis on explaining the principles of the algorithm rather than quoting the actual code.

Lastly, there are shown results of running the algorithm with real data from Belle II. The data were collected from March to May of the year 2020. It consists of the events of produced muon pairs in the mentioned time period. It is true that since then there were made adjustments to some of the measured quantities and also the instantaneous luminosity of Belle II has improved. This is not that important as the main task was to demonstrate the ability of the developed algorithm to determine the ideal sizes and positions of the calibration intervals in the given dataset.

The presented approach has its disadvantages. One of the most obvious ones is the big demand for computational power. The computations we did for demonstrative purposes took cumulatively around 60 to 70 hours. That is not optimal given the fact that we only worked with a maximum of 47 intervals. The needed time rises in a quadratic manner with the respect to the number of studied intervals. Most of the time was used by the maximum likelihood unbinned fitting of the invariant mass. The development of this fit is not a part of the thesis. Working on a smaller data sample would make it faster.

It is assumed that in the future the luminosity of the Belle II will be much higher. Thanks to this, calibrating shorter time periods with the same amount of data might be less demanding if the increase in luminosity does not also cause a bigger time dependence, because there would be a lower number of intervals we would start with.

The usage of the algorithm that fits the step function is not limited to calibrating a detector. In our implementation, we focused on how to preprocess the big data sample that follows a special time-dependent distribution and then we discussed how to interpret the results.

If we wanted to upgrade this method, we could assume, that not all the parameters from the centre-of-mass energy fit is equally dependent on time. Unlike the parameter m_0, time dependence of which we studied, other parameters might not change with time so much. If we could take this into the account during the fitting process, we might see more precise and faster results. However, choosing the correct parameters and passing them can be complicated and that is why we

did not occupy ourselves by this in the thesis.

The GitHub repository that contains the code is located at: https://github.com/DonnyPlease/belleII-calibration.

References

[1] Robert B. Mann. *An introduction to particle physics and the standard model*. Boca Raton, FL: CRC Press, 2010.

[2] E. Kou, P. Urquijo, et al. *The Belle II physics book*. Dec. 2019. DOI: 10.1093/ptep/ptz106.

[3] J. H. Christenson, J. W. Cronin, et al. "Evidence for the 2π Decay of the K_2^0 Meson". In: *Phys. Rev. Lett.* 13 (4 July 1964), pp. 138–140. DOI: 10.1103/PhysRevLett.13.138. URL: https://link.aps.org/doi/10.1103/PhysRevLett.13.138.

[4] 1980. URL: https://inis.iaea.org/collection/NCLCollectionStore/_Public/46/027/46027128.pdf.

[5] Andreas Höcker and Zoltan Ligeti. *CP violation and the CKM matrix*. 2006.

[6] P F Harrison. *THE PHYSICS OF B FACTORIES*.

[7] Jolanta Brodzicka, Thomas Browder, et al. "Physics achievements from the Belle experiment". In: *Progress of Theoretical and Experimental Physics* 2012.1 (Dec. 2012). 04D001. ISSN: 2050-3911. DOI: 10.1093/ptep/pts072. eprint: https://academic.oup.com/ptep/article-pdf/2012/1/04D001/11595832/pts072.pdf. URL: https://doi.org/10.1093/ptep/pts072.

[8] B. Aubert, A. Bazan, et al. "The BABAR detector". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 479.1 (Feb. 2002), pp. 1–116. DOI: 10.1016/s0168-9002(01)02012-5. URL: https://doi.org/10.1016%2Fs0168-9002%2801%2902012-5.

[9] URL: https://web.archive.org/web/20220121024842/https://www-public.slac.stanford.edu/babar/Nobel2008.htm.

[10] Kazuo Abe et al. "Observation of large CP violation in the neutral B meson system". In: *Phys. Rev. Lett.* 87 (2001), p. 091802. DOI: 10.1103/PhysRevLett.87.091802. arXiv: hep-ex/0107061.

[11] Bernard Aubert et al. "Observation of CP violation in the B^0 meson system". In: *Phys. Rev. Lett.* 87 (2001), p. 091801. DOI: 10.1103/PhysRevLett.87.091801. arXiv: hep-ex/0107013.

[12] URL: https://confluence.desy.de/display/BI/Belle+II+Luminosity.

[13] The BABAR Collaboration, J. P. Lees, et al. *Time-Integrated Luminosity Recorded by the BABAR Detector at the PEP-II e+e- Collider*. 2013. DOI: 10.48550/ARXIV.1301.2703. URL: https://arxiv.org/abs/1301.2703.

[14] Belle II collaboration, F. Abudinén, et al. "Precise measurement of the D^0 and D^+ lifetimes at Belle II". In: (Aug. 2021). DOI: 10.1103/PhysRevLett.127.211801. URL: http://arxiv.org/abs/2108.03216%20http://dx.doi.org/10.1103/PhysRevLett.127.211801.

[15] Chiara la Licata. "Belle II status and Physics prospects". In: *Proceedings of Science* 360 (2021). ISSN: 18248039. DOI: 10.22323/1.360.0006.

[16] Tetsuo Abe, Kazunori Akai, et al. “Achievements of KEKB”. In: *Progress of Theoretical and Experimental Physics* 2013 (3 2013). ISSN: 20503911. DOI: 10.1093/ptep/pts102.

[17] Yasmine Sara Amhis et al. “Averages of b-hadron, c-hadron, and τ-lepton properties as of 2018”. In: *Eur. Phys. J. C* 81.3 (2021), p. 226. DOI: 10.1140/epjc/s10052-020-8156-7. arXiv: 1909.12524 [hep-ex].

[18] Yukiyoshi Ohnishi, Tetsuo Abe, et al. “Accelerator design at SuperKEKB”. In: *Progress of Theoretical and Experimental Physics* 2013 (3 Mar. 2013). ISSN: 20503911. DOI: 10.1093/ptep/pts083.

[19] National Lab. for High Energy Physics. *KEKB B-factory design report*. 1995. URL: https://inis.iaea.org/search/search.aspx?orig_q=RN:27045036.

[20] Y. Suetsugu, H. Fukuma, et al. “Mitigating the electron cloud effect in the SuperKEKB positron ring”. In: *Physical Review Accelerators and Beams* 22 (Feb. 2019). DOI: 10.1103/PhysRevAccelBeams.22.023201.

[21] D. Besson and T. Skwarnicki. “υ spectroscopy”. In: *Ann. Rev. Nucl. Part. Sci.* 43 (1993), pp. 333–378. DOI: 10.1146/annurev.ns.43.120193.002001.

[22] Werner Herr and Bruno Muratori. *Concept of luminosity*. 2012. URL: https://cds.cern.ch/record/941318/files/p361.pdf.

[23] SuperB Collaboration. *SuperB: A High-Luminosity Asymmetric e+ e- Super Flavor Factory. Conceptual Design Report*. 2007. DOI: 10.48550/ARXIV.0709.0451. URL: https://arxiv.org/abs/0709.0451.

[24] URL: https://www2.kek.jp/accl/eng/topics/topics211224.html.

[25] Raghunath Sahoo. “Relativistic Kinematics”. In: (Apr. 2016). URL: http://arxiv.org/abs/1604.02651.

[26] T. Abe, I. Adachi, et al. *Belle II Technical Design Report*. 2010. DOI: 10.48550/ARXIV.1011.0352. URL: https://arxiv.org/abs/1011.0352.

[27] B. Shwartz. “Electromagnetic calorimeter of the Belle II detector”. In: *Journal of Physics: Conference Series* 928 (Nov. 2017), p. 012021. DOI: 10.1088/1742-6596/928/1/012021.

[28] M. Boronat. “DEPFET pixel detector for future e-e+ experiments”. In: *Nuclear and Particle Physics Proceedings* 273-275 (Apr. 2016), pp. 982–987. ISSN: 24056014. DOI: 10.1016/j.nuclphysbps.2015.09.154.

[29] B Spruck, F Abudinén, et al. *Belle II Pixel Detector Commissioning and Operational Experience*. 2019. URL: https://pos.sissa.it/.

[30] Katsumoto. 2016. URL: https://web.archive.org/web/20220422175059/https://belle2.jp/electromagnetic-calorimeter/.

[31] URL: https://web.archive.org/web/20220211042252/https://belle2.jp/muon-detection/.

[32] URL: http://www.hep.ph.ic.ac.uk/~dauncey/will/lecture02.pdf.

[33] URL: http://www.phys.ufl.edu/~korytov/phz5354/note_01_NaturalUnits_SMsummary.pdf.

[34] *Connecting the dots*. 2022. URL: https://indico.cern.ch/event/1103637/contributions/4844389/attachments/2453662/4204994/ConnectingDots_BelleII.pdf.

[35] URL: https://pdg.lbl.gov/2019/listings/rpp2019-list-muon.pdf.

[36] URL: https://web.archive.org/web/20211121143902/https://www.cs.cmu.edu/~cburch/survey/recurse/hanoiimpl.html.

[37] M Stone. *Cross-Validatory Choice and Assessment of Statistical Predictions*. 1974, pp. 111–147. URL: https://about.jstor.org/terms.

[38] Safae Sossi Alaoui, Yousef Farhaoui, and B. Aksasse. "Classification algorithms in Data Mining". In: *International Journal of Tomography and Simulation* 31 (Aug. 2018), pp. 34–44.

[39] Brian. Everitt and David C. Howell. *Encyclopedia of statistics in behavioral science*. John Wiley and Sons, 2005, p. 2208. ISBN: 0470860804.

[40] In Jae Myung. "Tutorial on maximum likelihood estimation". In: *Journal of Mathematical Psychology* 47 (1 2003), pp. 90–100. ISSN: 00222496. DOI: 10.1016/S0022-2496(02)00028-7.

[41] URL: https://root.cern/manual/root_files/.

List of Figures