DEVELOPMENT OF THE LHCb VELO MONITORING SOFTWARE PLATFORM*

Maciej Majewski

on behalf of the LHCb VELO group

Faculty of Physics and Applied Computer Science
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland

(Received April 7, 2017)

Corrected August 20, 2020
according to Errata ibid. **49**, 195 (2018); *ibid.* **51**, 1905 (2020)

One of the most important parts of the LHCb spectrometer is the VErtex LOcator (VELO), dedicated to the precise tracking close to the proton–proton interaction point. The quality of data produced by the VELO depends on the calibration process, which must be monitored to ensure its correctness. This work presents details on how the calibration monitoring is conducted and how it could be improved. It also includes information on monitoring software and data flow in the LHCb software framework.

DOI:10.5506/APhysPolB.48.1061

1. Introduction

1.1. LHCb detector

The LHCb detector [1] is a single arm forward spectrometer, dedicated for rare decays and CP violation in heavy-flavour quark sector. One of the most important parts of spectrometer is the vertex detector VELO (VErtex LOcator). The VELO is a crucial part of the LHCb tracking system and provides precise information on charged particles tracks close to the proton–proton interaction point. The VELO is composed of silicon microstrip sensors, which operate at a distance of 8 mm from the LHC beams. This makes the VELO susceptible to radiation damage effects which can have a significant impact on the physics performance. In this paper, the monitoring software platform that can be used for constant assessment of the detector conditions and maintaining the highest possible physics data quality is presented.

* Presented at the Cracow Epiphany Conference "Particle Theory Meets the First Data from LHC Run 2", Kraków, Poland, January 9–12, 2017.

1.2. Purpose of monitoring

The LHCb is one of the most complex scientific devices and, as a scientific tool, it needs constant monitoring and calibration. Precise knowledge of the current conditions of the respective sub-systems of the spectrometer is crucial for high-quality physics results. The Run 1 data taking has revealed that the calibration process may be prone to errors. Each detector must be carefully calibrated, before the proton–proton collision data can be taken. In the case of the VELO, this procedure may require a longer period of time, since a special data sample must be collected in order to determine the calibration constants. On a few occasions this process has been interrupted by external factors, which resulted in incorrect calibration. These mistakes are hard to notice upfront and usually they are noticed after the incorrect constants are already in use. In order to avoid such problems, a dedicated software package for monitoring of the state of the detector has been designed and implemented within the official LHCb framework. It features interactive graphical user interface (GUI) for data visualisation and involved analysis and trending modules for data quality assessment. There are also plans to instrument it with autonomous part that would not require a human operator to decide if the obtained data are of sufficient quality for physics analyses.

2. VELO

The VELO is located closest to the interaction point. It consists of two retractable halves that can be moved out during the LHC beam injection and setting. Both halves comprise of 21 modules, which in turn, have one R- and one Φ-type silicon micro-strip sensor. This allows the detector to register particles using a polar coordinate system.

Each VELO sensor has 2048 silicon strips, giving in total about 170 000 readout channels for the entire VELO. Every channel needs individual calibration to be performed. The silicon strips of VELO detector register a particle using the charge that is generated by its passage through active silicon bulk material. The charge collected on silicon strips is then digitised and expressed in ADC units [2]. The ADC counts are the basic dimension of the signal that is used for reconstruction and calibration. One ADC unit is approximately equal to $450\,e^{-}$.

2.1. Pedestals

Data that are read by different front-end chips (raw data) differ in mean ADC value, as seen in figure 1. This is caused by slight differences in physical properties of silicon and analogue hardware. To account this effect, the pedestal subtraction algorithm is used. The pedestal values (each channel has its own, independent pedestal value) are computed as a mean value of signal over time. The data for pedestal computation are taken in the absence of the proton beam. After the pedestal subtraction, all mean values computed for each channel should oscillate around zero ADC (in channels with no hits).

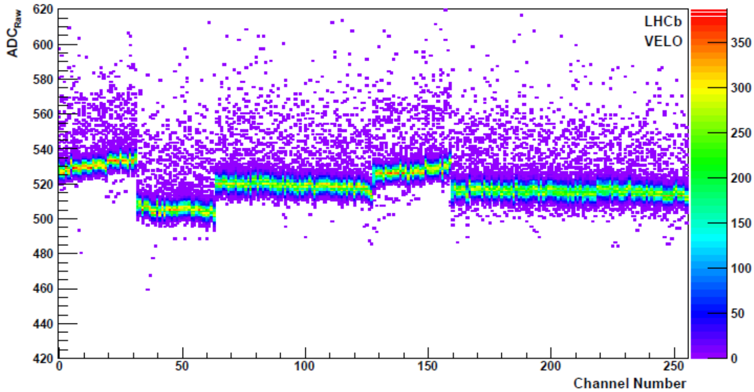

Fig. 1. Raw VELO data, before pedestal subtraction, plotted *versus* channel number. A significant variation is clearly seen.

2.2. Clustering

Two-threshold clustering algorithm is used in order to find the interpolated position of the particle hit. If the measured signal is higher than the high (or seeding) threshold, a hit is observed. The low (or inclusion) threshold, in turn, is used to select channels close to the seeding ones that can contribute to multi-strip clusters. Those two values are computed by using the noise measured for each readout strip. The value of the high-hit threshold for a channel i, $T_{\rm H}(i)$, is determined as $T_{\rm H}(i) = 6\,\sigma_n(i)$, where $\sigma_{\rm n}(i)$ is the noise measured in channel i without the proton beams. Respectively, the low threshold, $T_{\rm L}(i)$, is taken as $T_{\rm L}(i) = 0.4\,T_{\rm H}(i)$. Channels with signals that exceed any of those thresholds are grouped in clusters, which are subsequently used to determine the position of a particle hit on a sensor. Assuming the perfect linear charge sharing model, the hit position is calculated as weighed mean of charges measured in channels that contribute to a given cluster.

2.3. VELO callibration

In summary, there are three major calibration parameters:

— The high-hit threshold — which depends on the noise;

— The low-hit threshold (taken as 40% of the high threshold value);

— Pedestals (mean value over time of signal in channel).

Those parameters are crucial for proper data taking. Monitoring of those parameters is necessary to ensure the quality of the calibration procedure. For this purpose, the monitoring program Lovell was created.

3. VELO monitoring process

The current process of preparing the monitoring data stream for VELO consists of many steps. Some of them could be automated in the future. As seen in figure 2, in the current configuration, the raw VELO data are sent to the LHCb DAQ (data acquisition) system and processed by the TELL1 electronics boards [3]. For the purpose of calibration of the TELL1 processing parameters, the TELL1 boards can be set to output also the raw data. Vetra [4] (the emulation and calibration software) computes the appropriate calibration parameters using the raw data mentioned in Section 2.3. The parameters are stored in a file system. From this point, the calibration parameters can be uploaded to database, analysed by Lovell, or they can be used to create a recipe that would configure TELL1 boards. The calibration parameters are also necessary for the TELL1 emulation software that is used to perform bit-perfect raw data processing that can be compared to the one that is run on the TELL1 boards.

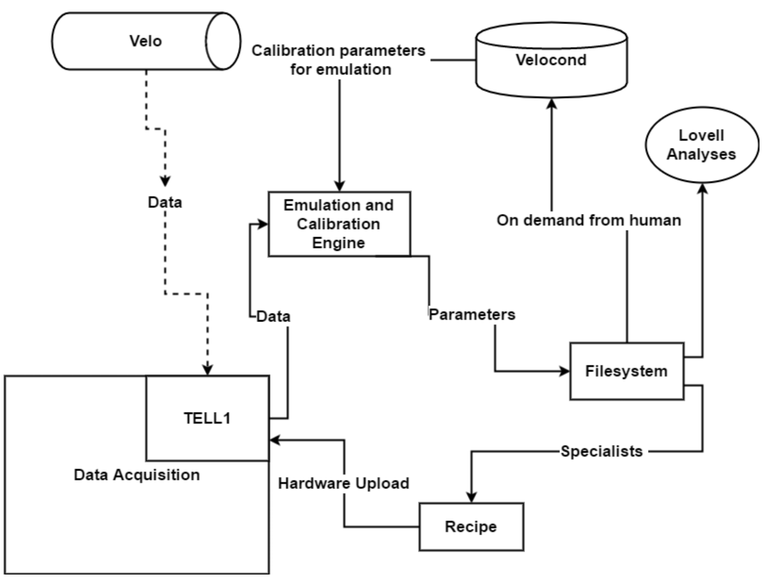

Fig. 2. Current VELO monitoring data flow (see detailed description in the text).

3.1. Lovell

Lovell is an analysis platform implemented in Python. The purpose of this application is to monitor the state of the VELO detector. Lovell was implemented using Python, PyQT, Matplotlib and Root. Recently, the project has been migrated to CERN's GitLab repository manager. Lovell is used to represent calibration parameters or raw data visually as graphs, and to choose between several different ways of representation. This monitoring platform consists of core analysis software and two types of graphical user interfaces; the off-line GUI interface and the web interface. All the data processing is done with the use of the core part of Lovell, and then passed on to graphical interfaces. Data are produced with Vetra software on regular basis. The program is constantly being improved with new features.

4. Future development

As a part of the future development, we propose to modify the process of calibration and monitoring, as seen in figure 3. The basic condition for improving the monitoring quality and the quality of data is to take the noise data and reevaluate the calibration constants on weekly basis. This procedure requires time, it is usually done during the periods with no beam, therefore no physics data are wasted. Fresh calibration data should be au-

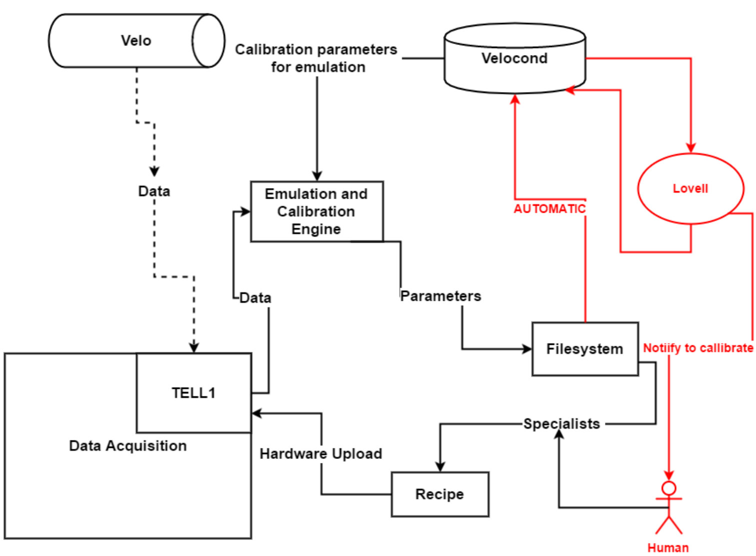

Fig. 3. Suggested VELO monitoring data flow modification. It includes the automation of VELOCOND database update with new calibration constants, calibration data analysis and their quality assessment.

tomatically uploaded to VELOCOND database for future reference. This would enable continuous monitoring of VELO and the changes occurring in data quality. Monitoring software should be equipped for autonomous decision of recalibration, and autonomous marking of calibration parameters as good or bad in VELOCOND database. The monitoring software should also notify the personnel if a recalibration is necessary.

5. Conclusions

The calibration process of VELO is highly important for continuity of precise measurements with the LHCb experiment. The calibration must be monitored to ensure its correctness, and if possible, the monitoring should be automated to minimize human involvement in the process.

This research was supported in part by the National Science Centre, Poland (NCN), grant No. UMO-2015/17/B/ST2/02904 and the Ministry of Science and Higher Education, Poland.

REFERENCES

[1] LHCb Collaboration *Int. J. Mod. Phys. A* **30**, 1530022 (2015) [arXiv:1412.6352 [hep-ex]].

[2] LHCb VELO Group, *JINST* **9**, P09007 (2014) [arXiv:1405.7808 [physics.ins-det]].

[3] G. Haefeli *et al.*, *Nucl. Instrum. Methods Phys. Res. A* **560**, 494 (2006).

[4] T. Szumlak, *J. Phys.: Conf. Ser.* **219**, 032058 (2010).