LHCb-INT-2014-014
LPHE-2014-001
June 19, 2014

ST TELL1 calibration in 2011 and 2012

F. Dupertuis[1].

[1] *Ecole Polytechnique Féderale de Laussane, Lausanne, Switzerland*

Abstract

The ST DAQ chain from the beetle chip to the TELL1 board is briefly summarised. The TELL1 processing and the calibration of the parameters used to perform the zero-suppression is presented. New beetle header bits effects are found and upgraded calibration strategy is presented.

Contents

1 Beetle chip, digitizer board and TELL1 board

The front-end chip that is used for both the IT and TT is version 1.3 of the *Beetle* [1] originally developed for the VELO. The Beetle is responsible for acquiring, shaping and storing signals into analogue pipelines with a given *Pipeline Column Number* (PCN) and transmitting signal information for L0-accepted events. Each Beetle reads 128 micro-strips and sends the data through four analogue ports to the *digitizer boards* [2] via 5m copper cables. Figure 1 shows the information chain sent by the Beetle at 40 MHz on the four ports and Fig. 2 the legend of the chain content. Each analogue port sends first a header and then the information of the 32 strips follows. The header information is given by the two voltage states corresponding to a *pseudo-bit*.

The digitizer board contains an *analogue-to-digital converter* (ADC). These boards are located in *service boxes* [3]. These service boxes are placed at the bottom of the IT

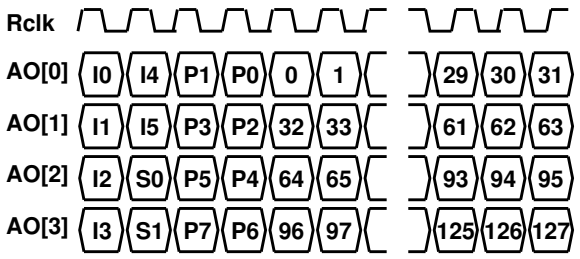

Figure 1: Layout of the signal chain sent by the Beetle through four analogue ports. [1]

Bit		Description
I0	LeadingBit	always active (= 1)
I2	ActiveEDC	1 indicates active error detection and correction (EDC) logic
I3	ParCompChTh	(even) parity of register *CompChTh* (reg. no. 20, cf. table 14)
I4	ParCompMask	(even) parity of register *CompMask* (reg. no. 21, cf. table 14)
S0		LSB of register *SEUcounter* (reg. no. 23, cf. table 14)
S1		bit 1 of register *SEUcounter* (reg. no. 23, cf. table 14)
P0		LSB of pipeline column number
P1		bit 1 of pipeline column number
P2		bit 2 of pipeline column number
P3		bit 3 of pipeline column number
P4		bit 4 of pipeline column number
P5		bit 5 of pipeline column number
P6		bit 6 of pipeline column number
P7		MSB of pipeline column number
special for ***Beetle1.3***:		
I1	ParPCN	(even) parity of pipeline column number (PCN)
I5	ParTpSelect	(even) parity of register *TpSelect* (reg. no. 22, cf. table 14)
special for ***Beetle1.4* and *Beetle1.5***:		
I1	ParTpSelect	(even) parity of register *TpSelect* (reg. no. 22, cf. table 14)
I5	ParPCN	(even) parity of pipeline column number (PCN)

Figure 2: Legend of the signal chain sent by the Beetle through four analogue ports. [1]

stations and at the side of TT, both outside the LHCb geometrical acceptance. The signal amplitude is coded in 8 bits. The output signal of the Beetle has a non-zero baseline and the gain is set such that the baseline corresponds to 128 ADC counts. The pseudo header-bit 0 (1) is 40 ADC counts below (above) the baseline. The average ADC count for the strip without signal is called the *pedestal* and its spread the *noise*. The properties of the Beetle chip were characterised and its parameters tuned during test beams periods. It has an I^2C *interface* to set the operating parameters of the chip with the slow control system. The reminder of the signal after 25 ns is about 30% of its peak voltage. The result of the digitizer board is further shipped to *TELL1 boards* using *Vertical-Cavity Surface-Emitting Lasers* (VCSELs) through optical fibres.

In the LHCb experiment, the acquisition, the *zero-suppression* and the *clusterisation* are performed by *TELL1 boards* developed at EPFL [4]. The zero-suppression is mainly the process that subtracts the pedestal in order to identify strips with signal. The clusterisation is the process that groups together strips with signals coming from the same particle, called *clusters*. The TELL1 boards are located in an area with low radioactivity. The ST group uses 90 TELL1 boards to process the data coming from the IT and TT sub-detectors via optical fibres. Twelve optical fibres are packed in one ribbon cable. One TELL1 board has two optical receivers using two ORX mezzanine cards. The data is split in four and processed by a *Parallel-Processing-Field-Programmable Gate Array* (PP-FPGA) each. Every PP-FPGA has twelve processing channels to perform the zero-suppression and clusterisation. The result of this process is sent to the trigger farms for event reconstruction. Figure 3 shows the full read-out chain used by the IT and TT. Table 1 summarises the

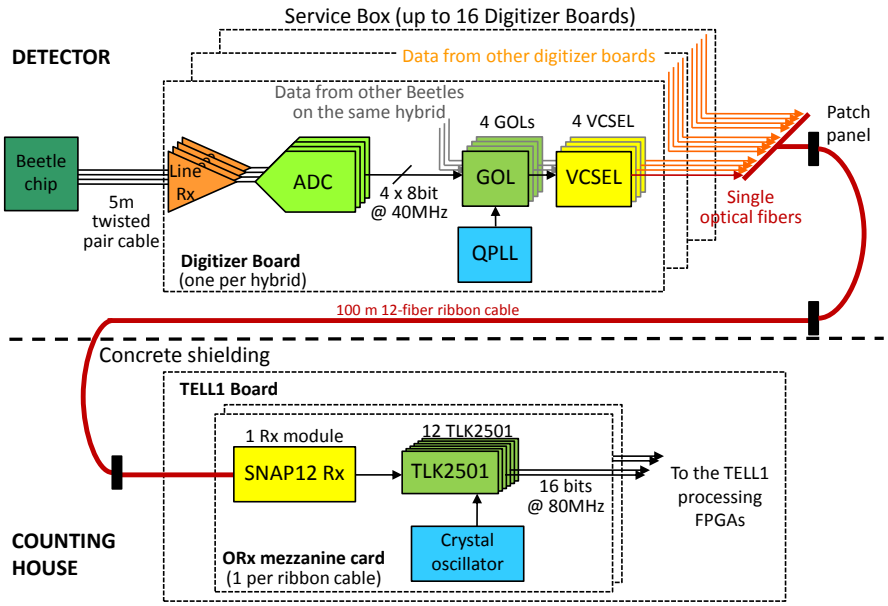

Figure 3: Read-out chain used by the IT and TT for the Beetle to the TELL1 board passing through the digitizer boards.

Table 1: The conversion table between the different granularities and their corresponding number of channels.

1 Strip	$\leftrightarrow$	1 channel
1 Beetle port	$\leftrightarrow$	32 channels
1 TELL1 PP-FPGA processing channel	$\leftrightarrow$	64 channels
1 Beetle/VCSEL	$\leftrightarrow$	128 channels
1 Read-out sector	$\leftrightarrow$	384/512 channels for IT/TT
1 TELL1 PP-FPGA	$\leftrightarrow$	768 channels
1 TELL1	$\leftrightarrow$	3072 channels

granularities present for IT and TT and the corresponding number of channels.

The TELL1 can be configured in two ways using either *configuration* (CFG) *files* or *PVSS recipes*, where *PVSS* is the software framework that is used to control the LHCb detector. The TELL1 recipes can be created from CFG files or XML files. The creation of TELL1 recipes from XML files is performed using a PVSS library and control interface developed at EPFL [5]. The XML files are called *config XML* files and they are stored in a conditions database called *COND*.

2 TELL1 board processing

For the zero-suppression and clusterisation processes of the TELL1, three consecutive steps are made. First, the *header correction*, followed by the pedestal subtraction, the *linear common mode subtraction* (LCMS) and finally, the clusterisation.

Before reviewing the different processing steps, the different sources that can form clusters should be identified. The first source is the signal, i.e. charge deposition from a charged particle crossing the silicon. The second one is noise due to thermal excitation of the silicon. The last source is *spillover*, also called *out-of-time* pileup in other experiments, where clusters contain signal from the previous bunch crossing.

In order to monitor the TELL1 processing and calibrate the clustering parameters, a bit-perfect emulator of the TELL1 processing was developed at EPFL [6], hereafter called the *TELL1 emulator*. It is written in C++. Four relevant types of data are available with the TELL1 emulator: *Non-zero suppressed ADC counts* (NZS ADCs), *pedestal-subtracted header-corrected ADC counts* (PedSub ADCs), *ADC counts after LCMS* (LCMS ADCs) and the *list of clusters*. Among those types, only the NZS ADCs and the list of clusters is available with the TELL1.

The three processes that the TELL1 is performing are described in the following lines.

2.1 Header correction and pedestal subtraction

One of the known features of the information sent by the Beetle chip is that the pedestal values of the first six strips, for each port, shift when any of the header bit changes its

status. This is known that 9 out of 16 header bits (P0 to P7 and I1 in Fig. 1) change their status continuously since they compose the PCN counter.

Two other header bits (S0 and S1 in Fig. 1) are expected to change their status as well, since they compose the counter for the *Single Event Upset* (SEU). The SEU is due to large ionisation by a particle hitting an electronics element of the Beetle chip and changing its status. This is a very rare phenomenon and changes of those header bits were not seen until a procedure was established for long-term monitoring of the ST data. Figure 4 demonstrates that the pedestal value shifts are correlated with at least the status of those two header bits. The remaining header bits are not supposed to change their status.

However, a careful look at results presented in Fig. 4 show that the header bit (I4) is changing as well and further studies showed that the header bit (I3) is also changing. Those two header bits are the parity bits of the two registers CompChTh and CompMask, which are not usually initialised when a data taking is resumed after the injection of new beams in the LHCb where the Beetle chips are switched off. Unfortunately, the registers do not come back to the same states after switching on the chips. Based on this discovery, it was decided to initialise all the Beetle registers after every switch-on. Despite this new measure, the shift of the pedestal values are inevitable as explained above for the header bits-S0 and S1. For the clustering, this header dependent shift of the pedestal values must be corrected in the TELL1, based on the transmitted header. Since the 9 header bits are continuously changing, corrections to the pedestal values are easily obtained from calibration runs where non-zero suppressed data are available. Statistics of the calibration data available for the other header bit cases are too small to obtain corrections. Therefore, corrections are made based on the available corrections for the case of the 9 headers. For this, we assume that the shift in the pedestal values are due to the charge of the pseudo-bit in the proceeding time bucket in the same analogue port. For example, the expected pedestal value for the Strip-0 data with the status of bit-I4 changed, in Fig. 1, is identical to that of Strip-1 data with the status of the bit-P1 changed. This assumption

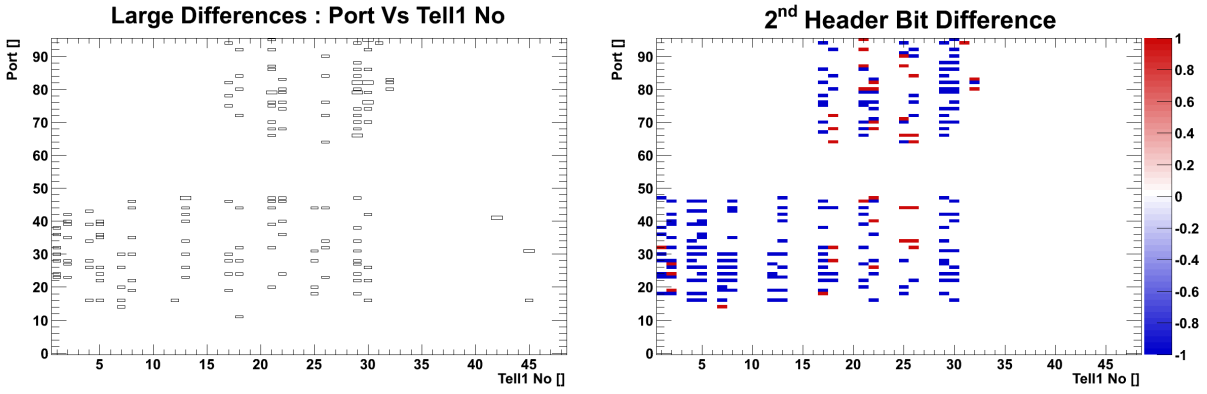

Figure 4: Shift in pedestal values between two set of calibration results by TELL1 and Beetle port numbers (*left*) and second header bit flips by TELL1 and Beetle port numbers (*right*) for the TT.

was successfully tested with the calibration data as shown in Figs. 5 and 6. Based on this assumption, pedestal value corrections for all the header configurations of the three last header bits are calculated and applied for the pedestal subtraction process.

For the pedestal subtraction process, the calibrated pedestal is subtracted for each channel from the ADC count obtained after correcting for header effects. This gives the PedSub ADCs dataset. In the case that the channel is masked, i.e. the *channel mask* is equal to 1, the PedSub ADC is set to 0.

2.2 Linear common mode subtraction (LCMS)

In order to perform the LCMS process and form clusters, thresholds are used. Each threshold is set as a coefficient times the noise level in ADC counts. These *threshold coefficients* correspond to given values of the *signal-over-noise ratio* (S/N) and they are summarised in Table 2. This gives a constant level of noise clusters whilst maintaining the signal hit efficiency. The thresholds coefficients were optimised using test beam data to find the best compromise between signal clustering efficiency and noise clusters rate.

During test beam studies, it was discovered that the Beetle chip has some large common

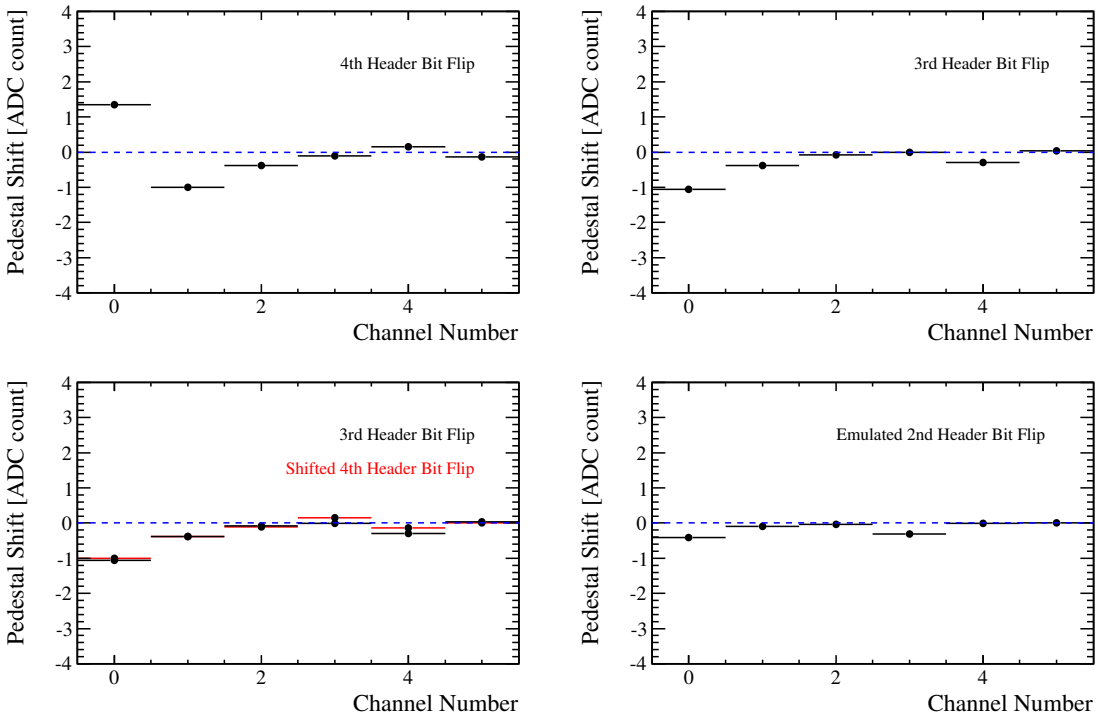

Figure 5: Pedestal shift between pseudo-bit 1 and 0 for the six first signal channels of IT for the last header bit (*top, left*) for the third header bit (*top, right*). Overlay of the two top plots where the first plot results are shifted by one channel (*bottom, left*). Emulated Pedestal shift between pseudo-bit 1 and 0 for the second header bit (*bottom, right*).

Table 2: Clustering threshold coefficients.

Threshold type	Coefficient
Hit	2.5
CMS	4
Confirmation	5
Spillover	10

mode noise. The common mode has two components, a global shift and a slope as a function of the ADC counts for every 32 channels corresponding to one analogue port of a Beetle chip. Therefore, the TELL1 first subtracts the average ADC counts of the 32 channels from each channel. To avoid that signal biases the average and slope estimates, the signal channels are identified by their ADC counts being larger than a threshold value ($> 4 \times$ LCMS noise), called *CMS threshold*. A temporary ADC count vector is formed with the ADC of the signal strips set to 0. Finally, the slope is computed using the temporary vector and subtracted from the ADC counts that includes signal. The result of the LCMS gives the LCMS ADCs dataset. The noise ADC value that one obtained after the LCMS

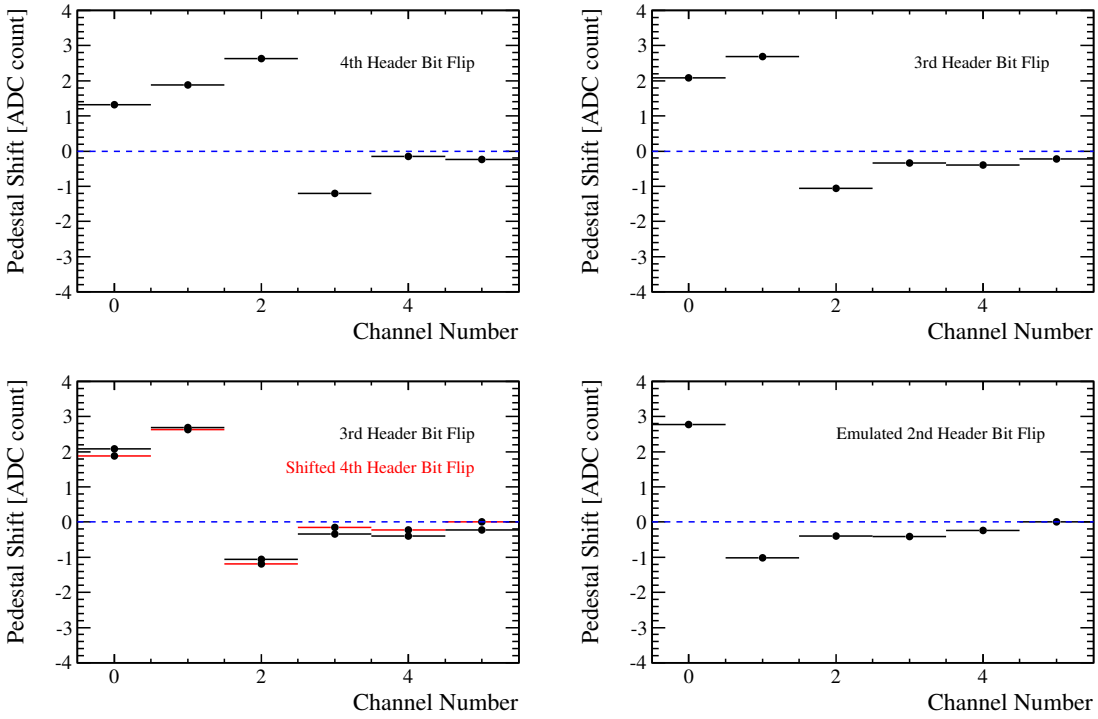

Figure 6: Pedestal shift between pseudo-bit 1 and 0 for the six first signal channels of TT for the last header bit (*top, left*) for the third header bit (*top, right*). Overlay of the two top plots where the first plot results are shifted by one channel (*bottom, left*). Emulated Pedestal shift between pseudo-bit 1 and 0 for the second header bit (*bottom, right*).

Table 3: Calibration parameters with their granularities.

1 Pedestal	/	channel
1 Hit threshold	/	channel
1 CMS threshold	/	channel
1 Channel masking	/	channel
1 Confirmation threshold	/	TELL1 PP processing channel
1 Spillover threshold	/	TELL1 PP processing channel
48 Header corrections	/	Beetle port

Table 4: Location of parameters used and resulting from the calibration in the COND and LHCBCOND databases.

COND	LHCBCOND
Channels masks	Strips masks
TELL1 analogue links masks	Beetles masks
TELL1 links masks	Sectors masks
Pedestals	Common mode noise
Header corrections	LCMS noise
Thresholds	Thresholds coefficients

process is called *LCMS noise*.

2.3 Clusterisation

The clusterisation starts from a set of LCMS ADCs. The first step is to search for signal by looking for *seeds*. Seeds are channels having their LCMS ADC counts larger than a threshold value ($> 2.5 \times$ LCMS noise), called the *hit threshold*. Clusters are then formed by combining adjacent seeds. The cluster size cannot exceed 4 strips, if this happens the cluster is split. Finally, the cluster is accepted if its charge is larger than a certain threshold ($> 5 \times$ LCMS noise), called *confirmation threshold*. If the cluster charge is lower than the *spillover threshold* ($< 10 \times$ LCMS noise), and higher than the confirmation threshold, the spillover flag is set to 1. The spillover flag is meant to indicate that the cluster could be generated by a charged particle from the previous bunch crossing. It can be used to discard these clusters from the previous spill by cutting on the number of hits on the track with spillover flag set to 1. The position of the cluster is determined from a weighted average of its ADC channel counts.

All the parameters needed for these processes are summarised in Table 3 with the required granularity for the parameters. In total, 3072 pedestals, masks, hit and CMS thresholds, 4608 header corrections and 48 confirmation and spillover thresholds need to be calibrated. Table 4 summarises in which database the relevant parameters are stored.

3 TELL1 board calibration

These are the particular contributions made to the calibration, operation and monitoring of the IT and TT during this research work over the period from 2011 to 2013:

- improvement and correction of the existing calibration codes.
- creation of scripts to mask strips, Beetle ports and Beetles.
- creation of a script that analyses the noise to look for problematic channels.
- creation of a script that launches the calibration chain in a single job.
- commissioning of the PVSS user interface to create TELL1 recipes.
- commissioning of inter-fill calibration data taking process.
- automatic offline monitoring using inter-fill calibration datasets.
- maintenance of the ST LHCBCOND and COND databases.

The calibration process is composed of five unique steps:

1. Channel masking
2. Pedestal calibration
3. Header correction calibration
4. Clustering parameters calibration
5. Monitoring

All these steps are described in the following sections.

4 Channel masking

An important ingredient to perform any calibration is to start from an up-to-date list of masked channels. The very first list was made after the module production, burn-in tests and quality checks. Every time a channel needs to be masked, the config XML is updated and new TELL1 recipes are created. Several sources of channel masking were identified during the three years of operation of the IT and TT: *Broken bonds*, *dead VCSELs*, *noisy strips* and *broken connector pins*.

Apart from the TELL1 processing, the masked list is important for cluster decoding, monitoring, simulation and measuring performance of the detectors such as the hit efficiency described in Sec. **??**.

The bond between the micro-strip and the pitch adapter can break. In order to keep a robust and reliable TELL1 processing, it is important to mask these bonds before the

next LHC refill. The diagnostic is to find and identify channels with very low raw noise of about 1 ADC count. To help identifying broken bonds, the calibration job was extended to include a software that looks for channels that have a significant drop in noise with respect to its neighbours within a Beetle port. All identified channels that are not already masked in the LHCBCOND are added to the calibration report for further manual check.

The IT and TT suffered from dying VCSELs from the beginning of LHCb operation at a rate of about five dead VCSELs per year for IT and TT. The diagnostic of a dead VCSEL is when error banks with the tag *TlkLinkLoss* are sent from the corresponding TELL1 for each event. Once this happens, the TELL1 link needs to be masked as soon as possible.

Few groups of channels can show increase of the noise up to the level of 90 ADC counts. This increases the number of noise clusters and those strips must be masked as soon as possible. The source of this issue is assumed to be due to a pinhole formation in the isolation between the micro-strip and the silicon bulk.

In IT, it happened at least twice that two sets of 32-channels had to be masked. This can be due to a dead Beetle port, but the more favoured explanation is that pins of the connectors at the IT detector box used to connect the long cables transmitting the data coming from the Beetle chips to the digitizer boards, are broken. These pins are fragile because they are tiny and densely packed in the IT to limit the material budget in the acceptance.

5 Pedestal calibration

The calibration of the pedestal values was previously performed using the automatic update of pedestal algorithm that can be used by the TELL1. This algorithm was found to be slow and only sensitive to the last few hundred events and not the entire dataset used by the calibration process. Therefore, a new algorithm, called STPedestalEstimator, was written that performs the iterative computation of an arithmetic mean for the event number n:

$$< \mathrm{ADC} >_n = \frac{\mathrm{ADC}_n + < \mathrm{ADC} >_{n-1} (n-1)}{n}. \quad (1)$$

STPedestalEstimator is significantly faster, has faster convergence and is sensitive to the whole calibration dataset but is not tolerant to outliers, e.g. charged particle passing through the silicon at the given place. This new algorithm is therefore not suitable for pedestal estimation during collision periods. However, regular calibration datasets are taken without any beam in the LHC, thus the STPedestalEstimator algorithm was used for 2011-2012 calibration. The estimated pedestals are stored in the COND database.

It was observed by the VELO group that the pedestals change depending on the L0 rate at which the dataset is recorded. For 2010 and 2011 data taking periods, the calibration was obtained from calibration datasets recorded at a very low L0 rate of few hundred hertz. In 2012, inter-fill calibration datasets were recorded centrally with a L0 rate larger than 900 kHz. The dependence of the pedestals on the L0 rate was re-discovered. Therefore, from early 2012 onwards, the calibration is performed based on high-L0-rate data.

6 Header correction calibration

The optimised procedure to obtain the corrections for the effect of header dependent pedestal shifts for the data coming from one analogue port of a Beetle chip are presented. Average pedestal values obtained from the previous procedure are subtracted from the ADC count for each strip. In this way, possible gain variation effects in the strip ADC counts are corrected. Two effects remain after pedestal subtractions: the header dependent pedestal shifts in the first six strips and a common mode affecting all the 32 strips. In order to isolate the header dependent shifts, the common mode must be subtracted first. This can be estimated from the 26 strips not affected by the header effect assuming that the shape is linear w.r.t. the strip numbers. This is done by calculating a mean of the ADC counts for strips-7 to 19 and 20 to 32. From the two averages, a linear extrapolation gives the common mode contributions to the first six strips which are thus subtracted from each ADC count. From those ADC counts, corrections for the header dependent pedestal shifts for the eight possible header patterns are obtained and stored in the COND database. The corrections can be as large as 7 ADC counts and therefore are coded in four bits. It must be noted that the pedestal values are appropriately shifted in order to exploit the four bit dynamic range available for the header correction process.

7 Clustering parameter calibration

As previously mentioned, the thresholds values used in the clustering are obtained from some coefficients and the estimated LCMS noise level. In the LHCBCOND database, it is not the raw and LCMS noise values that are stored but the *Common Mode* (CM) and LCMS noises, where the CM noise is given by

$$(\text{CM noise})^2 = (\text{RAW noise})^2 - (\text{LCMS noise})^2 \tag{2}$$

The common mode noise is needed for the digitisation performed in the simulation.

A new algorithm to estimate the raw and LCMS noises is written. It is significantly faster than the previous one where the pedSub ADCs and LCMS ADCs were stored in profile histograms to estimate their spread. The new algorithm uses the same philosophy as the estimate of the pedestal. The noise is the square-root of the variance of the ADC counts:

$$\text{Noise} = \sqrt{< (\text{ADC} - \overline{\text{ADC}})^2 >} = \sqrt{< \text{ADC}^2 > - \overline{\text{ADC}}^2} \tag{3}$$

The noise can be estimated by computing adaptive estimates of $< \text{ADC}^2 >$ and $\overline{\text{ADC}}^2$.

8 Monitoring

All the parameters obtained after the calibration process are constantly monitored by comparing the values in use and newly calculated ones from data collected during calibration

runs. A new program produces various monitoring plots such as the differences in pedestal, noise values and also the status of the second header bit for example.

9 Summary

Figure 7 summarises the calibration chain. A single BASH script was written to perform all these steps. In 2012, the code used by the VELO group to launch their own calibration process was modified to launch the ST calibration automatically from inter-fill calibration data. At the end of the job, all the summary plots are sent to the ST calibration expert, including the list of potentially problematic channels. These results are used to judge the quality of the current calibration and based on those results, the decision to change the calibration is made.

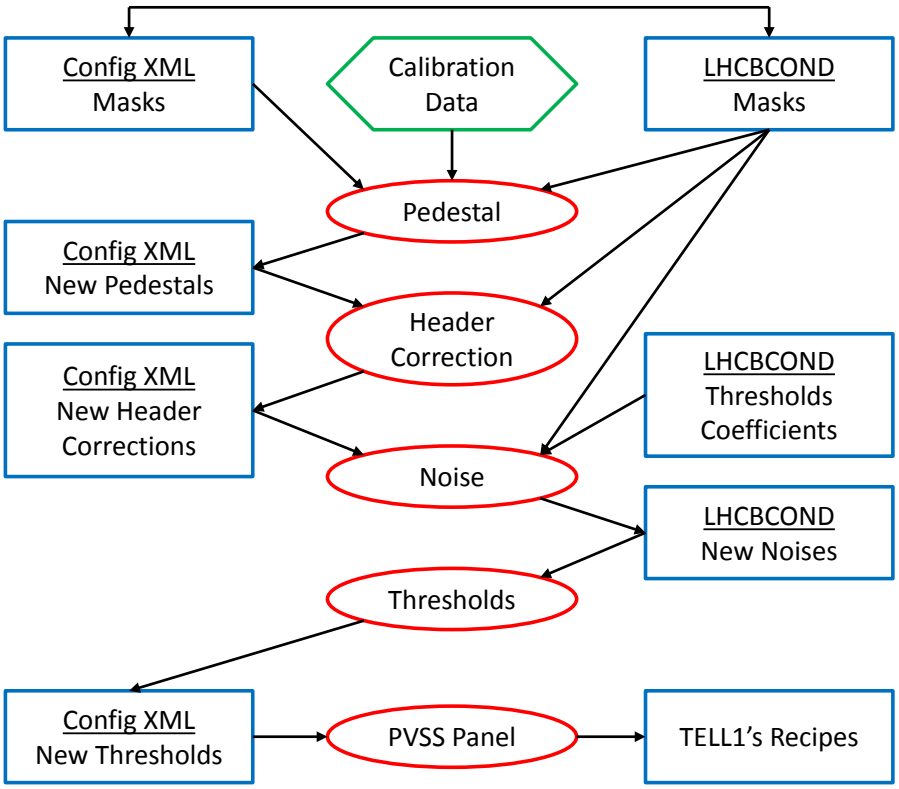

Figure 7: Schematic of the calibration chain.

References

[1] S. Lchner and M. Schmelling, *The Beetle Reference Manual - chip version 1.3, 1.4 and 1.5*, LHCb-2005-105.

[2] D. Esperante Pereira, *Design and Development of Electronics and the Control Software for the Silicon Tracker of LHCb*, PhD thesis, Santiago de Compostela University, 2010, CERN-THESIS-2010-238.

[3] A. Vollhardt, *An Optical Readout System for the LHCb Silicon Tracker*, PhD thesis, Zurich University, 2005, CERN-THESIS-2005-025.

[4] G. Haefeli *et al.*, *The LHCb DAQ interface board TELL1*, Nucl. Instrum. Meth. **A560** (2006) 494.

[5] J. Luisier, *Configuration of TELL1 boards from XML database with PVSS*, LHCb-INT-2011-017.

[6] A. Keune *et al.*, *Emulation of the ST TELL1 Algorithms*, LHCb-INT-2011-040.