

Department of Physics and Astronomy  
Experimental Particle Physics Group  
Kelvin Building, University of Glasgow,  
Glasgow, G12 8QQ, Scotland  
Telephone: +44 (0)141 330 2000 Fax: +44 (0)141 330 5881

## Application of the Pedestal Following Algorithm to the VELO Detector

Tomasz Szumlak<sup>1</sup>, Chris Parkes<sup>1</sup>

<sup>1</sup> University of Glasgow, Glasgow, G12 8QQ, Scotland

### Abstract

This note describes the performance of the algorithm used to determine and subtract the pedestal values in the TELL1. The algorithm is implemented in the TELL1 boards and emulated in the Vetra framework. This algorithm is performed as the first stage of the Non-Zero Suppressed data processing. Pedestal values determined in the Vetra emulation can be stored and later uploaded to the TELL1s. The time stability of the algorithm is assessed using data taken during commissioning. The monitoring of the pedestal values is also described.

# Contents

<b>1</b>	<b>Introduction.</b>	<b>3</b>
<b>2</b>	<b>Pedestal Following Algorithm</b>	<b>3</b>
2.1	Algorithm Description . . . . .	4
<b>3</b>	<b>Performance of the Pedestal Algorithm</b>	<b>4</b>
3.1	Pedestal Convergence . . . . .	4
3.2	Time Stability of the Pedestal Bank . . . . .	6
3.3	Performance for Time Alignment Events . . . . .	7
<b>4</b>	<b>Monitoring of the Performance of the Pedestal Following Algorithm.</b>	<b>8</b>
<b>5</b>	<b>Running with the Real Data - The Strategy.</b>	<b>10</b>
<b>6</b>	<b>Bit Perfect Emulation</b>	<b>13</b>
<b>7</b>	<b>Conclusions</b>	<b>13</b>
<b>8</b>	<b>Acknowledgments</b>	<b>14</b>

## List of Figures

1	Raw ADC values . . . . .	3
2	Pedestal values . . . . .	5
3	Pedestal subtracted ADC values and projection . . . . .	5
4	Convergence limit . . . . .	6
5	Pedestal oscillations . . . . .	6
6	Time stability of the pedestal bank . . . . .	7
7	Pedestal residuals . . . . .	8
8	Pedestal bank difference in time alignment events . . . . .	8
9	Pedestal bank difference for the central TAE sample . . . . .	9
10	Pedestal difference for TAE samples as a function of chip channel . . . . .	9
11	Pedestal subtracted data ADC values for TAE samples . . . . .	10
12	Mean value after pedestal subtraction . . . . .	11
13	ADC values after pedestal subtraction for control channels . . . . .	11
14	Mean of pedestal subtracted ADC values . . . . .	12
15	RMS of pedestal subtracted ADC values . . . . .	12
16	Mean of pedestal subtracted ADC values for each sensor and analogue link . . . . .	13
17	RMS of pedestal subtracted ADC values for each sensor and analogue link . . . . .	14

# 1 Introduction.

This note describes the use of the pedestal following and pedestal subtraction algorithms in the VELO. The algorithms are implemented in the TELL1 [1] firmware, and a bit perfect emulation implemented in the Vetra project [2]. Details of the hardware implementation of the algorithm are given in [3].

The algorithm has two components. First an algorithm is provided to follow the values of the pedestals by performing a type of running average. Second the determined values can be subtracted from the raw ADC values.

This note is structured as follows. Section 2 describes the algorithm that determines the pedestal values. Section 3 discusses the performance after pedestal subtraction. Example monitoring plots are presented and discussed in section 4. The proposed strategy for using the pedestal following algorithm and pedestal subtraction on physics data is presented in section 5. The final section, section 7, provides a summary and conclusions. Unless otherwise stated all plots and results use data taken in February 2009 during the commissioning of the VELO.

## 2 Pedestal Following Algorithm

The value of the pedestals in the VELO system are set to around 512 ADC counts<sup>1</sup>. This is half the full scale 10-bit range of the TELL1 ADC. However, there is significant variation in the pedestal values. Each Beetle chip has four outputs (analogue links) each with 32 channels, giving 128 channels per chip. Variations in pedestal from chip-to-chip, link-to-link and channel-to-channel are observed, see Figure 1. Hence, the pedestal value must be determined individually for each channel of the system. As there are 88 sensors each with 2048 channels, 180k 10-bit pedestal values are required.

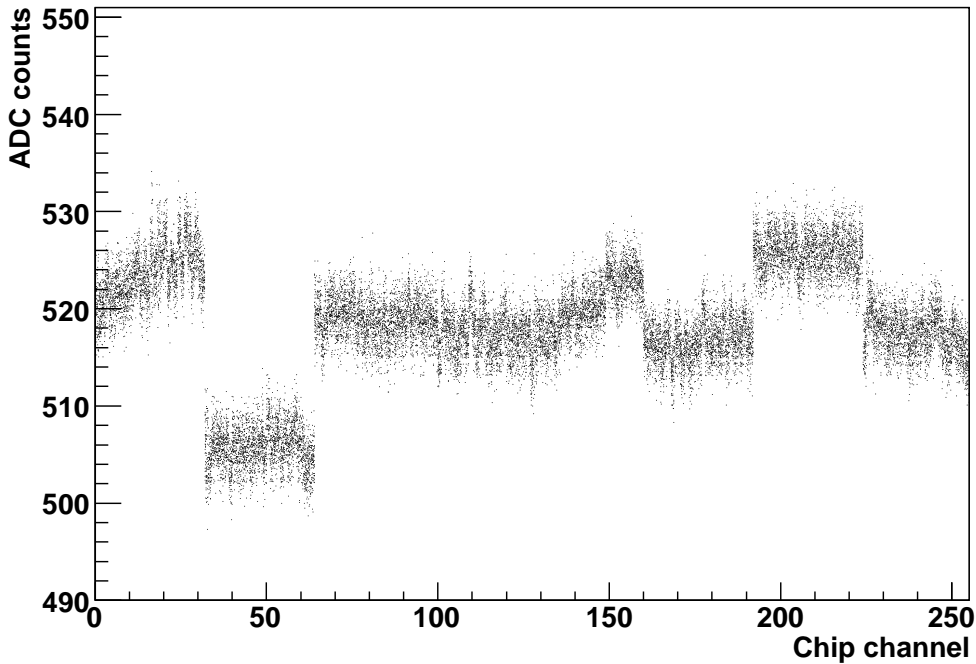


Figure 1: Typical raw (Non-Zero Suppressed) data ADC values. Data from two chips, each with four analogue links, is shown. Variations in the pedestal values at the chip, analogue link, and channel level are seen.

The following section describes the algorithm that is used to determine the pedestal values. The number of events needed to train the pedestal values is then analyzed.

<sup>1</sup>)As a rough guide one ADC count unit corresponds to a charge of approximately 450 electrons, so the signal will typically be around 50 ADC counts. The large available range allows for sizable variations due to common mode, although the dynamic range of linearity of the Beetle must be considered. The system noise is between 2 and 3 ADC counts.



## 2.1 Algorithm Description

The pedestal following algorithm used is described in [3]. The algorithm is repeated here for completeness and given in a form that emphasizes its function over its implementation.

The algorithm can be roughly described as a type of running average for the pedestal value of the channel. The pedestal value of a channel is updated using the difference between the value of the raw ADC counts in the current event and the pedestal calculated for the previous event, suppressed by a weighting factor. The pedestal value,  $p_i(n)$ , for a channel  $i$  and event  $n$  is calculated as:

$$p_i(n+1) = p_i(n) + \frac{\Delta_i(n+1)}{N}, \quad (1)$$

where  $\Delta_i(n+1)$  is the difference term between the ADC counts of the current event  $ADC_i(n)$  and the current pedestal,

$$\Delta_i(n+1) = ADC_i(n+1) - p_i(n). \quad (2)$$

Here,  $1/N$  is the weighting factor and  $N$  is set to  $1024^2$ .

In order to increase the stability of the pedestal following algorithm the allowed change of the pedestal between consecutive events is limited. This is done by limiting the maximum absolute value of the  $\Delta_i$  term:

$$|\Delta_i(n+1)| \leq 15. \quad (3)$$

If the value of  $\Delta_i(n+1)$  is greater than 15, it is set to 15. Hence if a particle passes through a channel, or a significant common mode fluctuation occurs, the pedestal in the next event is not overly distorted.

In the implementation used, the quantity stored is not the pedestal but rather the 'running sum'  $P_{Sum}^i(n)$ , so that:

$$p^i(n) = \frac{P_{Sum}^i(n)}{N}. \quad (4)$$

The initial value of the sum,  $P_{Sum}^i(0)$ , can be set via the conditions database in the emulation and uploaded via the the Experiment Control System (ECS). This value will typically be set from the value determined from a previous data taking run. If this is not available, then the default initial value of the pedestal sum is set to  $512 \cdot N$  ADC counts.

## 3 Performance of the Pedestal Algorithm

The second stage of the algorithm subtracts the pedestal values for each channel. The values to be subtracted in the TELL1 can either be: values determined immediately from the pedestal following algorithm described above; or previously determined values uploaded via the Experiment Control System (ECS).

In this section the performance of the pedestal following algorithm is assessed where the pedestal following algorithm has been first run on 4096 events (four cycles). Example Pedestal values obtained are shown in Figure 2. Typically, the pedestal following is then turned off and the subtraction is then performed and plots produced with an additional ten thousand events, taken immediately after the tuning sample.

Figure 3 shows the performance of the system after pedestal subtraction has occurred. The mean value of the pedestal subtracted data is consistent with zero. Once the pedestals have converged, no bias is observed either in the case that fixed pedestals are used or in the case that the pedestal following algorithm remains active. The negative and positive spikes observed every 32 channels in the figure are the consequence of cross-talk from the Beetle header signals, the correction for this effect will be discussed in a future note.

### 3.1 Pedestal Convergence

This section discusses the number of events needed in the pedestal following algorithm in order for the pedestals to converge to the correct value. This was studied by taking the same data sample but using different starting values for the pedestal following training. Starting values differing by up to 25 ADC counts from the final converged value were studied. This range corresponds to the maximum deviation observed in the VELO system from the average value. The results are illustrated in Fig. 4. As expected, the number of events taken to converge depends on this difference between the initial and final ADC values, and does not depend on whether the starting value is above or below the final value. Training over 4096 events provides convergence to within 1

---

<sup>2)</sup>A smaller number would allow faster training which may be useful for some purposes, and should be considered in future versions of the emulation. The division is implemented by bit-shifting and hence  $N$  is a power of two.

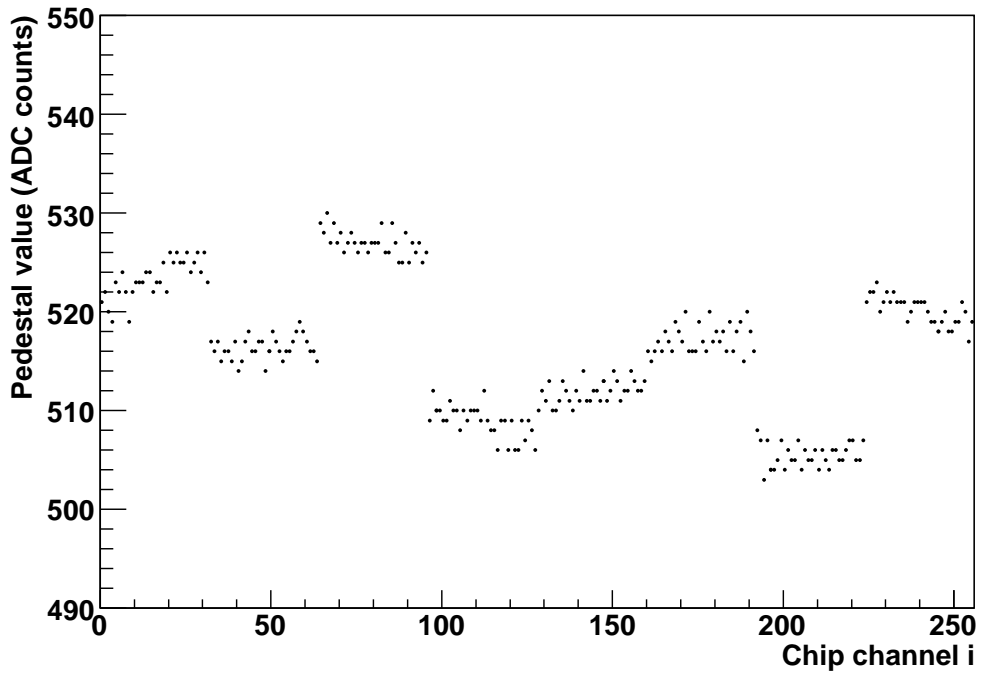


Figure 2: Example pedestal values. These values were obtained after training on a sample of 4096 events.

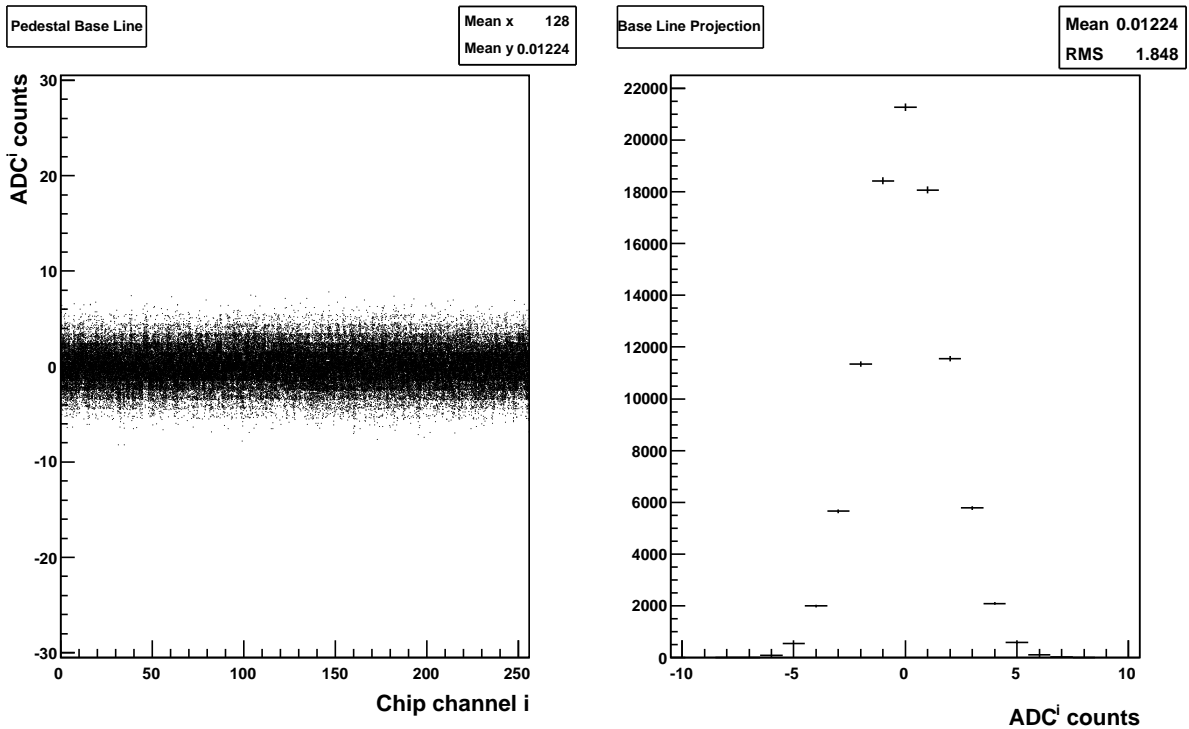


Figure 3: Pedestal subtracted ADC values (left hand plot). All values are well centred around the zero ADC count line. Projection of this plot on the vertical axis shows a typical noise Gaussian-like distribution (right hand plot).

ADC count of the final value over the full range of tested values. This is the proposed number of events to use for the pedestal tuning (see section 5).

After the pedestal values converge variations of one ADC count are still observed. This results from the pedestal following calculation being performed as integer precision operations. Further checks on the stability

have been performed using two data files (noise data only) that were taken consecutively. The pedestal values were then obtained from each sample (training over ten thousand events). If the pedestal following procedure is stable the maximum difference between the pedestal banks obtained for each data sample should be not greater than  $\pm 1$  ADC count. The results are shown in Fig. 5, and show the expected behaviour.

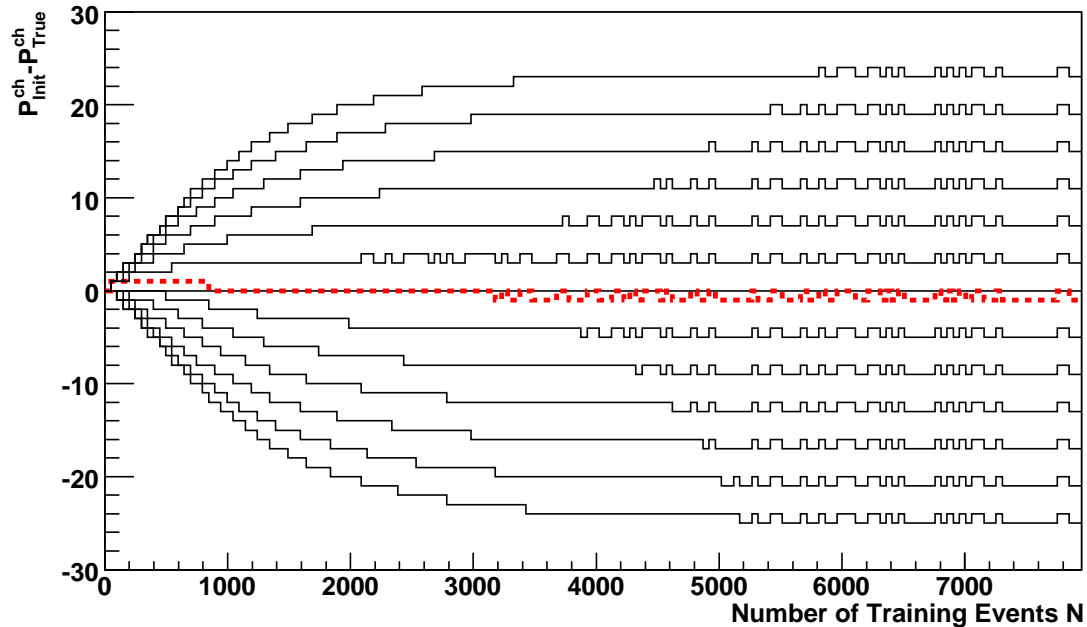


Figure 4: Evolution of the current pedestal value as a function of event number during the training cycle. The middle broken line shows the fastest convergence for the smallest value of the initial difference between the starting pedestal value and the true one. After the convergence is reached the final values vary within one ADC count.

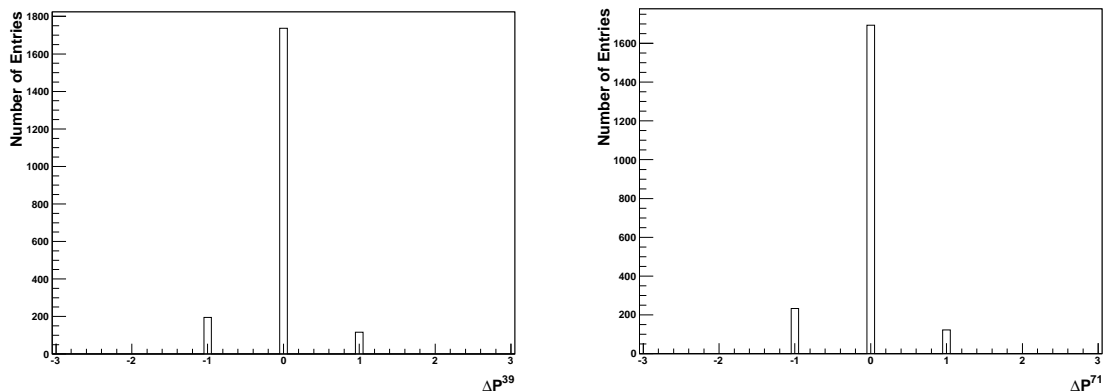


Figure 5: Difference between pedestal values calculated on two different data files. Results for two selected sensors, one R type (left) and one Phi type (right) are presented.

### 3.2 Time Stability of the Pedestal Bank

The stability of the pedestal values as a function of time has been studied. The time evolution of the pedestal bank for example channels on a given sensor are presented in Fig. 6. In Fig. 7 the difference in pedestal values for all channels on a sensor obtained from two runs are shown. The data used to obtain these plots has been taken during the ACDC3 test beam campaign that spanned over about a two week period. The observed pedestal value change for the channels over this period is no greater than 1 ADC count. Once stable conditions are

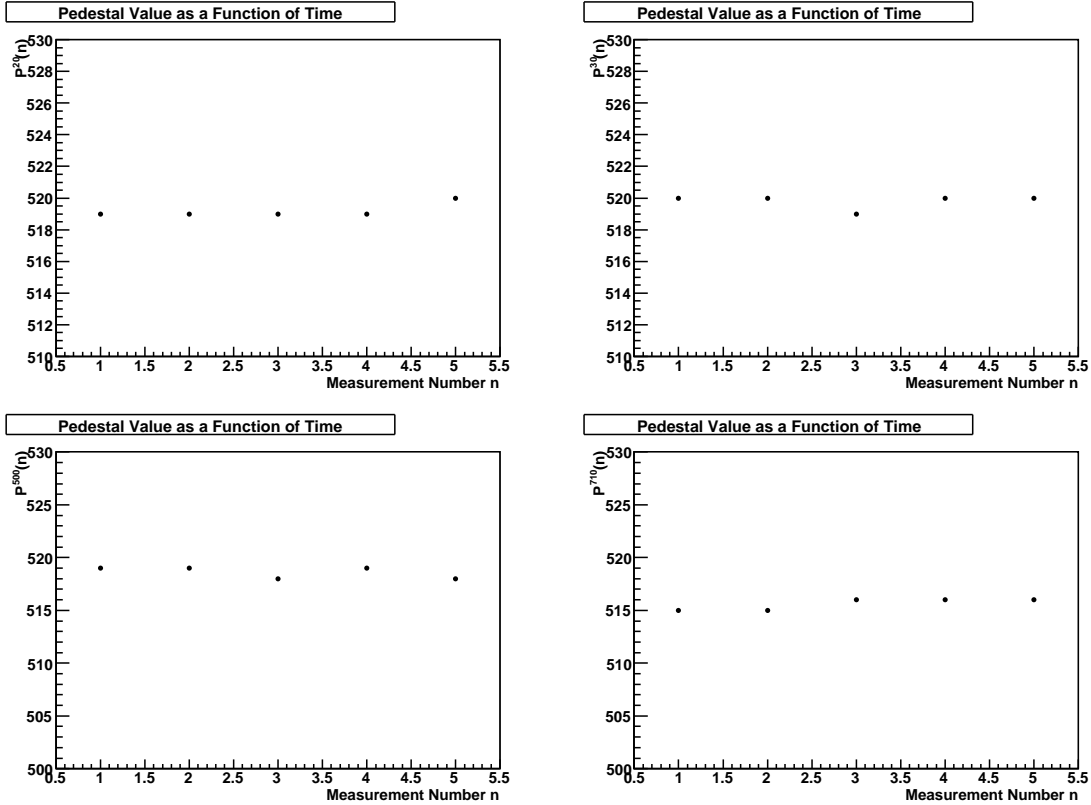


Figure 6: Evolution of pedestal values in time for four randomly chosen channels. Noise data taken during the ACDC3 test beam have been used to study the time dependency of the pedestals. The first file was taken on 08.11.2006 and the last one on 16.11.2006.

reached, similar tests will be performed for the data taken during VELO commissioning and then during the LHCb detector operation. However, it remains to be seen if the same situation will occur in the final experiment.

Our current knowledge would suggest that the pedestals do not need to be updated frequently, not more than once every few weeks. However, this needs to be studied once the full system is in operation. Any sudden changes in the pedestal values should be investigated, and hence the stability of the pedestals will be checked by the VELO shifters as part of the monitoring procedure.

### 3.3 Performance for Time Alignment Events

In standard data taking, each level 0 trigger will result in the readout of one value from each VELO channel. However, when commissioning the system and in particular when setting up the timing of the system an alternative mode can be used. In this time alignment events (TAE) mode multiple time samples (up to 15) each separated by 25 ns are read out for each channel.

A test was performed of the difference in pedestal values between the samples in time alignment events. The pedestals were tuned separately using data from each samples, and the results are shown in Fig. 8. Surprisingly differences between the pedestals in different samples of up to 4 ADC counts are seen. The pedestal values in the central time sample are in good agreement with the expected pedestals. This was checked by comparing the pedestal values in the central time alignment sample with those obtained from a run with a single sample, and is shown in Fig. 9.

Plotting the difference in the pedestal values between TAE samples as a function of channel number makes a structure over the 128-channels of the Beetle chip clearly visible for the R sensor (see Fig. 10). The pedestals in each TAE sample are constant, there is no evidence of an additional noise component in these samples. Tuning the pedestals for individual TAE samples and subtracting these values, gives the result shown in Fig. 11, where the performance is equivalent to that shown earlier in Fig. 3.

The scale of the observed deviations between TAE samples is such that it would be useful to take them into account in the TELL1 emulation of non-zero suppressed events when performing timing studies. It is neither possible nor required to take these variations into account in the TELL1 itself. Currently, Vetra is able to tune and subtract the pedestals separately for each TAE sample. However, as only one database tag can be used

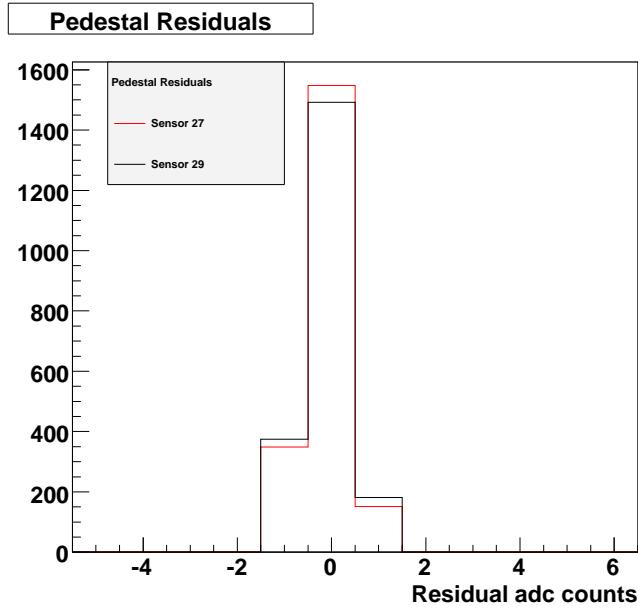


Figure 7: Difference in pedestal values for two runs taken two weeks apart, calculated for all the channels for two example VELO sensors.

in a single job, it is not possible to subtract different pedestals for different samples in a single job. Given the observed TAE behaviour this may be a useful extension, and will be considered in future versions.

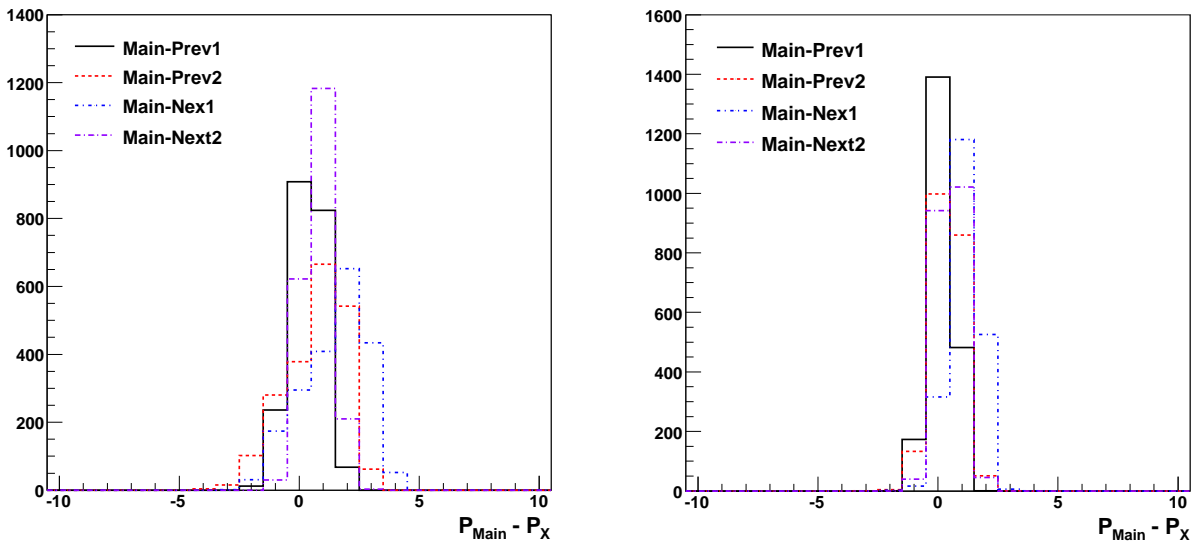


Figure 8: Difference in pedestal values for time alignment event samples. Results are shown for an example R sensor (left) and Phi sensor (right).

## 4 Monitoring of the Performance of the Pedestal Following Algorithm.

A monitoring package for the TELL1 algorithm parameters has been developed. This contains a set of plots that will be checked by the VELO shifters.

The pedestal monitoring will be performed using samples of non-zero suppressed data. These will be taken both as dedicated files and during standard physics running using a round-robin scheme. During standard physics running the main data format will be zero-suppressed (i.e. VELO clusters only). However, in addition,

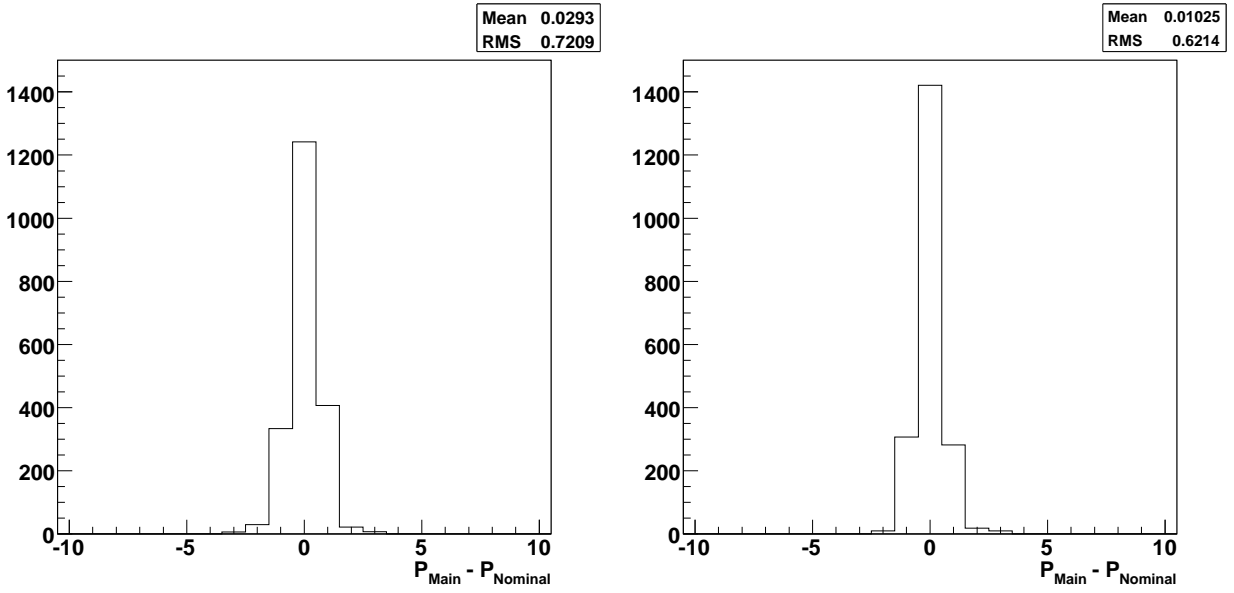


Figure 9: Difference in pedestal values for the central TAE sample ( $P_{\text{Main}}$ ) and standard single sample ( $P_{\text{Nominal}}$ ) data for R sensors (left) and Phi sensors (right).

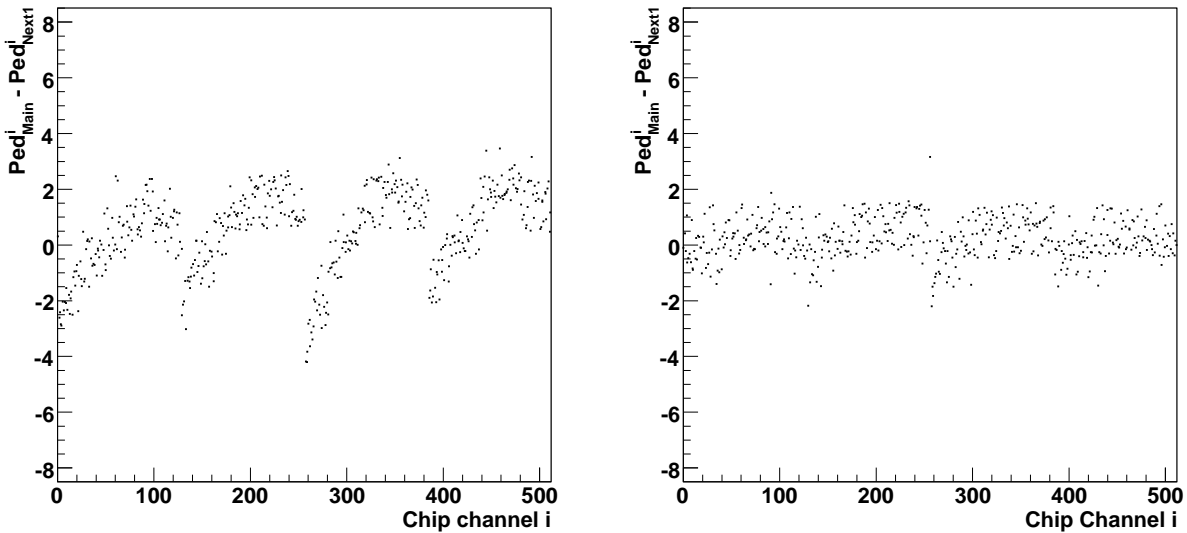


Figure 10: Difference in pedestal values between two TAE samples as a function of chip channel for R sensors (left) and Phi sensors (right).

in every few events a non-zero suppressed data bank will be sent out in addition to the zero-suppressed bank. This will be sent out at a rate (around 1Hz after the trigger for each sensor) sufficient to obtain several thousand non-zero suppressed data samples for each sensor in each fill. This data will be used for pedestal monitoring.

The detailed performance of the pedestal subtraction procedure can be checked using figures 1 and 3 showing the ADC values before and after pedestal subtraction. The pedestal values used can be read out in a pedestal bank, and were shown in Fig. 2.

More readily visible plots for monitoring the pedestals are produced by calculating the mean values for a few control channels for each analogue link. In Fig. 12 the mean values are plotted for four control channels in each analogue link of the data after the pedestal subtraction. Distortions of the mean value, due to the presence of signal, are suppressed in this plot by removing signals greater than 15 ADC counts from the average. All values are within  $\pm 1$  ADC counts as expected. An alarm message is issued by the monitoring when any of the mean values becomes comparable with the noise level or the RMS values increase or decrease. The checks performed are:

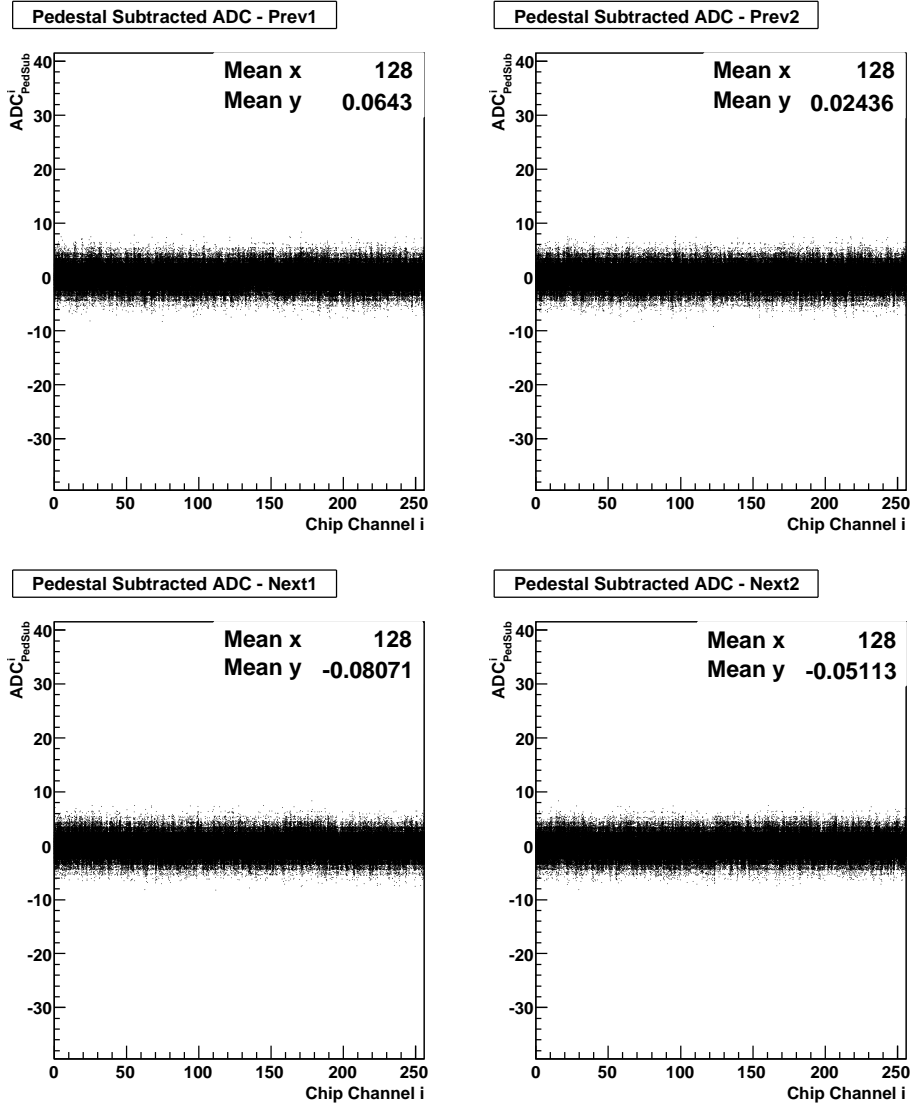


Figure 11: Pedestal subtracted ADC values for four different TAE samples as a function of channel number.

$$|\langle S_{PedSub}^{Mean} \rangle_j^i| < 0.5 \text{ ADC}, \quad (5)$$

where  $\langle S_{PedSub}^{Mean} \rangle_j^i$  is a mean value calculated for control point  $i$  within analogue analogue link  $j$ , and

$$0.1 \text{ ADC} > |\langle S_{PedSub}^{RMS} \rangle_j^i| > 1.5 \text{ ADC}, \quad (6)$$

where  $\langle S_{PedSub}^{RMS} \rangle_j^i$  is the RMS calculated for control point  $i$  within analogue analogue link  $j$ . The alarm values will be tuned with experience but starting values of 0.5 for the mean and 1.0 for the RMS are proposed. The plots presented in Fig. 12 are useful for the off-line monitoring (run by the shifters). Projections of these histograms will also be used as shown in Fig. 13. These should be centred around zero and with an RMS less than 1 ADC count. However, for the on-line monitoring a smaller number of histograms must be checked that can catch the most significant problems. The mean and RMS of the pedestal subtracted ADC counts are plotted in bins of sensor number in Fig. 14 and Fig. 15. Plotting the mean and RMS versus both the sensor and analogue link number, allows further diagnosis of problems. These plots are shown in 16 and 17.

## 5 Running with the Real Data - The Strategy.

This section describes the VELO strategy to set up the pedestal bank for data taking. The pedestal bank that will be used for the raw data suppression by the TELL1 acquisition boards will be calculated using the Vetra

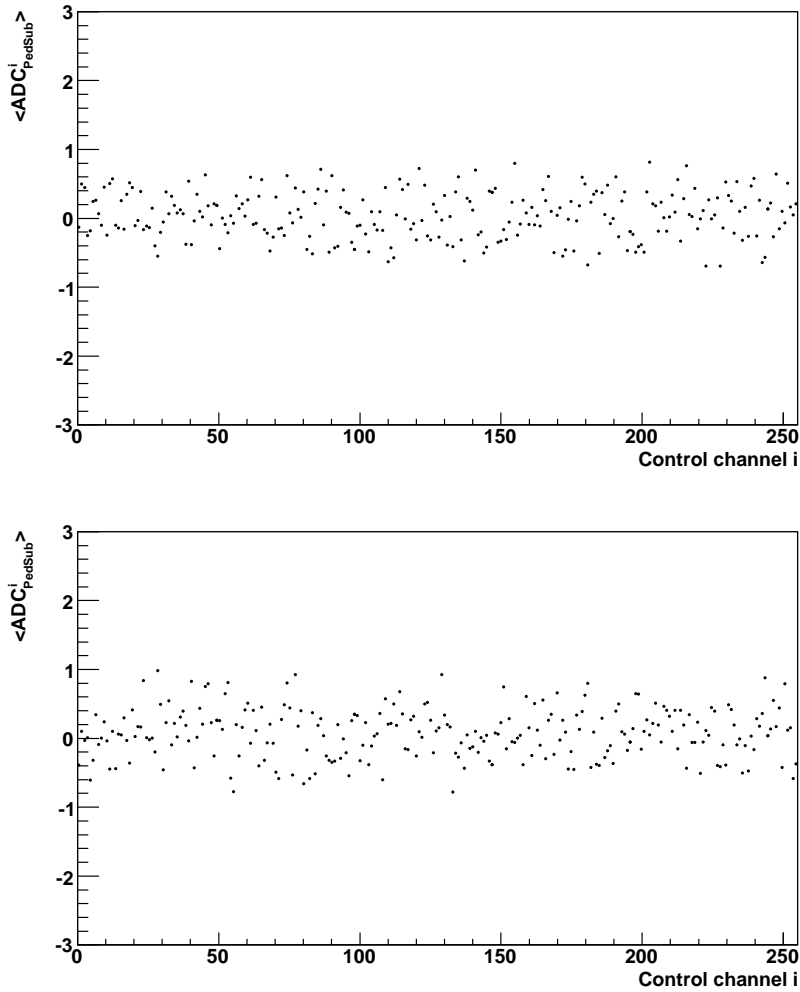


Figure 12: Mean ADC values measured in the control channels after the pedestal subtraction.

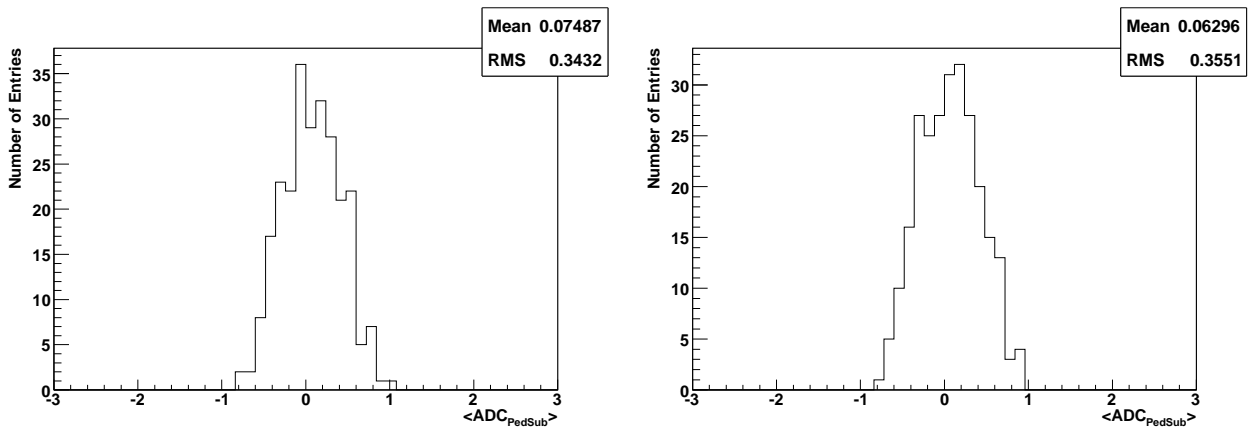


Figure 13: ADC values after pedestal subtraction for control channels from two example sensors.

TELL1 emulation. A non-zero suppressed data file of 20,000 events will be taken, without beam. 4096 events will be used to train the pedestal algorithm and 10,000 events to monitor the results. This tuning job will be run by a VELO expert. The pedestal values will be compared with the previous values. If the decision is taken to update the pedestals, then the new set of pedestals will be stored in an XML representation in the conditions database. These parameter values can then be uploaded via PVSS to the TELL1.

This baseline mode of operation assumes that the pedestals remain constant over a time period of weeks or



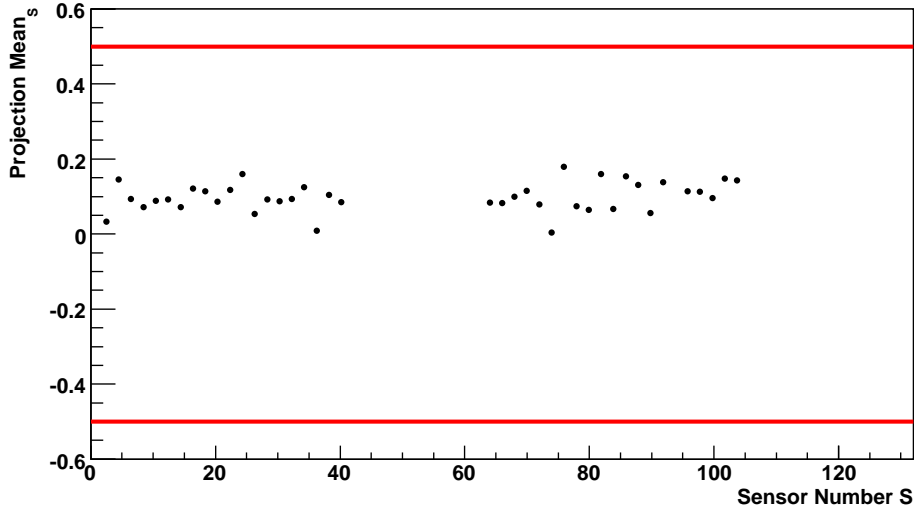


Figure 14: Mean ADC value after pedestal subtraction as a function of sensor number. Example alarm values are indicated as horizontal lines.

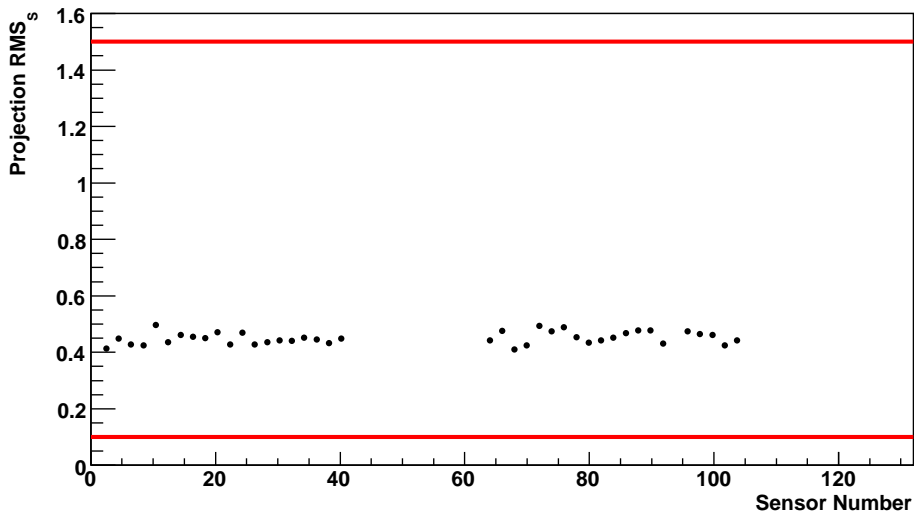


Figure 15: RMS of the pedestal subtracted ADC values as a function of sensor number. Example alarm values are indicated as horizontal lines.

longer (as has been currently observed) and do not differ between data taking with and without beam. This mode has the advantage that the exact pedestals used are known; facilitating a precise understanding of the response of the system through the comparison of non-zero suppressed and zero-suppressed data. However, if there is any significant variation in the pedestals compared with the noise of the system, an alternative mode of operation would be used. The pedestal following would be left turned on at all times in the TELL1. Although the exact pedestals used for any given event would not be known, the TELL1 emulation can still be used to monitor the algorithm performance on non-zero suppressed data files. In this case it would also be necessary to regularly read out from the TELL1 the pedestals currently being used in the pedestal bank (see [7]).

It is not expected that the pedestal training will be performed on the round-robin non-zero suppressed data samples, but rather on samples take in dedicated noise runs. However, as the pedestal training requires only a few thousand events this cross-check can also be performed.

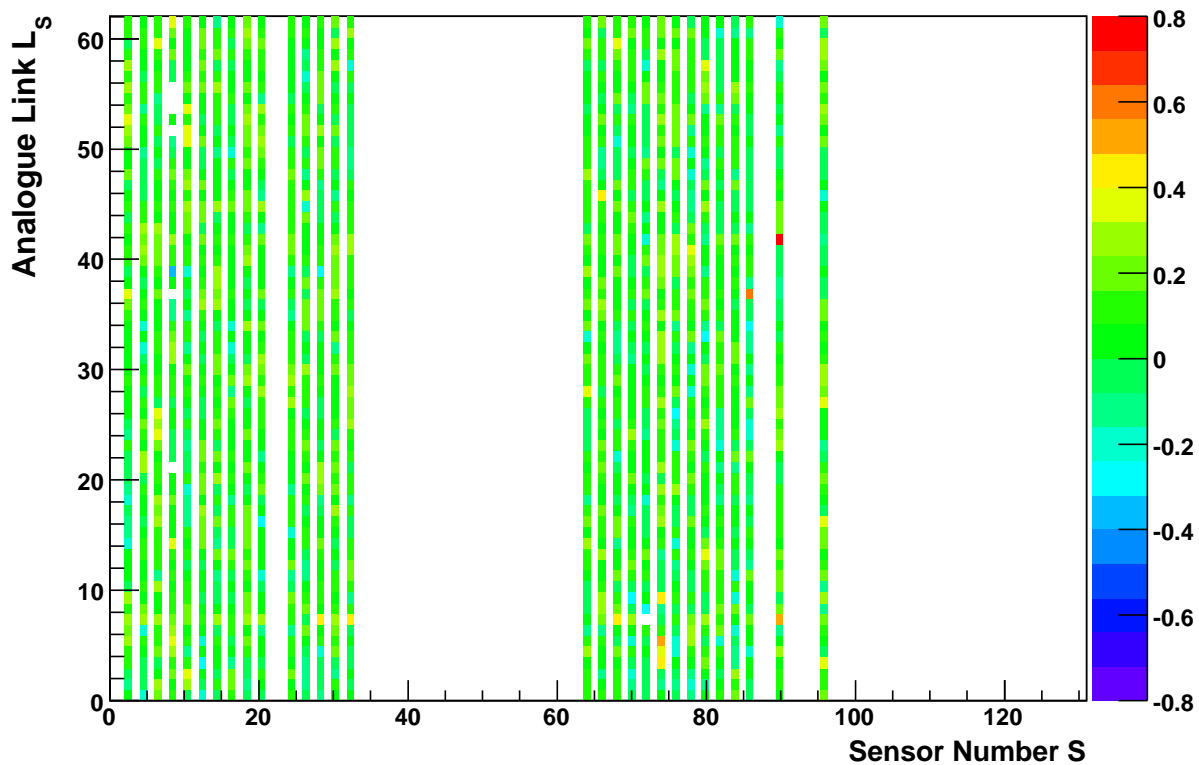


Figure 16: Mean of pedestal subtracted ADC values plotted in bins of sensor number and analogue link number.

## 6 Bit Perfect Emulation

This section describes the bit perfect comparison of the TELL1 algorithm and the emulation. VETRA v6r3 featuring the TELL1 processing engines from the tell1Lib v3.1 has been used to produce all the results presented in this note.

Bit perfect processing means that the emulated RawBank, with encoded VeloCluster objects, is identical to that produced by the TELL1. A bit perfect checking procedure is implemented via a python script that is a part of VETRA project. A number of 'observables' has been defined in order to check if both banks are identical. The emulation is regarded to be bit perfect if the following conditions are satisfied:

- the number of produced clusters is the same for the emulation and the TELL1;
- the encoded position of these clusters is identical (the centre strip number and the fractional position are compared);
- the ADC counts measured on all strips that have been used in the clusters are identical.

This procedure has been used to check that the emulation is bit perfect using ADC values that were generated in the TELL1s. The procedure has been used for the full chain of all TELL1 algorithms hence checking the pedestal algorithm. The test has been verified with both the pedestal following algorithm turned on, and with fixed pedestal values subtracted. Applying this procedure with generated data does not fully check the pedestal following procedure is bit perfect, and further tests will be performed.

## 7 Conclusions

In this note the pedestal correction procedure has been presented and its performance assessed. The pedestal following algorithm is found to measure the pedestals with an accuracy of 1 ADC count, and requires tuning on a sample of non-zero suppressed events of not more than four thousand events. The pedestals will be obtained using the TELL1 emulation in Vetra using a non-zero suppressed data file. These pedestals will then be uploaded

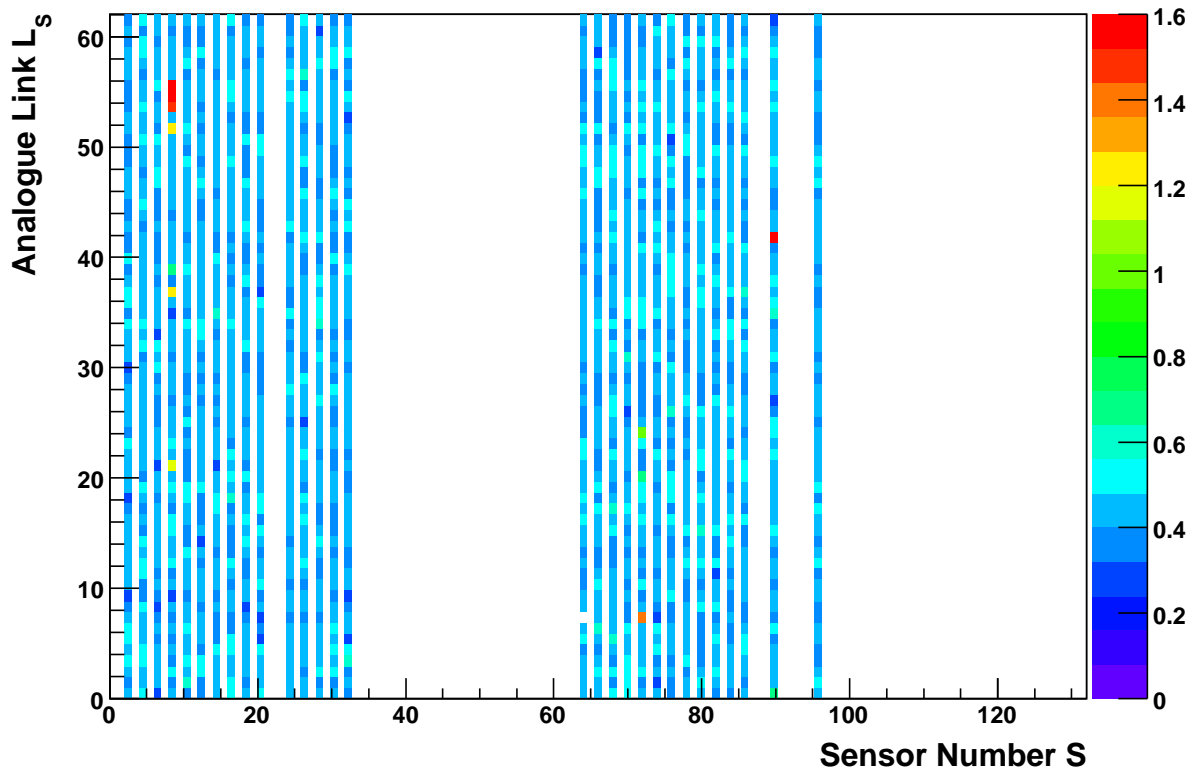


Figure 17: RMS of pedestal subtracted ADC values plotted in bins of sensor number and analogue link number.

to the TELL1. The values are currently expected to be constant for periods in excess of a week. The pedestals will be checked by the VELO experts using a standard set of monitoring plots.

## 8 Acknowledgments

We wish to thank Guido Haefeli and Alex Gong who have been responsible for developing and implementing the VHDL coding of the FPGA algorithms and the C-code emulating their performance. Thanks also to Kurt Rinnert for his useful advice and for developing the procedure in PVSS for parsing and uploading the parameters to the TELL1.

## References

- [1] G. Haefeli et al, 'TELL1 - specification for a common read out board for LHCb', LHCb-2003-007
- [2] T. Szumlak, C. Parkes, 'Description of the Vetra project and its application for the VELO detector', LHCb-2008-022
- [3] G. Haefeli, Contribution to the development of the acquisition electronics for the LHCb experiment, CERN-Thesis-2004-036 (PhD thesis).
- [4] G. Haefeli, A. Gong, 'VELO and ST non-zero suppressed bank data format', EDMS note 692431 v.2
- [5] G. Haefeli, A. Gong, 'VELO and ST error bank data format', EDMS note 694818 v.1  
G. Haefeli, A. Gong, 'VELO and ST pedestal bank data format', EDMS note 695007 v.1
- [6] S. Lochner, M. Schmelling, 'The Beetle Manual', LHCb-2005-105
- [7] T. Szumlak, C. Parkes, 'VELO event Model', LHCb 2006-054