Analysis, and machine learning anomaly detection of the VELO-LHCb calibration

Maciej Majewski
Ph.D. Student, Faculty of Physics and Applied Computer Science, AGH University of Science and Technology, Mickiewicza 30, Krakow, Poland
E-mail: majewski@fis.agh.edu.pl

Abstract. Silicon detectors are an extraordinary piece of equipment that has become the cornerstone of research in modern high energy physics. They are becoming increasingly useful in medical research as well. LHCb’s VELO detector is itself, a microstrip silicon vertex detector. It is the heart of LHCb spectrometer and plays a vital role in the reconstruction of particle tracks. The data gathered by VELO is used in LHCb studies of CP violation and heavy flavour physics. This work presents the studies on the calibration parameters of the VELO-LHCb. The inference from the analysis, with the use of probabilistic programming, is then used to create a machine learning based anomaly detection system in the space of VELO calibration.

1. Introduction
The LHCb [1] spectrometer is a single-arm forward spectrometer. Its main purpose is to study CP violation, as well as heavy flavour (beauty and charm) physics. The detector itself is located at Large Hadron Collider ring at CERN.

Since there is no single ideal way of getting all of the measurements that are necessary for the physics programme, the LHCb detector consists of sub-detectors (such as vertex detector, calorimeters and others). The actual physics data is an amalgamation of the readouts of the hardware. The key part in the process of combining the measurements is the ability to reconstruct the tracks of the particles. LHCb VELO [2] detector was designed for that specific task. In Fig. 1, you can see the VELO on the leftmost side of the LHCb spectrometer. It surrounds the interaction point, where particles collisions take place. This means that the detector is working under high particle fluence. The radiation damage from the beam is compensated mostly with voltage. However, the maintenance of the VELO still must be conducted cautiously and with great care.

1.1. VELO
The VELO detector is a silicon microstrip detector. It consists of 42 modules on retractable panels. Each module consists of R, and Phi type of sensors (in polar coordinates). Each sensor has 2048 silicon microstrips (they are also called readout channels). In Fig. 2 you can see the microstrips (in black). Notice the layout of the silicon strips. Both of the sensor types have their channels divided into sections, which is apparent in later figures. The signal from the detector is the charge collected in the silicon. Most of the signal is unnecessary noise, that does not carry meaningful information. To filter the signal the calibration is used.
1.2. **Calibration**

Calibration is calculated using Vetra software [3], and the data taken when the particle beam was not present. The most important calibration parameters are the noise parameters of the channels; the mean of the noise, and its standard deviation.
1.2.1. Mean of the noise The mean of the noise is used to calculate pedestals [4]. Pedestals are values that should be subtracted from the readout of the detector, to level its values to the zero. The actual output from the detector, varies in levels of ADC, as shown in Fig. 3. In order to unify the measurements, it is helpful to bring them to the same base of zero, by subtracting the mean of the signal, as depicted in Fig. 4. The value of the pedestal is the actual "zero" of the noise. In this paper, we mostly focus on the standard deviation and its meaning for the calibration.

\[ H_t = 5\sigma, L_t = 40\% \times H_t \] (1)

Figure 3. Example of raw signal from single VELO sensor, with no pedestals subtraction. Limited to 300 channels.

Figure 4. Example of signal from single VELO sensor, with subtracted pedestals. Limited to 300 channels.
These parameters are used to cluster the hits in the VELO [5]. If the ADC value of the channel exceeds the hit threshold, it means that there must have been a hit, and the supporting low threshold is used to select neighbouring channels to be used to calculate exact hit point. The values from the strips are used as weights to calculate the weighted mean of the hit placement. This process is performed for data from all the modules. That data combined together can reconstruct the track of the particles. This is essential to the data taking process.

2. The data and analysis
The analysis of the data included 31 calibrations between the dates 2010-08-16 to 2017-07-18. Each of the data points consisted of the pedestal value, low threshold and high threshold per each channel. That means that there were 170 000 parameters of one type in each calibration. That amount of data is hard to visualize. The preferred plot is 2D histogram, the channel numbers are on the X-axis, and the value of the variable is on the Y-axis, the colour denotes the number of occurrences of that particular value in a given set of data.

2.1. Pedestals
In Fig. 8 some small irregularities appearing every 32 channels are visible. This is expected since all channels are grouped together by readout chips, in groups of 32. Near the 1500th channel, you can see artifacts, that come from sensor 85, that sensor was discovered to be faulty, and was partially excluded from physics data taking. Also, near channel 1750 an irregularity is visible. It is caused by a cooling line that is placed nearby. Except for those effects, the histogram of pedestal values looks as expected, and is free from unusual artifacts. The pedestal values are centering closely to 520 ADC. We can say that the values of the pedestals are generally stable, although some of the local changes are still studied.

2.2. Thresholds
In Fig. 6 you can see a summary plot of all high thresholds. Here, we focus on High Thresholds only, since Low Thresholds are based on High Threshold. You can already see some regularities. Some of the distributions in the plot repeat every 512 channels. This is due to how the physical strips are connected and numbered, which you can observe in Fig. 2. You can also see horizontal lines appearing in the histogram. This effect was known to VELO-group as ”Header cross-talk”. This is a hardware issue that has been addressed, and those channels were suppressed in further data taking, so although they are visible in the monitoring plots, they are skipped in the analysis and calculations. The difficulty of the analysis of this data comes from the dimensionality of the problem. Although 31 data points are not much, 170 000 different parameters create a lot of possible approaches to the data.

It was known that on two particular dates the calibration was conducted in improper conditions. We can see those calibrations isolated from rest on the Fig. 9 and 11, and all other calibrations in the Fig. 10, 12.

Notice the difference of the split for R- and Phi-type sensors. For the sake of clarity, we will focus on the R-type ones in the rest of this paper. After the separation, we can still observe some suspicious values in the plots. After further investigations, we classify another two dates as outlying calibrations, depicted in Fig. 13, 14. It is worth noting that they are not ”improper calibration”, but calibrations conducted in the conditions that might not have been ideal, thus are treated as outliers in the data set.

3. Anomaly detection
The usual approach in machine learning is to use big amounts of data, and feed them into the black box. This approach, although useful in some areas, might not be ideal when dealing
with smaller data sets, and processes that we already have good knowledge about. This paper presents the approach that uses Probabilistic Programming, and Pythons library called Pymc3 [6]. We build a probabilistic model of the calibration and then use Markov chain Monte Carlo to learn model parameters.

3.1. Probabilistic model
The non-outlying calibrations are depicted in Fig. 7. Except for some artifacts, it can be approximated quite well with Gaussian distribution (per each channel). Gaussian might not be the best distribution, but it is good enough approximate. Other might be tested in the future. This model can be expressed as follows:

\[ T_n \sim \text{Gaussian}(\mu = \mu_n, \sigma = \sigma_n) \]  

(2)

Where \( n \) is the n-th channel. Assuming that we reduce the dimension of the data, by treating all the modules as the same, we can easily calculate these simple models parameters using non-outlying calibrations. Notice that \( \mu_n \) and \( \sigma_n \) is calculated per channel.

| Calibration date | X |
|------------------|---|
| 2011-03-07       | 1 |
| 2012-08-02       | 3 |
| 2012-07-30       | 10|
| 2012-08-01       | 10|
| all others       | 0 |

Table 1. Outlierness calibration dataset values.

Now in order to create a single metric for the "outlierness" of the calibration, we create artificial parameter \( X \), that we also call "outlierness". This metric is completely artificial and only based on expert knowledge. The core meaning of "outlierness" is "the higher it is, the higher the probability of particular calibrations is to be an outlier". The table 1 contains the arbitrarily created values of \( X \), that were assigned to the calibrations.

This allows us to create a model that includes outlying calibrations (eq. 3).

\[ T'_n \sim \text{Gaussian}(\mu = X \ast \mu_n + \mu_n, \sigma = X \ast \sigma_n + \sigma_n) \]  

(3)

This model is a Gaussian that is linearly dependant on the outlierness of the calibration, where \( \mu_n \) and \( \sigma_n \) are slopes, and \( \mu_n \) and \( \sigma_n \) are intercepts. The intercepts for the model have been calculated in the previous model. Notice that if \( X \) is zero, then all that’s left is the previous model \( T_n \). We have assigned \( X \) to the calibration in the dataset (table 1) so we can use it to calculate \( \mu'_n \) and \( \sigma'_n \). This completes the model.

Using the complete set of models parameters we can approximate the value of \( X \). Using the same parameters we can not only get the outlierness for the whole calibration set, but also for any subset of channels in the detector. Although when applied that way, it is best to take a look at the trend, and values of outlierness for the historical data, as the exact values of outlierness might change depending on the specific subset of channels. It is worth mentioning that this model of anomaly detection was created bearing in mind that ideally it should be applied to the new VELO detector, scheduled to be inserted in LHCb after the technical stop [7]. It is a good idea to search for the broadest and universal model, that could be compatible not only
with the upgraded VELO but also possibly with other silicon detectors that rely on the noise metrics in their measurements. This is why this approach was chosen over more direct methods, like means of the thresholds, or some other simple statistical metrics.

Another advantage of using Bayesian modelling is that we can create artificial calibrations, and extrapolate the insight from the data, for example, to go outside the scope of our dataset, and see what the calibration with the value of $X = 12$ would look like.

### 3.2. Application of outlierness in anomaly detection

![Figure 5. Screenshot from Lovell monitoring software.](image)

The goal of the outlierness detection is to in a simple manner, warn about some unusual behaviour of the detector. This model has been included as a part of Lovell monitoring [8-10] software for the VELO. In Fig. 5 you can see a screenshot of the plot from Lovell, and box plot as a trend of outlierness. This specific screenshot shows the outlierness trend for sensor 21, and you can see that since 2017-10-12 there is a small rise in outlierness. This was because of the change in the cooling system. The software for outlierness calculation was put in Calina package [11], and can be viewed in the repository.

### 4. Summary

The calibration of the VELO detector is a critical part of its maintenance and operation procedure. When conducted under improper conditions, it can have a negative influence on the data taking process. We have shown initial insights from analysis of calibration data. The values of the pedestals are stable in time, but the thresholds vary and can be used to model the anomalies. We propose a model describing this behaviour. The proposed machine learning approach can help with the early detection of anomalies in its calibration data stream, and has already been incorporated into the detector monitoring software. The robustness of the model may allow its application to other silicon detectors. A number of detailed analyses are currently ongoing in order to evaluate the possibility of an autonomic decision making by the software
platform based on reduced metrics obtained from both the raw and calibration data streams respectively. Also, a new software infrastructure is being currently designed for the upgraded vertex detector that will employ similar solutions described here.

We acknowledge support from National Science Centre (Poland), UMO-2017/27/N/ST2/01880. This research was supported in part by PLGrid Infrastructure.

**Figure 6.** Total high threshold distribution, across all modules, and all calibrations.

**Figure 7.** High threshold distribution, across all modules, non outlying calibrations, plot limited to 500 channels

**Figure 8.** Total distribution of pedestals, across all modules, and all calibrations.
Figure 9. R sensors outlying calibrations.

Figure 10. R sensors calibration with outlying calibrations excluded.

Figure 11. Phi sensors outlying calibrations.
Figure 12. Phi sensors calibration with outlying calibrations excluded.

Figure 13. R sensor thresholds distribution on 2011-03-07.

Figure 14. R sensor thresholds distribution on 2012-08-02.

References
[1] Aaij R et al. (LHCb) 2015 Int. J. Mod. Phys. A30 1530022 (Preprint 1412.6352)
[2] Aaij R et al. 2014 JINST 9 P09007 (Preprint 1405.7888)
[3] Szumlak T 2010 J. Phys. Conf. Ser. 219 032058
[4] Parkes C and Szumlak T 2009 Application of the Pedestal Following Algorithm to the VELO Detector Tech. Rep. LHCb-2009-036. CERN-LHCb-2009-036 CERN Geneva URL http://cds.cern.ch/record/1166955
[5] Parkes C, Ruf T and Szumlak T 2007 Reconstruction of Cluster Positions in the LHCb Velo Tech. Rep. LHCb-2007-151. CERN-LHCb-2007-151 CERN Geneva URL https://cds.cern.ch/record/1074928
[6] Salvatier J, Wiecki T V and Fonnesbeck C 2016 PeerJ Computer Science 2 e55 URL https://doi.org/10.7717/peerj-cs.55
[7] Collaboration L 2013 LHCb VELO Upgrade Technical Design Report Tech. Rep. CERN-LHCC-2013-021. LHCB-TDR-013 URL https://cds.cern.ch/record/1624070
[8] Bel L, Crocombe A C, Gersabeck M, Pearce A, Majewski M and Szumlak T 2017 J. Phys. Conf. Ser. 898 092046
[9] Majewski M 2017 Acta Physica Polonica B 48 1061 URL https://doi.org/10.5506/aphyspolb.48.1061
[10] gitlab repository of lovell https://gitlab.cern.ch/lhcb/Lovell accessed: 2018-05-20
[11] gitlab repository of calina https://gitlab.cern.ch/mmajewsk/calina accessed: 2018-05-20