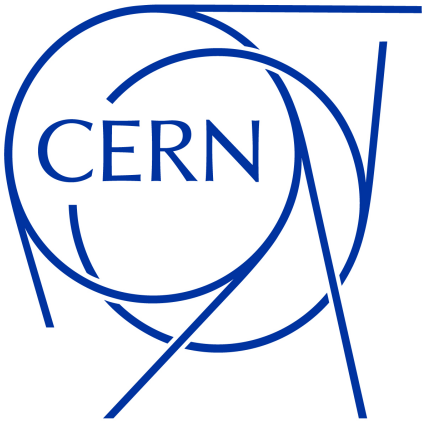
SUMMER STUDENT PROJECT

Luminosity calculation and threshold scan for the VELO Pix

August 2023

Léa Girard, supervisor Nathan Jurik



ABSTRACT

During my summer project I calculated the rates of the MonB counters from the VELO Pixels. Using these rates, I then studied the luminosity of the collisions in the LHCb detector. Studying archiving of the MonB data from past runs we realized that the rates and luminosity have a linear relation and the linear fit parameters can be used to calculate luminosity in real time.
Secondly, the VELO team has a goal of setting up all of the ASIC DAC threshold so that the noise rates are all close to a goal rate. I developed an algorithm to set these threshold automatically in a single scan.

1 INTRODUCTION

The LHC (Large Hadron Collider) is a hadron accelerator located at CERN (Conseil Européen pour la Recherche Nucléaire, now European Organization for Nuclear Research). It is composed of two rings where hadrons can be accelerated using supraconducting magnets and radio frequence cavities in a 27km long tunnel about 100m deep underground. There are four points where the two counter rotating beams collide, which correspond to the four main experiments at LHC : ATLAS, CMS, ALICE and LHCb.
The LHCb (Large Hadron Collider beauty) experiment is located at the interraction point 8 and is an experiment dedicated to heavy flavour physics. It is used for many areas of High Energy physics studies, but was designed for studies of the CP violation and rare events involving charm and beauty quarks. One of the main physics goals for this experiment is to explain the beyond-standard-model levels of CP violation that correspond to the amount of matter in the universe. Thus, rare decays must be studied with an experiment capable of producing large amounts of b and c hadrons. With the high luminosity made available by the LHC, the LHCb needs to have an efficient event trigger sensible to many different final states, excellent vertex and momentum reconstruction to identify many different types of particles and finally a robust data acquisition system.
Contrary to the more traditional setups of the ATLAS and CMS experiments that completely cover the interaction point, LHCb is a forward angular detector. Only particles with angles of 10 mrad to 300 mrad (250 mrad on the non-bending plane) can be reconstructed. This is explained by the fact that high energy b hadrons will be preferably emitted in this forward region.

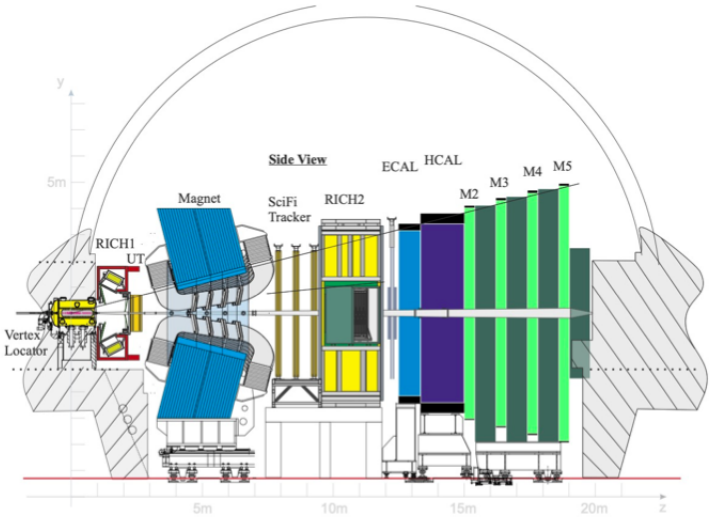


Figure 1 – LHCb detector

The VELO (Vertex Locator) is the first sub-detector in the LHCb. It is a silicon detector tracking ionizing particles to calculate the location of the initial collision and the decay vertices of B hadrons. It is the first step in the LHCb event reconstruction programs as well as providing information for event selection.

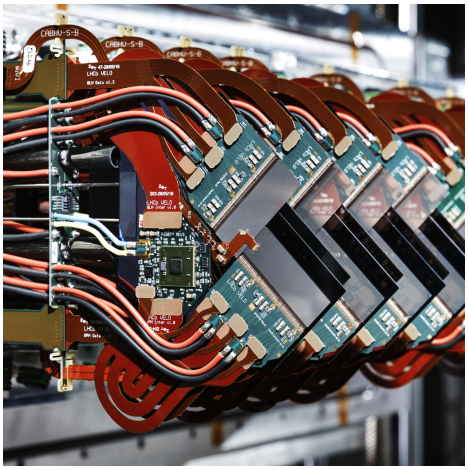


Figure 2 – The VELO modules at the final stages of construction

1.1 VELO

The VELO aims to be as close to possible to the beam line to detect the particles precisely and detect the particles that will later enter the LHCb detection region and be visible by the later detectors. The goal distance from the interaction point is at about 3 mm for the interface with the beam pipe, which places the silicon detectors at about 5 mm from the beam. To make this possible, the VELO is a movable device : it moves out during the injection and acceleration proces of the LHC and only closes in when the beam is colliding with the

desired stability.
The pixelated tiles, the ASICs (Application Specific Integrated Circuits) and their services are assembled into a series of identical modules, arranged perpendicular to the beam line.The detector is composed of 52 modules, separated equally with the even numbers on the C side and the odd on the A side and spread out unevenly around the interaction point, with a higher density closer to the collision and a few forward and backward modules situated further out. Each module is composed of 4 sets (named VP) of 3 aligned ASICs. Two VPs are set in an L shape on either side of a module.

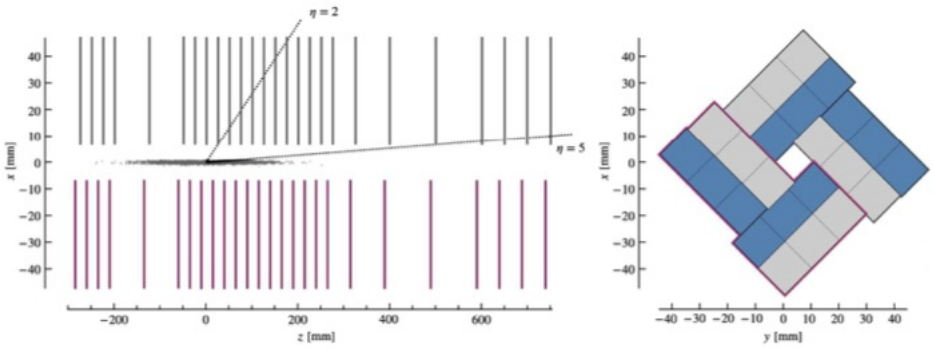


Figure 3 – The VELO modules position with luminous areas represented (right) and layout on the ASICs in the closed configuration (left) with the upstream modules in grey and C side modules highlighted in purple

The VELO pixel chips (VeloPix) are ASICs specifically designed for the VELO, with a pixelated silicon sensor relaying data to the rest of the electronic set-up. They are cooled by microchannels with liquid CO2. The VeloPix are the part of the VELO sub detector my project focused on.

1.2 Context of this work

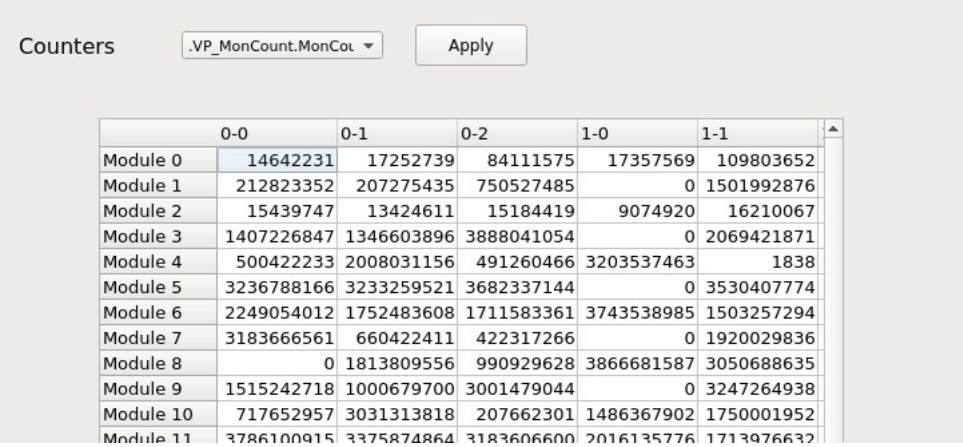
For my project I have worked with the VeloPixel counters. The ASICs come with a wide range of 32 bits counters to monitor various quantities, as well as other information important to their services like temperature. One of these counts the number of packets sent out corresponding to "hits" in the pixel matrix and is referred to as the MonB counter. This is the counter I specifically studied and used in my project.

These counters and other information given in real time by the VELO pixels can be accessed and used thanks to SIMATIC WinCC, a SCADA (Supervisory Control and Data Acquisition) system. It is used by the VELO team to create graphical interfaces and run calculations on real time data. WinCC is used to create panels, which correspond to a window visible on screen, with graphical elements and code written in a C++ like language with added functions specific to WinCC.
There are many control panels used in the control room to configure the VELO, monitor different parameters during the run and create or acknowledge alerts.
WinCC handles data through Data Points, a set of values grouped under a name that are constantly modified when the electronics read out data. These are collected in data point types that define the values (data point elements) that will exist for each data point. For the VELO, the individual data points tend to represent the 624 ASICs. These data points can be archived or not and the amount of data archived can be controlled thanks to smoothing.
For my summer student project, I worked specifically with the MonB counters to calculate their rates and from then create WinCC panels useful to the VELO team.

2
RATES COUNTERS

2.1 Counters table panel

The counters I accessed in my project are separated in four systems, separated by A side and C side as well as two groups of modules of numbers under or over 25. The counters table panel is simply a display of the chosen counter from the Velo Pix counters.



	0-0	0-1	0-2	1-0	1-1
Module 0	14642231	17252739	84111575	17357569	109803652
Module 1	212823352	207275435	750527485	0	1501992876
Module 2	15439747	13424611	15184419	9074920	16210067
Module 3	1407226847	1346603896	3888041054	0	2069421871
Module 4	500422233	2008031156	491260466	3203537463	1838
Module 5	3236788166	3233259521	3682337144	0	3530407774
Module 6	2249054012	1752483608	1711583361	3743538985	1503257294
Module 7	3183666561	660422411	422317266	0	1920029836
Module 8	0	1813809556	990929628	3866681587	3050688635
Module 9	1515242718	1000679700	3001479044	0	3247264938
Module 10	717652957	3031313818	207662301	1486367902	1750001952
Module 11	3786100915	3375874864	3183606600	2016135776	1713976632

Figure 4 – The counters table panel

In all of this report the WinCC screenshots have been taken during the LHC shutdown of august 2023 caused by a damaged power line. During this shutdown the LHCb team was still turning on the sub detectors to take advantage of this down time to work out issues and prepare for a heavy ion run. The VELO in particular was injecting data from a past collision as if it were real time data. Thanks to this, I was able to test my panels as if the data was physics data from real time collisions, however some details might be different from a real run situation.

2.2 Rate calculation and data points

The rates of the MonB counter are calculated using a script : lbECS/Counters.ctl. This allows it to be calculated all the time. The rates are recorded in the VP Counters data point type in data points with names of the type "VEECS :Module0_VP1_2".
This data point type has 4 attributes : luminosity (which will be explained later in this report), rate, PreviousTime and PreviousValue. While luminosity and rate are useful values that will be displayed, the last two are used only to calculate the rate.

The rate is calculated every time the value of the counters change by dividing the difference in value by the lapse of time since the last counter update. This calculation is only done if the ASIC is in state "ready" to make sure no wrong values are being calculated when the ASIC is in error.
The script also calculates sums of these rates : the "VEECS :Averages" data point's rate value is a global sum over all the rates. The ASIC in state not ready will not cause major changes to the value of this sum as they are still being represented by the last value of the rate in state Ready.

The data points of the type "VEECS :Downstream_VP1_2" exist for the three ASICs of VP 1 and 3 as well as VP0_2 and 2_2 and their rate value is the sum of the rates of the downstream modules 44 to 52 on the specific ASIC. This choice aims to get specific values of rate for the ASICs with less hits : the ASICs further away from the interaction point because they are part of the most downstream modules as well as further out from the center of the beam pipe on each of these modules. This choice was motivated as the lower occupancy is beneficial for luminosity counters, as well as for long-term stability due to lower radiation damage.

2.3 The rates panel

This panel displays the MonB counter and the MonB rates in two tables. Colours are used to highlight concerning values of the rates : a light orange denotes a low value (probably an ASIC being off) and the red notes a surprisingly high rate. The red color limit is defined as a certain percentage over the average rates of the VP closest to the interaction point.

A button creates a child panel that displays the plots of rate over time for all the ASICs of two modules. The trending plots are based on pre-existing code from the VELO team with modifications to adapt to my own data points.

Figure 5 – The rates panel

Figure 6 – The trending child panel

3 LUMINOSITY STUDY

One of the goals of this summer student project was to use the VeloPix counters to calculate the luminosity in the LHCb. The approach chosen was to study the archiving of MonB counter data from past mu runs which had a recorded luminosity value to find a link between counting rate and luminosity.

3.1 Study of past mu runs

The MonB counters values of past runs are recorded in one text file per module and per run, featuring one line for each time one of the ASIC's counter changed. On each line the time and then the values of the MonB counters of each ASIC are written, with the first line recording the name of each of these variables in order.

```
1.  | mu  target | mu  measured | run    |
2.  | ---------- | ------------ |--------|
3.  | 1.1        | 1.06         | 268205 268213|
4.  | 1.6        | 1.46-1.77    | 268243 268246 |
5.  | 4.0        | 4.00         | 268248 268249 268253 |
6.  | 6.6        | 6.4          | 268257 |
7.  | 8.6        | 8.45         | 268260 |
8.
9.
10. 264589 : noisy
```

Figure 7 – List of the runs used to study the luminosity

The files can easily be read with a Python code. For each ASIC, I calculated the rate when each value changed (not including the first value because the associated time, that of the beginning of the run, is arbitrary). I calculated the average of all of the rates seen in the runs with the same luminosity value.
I then plotted these rates against the luminosities : results did seem very linear, as we had hoped, so they could be fitted with a linear function. Using different runs with the same luminosity value, when they were available, turned out to be very useful in order to get a more meaningful average rate and improve the quality of the fit.

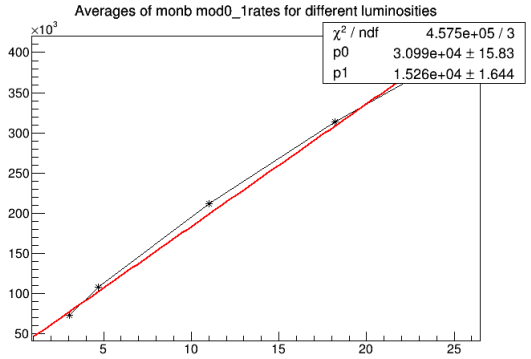
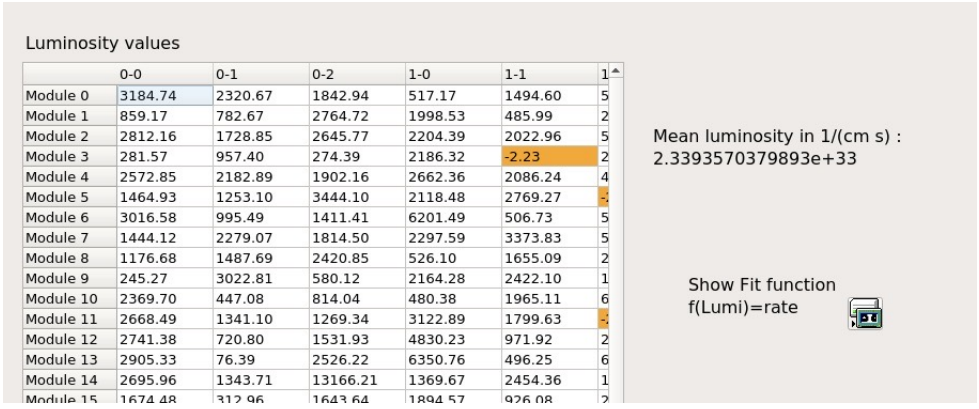


Figure 8 – The plot of rate depending on luminosity and its fit function

3.2 Luminosity calculation

The fit parameters from the study of past mu runs are written to a file that can be accessed from WinCC and read easily. The fit function is used to calculate the luminosity from the current rate in real time. This once again is done in a script so the process is always running.
The archiving for the luminosity data point values can be turned on and off depending on the current needs and the value of the luminosity average can be communicated to the LHC.
The luminosities calculated are displayed in the **calculating_luminosity_VeloPix** panel. The average luminosity over all ASICs is also displayed to make it easier to get an idea of what the luminosity is : while the full table of 624 values is very interesting to observe values in specific ASICs closer or further from the beam line, or in different groups of modules, a quick glance is also important. A button also displays a table with the fit functions $f(Lumi) = rate$ so they can be referred to and used to check orders of magnitude.



	0-0	0-1	0-2	1-0	1-1	1
Module 0	3184.74	2320.67	1842.94	517.17	1494.60	5
Module 1	859.17	782.67	2764.72	1998.53	485.99	2
Module 2	2812.16	1728.85	2645.77	2204.39	2022.96	5
Module 3	281.57	957.40	274.39	2186.32	-2.23	2
Module 4	2572.85	2182.89	1902.16	2662.36	2086.24	4
Module 5	1464.93	1253.10	3444.10	2118.48	2769.27	-
Module 6	3016.58	995.49	1411.41	6201.49	506.73	5
Module 7	1444.12	2279.07	1814.50	2297.59	3373.83	5
Module 8	1176.68	1487.69	2420.85	526.10	1655.09	2
Module 9	245.27	3022.81	580.12	2164.28	2422.10	1
Module 10	2369.70	447.08	814.04	480.38	1965.11	6
Module 11	2668.49	1341.10	1269.34	3122.89	1799.63	-
Module 12	2741.38	720.80	1531.93	4830.23	971.92	2
Module 13	2905.33	76.39	2526.22	6350.76	496.25	6
Module 14	2695.96	1343.71	13166.21	1369.67	2454.36	1
Module 15	1674.48	312.96	1643.64	1894.57	926.08	2

Figure 9 – The Luminosity panel running

4
NOISE THRESHOLD SCAN

The second part of my project was aiming at creating an algorithm to perform a noise threshold scan on the VeloPix ASICs. The goal of this scan is to have a noise rate (rate of the MonB counters) as close as possible to a certain goal rate.
This is done by changing the DAC threshold for each ASIC until the rate is correct. It is important to note that this scan is to be done when the detector is not in use, so that the MonB counters do represent pure noise. As the conditions needed for this scan corresponded to the LHC status during my stay at CERN I was able to test it fully.

The WinCC architecture supports specific data points that communicate both ways with the VELO ASICs, so a WinCC panel can act on the electronics of the VELO to change the DAC threshold.
The first step was to create a panel so I could manually change the threshold of one ASIC and observe how the MonB rate reacted. The rates panel I had made previously was very useful and I could observe the trending of the rate for the specific ASIC I was working on and see the impact of the threshold value directly. However, with 624 ASICs to calibrate, this manual process was way too long and difficult, so I worked on creating an algorithm to scan all of the ASICs with one click. This is done in the **NoiseThresholdScan** panel.

4.1 The algorithm and the panel

The algorithm chosen works in a few steps, noted on figure 10 (the changes in the rate correspond to changes of the threshold). It begins by setting the threshold to the value in the settings, then goes through the following steps :
1 : If the rate is higher than the goal, the threshold is increased until the rate drops
2 : The threshold is then decreased by larger increments until the rate goes up
3 : At the end of the second step of the algorithm, the rate goes up very sharply. The threshold is then increased to get back to the last situation before the rate went over the goal value.
4 : The rate goes back down, the threshold is now decreased by very fine increments until the rate reaches the desired value

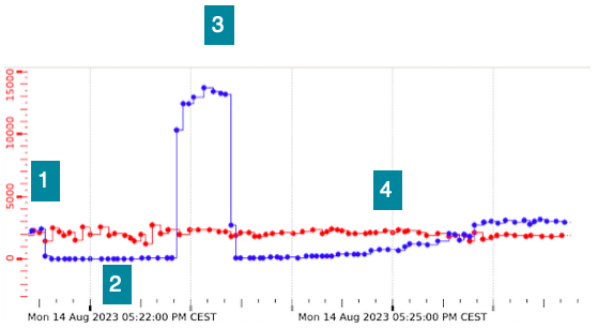


Figure 10 – Rate evolution during the Threshold Scan

Running this algorithm takes about 5 minutes as there needs to be a delay between each time a new threshold is set and the next rate read. This delay allows time for the new threshold to be taken into account by the DAC and for the MonB counters to be updated twice, in order to have a rate reflecting the current threshold.

To make the scan of all 624 ASICs faster, we decided to use threads to parallelize the scan on 12 ASICS at a time : this way, all the ASICs of one module can be calibrated at a time and the scan of the whole VeloPix takes less than 5 hours.

The final panel has three buttons : one doing the calibration for only one selected module, one running the calibration on all of the modules, and the last one resetting all thresholds to the settings value. A desired noise rate can be entered before starting any of these processes, the default rate if none is given by the user is 2000Hz. To make sure that only one process is running at a time and that the number of threads opened remains manageable, the buttons are turned off until all of the threads opened have finished running.
The lower half of the panel displays the module currently being calibrated (this can be used as an approximation of the remaining run time during a full scan) and a table displaying the rates for each of the ASICs being calibrated.

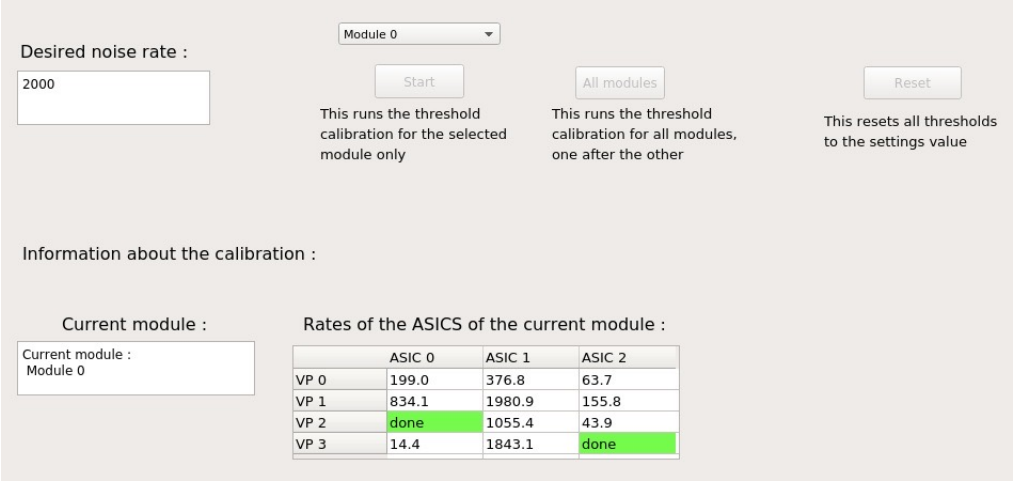


Figure 11 – The NoiseThresholdScan panel during a scan

4.2 Results

After the threshold calibration process, most of the ASICs have rates around 1000Hz from the goal rate. However, some of the ASICs show another pattern of noise : they have alternating very high and low rates which create spikes of noise. This noise characteristic cannot be resolved by any change in the DAC threshold and remains consistent over time, it seems to be a characteristic of the ASIC itself.

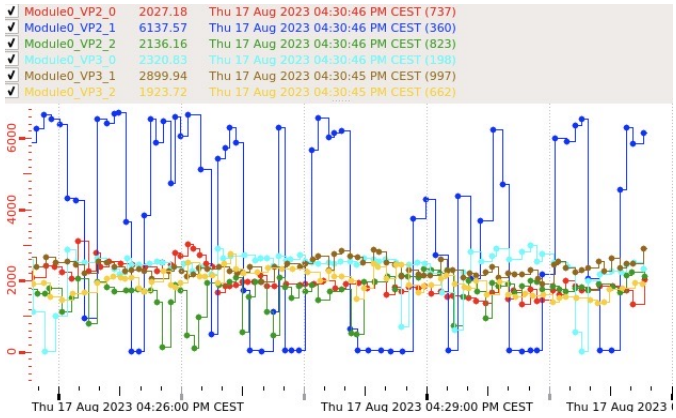


Figure 12 – Results of the noise threshold calibration process with one noisy ASIC

These noise spikes need to be detected during the threshold scan in order to protect the process. Indeed, if a very high or very low rate coming from a spike was to be considered as the average rate with this threshold, the scan could end up with a wrong result.
This protection system also works to detect spikes : since having such a noise pattern seems to be inherent to the ASICs, the ASICs where noise spikes are detected are recorded in a file. This may be used later to keep the noisy ASICs out of some calculations, such as a luminosity average, or to study them in more detail to figure out the origin of this unusual behaviour.

In addition to recording which ASICs are noisy, the chosen thresholds are recorded at the end of the calibration process. With this file, the new thresholds could be set as default in the settings so the scan does not have to be run every time the VELO is turned off.

5 CONCLUSION

To conclude, during my time as a summer student at CERN, I have worked on the MonB counters of the VELO Pixel in order to calculate their rates, which have then been used to calculate the luminosity and to run noise threshold scans.

I have created data points in the VPCounters data point type in the VEECS system that are being updated by the lbECS/Counters script. They record the MonB rate and luminosity of all the ASICs as well as relevant averages. I have created four WinCC panels for which this report can serve as documentation in addition to the commentaries in their codes.

The **Counters table** panel displays all of the VP counters available.
The **MonB rates** panel displays the rates of the MonB counter and can create plots of these rates over time in a child panel.
The **Luminosity calculation** panel displays the luminosities calculated for each ASIC based on a linear fit of the relation between MonB rate and luminosity from past runs.
Noise Threshold Scan is the last panel I created, it allows the calibration of DAC thresholds for each ASIC in order to have the desired noise rate.

Thank you to the VELO team and my supervisor Nathan Jurik for giving me the opportunity to work on this amazing project, I hope the panels I developed will be useful to the VELO and LHCb teams.

REFERENCES

References for the introduction paragraph and the included pictures and diagrams :
Lyndon Evans and Philip Bryant 2008 JINST 3 S08001
The LHCb Collaboration et al 2008 JINST 3 S08005
T. Poikela et al 2017 JINST 12 C01070
The LHCb Collaboration LHCb-DP-2022-002