The Online Calibration, Operation and Performance of the CMS Pixel Detector

B. Kreis

Abstract

The CMS pixel detector consists of approximately 66 million silicon pixels whose analog signals are read out by 15,840 programmable readout chips. With the recent startup of the LHC, the detector is now collecting data used for precise vertexing and track-finding. In preparation for data taking, the detector’s readout chips and their supporting readout and control electronics were calibrated. We will describe the calibration that has taken place since the detector’s installation in the summer of 2008, focusing on the optimization of the readout chips’ thresholds and analog response. We will then describe some aspects of the operation of the detector during the early running of the LHC.

Keywords: Pixel detector, Readout chip, Vertexing, Tracking, CMS

1. Introduction

The CMS pixel detector is the innermost tracking device of the CMS experiment located at the Large Hadron Collider (LHC) at CERN. The detector consists of approximately 66 million “n on n” silicon pixels which are arranged in three barrel layers (BPix) and four forward disks (FPix) to provide coverage over the pseudorapidity range |\(\eta|<2.5\). The detector reads out analog pulse heights so that signal interpolation can be used to achieve the hit position resolution required for CMS vertexing and track-finding [1, 2].

The pixel detector was installed and commissioned in 2008 [3]. The detector was calibrated in the time leading up to and throughout the commissioning in 2008, and its performance was studied with cosmic ray muons [4]. Further calibrations were performed in 2009 as the LHC prepared to deliver collisions. These calibrations, which are described in Sections 3 and 4, mainly addressed the thresholds and analog response of the detector. The detector has been operating since collisions began in December 2009. Parts of the data acquisition system are recalibrated on a regular basis to account for environmental changes and to monitor the detector’s status. These calibrations are described in Section 5.

2. Pixel Data Acquisition

Groups of 4,160 pixels, making up 80 rows and 26 pairs of columns, known as double-columns, are readout by the PSI46V2 Readout Chip (ROC) [5]. After amplification and shaping, zero-suppression is performed on the ROC with a comparator for each pixel. When a pulse crosses the comparator’s threshold, it is considered as a hit, and the analog pulse height, the address of the pixel and the bunch crossing number are stored in buffers dedicated to its double-column for the latency time of the CMS first level trigger. The ROC reads out single bunch crossings; hits are validated by the trigger and sent on to the pixel data acquisition system if the bunch crossing number of the hit and the trigger match.

The ROC contains 21 8-bit digital-to-analog converters (DACs), five 4-bit DACs and one 3-bit DAC, which influence various aspects of the readout. In addition, each pixel has four trimbits, which influence the comparator’s threshold, and one bit to mask the pixel if needed. The DAC settings are programmed before running the detector. Calibration signals can be injected with an 8-bit DAC known as VCal. On average, one VCal DAC unit corresponds to 65.5 electrons, with an offset of -414 electrons [3].

The ROCs are readout and controlled by VME devices in the CMS electronics room. The analog elec-
trical signals from groups of ROCs are converted to an optical signal and sent to the Front End Driver (FED) VME device [6]. The FED digitizes the signal when it is triggered by the CMS first level trigger.

3. Threshold Calibrations

The thresholds of the detector are important performance parameters because they influence the cluster size, and therefore, the hit position resolution. The threshold of a pixel’s comparator depends on the four trimbits and two 8-bit DACs on its ROC known as VcThr and Vtrim. The impact of these settings on the threshold of pixel i’s comparator, Thr_i, is given by,

\[ Thr_i \sim -C_0 VcThr - C_1 Vtrim (15 - \text{trimbits}_i), \]

where C_0 and C_1 are positive constants. As seen in Equation 1, VcThr applies an offset to the threshold of every pixel on the ROC, and Vtrim determines how much influence the trimbits on the ROC have.

Due to timewalk, the smallest signals that cross threshold may only do so in a bunch crossing following the triggered one in which the charge was actually deposited. For this reason, two thresholds are defined for each pixel. The first is the absolute threshold, which is the charge required to cross threshold independent of the time at which it does so. It is precisely equal to the comparator’s threshold. The second is the in-time threshold, which is the charge required to cross threshold in the same bunch crossing as the one in which the charge was deposited. The absolute threshold is relevant when studying occupancy, noise and cross-talk, and the in-time threshold is relevant when studying hit reconstruction.

The true in-time thresholds depend on the timing of the detector’s clock with respect to the LHC’s collisions, however it can be estimated using charge injection. The timing of the charge injection is set so that the maximum injected charge crosses threshold approximately five nanoseconds into the bunch crossing. Both the absolute threshold and approximate in-time threshold of a pixel are measured using so-called S-Curves. An S-Curve is the efficiency for injected charge to cross threshold in a specified bunch crossing versus injected charge. The in-time threshold is taken as the location of the turn-on of the S-Curve from the bunch crossing in which the charge was injected, or in-time bunch crossing (see Figure 1). The absolute threshold is taken as the location of the turn-on of the sum of S-Curves from the in-time bunch crossing and the following one (see Figure 1). The location of a turn-on is taken as the injected charge at which an error function fit to the (summed) S-Curve(s) reaches 50% efficiency.

3.1. Threshold Trimming

In the first step of the threshold calibration, the thresholds on each ROC, either the absolute or in-time thresholds, were adjusted, or trimmed, to the same VCal value by tuning VcThr, Vtrim and the trimbits. In the BPix, the absolute thresholds were trimmed using an algorithm described elsewhere [7]. In the FPix, the in-time thresholds were trimmed using an alternative algorithm.

The FPix trimming algorithm is an iterative algorithm. In each iteration, changes in the in-time thresholds resulting from small changes in VcThr, Vtrim and the trimbits are measured using S-Curves. The next values for VcThr, Vtrim and the trimbits are solved for using a first order Taylor expansion of Equation 1 for each pixel and some additional requirements to constrain the problem. Approximately four iterations are performed on a small but representative subset of the pixels on each ROC (~2% distributed across the ROC) to save time. In the final iteration, the in-time threshold of every pixel is measured, and then based on the average influence of the subset of trimbits measured in the previous iterations, the final trimbit value of every pixel is chosen.

A histogram of the ROC absolute threshold RMS is shown in Figure 2. All of the RMSs are well below the variation in VCal. The FPix RMSs are slightly larger than those of the BPix because the in-time thresholds, rather than the absolute thresholds, were trimmed.
3.2. In-time Threshold Calibration

The in-time thresholds depend on the amount of timewalk introduced in the amplification and shaping that occurs before the signal reaches the comparator. This depends on Vana, an 8-bit DAC that regulates the voltage applied to the analog part of the ROC. Vana was set in such a way that balanced the desire to minimize timewalk with the need to keep the analog current drawn by the detector at a reasonable level. The Vana setting for each BPix ROC was determined during module testing by directly measuring the analog current drawn by the ROC as a function of Vana and then choosing the Vana that corresponded to 24 mA.

The Vana setting for each FPix ROC was determined without a direct measurement of its analog current. On these ROCs, Vana was set so that the difference between the average in-time threshold and approximate absolute threshold was 12 VCal. This was done in an iterative procedure, where the next Vana was chosen based on the current difference. In each iteration, the absolute threshold and charge injection timing were recalibrated because they also depend on Vana. The impact of Vana on this quantity, which quantifies the timewalk, is shown in Figure 3. A difference of 12 VCal was chosen because it was found to make the average analog current drawn per ROC near the FPix target of 25 mA. Figure 4 shows the resulting analog currents.

As shown in this issue [8], various offline methods using collision data show that the true in-time thresholds are 700-1000 electrons higher than the absolute thresholds. This is consistent with our expectations from charge injection.

3.3. Lowering the Absolute Thresholds

Increasing the cluster size from one to two pixels so that signal interpolation can be used increases the hit position resolution of the detector. The cluster size was maximized by minimizing the timewalk (as described in Section 3.2) and then lowering the absolute thresholds. The absolute thresholds were lowered in a ROC-based approach, which works well after trimming has been performed. The absolute thresholds on a ROC should not be set below the level of cross-talk on the ROC. When the absolute thresholds are set below this level, the ROC is overwhelmed by spurious hits, which overfill the double-column buffers and prevent real hits from being read out. The absolute thresholds were set just above this failure point.

To begin the absolute threshold calibration, the absolute thresholds were set to modest values so that none of the ROCs were failing. This can be done quickly for most ROCs by adjusting the VcThr setting on each ROC until a representative subset of its pixels are 100% efficient to an injected charge of ~ 50 VCal, for instance.

Next, the absolute thresholds were lowered away from the working point found in the first step by raising the VcThr setting on all ROCs in five steps of 2 DAC units. At each step, the ROC was checked for failure by looking for failed S-Curve fits, which is one artifact of the absolute thresholds being too low. S-Curves from a small but representative subset of the pixels on each ROC (~ 2% distributed across the ROC) were considered to save time. The majority of ROCs fail within five steps. The VcThr setting of each failing ROC was
then moved away from its failure point by 4 DAC units. The VcThr setting of the ROCs that never failed were set 2 DAC units away from the lowest threshold setting tested.

After the absolute thresholds were lowered, the VcThr settings of a small number of ROCs required manual tuning due to some dependence between ROCs. First, any ROC featuring failing S-Curve fits was manually tuned. Then, in a second, independent test known as PixelAlive, the hit efficiency for charge well over threshold was measured for every pixel. Any ROC featuring inefficiencies due to the absolute thresholds being too low, such as inefficient double-columns resulting from filled buffers, was manually tuned.

The final absolute threshold distribution is shown in Figure 5. The mean absolute threshold is 2457 electrons, which is approximately 1/10th of the charge collected from a minimum ionizing particle. This threshold was achieved without introducing a significant number of inefficient or noisy pixels. The final number of bad pixels on ROCs included in the readout was measured in several ways. Noisy pixels were identified and masked during cosmic ray data taking. Pixels inefficient to charge injection were identified using the PixelAlive test; this test does not identify dead sensors or poor connections between the sensor and ROC. Dead pixels were identified by their lack of hits in high-statistics collision data. The final numbers are shown in Table 1.

### 3.4. Noise

The noise of a pixel is equal to the width of the turn-on region of its S-Curve, which is taken as the RMS width of the Gaussian function that would result from differentiating the error function fit. The BPix and FPix noise distributions are shown in Figure 6. As seen in Figure 6, the noise is well below the absolute thresholds, which are set just above the level of cross-talk, so it does not negatively impact the performance of the detector.

### 4. Analog Response Calibrations

The analog response of each ROC was optimized by maximizing the linearity and range of the gain. The gain of a pixel is defined as the pulse height versus injected charge. It is important for the gain to be as linear as possible because it is parameterized with a linear function by the offline reconstruction to limit the number of parameters it must store.

The linearity of the gain was maximized through the calibration of two 8-bit DACs known as VHldDel and Vsf. VHldDel determines a delay applied to each pulse...
before its height is sampled and held in a capacitor until
the double-column is readout. VHldDel was set so that
the maximum of a pulse is sampled. Vsf regulates the
time applied to the sample and hold circuit. Vsf was
calibrated differently in the BPix and FPix. The BPix
algorithm is based on the observation that linearity in-
creases as Vsf is increased. Vsf was increased on each
BPix ROC until the average linearity reached a target
value or a digital current limit was reached. In the FPix,
Vsf was set so that the pulse height when VHldDel is
set to its minimum is equal to the pulse height when
VHldDel is set to its maximum; this was also observed
to produce good linearity.

The range of the gain of one pixel is defined as the
difference between the pulse height due to the maxi-
mum injected charge and the pulse height due to in-
jected charge just above threshold. The range depends
on several DACs, but it can be maximized by calibrat-
ing only two of them on a ROC-by-ROC basis after
the rest have been set to compatible values. Vbias_PH
and VoffssetOp, both 8-bit DACs, were calibrated. Vbias_PH applies a gain to the pulse height, and VoffssetOp
applies an offset. The ranges of a small but representa-
tive subset of pixels on each ROC were measured in a
two-dimensional scan of these DACs, and the settings
that produced the largest ranges within the range of the
FED’s ADC were chosen.

The final gain parameters resulting from these cali-
brations are presented in this issue [8].

5. Regular Recalibrations

The vast majority of the settings obtained before the
2009-2011 LHC run began, including all of the ROC
DAC settings, will not be changed until after the run.

There are several FED parameters that are recalib-
rated on a regular basis to account for environmen-
tal changes and to monitor the detector’s status. The
most frequently changed parameters are offsets in the
optical receiver of each FED channel. These are recal-
ibrated approximately once per week, when tempera-
ture changes at the laser drivers near the front end shift
the signal beyond what can be handled by an automatic
correction in the FED. The automatic correction can ac-
count for temperature changes of 2 – 3 °C.

Approximately two to four times per month, the pa-
rameters necessary to decode the addresses of hit pixels
are remeasured. The address parameters are relatively
stable, so they are often remeasured only to check for
problems. Finally, approximately once per month, the
optimal phase of the FED’s ADCs are remeasured. The
phase parameters are stable, and therefore, this calibra-
tion is performed mostly as a check.

For more detail on these calibrations, see [3].

6. Online Experience with Beam Background

We observe showers of particles that graze the de-
tector along the beam axis and give rise to occupancies
much larger than those expected from collisions at the
LHC’s full design luminosity. These events are consist-
tent with expected interactions between beam and gas
in the LHC beam pipe.

On the order of 10k–100k pixels can register a hit in
just one beam background event. When they are coinci-
dent with a trigger, all of the hits must be read in by the
corresponding FED channels, which takes a long time.
This imposes a challenge to maintaining event synchro-
nization because the events that follow come long af-
ter the affected FED channels expect them. We have
implemented a two-part solution to maintain synchro-
nization when this occurs. First, a mechanism to ignore
delayed events, rather than confuse them with ex-
pected events, was implemented. The number of events
that must be ignored increases with the trigger rate. The
second part of the solution was implemented to prevent
this number from climbing too high. When the number
of events that must be ignored is greater than a config-
urable number, N, CMS triggers are stopped until the
affected FED channels can regain synchronization by
ignoring the next N events.
7. Conclusion

The performance of the detector has been optimized for the first LHC run. Calibrations performed in 2009 improved the threshold and analog response of the detector. A few FED parameters are recalibrated on a regular basis to account for environmental changes and to monitor the detector’s status. A mechanism to maintain event synchronization in the FED while it reads in large beam background events has been implemented.

Acknowledgments

The author would like to thank Anders Ryd, Danek Kotliński, Will Johns and Gino Bolla for helpful discussions. This work was supported by NSF PHY-0846388, NSF PHY-0757894 and (Simon COLA).

References

[1] CMS Collaboration, JINST 3 S08004 (2008).
[2] CMS Collaboration, CERN-LHCC-98-006 (1998).
[3] D. Kotliński, JINST 4 P03019 (2009).
[4] CMS Collaboration, JINST 5 T03007 (2010).
[5] H.Chr. Kästli et. al., Nucl. Instr. and Meth. A 565 (2006) 188-194.
[6] M. Pernicka et. al., Topical Workshop on Electronics for Particle Physics, Prague, Czech Republic, 03 - 07 Sep 2007, 487-491.
[7] A. Starodumov et al., Nucl. Instr. and Meth. A 565 (2006) 67.
[8] U. Langenegger, in this issue.