2010

Threshold Tuning of the ATLAS Pixel Detector

P. Behara

G. Gaycken

C. Horn

A. Khanov

D. Lopez Mateos

ATLAS NOTE

ATL-COM-INDET-2010-044

July 28, 2010

Threshold Tuning of the ATLAS Pixel Detector

Prafulla Behara[1)], Goetz Gaycken[2)], Claus Horn[3)], Alexandre Khanov[4)], David Lopez Mateos[5)6)], Lucia Masetti[2)], Klemens Müller[2)], Kerstin Perez[5)6)]

Abstract

The threshold of each pixel in the Pixel Detector, measured in electrons (e), must be properly set in order to optimize position resolution and particle detection efficiency. This note details the threshold tuning performed for the full pixel detector over a 4-month period in the fall of 2008. At the end of this period, 94% of the detector was tuned to a measured mean threshold value of 3939 e, which is offset from the true threshold value by ∼60 e, with an RMS of 37 e. The remaining 6% of the detector was not measurable due to software or hardware problems.

[1)] Iowa State University, Iowa City, Iowa 52242, USA
[2)] Universitat Bonn, 53113 Bonn, Germany
[3)] SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA
[4)] Oklahoma State University, Stillwater, OK 74078, USA
[5)] California Institute of Technology, Pasadena, CA 91125, USA
[6)] Columbia University, New York, NY 10027, USA

1 Introduction

The Pixel Detector is the innermost tracking sub-detector of the ATLAS detector. It is composed of over 80 million silicon pixels, distributed over 1744 modules. These modules are arranged in 3 barrel layers, denoted Layer-0, Layer-1, and Layer-2, and two endcaps, each of which has 3 disks. Together, the barrel and endcaps provide coverage up to $|\eta| < 2.5$.

The goal of the Pixel Detector is to provide primary vertex detection and precision tracking of charged particles. To accomplish this, it is designed to have a spatial resolution in r-ϕ of less than 15 μm and a hit detection efficiency greater than 97% [1]. It must also be able to operate within the high-radiation environment of the LHC.

In the binary pixel readout system, each pixel has an adjustable threshold, measured in electrons (e). Proper setting of this threshold for each pixel is necessary to attain both high efficiency and good position resolution. This threshold should be high enough to avoid registering fake hits caused by noise in the readout electronics. If there are enough of these fake hits they can even overwhelm the readout buffers on the module, causing all information for all pixels on this module to be lost. This occurs if an End-Of-Column buffer, which holds the readout information for 320 pixels organized in a pair of columns, registers more than 64 pixels with hits, or if the Module Controller Chip (MCC) buffer, which holds readout information for the 16 front-end chips on a module, registers more than 128 pixels per front-end chip with hits [2]. On the other hand, the threshold should be low enough to register the charge from a minimum ionizing particle (mip) with high efficiency. A mip typically deposits approximately 20 ke while perpendicularly traversing the silicon sensor, but this charge is usually distributed among several neighboring pixels, producing a cluster of hits. If the threshold is too high, some of these hits will be lost and the clusters can be split or mis-measured, degrading the position resolution [3].

To balance these effects, in autumn 2008 the threshold was tuned to a target value of 4000 e. This setting was verified during test beam and production to have low noise while maintaining a high efficiency in charge collection. With this threshold, the Pixel Detector has been measured to have a detection inefficiency less than 0.1% and a noise occupancy of $\sim 10^{-10}$ per pixel per event per bunch crossing [4]. As the detector becomes irradiated due to LHC operation, the signal observed due to a mip traversing the silicon will decrease, and the threshold may need to be lowered [2].

The threshold for each pixel is controlled by two discriminator stages located in each pixel cell of the front-end chips [2], as shown in Figure 1. The first discriminator controls the threshold for an individual pixel, and the value of this threshold is set by a 7-bit trim digital-to-analog converter (TDAC). The second discriminator imposes a threshold that is common for all pixels read out by the same front-end chip. This value is set by a 5-bit global digital-to-analog converter (GDAC). Both the per-pixel TDAC and the per-front-end chip GDAC can be adjusted in the range of approximately 0-1 fC, giving a total threshold range of approximately 0-9000 e. There are uncertainties in the exact value of the injected calibration charge, and hence the value of the measured and calibrated threshold, though these uncertainties have been shown to be negligible [5] [6].

Section 2 of this note describes the principles of threshold measurement and tuning. Section 3 discusses the tuning procedure that was used during the calibration period in autumn 2008 and the results of this tuning. A summary of improvements to the tuning procedure, as well as the detector hardware and software, for 2009 is presented in Section 4.

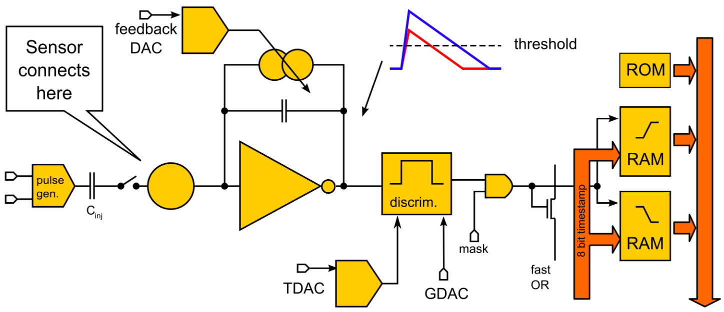


Figure 1: Schematic of the front-end chip readout for a single pixel cell.

2 Threshold Measurement and Tuning Algorithms

2.1 Threshold Measurement

The threshold value of each pixel is measured using an algorithm that is known, in the software, as a THRESHOLD_SCAN. Henceforth, the word "scan" will be used to refer to the process of executing an algorithm over the detector in order to measure or tune the threshold.

In a THRESHOLD_SCAN, a fixed charge is injected into a preamplifier just upstream of the first pixel discriminator on the front-end chip. This injection is repeated multiple times, and the percentage of injections that result in a hit being readout is recorded by the low-level Digital Signal Processor (DSP) code [7] that runs on the Read-Out Drivers (RODs). The value of the injected charge is then increased by a discrete voltage step, and the process is repeated until a specified charge range has been covered. The number of injections at each charge value is known as the number of events per scan point, and the voltage steps are known as VCAL steps. Typical parameters for a THRESHOLD_SCAN are listed in Table 1.

Ideally, this process would produce a step function, with zero injections resulting in a hit for any charge below threshold and all injections resulting in hits for any charge above threshold. However, due to the electronic noise of each front-end channel, this step function is smeared into an S-curve shape, as shown in Figure 2. The S-curve is fit with a Gaussian error function by the DSP code. The mean value of this fit is recorded as the threshold for each pixel. The σ of the fit is defined to be the noise of each pixel. The mean, σ, and χ^2 of the fit are stored in histograms titled SCURVE_MEAN, SCURVE_SIGMA, and SCURVE_CHI2, respectively.

An important gauge of the accuracy of the threshold measurement is the variation in the measured threshold and noise between two consecutive scans. Figure 3 shows the difference between the measured threshold and noise values obtained from two scans, listed in the Appendix. The mean threshold and noise difference are both 0 e, with an average variation of 18 e in threshold and 17 e in noise.

2.2 TDAC Tuning

The TDAC is a 7-bit digital-to-analog converter that determines the fine-adjustment of threshold values for each pixel. The variation of threshold value with TDAC setting is approximately linear in the middle of the TDAC range, from 40 to 100, as shown in Figure 4(a) for 8640 pixels on one module. In this range, one step in TDAC is approximately 75 e. A TDAC setting greater than 120 effectively disables the readout of the pixel by setting the threshold higher than the expected charges.

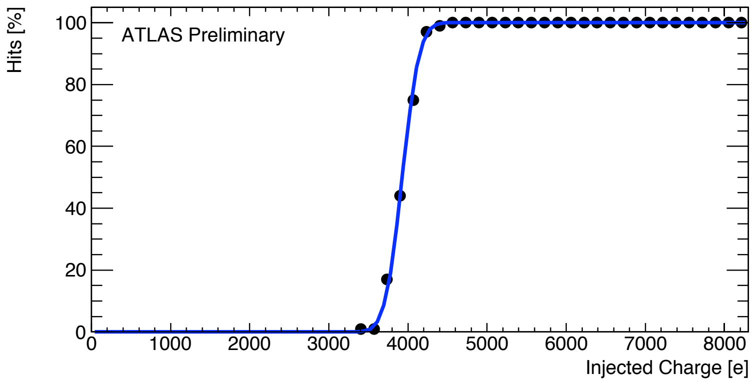


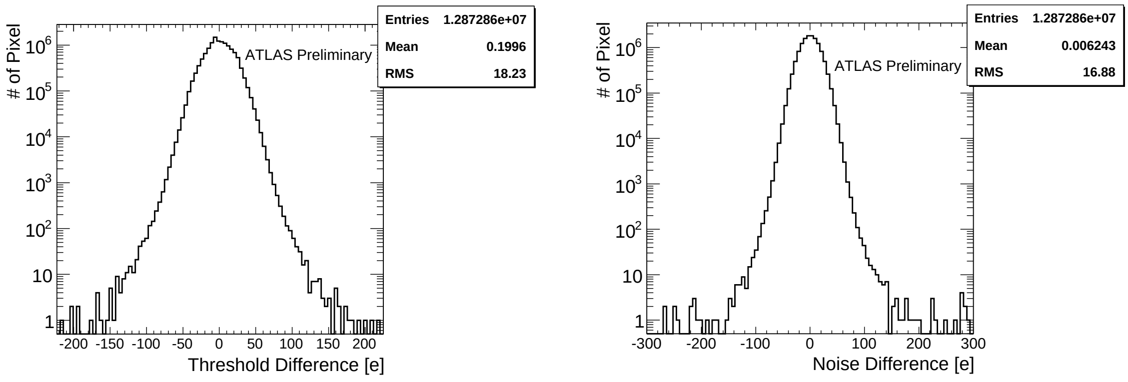
Figure 2: Illustration of a typical S-curve that results from a threshold scan of one pixel. This S-curve is fit by a Gaussian error function. The mean value of this fit is regarded as the threshold for this pixel. The width of the fit, defined as the difference between the charge values that correspond to 16.5% and 83.5% hit efficiency, is regarded as the noise of the pixel.

Table 1: Threshold scans and their default parameters.

Scan Type	THRESHOLD_SCAN	TDAC_FAST_TUNE (full scan)	TDAC_FAST_TUNE (fine scan)	GDAC_TUNE
Events/scan point	50	50	50	25
Step parameter	VCAL	TDAC	TDAC	GDAC
Initial Value	0	64	current value	10
Step range	0-200	11-117	current value ± 7	10-30
# of steps	101	8	3	3
Step pattern	0, 2, 4, ..., 198, 200	0, ± 16, ± 12, ± 10, ± 8, ± 4, ± 2, ± 1	± 4, ± 2, ± 1	10, 20, 30

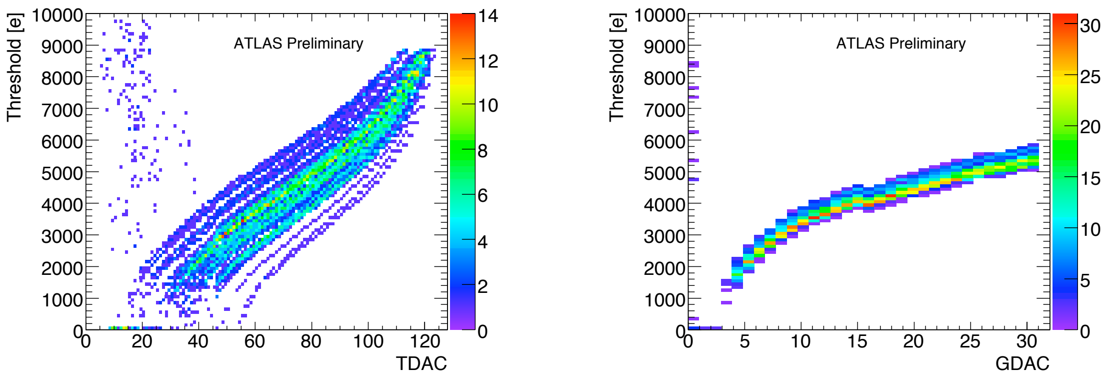
To tune the TDAC of each pixel to a desired threshold value, the TDAC_FAST_TUNE algorithm is used. First, a charge equal to the desired threshold value is injected into the preamplifier on the front-end chip. This injection is repeated multiple times, known as the number of events per scan point. If the percentage of injections that result in a hit being read out is less than 50%, then the TDAC setting is decreased by some value, known as the step size; if this percentage is more than 50%, then the TDAC setting is increased. This process is performed a fixed number of times, known as steps, with the step size typically decreasing at each step. The final TDAC value achieved at the end of this iterations is chosen as the tuned setting.

The step sizes must be chosen to avoid that a large number of pixels simultaneously have thresholds below the typical noise value during the tuning process. This can occur, for example, if the algorithm determines that the TDAC setting must be decreased, but the step size down is so large that the TDAC is then set to a very low threshold. When this happens, noise hits fill the End-Of-Column and MCC buffers on the module, thus blocking the readout of any hits. This can be seen in Figure 4(a), where for many pixels the s-curve fit returns a false high threshold value for low TDAC settings. This starts to occur for TDAC values < 40, corresponding to an actual threshold < 2500 e. Since no hits are registered, the tuning algorithm then lowers the threshold further. This process repeats at each step in the tuning, until the lowest TDAC value attainable with the chosen step sizes is reached for many pixels in the same



(a) (b)

Figure 3: Difference between the measured value of threshold per pixel (a) and noise per pixel (b) between two consecutive threshold scans over 280 modules.



(a) (b)

Figure 4: Dependence of threshold on TDAC (a) and GDAC (b) setting for 8640 pixels on one module.

module.

In the threshold tuning procedure two different sets of step sizes and range are used, both chosen to avoid having noise hits overwhelm the readout. One set is used to cover a maximal range of TDAC settings, and is used for an initial TDAC tuning. For this, the initial TDAC value is set to the middle of the range, i.e. 64. The TDAC is then adjusted in 8 steps, including the initial setting, with the decreasing step sizes of 16, 12, 10, 8, 4, 2, and 1. Pixels that still end the tuning with the lowest reachable TDAC setting, determined by the initial value and the step sizes, are then reset to a TDAC value of 113, effectively disabling these pixels. Another set of step sizes covers a smaller range around the current TDAC setting, and is used to refine a previous TDAC tuning. For this, the initial TDAC value is kept at the current value. The TDAC is then adjusted in just 4 steps, with the decreasing step sizes of 4, 2, and 1. These two sets of parameters are detailed in Table 1.

2.3 GDAC Tuning

The GDAC is the 5-bit digital-to-analog converter that controls the threshold for an entire front-end chip. The full range of GDAC values is 0-31, covering a range in threshold of approximately 0-5000 e. A typical GDAC value for a target threshold of 4000 e is between 10 and 30. In this range, one step in GDAC is equal to approximately 100 e. The dependence of threshold on GDAC setting for all pixels on one module is shown in Figure 4(b).

Tuning of the GDAC is only necessary if the range of available thresholds during the TDAC tuning is not sufficient to reach the target threshold on a particular front-end chip, as indicated by a large number of TDACs being set to very high or very low values. Shifting the global threshold value for a front-end chip correspondingly shifts the range of individual pixel thresholds available for a TDAC tuning. To ensure that after the GDAC tuning the target threshold will be attainable in the range of TDAC settings, the GDAC tuning is performed with all TDACs set to the middle of their range, i.e. 64.

The GDAC is tuned to a target threshold using the GDAC_TUNE algorithm. Typical parameters of this algorithm are listed in Table 1. The GDAC_TUNE algorithm first performs a THRESHOLD_SCAN for a set number of GDAC settings and records the average threshold over the front-end chip at each step. This process is very time consuming, as it requires a full THRESHOLD_SCAN to be performed at each step (the typical time for one THRESHOLD_SCAN is 41 minutes, as discussed in Section 3.1). Because of this, usually only very few GDAC settings, typically 3, are used, and only a fraction of the pixels on every front-end chip, typically $\sim$ 10%, are scanned. The two GDAC settings that yield an average threshold closest to the desired threshold are then selected. These two GDAC settings and their corresponding average threshold results are then used to create a linear function that maps GDAC settings to threshold values. This linear function is used to select the GDAC value that corresponds most closely to the desired threshold.

3 Threshold Tuning During the 2008 Calibration Period

3.1 Threshold Tuning Procedure

The calibration period in the fall of 2008 was the first time the threshold tuning was performed on the full Pixel Detector. At the beginning of this period, all TDACs and GDACs were set to the values that had been determined during module production [8], referred to from now on as the production tuning. During LHC operation, radiation damage will increase the dispersion of threshold values, and periodic retuning will be necessary.

The following summarizes the threshold tuning procedure in 2008. All scans were performed with the settings listed in Table 1.

1. THRESHOLD_SCAN : to establish the initial threshold settings of each pixel.
2. TDAC_FAST_TUNE : to tune the TDAC values of each pixel to a threshold of 4000 e.
3. THRESHOLD_SCAN : to verify the tuned threshold value of each pixel.
4. GDAC_TUNE : to tune the GDAC values of front-end chips that failed the initial TDAC tuning.
5. TDAC_FAST_TUNE : on only those front-end chips that failed initial TDAC tuning.
6. THRESHOLD_SCAN : to verify the tuned threshold value of each pixel.
7. Time-over-threshold (ToT) tuning procedure [9].

8. TDAC_FAST_TUNE : using a restricted number of steps to refine the threshold tuning after the ToT tuning.

9. THRESHOLD_SCAN : to verify the tuned threshold value of each pixel.

A module was determined to have failed the initial TDAC tuning if its average TDAC was > 74 or < 55, as determined by a software tool that directly inspects the module configuration stored in the database. This indicated that the GDAC values for its front-end chips are too low or too high. In this case, a GDAC tuning was performed in order to allow the TDACs to be tuned closer to the middle of their range.

Due to interference between the threshold tuning and the ToT tuning for each pixel, TDAC tuning was repeated after the ToT tuning procedure. The ToT tuning changes the value of the feedback current across the preamplifier. This changes the maximum pulse height, and thus affects the threshold setting that corresponds to a particular charge. It was observed during the 2008 calibration period that the ToT tuning increased the average threshold by $\sim 0.04\%$ and the RMS of the threshold distribution by $\sim 40\%$. The threshold tuning, on the other hand, influenced the ToT tuning much less. The threshold tuning increased the average ToT by $\sim 0.01\%$ and decreased the RMS of the ToT distribution by $\sim 3\%$ [9]. This may be due to the fact that the ToT only changed slightly during this tuning, by $\sim 2\%$, while the threshold changed more significantly, as discussed in Section 3.2.

Since the threshold is only slightly affected by the ToT tuning, the TDAC_FAST_TUNE that is run after the ToT tuning uses the 4-2-1 step pattern, as described in Section 2. This setting is used whenever a small refinement to a previous tuning is necessary, for example at new temperature settings, after radiation damage, or to reduce the RMS of the threshold settings across a module.

In order to quickly and efficiently analyze the data obtained from the threshold scans, an online analysis, called "THRESHOLDanalysis", was implemented. This analysis allows the user to define limits on certain parameters in order to assess the tuning of a module or front-end chip. The user can then combine the results of this analysis with data from the histograms produced by a THRESHOLD_SCAN and knowledge of the history of each module to determine whether or not the tuning procedure should be repeated for that module. The parameters, and typical values of these parameters for a target threshold of 4000 e, that are used to assess the quality of the tuning are:

- Percentage of pixels on a front-end that are within a minimum and maximum threshold value. Typically 95% of pixels are required to be above 3000 e and below 5500 e for a front-end chip to be classified as passing the analysis.

- Minimum and maximum value of average threshold over a front-end. Typically the average threshold is required to be above 3200 e and below 5300 e for a front-end chip to be classified as passing the analysis.

- Percent of pixels on a front-end, separated by pixel type, that fail a maximum noise cut. Typically 90% of pixels are required to have a noise value below 400-450 e for a front-end chip to be classified as passing the analysis. The exact cut value varies between types of pixels because of their varying typical noise values, as discussed in Section 3.2.

- Maximum RMS of threshold for pixels on a module or front-end. Typically the RMS must be less than 400 e for a front-end chip to be classified as passing the analysis.

- Number of "bad" pixels on a front-end chip, where a pixel is designated as "bad" if it does not return a threshold and noise measurement. This failure could occur if a pixel is legitimately dead or damaged, or if the S-curve fitting procedure during a threshold measurement fails, as discussed in Section 3.2. Typically a front-end is required to have less than 200 bad pixels to be classified as

Table 2: Execution time for threshold scans on Layer 0, Layer 1, Disks, and half of Layer 2.

Scan	Time (L0, L1, Disks, half of L2)
THRESHOLD_SCAN	41 m
TDAC_FAST_TUNE	1 h 20 m
GDAC_TUNE	1 h 10 m

passing the analysis. However, unlike the other analysis parameters, a failure on this cut typically does not indicate a bad threshold tuning, but instead a poor threshold measurement.

During 2008, the version of the DSP code being used had memory constraints that limited the number of modules connected to a ROD which could be included in a single scan. This introduced complications for RODs connected to modules on Layer-2 of the pixel barrel, which have 26 modules per ROD, as opposed to RODs connected to other segments of the detector, which have at most 13 modules per ROD. Thus in order to cover the full detector, each scan was performed once for part of the detector containing half of the Layer-2 modules, then again for a portion containing the other half.

The approximate times needed to perform these scans are listed in Table 2. Note that because only half of all Layer-2 modules could be included in the same scan, the total time needed to scan the full detector was twice the time listed.

3.2 Threshold Tuning Performance

Figure 5 shows the module average threshold value for 1642 modules for which reliable threshold measurement data exists. The scans used to obtain this data are listed in Table 3 of the Appendix. The mean module measured threshold is 3939 e, with an RMS of 5 e. These measured threshold values are offset from their actual threshold values by a known bug of the 2008 DSP code, which incorrectly mapped the VCAL setting to the measured charge value. It should be emphasized that this offset is a fault in the measurement of threshold values, not in the tuning of the threshold.

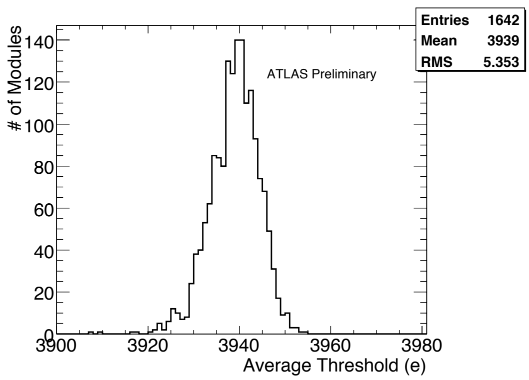


Figure 5: Average module threshold for 1642 modules with the 2008 tuning.

1642 modules corresponds to $\sim$ 94% of the Pixel Detector. The remaining modules were either not tuned or not measurable due to hardware or software faults. Specifically,

- 51 modules were inoperable due to hardware failures. Of these, 36 modules were located on cooling circuits that were not operated in 2008 due to leaks, and 15 modules had hardware failures on the module or its optical services, such as open low voltage lines or broken optical connections. Another 6 modules were operable, but failed during optical communication or digital injection tests. 3 modules for unknown reasons consistently failed to return results during a threshold scan.

- 20 modules had known or suspected broken high-voltage supply lines. The high voltage on a module is used to provide the bias voltage across the silicon sensor. If no high voltage is provided, then the sensor is not depleted, resulting in an increased capacitance across the sensor. This increased capacitance causes additional noise across the readout electronics that interferes with both threshold calibration and measurement. Since measurements for these modules are inherently inaccurate, they are also excluded from the results discussed here.

- The remaining 22 modules failed to be measured due to an error in the S-curve fitting procedure that is performed by the DSP code during a threshold measurement. Due to a bug in the 2008 DSP code, the S-curve fitting procedure sporadically fails for many pixels on some modules during a THRESHOLD_SCAN, returning a value of zero for the threshold and noise of all except approximately 1000 pixels. In a subsequent scan, however, the module may return normal results. In order to eliminate these failures, modules on which less than 2000 pixels return non-zero values are excluded from all results shown here.

Figure 6 shows the single-pixel threshold and noise distributions for pixels on 1527 modules with production tuning and 1642 modules with 2008 tuning. The production tuning was performed at three different sites, before the modules were initially loaded onto support structures. This tuning used variations of the TDAC and GDAC tuning algorithms described in Section 2 to set a target threshold of 4000 e. The two peaks in threshold value seen in the production tuning are possibly due to differences between the testing infrastructures at different production sites. The 2008 tuning merges these threshold settings to a measured central value of 3939 e.

Threshold tuning should not affect the electronic noise of the pixel channels. The increase in the number of pixels with noise above 500 e in the 2008 tuning with respect to the production tuning is the result of setting lower TDAC values on pixels that had TDAC settings of 127, and hence were effectively disabled, in the production tuning. This corresponds to a 0.01% ($\sim$ 8000 pixels) increase in the number of noisy pixels observed during data taking with a random trigger [4].

Figure 7 shows the distribution of all TDAC and GDAC values over the full detector in the 2008 tuning module configuration. The mean TDAC value is 67, and the mean GDAC value is 18. A spike can be seen at a TDAC value of 106. This is due to noisy pixels that had their TDAC value set to 113 during the initial TDAC tuning (the maximum TDAC reachable), as described in Section 2.2, but then had their TDACs lowered by 7 steps (the maximum decrease) during the TDAC tuning that was performed after the ToT tuning. The small number of pixels with a TDAC value of 0 are pixels that were never tuned during 2008. The dip in the number of pixels with TDAC of 64-65 is due to the tuning algorithm initially setting the TDAC to 64, and then iterating away from this value.

The threshold and noise behavior of individual pixels varies according to the size of the sensor cell and the manner in which the pixels are connected to the front-end chips [2]. Figure 8 shows a map over the module of the average noise value for all pixels in the 2008 tuning, and Figure 9 shows the threshold and noise distributions of these pixels, separated according to their connection type. Most pixels are normal pixels, which are connected directly to the front-end chip for readout, have a size of $50 \times 400 \mu\text{m}^2$, and have an average noise of 160 e. Long pixels, $50 \times 600 \mu\text{m}^2$ in size, are also bonded directly onto a front-end chip, but are longer in order to cover the approximately 400 μm side gap between front-end chips on a module. These can be seen in Figure 8 as the columns of pixels in between the front-end chips. Their larger size gives a larger capacitance, and hence a slightly higher noise value than normal pixels. On average, long pixels have a noise of 185 e. Other $50 \times 400 \mu\text{m}^2$ pixels are also located between facing front-end chips. As these cannot be connected directly, they are instead ganged by a metal strip on the sensor to pixels which are connected directly to a front-end chip. Both these pixels and the pixels to which they are connected are called ganged pixels. Due to the increased capacitance across the sensor, ganged pixels have a mean noise of 280 e, which is higher than the noise for pixels that are directly connected to the readout. Pixels between two ganged pixels are called inter-ganged pixels. Inter-ganged pixels pick up some of the noise from ganged pixels, and thus also have slightly high noise value, on average 172 e. Both ganged and inter-ganged pixels can be seen in Figure 8 as the rows of pixels in the middle of the module, between the upper and lower front-end chips. Pixels that are both long and ganged (interganged) have a mean noise approximately 37 e (20 e) higher than regular ganged (interganged) pixels.

The mean threshold value is within 13 e for all pixel types. However, higher noise interferes with the tuning procedure, producing a larger spread in threshold values for non-normal pixels. This can be seen in the RMS of Figure 9(a). The dispersion of normal pixels is 37 e, but can be as high as 51 e for ganged pixels.

4 Improvements for 2009

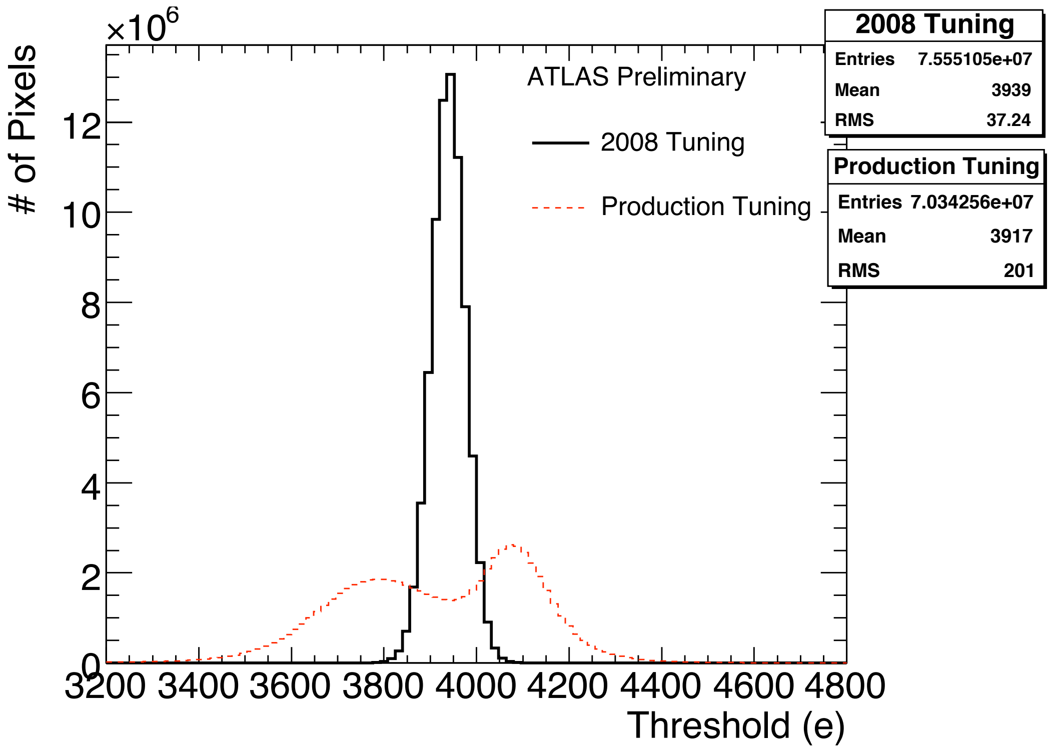
Several changes have been made to the software, hardware, target thresholds, and tuning algorithms that are used starting in summer 2009. A new version of the DSP code is available, and it improves the performance of threshold measurements and tunings. In the 2009 DSP code, correct mapping of the VCAL setting to charge value eliminates the roughly 60 e offset in the measured value of the threshold reported by the THRESHOLD_SCAN. Failures during the S-curve fitting procedure have been reduced by upgrading the DSP router firmware to better handle corrupted data. Due to improvements in the allocation of memory on the ROD, both threshold measurement and tunings can be performed on all modules on the detector in parallel, reducing by a factor of two the amount of time needed to scan the full detector. The TDAC_FAST_TUNE algorithm was also changed to set pixels that reach the minimum possible TDAC value to a TDAC value of 127.

Threshold measurements can be performed on approximately 97.9% of the detector for fall 2009 calibration, as opposed to the 94% for which data was available in 2008. This improvement encompasses both the improvements to the measurement procedure in the DSP software and recovery of cooling loops and optical connections. The remaining problematic modules are either inoperable due to hardware problems, such as open high voltage or low voltage lines, or cannot be threshold tuned due to failures during optical tuning or digital injection.

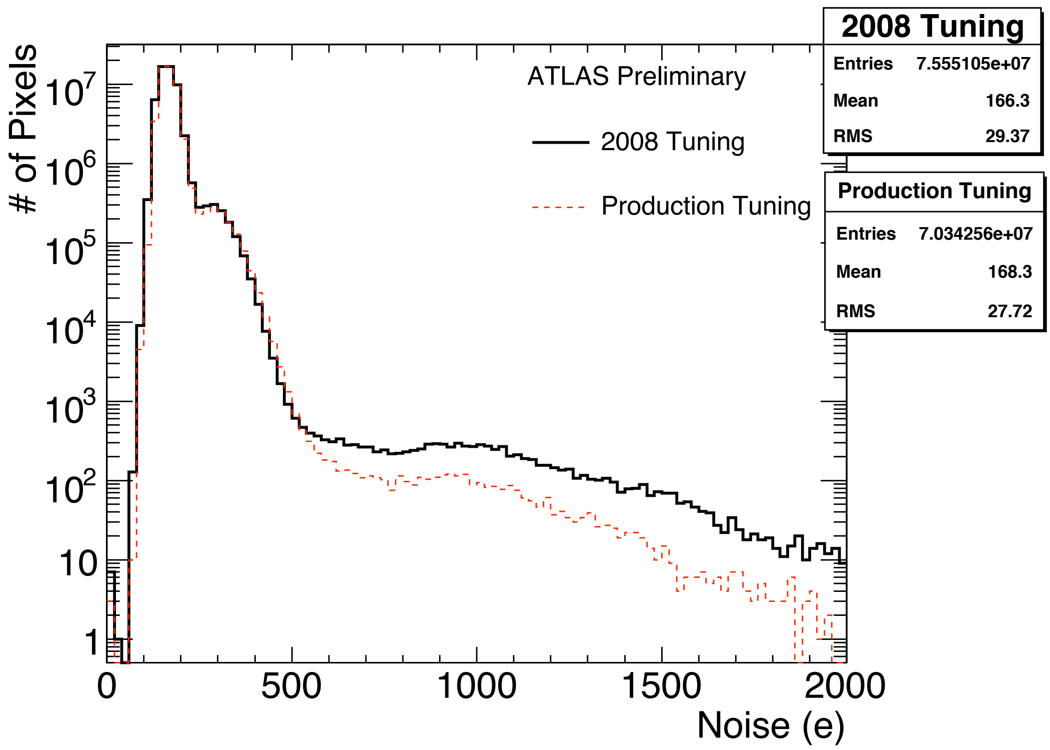
In 2009, threshold tuning will be performed with the target values of 3500 e and 3000 e, in addition to 4000 e. It is hoped that these lower threshold values will have a reduced number of split clusters and improve the position resolution with respect to 2008 cosmic data, without a noticeable loss in efficiency, which is presently near 100% for working pixels in the barrel region.

5 Conclusion

In autumn 2008, the threshold tuning procedure was performed on the full Pixel Detector for the first time. The target threshold of 4000 e was chosen as a reasonable set point to have both good hit detection efficiency and good cluster position resolution. Approximately 94% of the detector was successfully tuned to a measured mean value of 3939 e, which is offset from the true threshold value by $\sim$60 e, with a RMS over the full detector of 37 e. The remaining 6% was either inoperable, remained with the production tuning, or was correctly tuned, but could not be measured. Improvements to both the software and hardware make approximately 97.9% of the Pixel Detector available for threshold tuning in 2009.

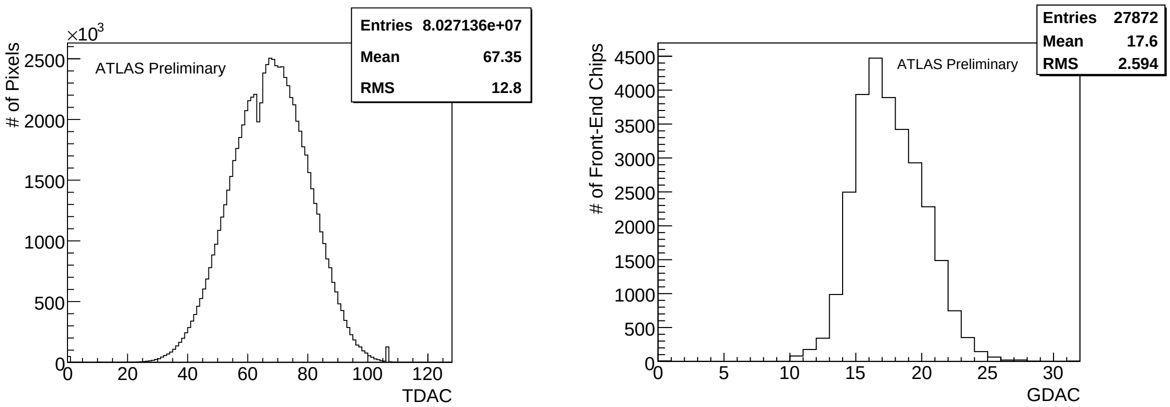


(a)



(b)

Figure 6: Threshold (a) and noise (b) distributions as measured for 1527 modules with the production threshold tuning (dashed) and 1642 modules with the 2008 tuning (solid).



(a) (b)

Figure 7: Distribution of TDAC values (a) and GDAC values (b) for all 1642 modules in the 2008 tuning. Note that this includes modules that were not successfully TDAC-tuned or do not have reliable threshold measurement data.

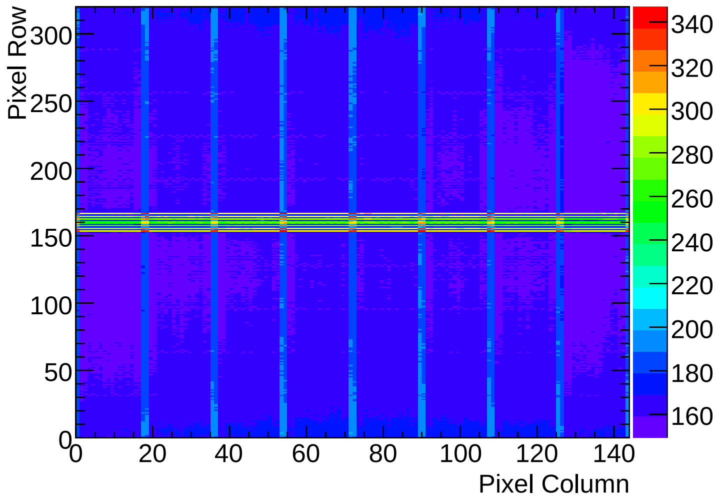
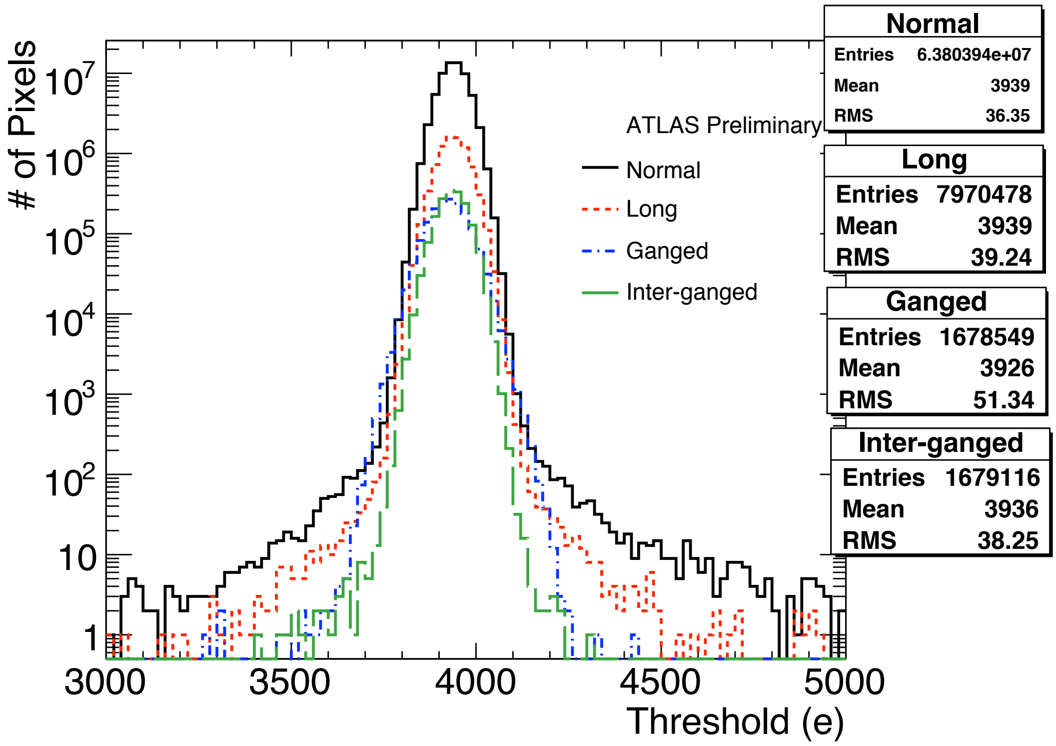
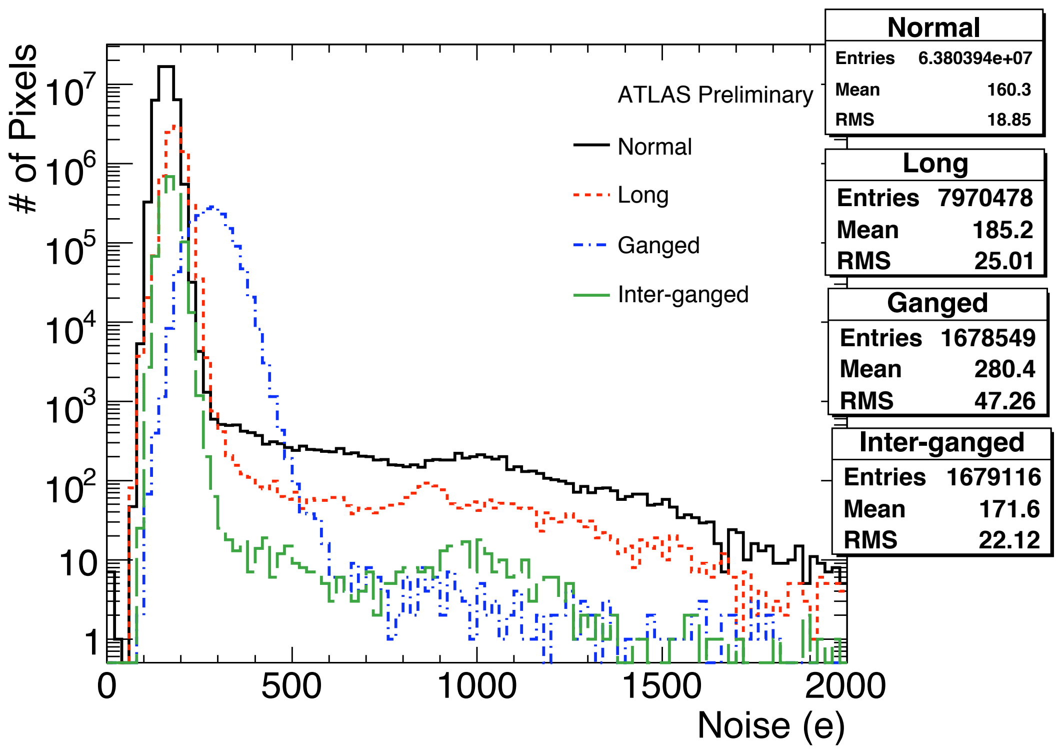


Figure 8: Map of average noise per pixel for the 1642 modules with the 2008 tuning. Most pixels are normal pixels, with an average noise of 160 e. Long pixels can be seen on the columns between front-end chips, with an average noise of 185 e. Ganged and inter-ganged pixels can be seen in the middle rows of the module, between the upper and lower front-end chips, with an average noise of 280 e and 172 e, respectively.



(a)



(b)

Figure 9: Threshold (a) and noise (b) distributions for normal, long, ganged, and inter-ganged pixels on 1642 modules with the 2008 tuning.

6 Appendix

Table 3: THRESHOLD_SCANs used in this document.

Scan #	Comment
11880	production tuning
11881	production tuning
13902	final tuning
13912	final tuning
13922	final tuning
13654	final tuning, consecutive scans
13657	final tuning, consecutive scans

References

[1] The ATLAS Collaboration, *ATLAS pixel detector: Technical Design Report, 1*, CERN-LHCC-98-013, CERN, Geneva, May, 1998.

[2] G. Aad et al., *ATLAS pixel detector electronics and sensors*, Journal of Instrumentation **3** (2008) no. 07, P07007.

[3] S. Montesano et al., *Calibration of the Charge Sharing Algorithm for the ATLAS Pixel Detector using Cosmic Data*, In publication.

[4] G. Aad et al., *Cosmic Ray Detection Efficiencies for the ATLAS Pixel Detector*, In publication.

[5] dell'Asta, L. et al., *Measuring the Charge Scale of the Pixel Detector with Cosmic Ray Data*, ATL-INDET-INT-2010-003, CERN, Geneva, February, 2010.

[6] J. Grosse-Knetter, *Vertex Measurement at a Hadron Collider - The ATLAS Pixel Detector.* PhD thesis, Universitat Bonn, 2008.

[7] Biesiada, J. et al., *The Implementation and Performance of ROD DSP Software in the ATLAS Pixel Detector*, ATL-INDET-INT-2010-006, CERN, Geneva, February, 2010.

[8] Andreazza, A. et al., *ATLAS Pixel Module Electrical Tests Description*, ATL-IP-QP-0144, CERN, Geneva.

[9] Garelli, N. et al., *The Tuning and Calibration of the Charge Measurement of the Pixel Detector"*, ATL-COM-INDET-2010-017, CERN, Geneva, November, 2009.