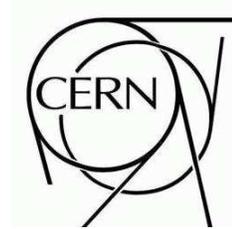


ATLAS NOTE

ATL-ATL-INDET-PUB-2007-xxx, V3.0

October 21, 2007



Pixel Offline Analysis for EndcapA Cosmic Data

The ATLAS Pixel Group

Abstract

This document summarizes the results of the offline analysis for the pixel endcapA system test cosmic data. The setup consists of one pixel endcap of three disks, for a total of 144 modules and 6.6 million channels, about the 8% of the full detector with the goal of exercising the readout system, data taking and testing the full chain of offline reconstruction. The observed noise occupancy per readout (BCID) is found to be 10^{-10} after removing all noisy pixels. Comparison with the detector characterization performed during the detector assembly shows that approximately 90% of these noisy pixels were already flagged as “special” during the production test and that the total fraction of special pixels is below 0.1%. The tracking studies, especially relating to tracks passing in the overlap regions between adjacent modules in the same disk, have been very useful in spotting problems in our geometry description and can be used for the relative alignment between the adjacent modules with the residuals of overlap hits. The characteristic of pixel clustering in the data are checked and agree well with Monte Carlo simulation. The pixel cluster efficiency has been measured to be >99.0% using the hits in the overlap region. When using the nominal geometry, an initial resolution of $23.0 \mu\text{m}$ is obtained. After a preliminary alignment this improves to $16.0 \mu\text{m}$, which is not far from $14.0 \mu\text{m}$ expected from the Monte Carlo simulation.

1 Introduction

In December 2006 an endcap of the ATLAS pixel detector was used to perform a commissioning test of the detector. For the first time, part of the pixel detector consisting of 144 modules, out of a total of 1744, was completely equipped with services and managed by an initial production of the ATLAS DAQ system components.

The Endcap has been operated in an environmental chamber withing the clean room used for the ATLAS Inner Detector assembly in the CERN SR1 building.

This operation has been an important step in the commissioning of the detector, showing that the full services, the communication chain between the on-detector and the off-detector electronics, the DAQ and DCS hardware, and software systems are properly integrated and capable to drive the detector.

The description of the components of the pixel detector and the readout electronics and Detector Control System will be described in detail in a Pixel Detector paper under preparation [1].

The experience gained in SR1 running is applied to the assembly of the final services. The results from the analysis of the performance of services are the subject of an accompanying ATLAS note [2]. That note will also contain the tests performed in order to calibrate the whole detector and prepare it for the data taking.

This note describes the results of the running of the system and the analysis of the data collected by the detector in two running modes:

1. with random triggers, in order to measure noise occupancy;
2. using a scintillator system to trigger on cosmic rays.

Noise measurements with random trigger have been taken in different conditions and are used to derive useful information for understanding detector properties and tuning the simulation:

1. the amount of *fixed pattern noise*, i.e. channels with higher than normal occupancy, and the correlation between these channels and the one detected as *special* during the module acceptance tests;
2. the rate of random noise. Previous test beam operation could only put an upper limit on the level or random noise [3].
3. Time over Threshold (TOT) spectrum for noise: differently from naïve expectation, this has proven not to have a Gaussian distribution and a more complex model needs to be setup;
4. dependence of noise rate from operational conditions: trigger rate, depletion voltage, and threshold settings.

Running with cosmic rays is used to derive information that needs a physics signal. Unfortunately, the time devoted to cosmic runs was not enough to collect the statistics needed to check functionality of every single channel. However, the amount of data written on disk is useful for:

1. checking the overall resolution and tracking reconstruction;
2. validating the simulation by comparing cluster size, ToT spectrum and timing distribution with what can be observed on the data and the calibration information;
3. exercising the alignment algorithms using tracks passing through overlapping regions between modules.

This effort is also the base for the next commissioning run with cosmics in the pit.

The note is organized as follows: at first a description of the setup geometry and cosmics trigger is given, then will review the results obtained with noise runs. After that the modification to the ATLAS tracking to reconstruct cosmics tracks will be described. Also overall tracking efficiency and rate will be computed, in addition to Monte Carlo validation using cosmics tracks. Finally, results on alignment and resolution will be presented.

2 Pixel EndCap A Cosmic Muon Setup

The pixel endcap A cosmic muon test [4] is using the same setup as the pixel system test in the SR1 building at CERN [5]. The cosmic test is a logical continuation of the system test with the goal of exercising the readout system, data taking and testing the offline reconstruction chain.

The aim is to collect a larger cosmic muon sample to reconstruct tracks passing through the detector, study their properties and perform a simple alignment of the detector using overlap regions on the pixel disks.

2.1 Pixel EndCap A Geometry

The pixel endcap A is one of the two end sections of the pixel inner detector, its geometry is described in [7]. It consists of three disks placed at 49.5 cm, 58.0 cm and 65.0 cm in the z -direction. There are 48 modules on each disk, i.e. 144 modules in the entire pixel endcap A. Each disk has 24 modules on the even (odd) side respectively. The even side is closer to the interaction point. The modules centers (i.e. the middle of the silicon wafer) are displaced by 4.276 mm according to the survey (the thickness of the disk implemented in the reconstruction software is only 4.2 mm however, the discrepancy will be discussed further in this note). The first module is positioned at 3.75° in φ , every other module is rotated by 7.5° in $r\varphi$ plane with respect to the previous module. The centers of modules are placed at 119.17 mm in radius.

The module consists of 16 front-end chips bump bonded to the silicon wafer (average thickness of $256\mu\text{m}$, the area $(x,y) = 1.88\text{ cm} \times 6.3\text{ cm}$, that includes all guard rings), there are two rows of eight FE chips on each module. The active area of the sensor is $(x,y) = 1.64\text{ cm} \times 6.08\text{ cm}$, each front end chip covers $0.76\text{ cm} \times 0.82\text{ cm}$, it has 16 columns of $400\mu\text{m}$ and 2 columns of $600\mu\text{m}$ (so-called *long*) pixels, and 160 normal plus 4 *ganged* rows of $50\mu\text{m}$ pixels. The geometry of the module and its dimensions are well described in [8]. Thus, the short side (local X direction) of the module has a $50\mu\text{m}$ pitch and the long side (local Y direction) has a $400\mu\text{m}$ pitch with the only exception of ganged pixels. They are cross connected and receive a special treatment in the simulation/reconstruction code. The position of a module on a disk is defined by the module η index (it is the same for the even and odd side modules of a particular layer, it is 0, 1 and 2 for disk 1A, disk 2A and disk 3A respectively) and the module φ (it goes from 0 to 47, and we use it to distinguish front (=even), back(=odd) modules on one disk). The module φ id can be used to calculate the φ of the module center by using Eq. 1.

$$\varphi_c = (\varphi_{mod} + 0.5) \times 7.5 \times \pi / 180 \quad (1)$$

The properties, position and status (ON=in readout, OFF=out of readout) of pixel endcap A modules is given in Figs. 1, 2, 3, 4, 5 and 6.

The (x,y) position of a pixel on pixel endcap module is defined by the η index (local Y-axis, η direction, column number) and φ index (local X-axis, φ direction, row number). There are 144 columns (coarse pitch) and 328 rows (fine pitch) on the module. The η index increases with an increasing radius for both the both even and odd side modules while the φ index increases with increasing (decreasing) global φ for the even (odd) side modules. The same rule is also applied for the local coordinate system

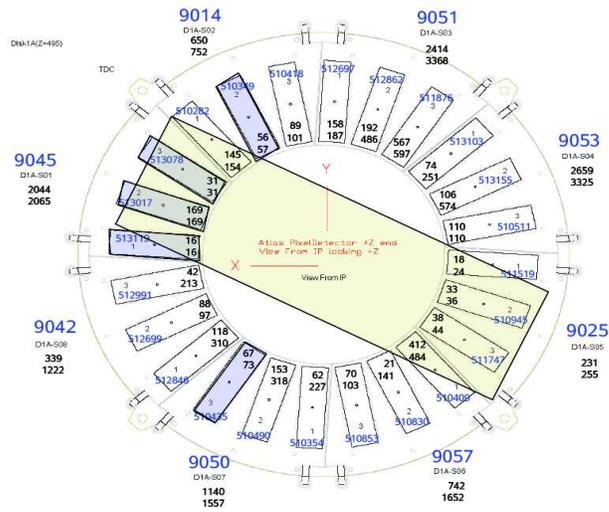


Figure 1: Disk 1A even side modules.

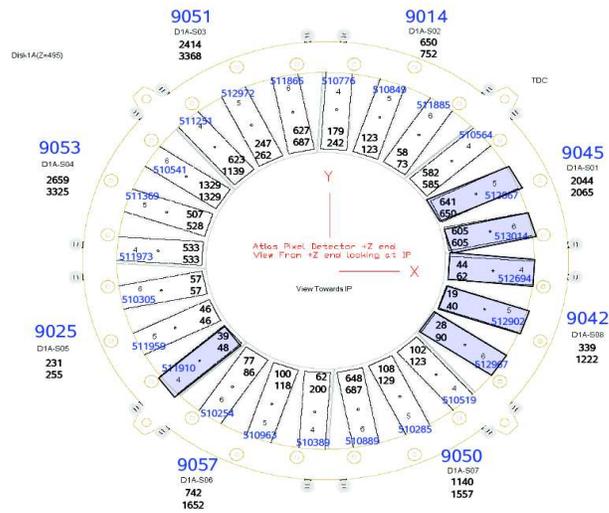


Figure 2: Disk 1A odd side modules.

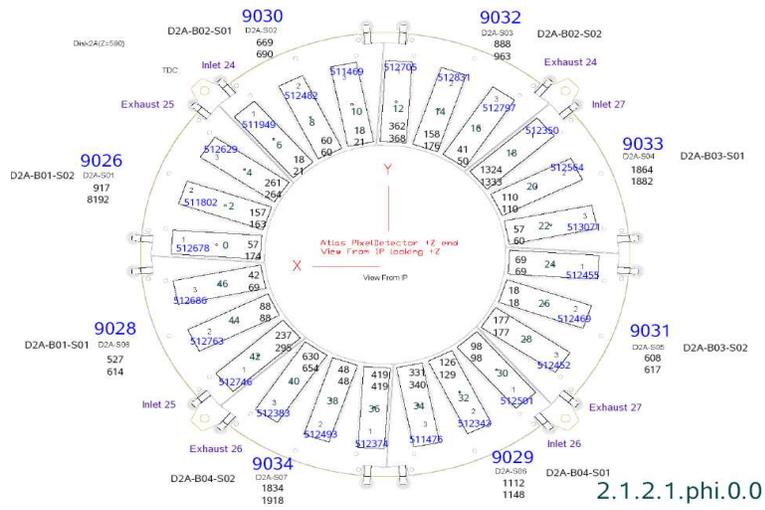


Figure 3: Disk 2A even side modules.

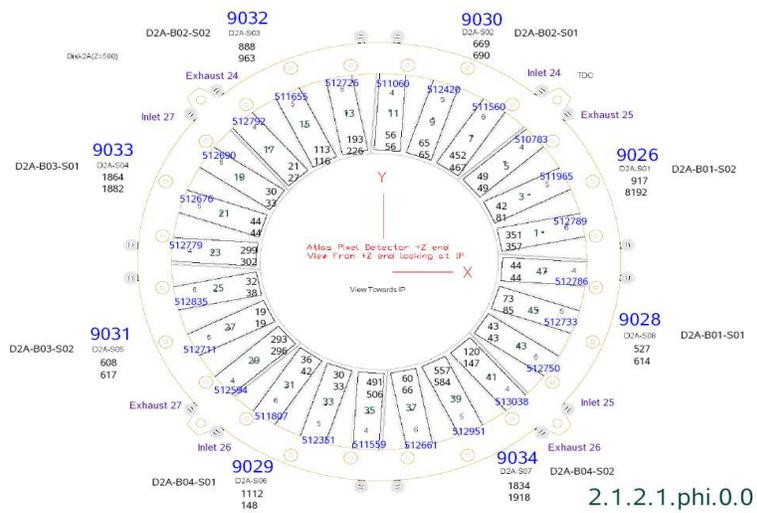


Figure 4: Disk 2A odd side modules.

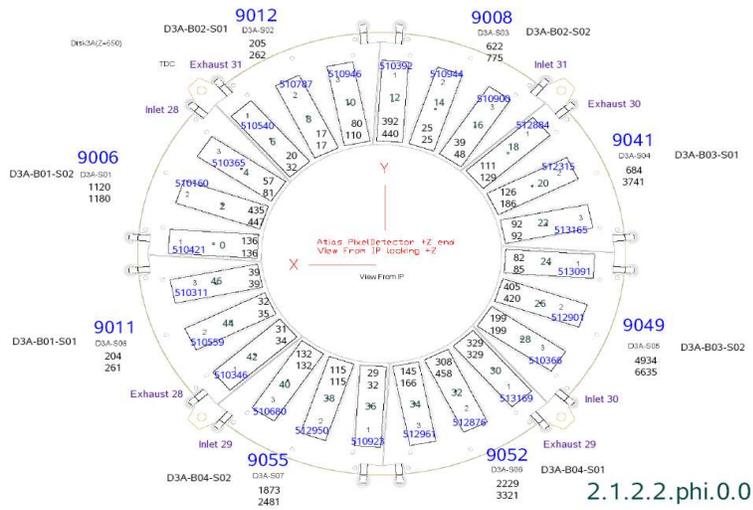


Figure 5: Disk 3A even side modules.

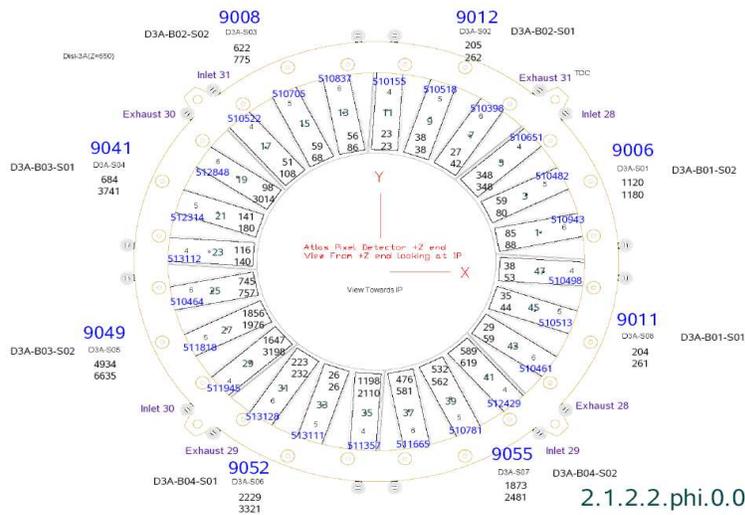


Figure 6: Disk 3A odd side modules.

on the pixel endcap module with the coordinate center defined as a center of the module (that is the actual center of the silicon sensor). Note that the even module local X coordinate orientation is the same as the local X of an odd module.

The global coordinate system is defined in Figs. 1, 2, 3, 4, 5 and 6.

2.2 Cosmic Setup

The pixel endcap A cosmic setup consists of the pixel endcap A that is hooked up inside the dry box providing the required environment for the pixel endcap operation (dry air mainly), and a prototype service quarter panel (PSQP) connected to all outside services (cooling, low voltage and high voltage distribution and regulation, readout, environmental information etc.). Both the pixel endcap and PSQP are placed inside the dry box. The endcap hangs vertically inside the dry box (i.e. its axis is perpendicular to the table top), that is an obvious requirement to maximize the flux of cosmic muons passing through the pixel endcap fiducial volume as well as to maximize the number of at least 3-hit tracks.

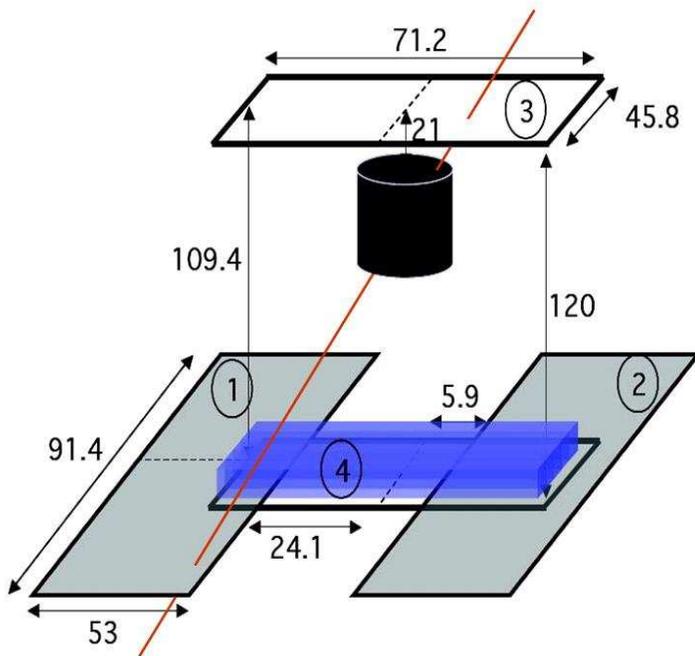


Figure 7: Schematics of a pixel endcap A cosmic setup.

2.2.1 Mechanical design

The trigger system [9] (dimensions of the system are given in Fig. 7) consists of four SLAC scintillators, two smaller scintillators are placed above each other (the top scintillator only 21 cm above the end section, i.e. 23.5 cm above disk 3A or 39 cm above disk 1A and the bottom scintillator 120.0 cm below the top one, i.e. 96 cm below the disk 3A). The small scintillators (45.8 cm \times 71.2 cm) are centered around the z-axis of the end section. The top scintillator is referred to as scintillator no. 3 and the bottom one as scintillator no. 4. These two scintillators are the bare minimum to have a good coincidence circuit (described in Fig. 9), the top scintillator is required to trigger the cosmic muon and the bottom scintillator defines geometrically the acceptance of the trigger system. There are many muons that leave the signal in scintillators 3 and 4 but never pass through the pixel endcap or leave only one/two hits. The endcap A is rotated by $-\pi/8$ with respect to the y-axis of the dry box (the y-axis of the dry box is parallel to the long side of the dry box table, i.e. axis of the PSQP). This rotation can be seen in Fig. 1.

Additionally, we had a luxury of additional two large scintillators (53.0 cm \times 91.4 cm) under the table top of the dry box that improve the trigger efficiency of the system.

Whereas the placement of top and bottom small scintillator is naturally determined by the length (or height in this particular case) of the endcap, the height of the PSQP and by the access essentially required to connect exhaust copper extension tubes to the endcap cooling pipes, in order to decide what is the optimal placement of large scintillators, we had to run a toy Monte Carlo simulation to make a decision. The results of this simulation are presented in Fig. 8, maximizing the weighted hit density coverage on the surface of the bottom large scintillators (left plot in Fig. 8).

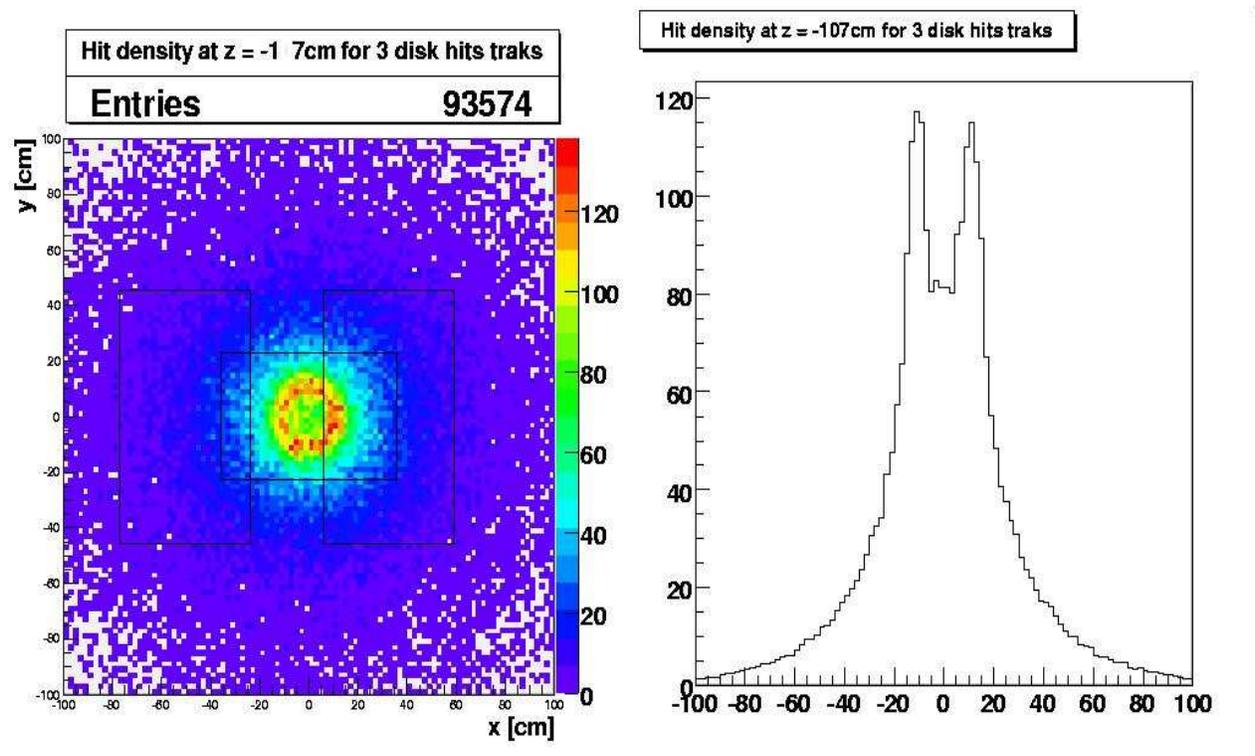


Figure 8: Toy Monte Carlo study dedicated to optimize the large scintillator placement.

To remove low momentum cosmic muons which suffer most from the multiple scattering, we have added a ~ 12 cm thick layer of iron between two small scintillators, directly below the endcap. That

helps to filter the muons out below 140 MeV.

2.2.2 Coincidence circuit

The layout of a coincidence circuit is given in Fig. 9. The starting point is to require a signal in the top scintillator (scintillator no. 3) and require logical *AND* with logically *OR* of all bottom scintillators (one of them is the small bottom scintillator, and two are the large ones in the front and back of the setup).

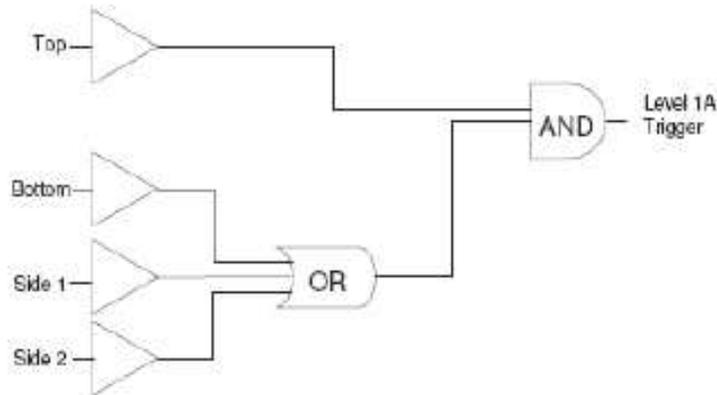


Figure 9: Coincidence circuit.

In the cosmic trigger and readout system we had three crates (the following describes an ideal signal path):

- *NIM crate*: it was a crate with all the HV power supplies for the PMT tubes in the setup and the logic electronics (discriminators, delay units and AND/OR units). The final output of the AND logical unit comes out of the crate as an input for TDC in the VME crate in the rack area, it is actually equivalent to L1A Trigger except in the real experiment.
- *VME crate*: this is a crate that holds SBCs, LTP, TTC and a BUSY unit (OR). The TDC receive TTL signal from discriminators and AND/OR logic unit, and in principal we could use this information for some timing study. However, we never got a chance to do it and this information is not available offline. This is why only the output of the AND/OR unit is received by LTP, transferred to TTC and finally to TIM in the ROD/DAQ crate.

- *DAQ crate*: it consists of several RODs (12 RODs needed to readout all 144 modules in endcap A) and TIM. The trigger signal from TTC (in VME crate) goes to TIM and then gets distributed among RODs. In the reality we had 2 DAQ crates, one with 8 RODs and the other with 4 RODs.

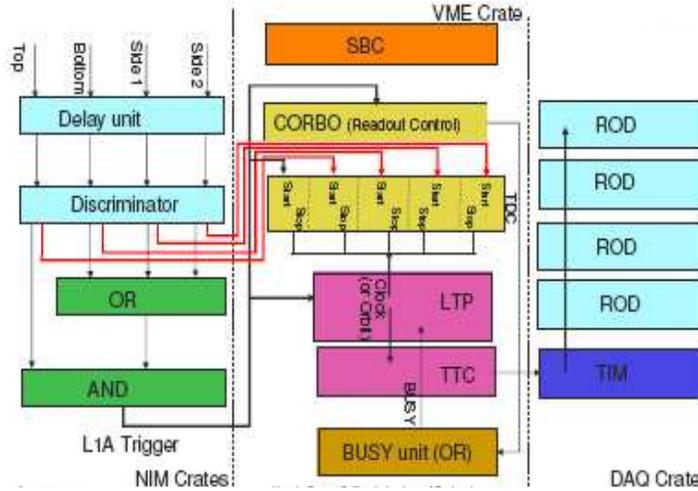


Figure 10: Ideal signal path.

2.2.3 Cosmic muon rate

The expected cosmic muon rate was studied extensively using the toy Monte Carlo simulation, full ATLAS Pixel Detector simulation and cross-checked with the back of the envelope calculations.

The integral intensity of vertical muons above 1 GeV/c at sea level is $70 \text{ m}^{-2}\text{s}^{-1}\text{sr}^{-1}$ [10]. It is an equivalent of 1 cosmic muon per cm^2 every second. The distribution of cosmic muons is roughly $\cos^2\theta$ of the incidence angle, and symmetric in ϕ . The mean of the cosmic muon momentum at the sea level is $\sim 4 \text{ GeV}/c$.

The toy Monte Carlo start with a randomly distributed cosmic muon passing through the top level scintillator. It assumes the $\cos^2\theta$ incidence angle distribution, and a full symmetry in the ϕ angle. It has a flat momentum distribution and no detector effects assumed. The overall trigger efficiency is expected to be close to 85 % and the disk hit efficiency is roughly 90 %. We count how many times the cosmic muon will pass through the bottom scintillator(s) and how many hits will be associated to a muon traversing the fiducial volume of the endcap given the geometry of the detector.

The full ATLAS pixel detector simulation takes all the detector effects into account. We start with the cosmic muon generated in cosmic muon generator (it has all angular and momentum distributions simulated correctly), count how many times do we have a coincidence in the top and bottom scintillators, simulate the response of the detector to cosmic muon passing through the sensitive layers, build space points/hits and reconstruct tracks. We also count the number of tracks and hits on the tracks to estimate the trigger rate and the track reconstruction efficiency (convoluted with the detector acceptance indeed).

The toy Monte Carlo estimates the cosmic muon rate through the top scintillator is 54 Hz and the trigger rate with all four scintillators is 16 Hz. The full detector simulation gives 6 Hz for the coincidence

of top and bottom scintillator and 18 Hz for all four scintillators. These two values are in a good agreement with the trigger rate measured in the DAQ system, we have recorded cosmic data with a data taking frequency of about 15 Hz.

The track reconstruction efficiency (again, convoluted with the detector and setup acceptance), if all 144 modules are readout, estimated from the simulated data was expected to be in the vicinity of 6 %, i.e. we would expect roughly one three hit cosmic muon track every second. In the reality, only 115 modules were readout, some disabled modules have clustered in space (e.g. the whole sector, i.e 6 modules, were disabled) and that is why the track reconstruction should be 20-30% lower than that, i.e. $\sim 4\%$. This is really what we have measured in the data, $\epsilon = 3.8\%$ (see following sections of this note). The ideal cosmic muon rate for at least 3-hit tracks is 1.5 Hz, and probably about 1 Hz for the detector with 29 disabled modules.

The track properties are affected by our choice of the cosmic trigger system layout. For example, the fact that the small scintillators are rectangular will deform the initial flat angular φ distribution into a sin wave distribution. The fact that the scintillators are rotated with respect to the end section by $\pi/8$ will shift the φ angular distribution by this amount to the negative side. Another interesting feature is that modules missing in the readout will create dips in the angular φ distribution (see the following section of this note for details). There were many reasons for having that many modules disabled from the readout: disconnected cooling loop, malfunctioning opto board, missing bias voltage, missing NTC reading, etc. All these effects were also implemented in the detector simulation, see Fig. 11.

2.3 Cosmic setup simulation

The goal of the simulation chain is to preserve existing ATHENA structure and geometry of packages as much as possible without introducing too many changes. The reason is that one of the cosmic test priorities is to validate the simulation/reconstruction chain from the beginning to the end, i.e. from simulation, digitization, bytestream converter all the way to reconstructed tracks and alignment. This is why if we find any abnormalities, bugs or mistakes, we would like to correct them, and make them part of ATHENA release 13. We have quite successfully managed all that with the only exception of using the combined test beam standalone tracking code with no magnetic field.

The full ATLAS cosmic setup simulation is done in the following five steps:

- *cosmic muon generator*: cosmic muons get randomly generated according to all distributions described in [10] in *CosmicGenerator* package [11]. The core of the package is an old Fortran cosmic muon generator inherited from previous experiments that is wrapped up inside the C++ code to provide the necessary interface with other ATHENA simulation packages.
- *pixel endcap A geometry*: the pixel endcap A geometry is identical to the one implemented in *PixelGeoModel* [12]. The only difference is that one has to switch off endcap C, the barrel of the pixel detector, pixel support tube, frame and services. The whole pixel detector is essentially reduced to endcap A only.
- *GEANT 4 detector simulation*: the GEANT 4 (G4) is described mainly in *G4AtlasApps* package [13]. It contains definition of all setup positions and dimensions (pixel endcap position, scintillator and iron positions/dimensions etc.). The pixel endcap A (or the pixel detector that is reduced to pixel endcap A) is positioned in its nominal position.
- *trigger system simulation*: the scintillators are placed in the right position in *G4AtlasApps*, when the cosmic muon passes through their fiducial volume, the energy is deposited in that volume

and if it is above some minimum amount it is considered to be a hit in the scintillator system. The logic coincidence between two scintillators is implemented in *InDetCosmicSimAlgs* package [14]. Events where there is no coincidence between top and logical OR of bottom scintillators are skipped, only triggerable events are passed further to digitization.

- *digitization*: the digitization code is the same as it is currently implemented in release 13, the only difference is that one has to make sure that hits from modules that are not present in readout (that includes pixel endcap C and barrel modules) are not being digitized. That would cause runtime errors.

The whole simulated pixel endcap A cosmic setup can be seen in Fig. 11 bottom, and 12. The how to run the simulation code instructions are described in [16].

3 Study of Noise Data

For the study of the noise in endcap A several runs were performed with various trigger signals that initiated by either a cosmic trigger or an external clock(random trigger). The data from several of these runs was analysed for characteristics of the noise signal. Maps of the hits on each module were used to study the correlation between the positions of noise hits in the data and the positions of pixels that had been marked as special during the production tests of the individual modules. Figure 13 shows a comparison of the noise levels for the modules of endcap A determined from threshold scans during the production and system tests. One can see that while the noise level in the system test is slightly higher than in the production test the results generally agree well with each other.

3.1 Results with Cosmic Triggers

Data from the cosmic rays are collected through the use of a triggered scintillator system. When a cosmic muon fires a trigger, the data are recorded with a unique bunch crossing identifier (BCID) and read out 16 times that $BCID=5$ corresponds to the event that contains a real cosmic signal. This characteristic provides a distinction separation for the pixel hits between the random noise, which can have any BCID, and the typical cosmic signal. With a cut of $|BCID - 5| < 2$ for cosmic signals, pixel hits outside this range were classified as noise hits ($|BCID - 5| \geq 2$).

The pixel occupancy, the number of noise hits of a given pixel per event readout, was investigated using a threshold of 10^{-5} . A pixel was classified as hot according to its pixel occupancy as shown in Figure 14 from Run 1129. Pixels with an occupancy of 10^{-5} or greater were defined as hot pixels; there were 1547 pixels identified as hot for run 1129, which is roughly 0.023 % of the total pixels in EndcapA.

Figure 15 shows the number of hot pixels per module, which reveals some of the noisier modules (layer 0 module 34 and layer 2 module 32). Further investigation of the local position of the hot pixels on module 32 of layer 2 shows an increased density along the boundaries of individual chips in Figure 16.

3.1.1 Run 1125

The *module* occupancy is defined as the fraction of pixels per module, event, and BCID read out, and it was computed for each of the three endcap A layers, as shown in Figures 17, 18, and 19. There are large fluctuations in occupancy, in particular for events not trigger by the cosmic muons, due to the presence of hot (noisy) pixels.

The *BCID* of all selected (good) and hot pixels is shown in Figures 20, and 21. As expected, hot pixels generated by noise are not correlated with any L1-trigger, whereas all signal pixels cluster around the cosmic peak corresponding to $BCID = 5$.

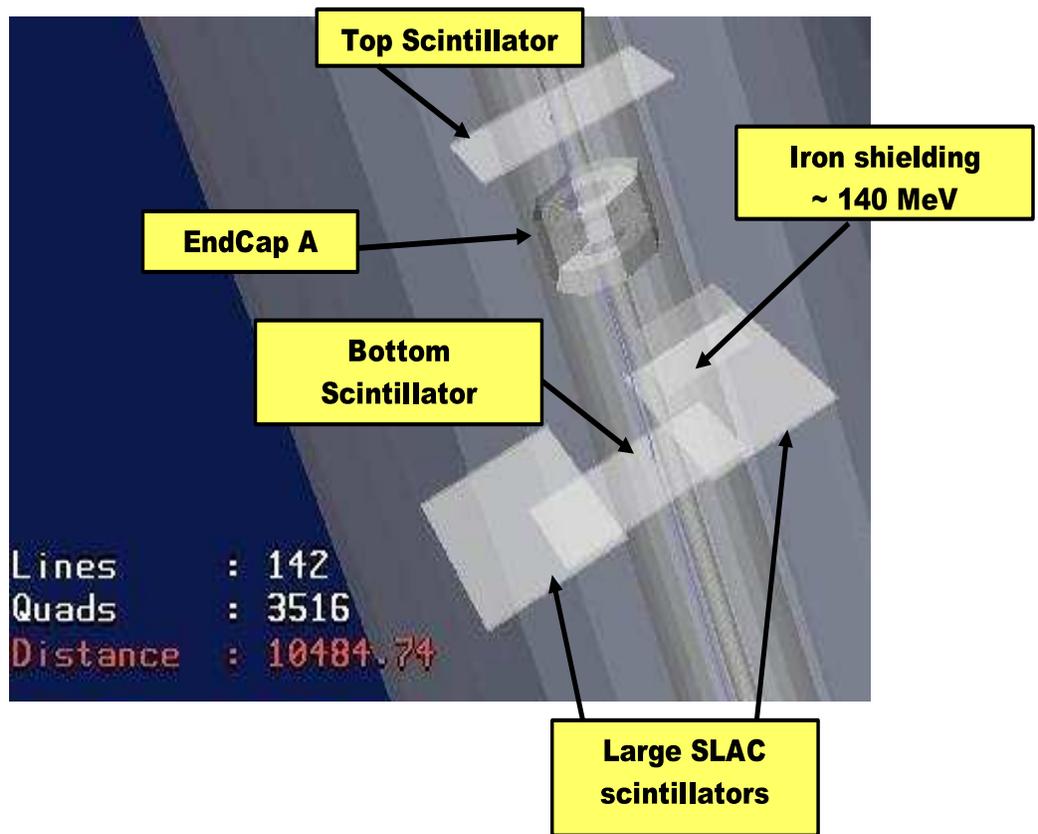


Figure 12: Simulated pixel endcap A cosmic test setup with a cosmic muon passing through it.

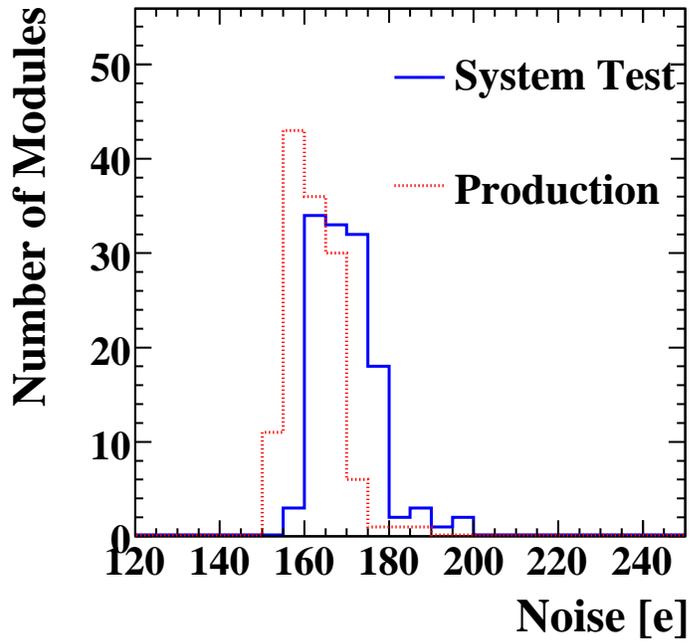


Figure 13: Noise in electrons for the modules of endcap A as determined from threshold scans during the production and system tests

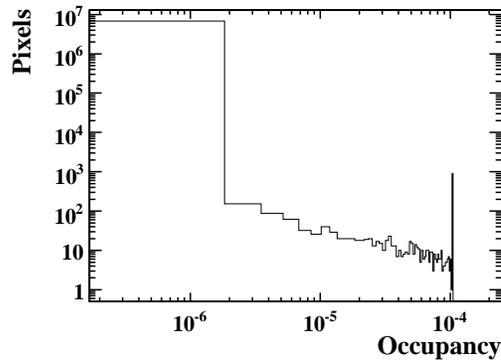


Figure 14: Pixel occupancy per event readout in run 1129.

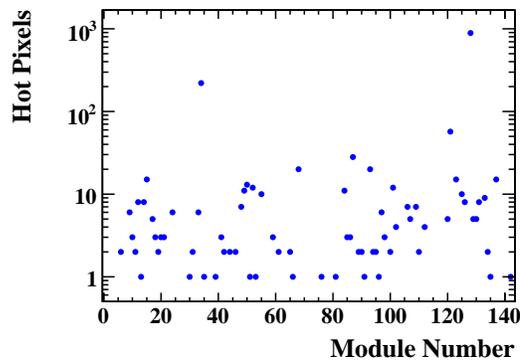


Figure 15: The number of hot pixels per module in run 1129.

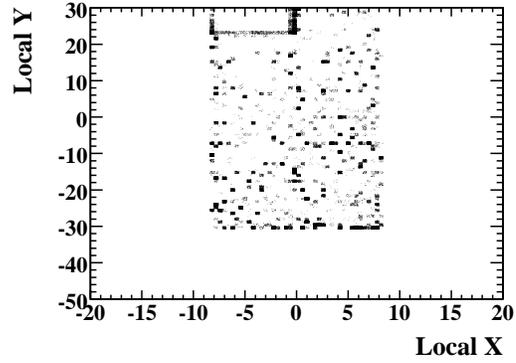


Figure 16: The local position of the hot pixels on one of noisy module 32 of disk 2.

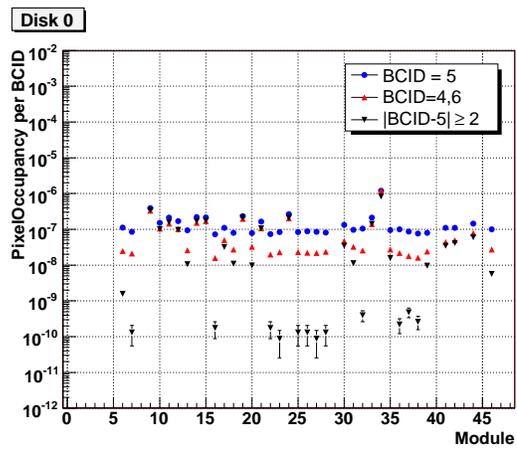


Figure 17: Module occupancies in layer 0 of endcap A during run 1125, for different L1-triggers.

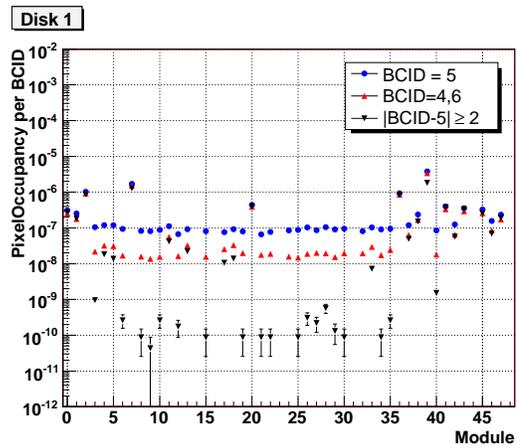


Figure 18: Module occupancies in layer 1 of endcap A during run 1125, for different L1-triggers.

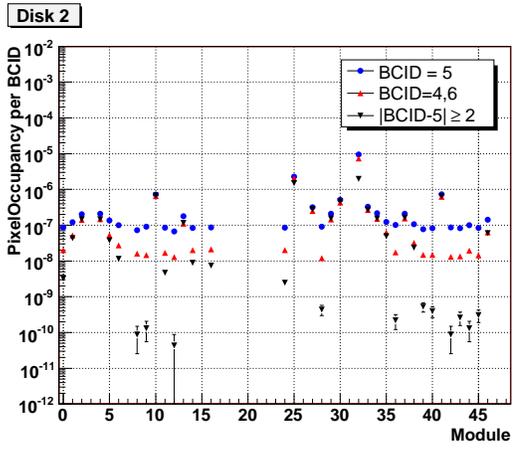


Figure 19: Module occupancies in layer 2 of endcap A during run 1125, for different L1-triggers.

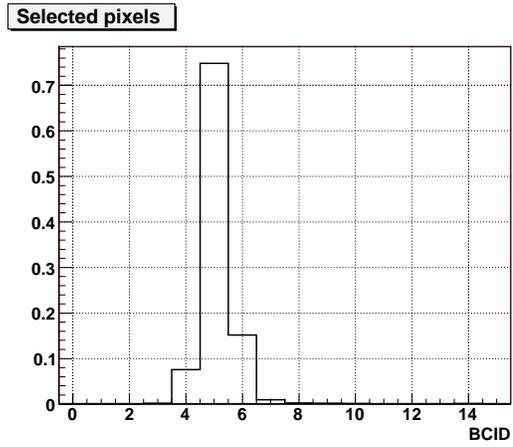


Figure 20: Bunch Crossing ID for all selected pixels not masked as hot.

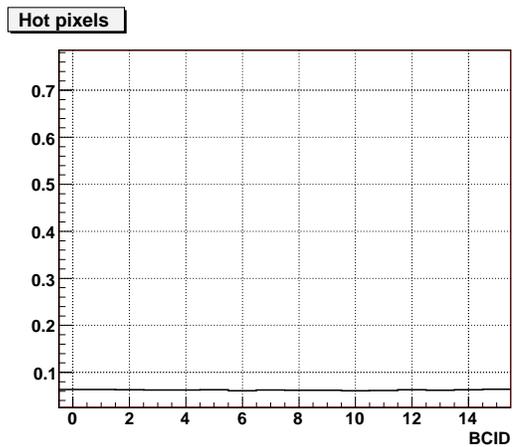


Figure 21: Bunch Crossing ID for hot pixels.

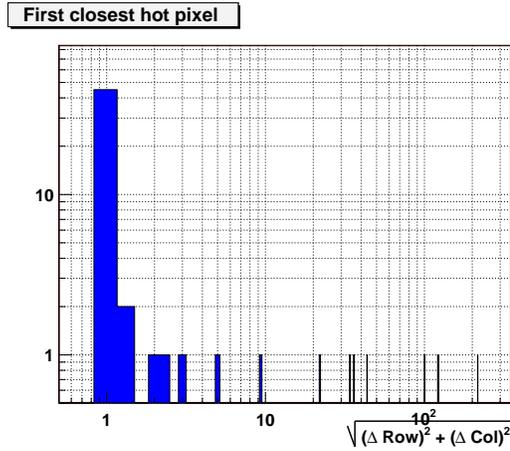


Figure 22: Distance between a hot pixel and the closest hot pixel within a module.

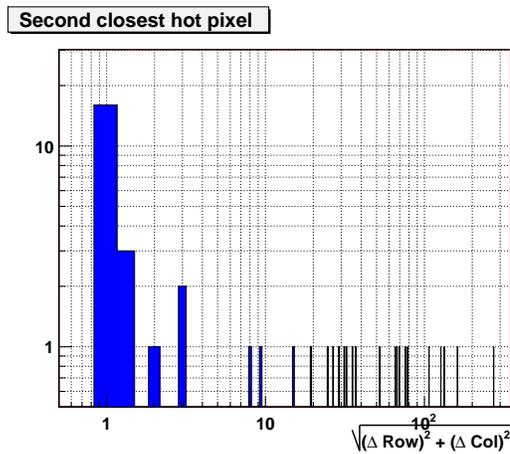


Figure 23: Distance between a hot pixel and the second closest hot pixel within a module.

The topology of the hot pixels within a module was investigated by looking at the distance (in units of row and column) between a hot pixel, and the closest and second closest hot pixel, as shown in Figures 22, and 23. The significant peaks at 1 indicate the presence of small *clusters* of nearby noisy pixels.

An example of such cluster of hot pixels in module 34 of layer 2 can be seen in Figure 24.

After removing all hot pixels in this run, the pixel module occupancy becomes very uniform within modules of a same layer, as it can be seen from Figures 25, 26, and 27. The pixel noise occupancy per module is of the order of 10^{-10} .

The distribution of number of pixels per event before and after hot pixel removal is shown in Figures 28, 29, and 30. On average, there were 24 pixels read out per event, of which 23 were hot pixels.

3.1.2 Run 1129

The noise occupancy was also investigated using a high statistics run 1129. There are large fluctuations in the module occupancy due to the hot pixels, as shown in Figure 31. After removing all hot pixels, the module occupancies in Figure 32 become much more consistency cross modules and reduced to the order of 10^{-10} for the hits outside the cosmic trigger.

Out of the 319 hot pixels on modules not characterized as noisy, 295 were found in a special pixel map denoting certain hardware issues as described in the following section. Most of these pixels have problems relating to the detection of particles and tuning the charge. The impact of these pixels was investigated further through cluster and track studies along with clusters completely composed of pixel hits outside the 4,5,6 BCID range. Only 62 tracks out of 15369 were found to contain one or more cluster

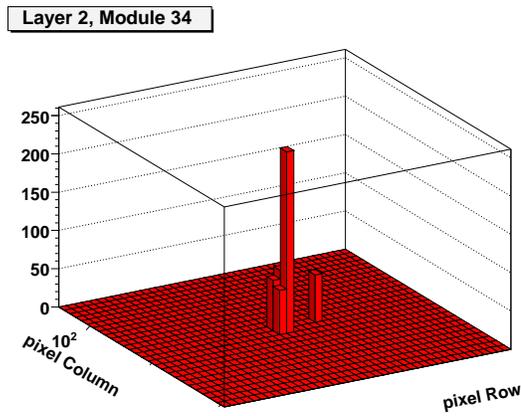


Figure 24: Example of a cluster of hot pixels in module 34 of endcap A layer 2.

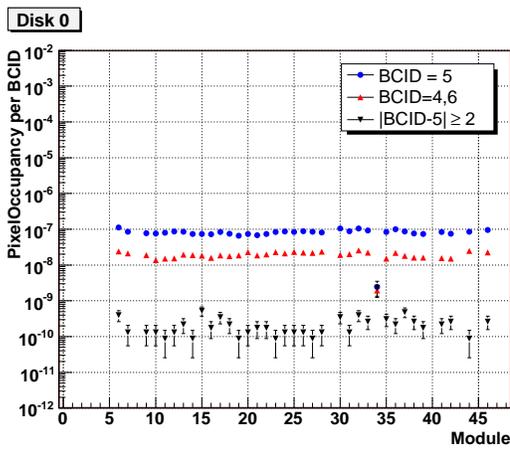


Figure 25: Module occupancies in layer 0 of endcap A during run 1125, for different L1-triggers and after hot pixel removal.

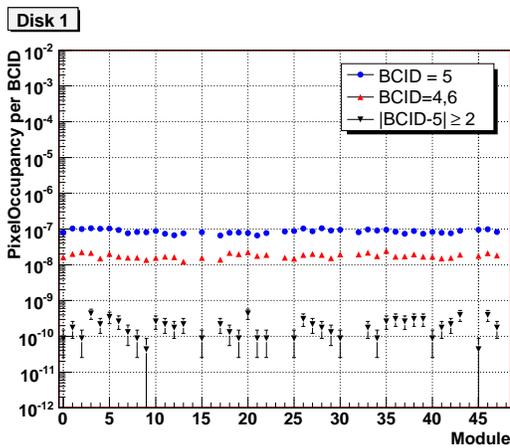


Figure 26: Module occupancies in layer 1 of endcap A during run 1125, for different L1-triggers and after hot pixel removal.

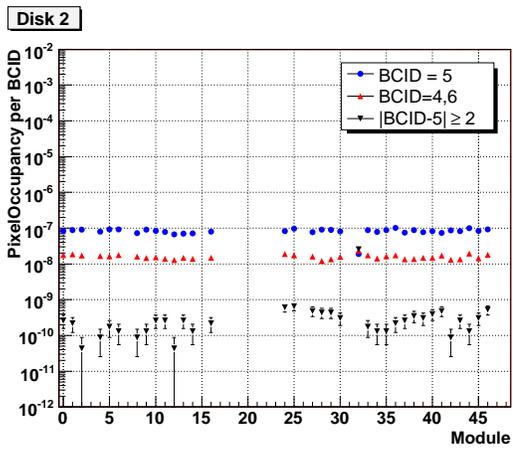


Figure 27: Module occupancies in layer 2 of endcap A during run 1125, for different L1-triggers and after hot pixel removal.

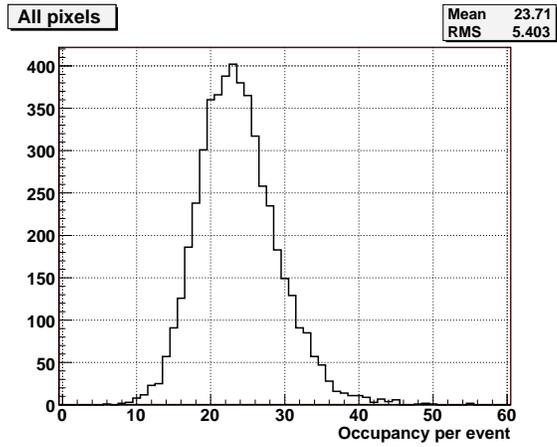


Figure 28: Distribution of total number of pixels per event.

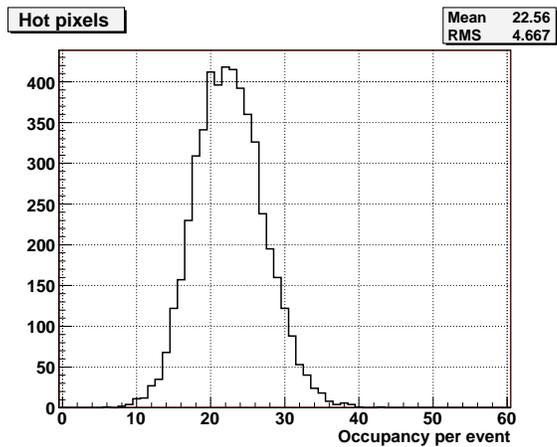


Figure 29: Distribution of number of hot pixels per event.

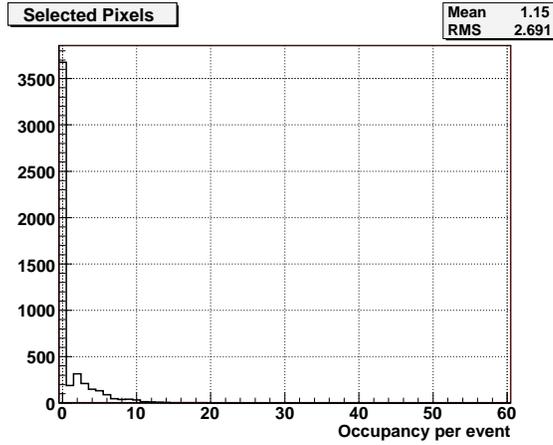


Figure 30: Distribution of number of selected pixels per event.

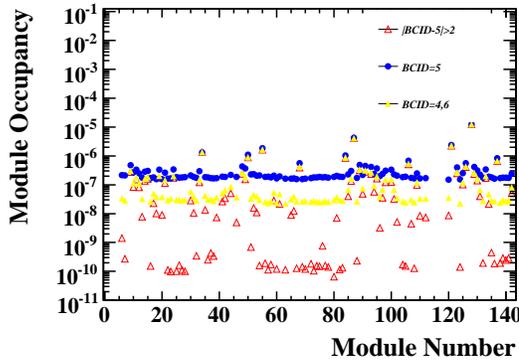


Figure 31: Pixel module occupancy per event readout in run 1129 before removing all hot pixels.

of this type, which is about 0.4 %. Once these tracks were removed, the overlap efficiency for all three layers rose about about 0.2%: a significant raise for such a small removal of tracks.

3.2 Run 1131 with Random Triggers

Run 1131 was performed with an external trigger signal at a frequency increasing from 10 Hz to 15 kHz and then decreasing back to 10 kHz. A single level 1 accept signal was used. The number of events in the run is 14 147 494.

The distribution of the occupancies for the individual pixels is shown in figure 33, with the occupancy being defined as the number of hits in a component, in this case one pixel, divided by the number of channels in the component and by the number of events in the run. Taking into account that approximately 20 modules were disabled for the run, most of them due to problems with the tuning of the optical readout chain, the occupancy for the endcap in this run is $2.5 \cdot 10^{-7}$.

Using the occupancy information it is possible to define pixels as being “noisy”. In the following, all pixels with an occupancy greater than 10^{-4} are defined to be noisy pixels. With this definition there are 871 noisy pixels in endcap A. By excluding the noisy pixels from the analysis the occupancy for the endcap can be reduced by more than 98 % to $4.8 \cdot 10^{-9}$.

There were two modules with unusually high noise levels in this run, modules [2,1,2,0,34,0,0]¹⁾ and [2,1,2,2,32,0,0]²⁾ which together contained 568 of the 871 noisy pixels. Maps of the hits for those

¹⁾offline ID [2,1,2,0,34,0,0], serial number 510853, geographical IDs D1A-S06-M3, D1A_B04_S1_M3

²⁾offline ID [2,1,2,2,32,0,0], serial number 512876, geographical IDs D3A-S06-M2, D3A_B04_S1_M2

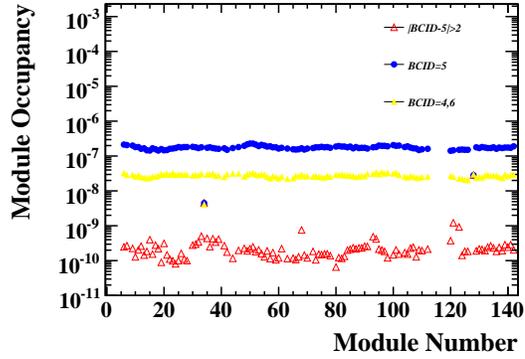


Figure 32: Pixel module occupancy per event readout in run 1129 after removal the hot pixels.

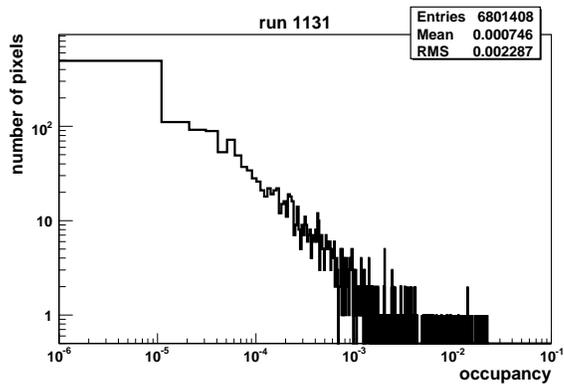


Figure 33: Distribution of the occupancies for individual pixel in endcap A during run 1131. The occupancy is defined as the number of hits in a pixel in the whole run divided by the number of events in the run.

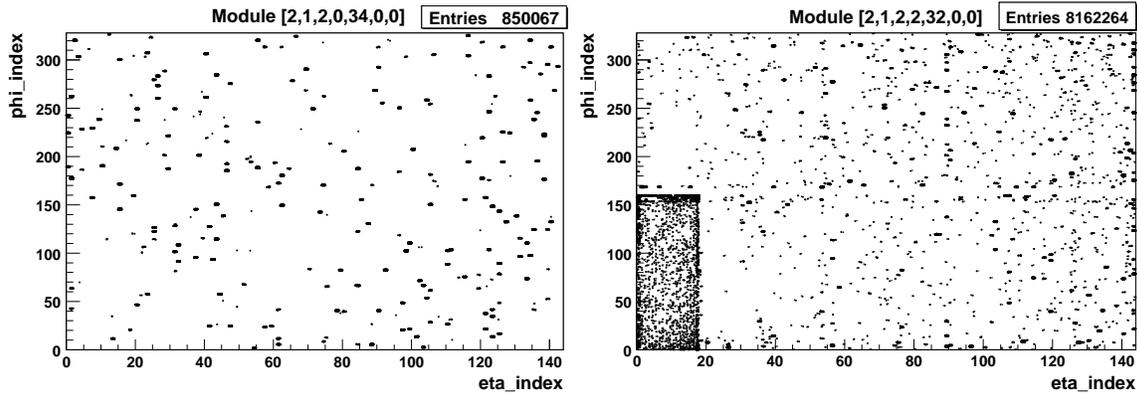


Figure 34: Hit maps for two modules with high noise levels in run 1131.

two modules are shown in figure 34. More than 40 % of the hits in the run are found on module [2,1,2,2,32,0,0].

The positions of noisy pixels were compared to the positions of pixels that had been marked as being special during the production (“FLEX”) tests of the individual modules. Table 1 gives an overview of the convention for the description of the pixel status used in the offline software and the numbers of pixels with each condition as determined in the production tests. The comparison shows that most of the noisy pixels had been marked as special in the production tests. Excluding the two modules with high noise levels, 283 of the remaining 303 noisy pixels are special, which corresponds to 93 %. 273 of those pixels have bits 0 and 13 set, meaning that the threshold is not tunable and that they do not yield useful data. Table 2 gives an overview of the numbers of status bits for the 303 noisy pixels.

A distribution of the time-over-threshold for all the hits in the endcap is shown in figure 36. One can see that the distribution has a maximum at 5 bunch crossings, or 125 ns, and a tail up to 30 bunch crossings, or 750 ns, and more.

3.3 Run 1138 with Random Triggers

The run 1138 was taken with an external trigger at a frequency up to 20 kHz and an effective trigger frequency of ~ 13 kHz. 15 231 074 events in this run were analyzed and the results were compared with the special pixel map taken from the production test. 19 modules were masked during the run. Module [2.1.2.0.34.0.0] was very noisy, and is excluded from the results of the analysis. A total of 23.7 MHits were registered during this run which corresponds to a total occupancy of $2.7 \cdot 10^{-7}$ per pixel, excluding the 19 masked modules and the noisy one. 51 modules, including the 19 masked ones, didn’t get any hit.

3.3.1 Comparison with the special pixel map

Figure 37 represents the occupancy of pixels both flagged and not flagged as special in the special pixel map. One can see the large occupancy for some of the flagged pixels in the special pixel map. Based on this plot, one can define a noisy pixel as a pixel that has an occupancy greater than 10^{-5} . With this definition there are 469 noisy pixels of which 89% were already flagged in the special pixel map. Table 3 shows the number of noisy pixels for different type of pixels according to the special pixel map definitions. If we take into account only modules that have at least one hit⁵⁾, we can compute the noisy pixel fraction for each special pixel type. Table 4 summarizes this fraction.

³⁾offline ID [2,1,2,1,39,0,0], serial number 512951, geographical IDs D2A-S07-M5, D2A_B04_S2_M5

⁴⁾offline ID [2,1,2,2,41,0,0], serial number 512429, geographical IDs D3A-S07-M4, D3A_B04_S2_M4

⁵⁾During data taking it was not clear that all modules were working without any problems. So one prefers doing this calculation only for modules that have at least one hit to be sure that they were sending data.

bit	meaning	description	numbers of pixels	
			endcap A	detector
0	use code	0 =useful data, 1 =not useful data: black out pixel in reconstruction	4938	129231
1	off for data	set to one if pixel is masked by DAQ	326	5908
2	off for calibration	set to one if pixel is masked during calibration runs	0	0
8	digitally dead	bit 0 of ModuleAnalysis mask: must trigger bit 0	71	57345
9	disconnected bump	bit 1 of ModuleAnalysis mask	1528	29511
10	merged bump	bit 3 of ModuleAnalysis mask	173	1437
11	dead with particles	bit 5 of ModuleAnalysis mask: must trigger bit 0	4173	61852
12	low efficiency with particles	bit 6 of ModuleAnalysis mask: must trigger bit 0	4255	62731
13	threshold not tunable (analog dead)	bit 11 of ModuleAnalysis mask: must trigger bit 0	2675	42567
14	ToT not tunable	bit 14 of ModuleAnalysis mask	2543	37257
15	noisy pixel	bit 16 of ModuleAnalysis mask: must trigger bit 0	17	634
16	unknown dead	any pixel with bit 0 set and bits 8-15 not set	0	2144
25	bottom neighbour special	bottom=smaller row number	3179	90192
26	top neighbour special		3179	90192
27	left neighbour special	left=smaller column number	1624	69386
28	right neighbour special		1624	69386
any	special	pixels that have at least one bit set	5627	141189

Table 1: List of the status bits used to describe the pixel status in the offline software. The pixel status is stored in an unsigned integer, bit i in the list corresponding to the position 2^i in the integer. The corresponding bits used in ModuleAnalysis, where applicable, are given in column 3. For each bit the number of pixels that were assigned the corresponding status in the production (“FLEX”) tests is given, both for endcap A and for the whole pixel detector. Bits that are not present in the list are not used at the moment.

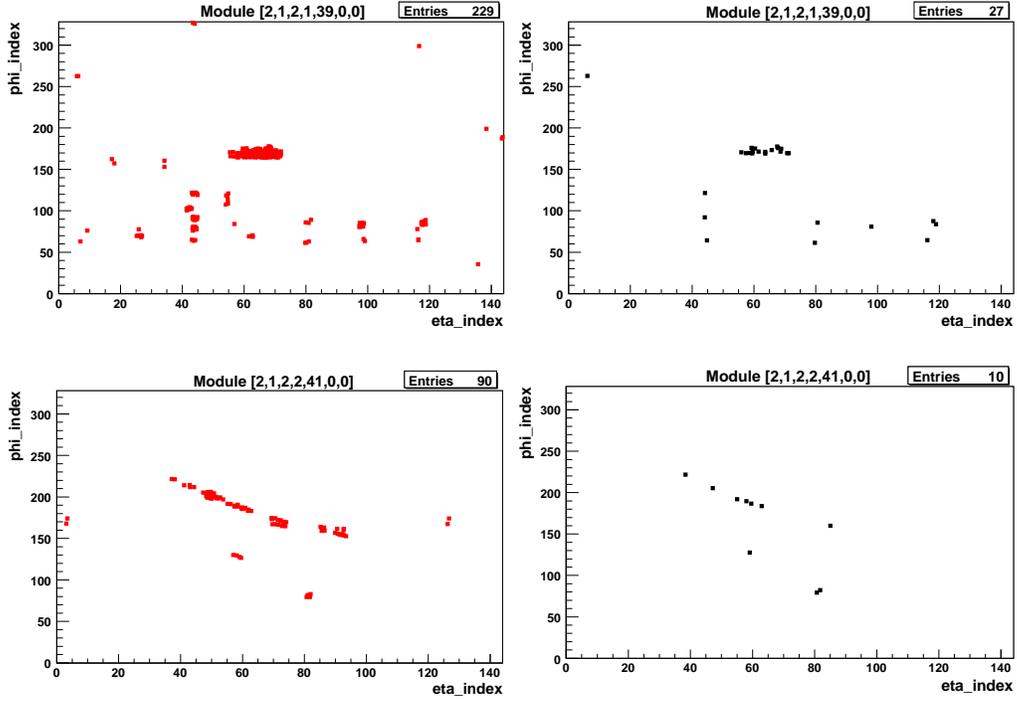


Figure 35: Scatter plots for two modules³⁾⁴⁾ of pixels marked as special in the production tests (left) and of noisy pixels in run 1131 (right).

status bit	number of noisy pixels	fraction of all noisy pixels
any bit, special	283	93 %
0, not giving useful data	273	90 %
1, off for data	65	21 %
8, digitally dead	0	0 %
9, disconnected bump	0	0 %
10, merged bump	3	1 %
11, dead with particles	205	68 %
12, low efficiency with particles	205	68 %
13, threshold not tunable	273	90 %
14, ToT not tunable	222	73 %
15, noisy	0	0 %

Table 2: Numbers of noisy pixels in endcap A in run 1131 with an occupancy cut of 10^{-4} , excluding two modules with unusually high levels of noise, for the different status bits as determined from the production test data.

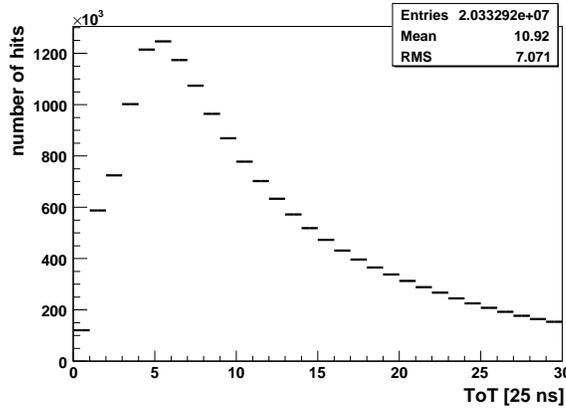


Figure 36: Distribution of the time-over-threshold for all hits in endcap A during run 1131.

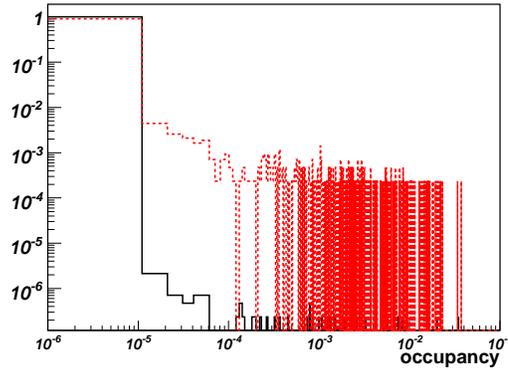


Figure 37: Number of hits per event per pixel. In black solid line pixels that are not flagged in the special pixel map and in red dashed line, for the pixels that are flagged in the special pixel map.

Special pixel map status bit	Number of noisy pixels	Fraction of all noisy pixels
Any bit, special	417	89%
1, off for data	95	20%
8, digitally dead	0	0%
9, disconnected bump	1	~0%
10, merged bump	7	1%
11, dead with particles	299	64%
12, low efficiency with particles	299	64%
13, threshold not tunable	393	84%
14, ToT not tunable	328	70%
15, noisy	1	~0%

Table 3: Number of noisy pixels in Run 1138 for the different status bits in the special pixel map and the fraction of these pixels w. r. t. all noisy pixels.

Special pixel map status bit	Noisy fraction of special pixels
Any bit, special	10%
1, off for data	44%
8, digitally dead	0%
9, disconnected bump	~0%
10, merged bump	5%
11, dead with particles	9%
12, low efficiency with particles	9%
13, threshold not tunable	19%
14, ToT not tunable	16%
15, noisy	10%

Table 4: Fraction of noisy pixels w. r. t. all special pixels for the different status bit in the special pixel map.

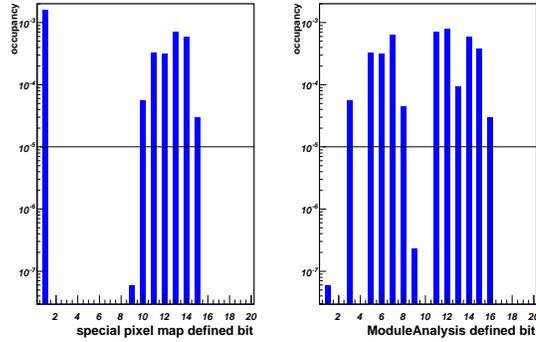


Figure 38: Occupancy per type of pixel for different status bit in the special pixel map and MA.

3.3.2 Comparison with ModuleAnalysis (MA) status bit

Not every type of special pixels in the production test data are copied into the special pixel map. Definition of these types is done using MA. Table 5 summarizes the different status bits in MA and their meaning. The correspondance between MA bits and the special pixel map bits can be found in Table 1. In order to see if one needs to add some other types of pixels to the special pixel map, one has to compare the noisy pixels with the remaining types of pixels that are not copied to this map. Figure 38 shows the pixel occupancy by status bits as defined by both the special pixel map and MA. One can deduce the type of pixels with high noise level. There are 5 types of pixels in MA with a high noise level which are not copied to the special pixel map. This corresponds to bits 7, 8, 12, 13, 15 according to MA definition. If we investigate further, one can see that most of noisy pixels that have these bits set also have another bit that is already in the special pixel map. Table 6 summarizes the fraction of noisy pixels in these bits and the fraction of pixels that have these bits set with no other bit in the special pixel map. For exemple, we can see that for bit 8 (noisy in source test) we have 87% of pixels with no other bit in the special map set. But this number is reduced to 1% if we just look for noisy pixels. That means that the cut used to set this bit was very loose and should not be used it in the special pixel map. All noisy pixels in the other bits are also well covered by the bits of the special pixel map.

3.3.3 Occupancy definition for noisy pixels

In this run, we define noisy pixels as pixels with an occupancy greater than 10^{-5} . This definition is rather arbitrary, and the separation between noisy and non-noisy pixels is not clear. If we define noisy pixels as the ones with an occupancy greater than 10^{-4} , instead we get 384 noisy pixels that correspond

MA status bit	meaning
0	hits (digital scan)
1	noise difference (HVon -HVoff)
2	noise (or with above)
3	Xtalk fraction (merged bump)
4	Xtalk noise (or with above)
5	source hits
6	source max ToT
7	source masked
8	noise (source scan)
9	difference from mean threshold
10	mean ToT (digital scan)
11	threshold S-fit χ^2
12	leakage current
13	crosstalk
14	ToT - calibration χ^2
15	overdrive
16	source rate ration

Table 5: Meaning of the different status bit as defined in MA.

Module Analysis status bit	Fraction of noisy	All pixels not covered in the special pixel map	Noisy pixels not covered in the special pixel map
7, source masked	18%	$\sim 0\%$	0%
8, noisy (source scan)	3%	87%	1%
12, leakage current	20%	12%	$\sim 0\%$
13, crosstalk	10%	9%	1%
15, overdrive	10%	50%	$\sim 0\%$

Table 6: Status bit as defined in MA and not copied to the special pixel map that contains noisy pixels, with the fraction of pixels that do not have any other bit set in the special map.

Special pixel map status bit	occupancy $> 10^{-5}$	occupancy $> 10^{-4}$	$10^{-5} < \text{occupancy} < 10^{-4}$
1, off for data	23%	22%	28%
8, digitally dead	0%	0%	0%
9, disconnected bump	$\sim 0\%$	$\sim 0\%$	1.5%
10, merged bump	1.7%	1.7%	1.5%
11, dead with particles	72%	73%	66%
12, low efficiency with particles	72%	73%	66%
13, threshold not tunable	94%	96%	86%
14, ToT not tunable	79%	78%	81%
15, noisy	$\sim 0\%$	$\sim 0\%$	0%

Table 7: Fraction of pixels in different bin of occupancy for the different status bits in the special pixel map w. r. t. all pixels in the same occupancy bin.

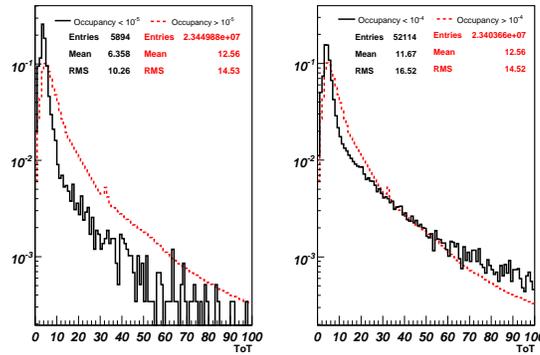


Figure 39: ToT for noisy pixels, in black solid line, and non noisy pixels, in red dashed line. On the right, noisy pixels are defined as the ones that have an occupancy higher than 10^{-5} and on the left, noisy pixels are defined as the ones that have an occupancy higher than 10^{-4} . All distributions are normalized to 1.

to 82% of noisy pixels from the first definition. 92% of these pixels are already flagged in the dead pixel map. This number is in complete agreement with the results from run 1131. Table 7 lists the fraction of noisy pixels for different types of special pixels for different occupancy ranges. This table shows that there is no real changes in type of pixels with the occupancy range definition. Figure 39 shows the ToT distribution for noisy and non-noisy pixels for the two occupancy definition cuts. It is clear that the noisy pixels have a higher ToT than normal ones. The discrepancy is very clear for the 10^{-5} definition, while we can see large tails for the non-noisy pixels that seem to be closer to noisy pixels for the 10^{-4} definition.

To investigate further the noisy pixels definition, one can divide pixels in different ranges of occupancy. Figure 40 shows the correlation with the special pixel map in different range of occupancy. We can see that almost all pixels with very high occupancy are already flagged in the special pixel map. The fraction of special pixels decreases with the occupancy and there is a large gap at an occupancy of 10^{-6} . One can think that the cuts in the production data test that are used to build the special pixel map correspond to a tight cut on the occupancy (10^{-6}) rather than a loose one. Figure 41 shows the distribution of ToT for pixels in different occupancy bins. As expected, one can see that pixels with higher occupancy have a higher ToT. This is not the case for bin $10^{-5} - 10^{-4}$ where the ToT is higher than expected. If we compare this distribution with the one from bin $10^{-4} - 10^{-3}$ as shown in figure 42, we can see that the main difference is coming from the large tails in the $10^{-5} - 10^{-4}$ bin. A fit excluding the tails shows that the mean values in the peak region are comparable. These tails can be explained by one or more pixels that have a higher ToT distribution. This can also explain the tails in the ToT distribution for the pixels with an occupancy $< 10^{-4}$ in figure 39.

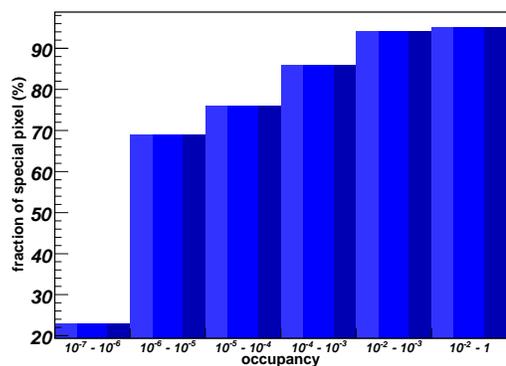


Figure 40: Fraction of pixels that are already flagged in the special pixel map in different occupancy ranges.

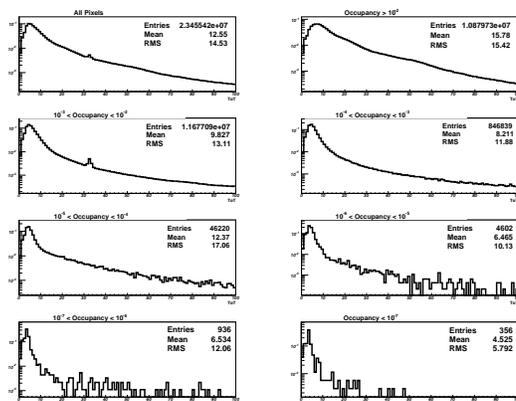


Figure 41: ToT distribution for pixels in different occupancy bins. All distributions are normalized to 1.

The peak in the ToT distribution in bin $10^{-2} - 10^{-3}$ is due to one pixel with a strange behavior. The ToT distribution of that pixel is shown in figure 43 for 100000 events.

3.4 Run 1153 and 1144 with Special Setting

Run 1153 was performed with an effective trigger rate of about 12 kHz. A single level 1 acceptance signal was used. The number of events in the run is 16 776 587. In this run, two modules, $[2,1,2,0,14,0,0]$ ⁶⁾ and $[2,1,2,2,27,0,0]$ ⁷⁾, had the threshold lowered to TDAC-25, corresponding to a change with respect to the reference value of about 1581 electrons. Module $[2,1,2,0,14,0,0]$ shows an above average number of noise hits, but still has no hits in most of the pixels. Module $[2,1,2,2,27,0,0]$, for which a map of the hits is shown in figure 44, shows a number of about 13 000 hits in almost every pixel. The pixels in the first four rows of every chip, corresponding to a ϕ -index from 0 to 3 and from 324 to 327, show lower than average numbers of hits, with about 6000 hits on average for the first two rows of each chip. This can be seen clearly in figure 45. In a subsequent analysis of run 1144, in which the threshold had been lowered to TDAC-20 for all modules, three modules⁸⁾⁹⁾¹⁰⁾ show a similar behaviour in that they have a high number of evenly distributed hits with about half as many hits in the first two to four rows. One

⁶⁾offline ID $[2,1,2,0,14,0,0]$, serial number 512862, geographical IDs D1A-S03-M2, D1A_B02_S2_M2

⁷⁾offline ID $[2,1,2,2,27,0,0]$, serial number 511818, geographical IDs D3A-S05-M5, D3A_B03_S2_M5

⁸⁾offline ID $[2,1,2,1,34,0,0]$, serial number 511476, geographical IDs D2A-S06-M3, D2A_B04_S1_M3

⁹⁾offline ID $[2,1,2,1,42,0,0]$, serial number 512746, geographical IDs D2A-S08-M1, D2A_B01_S1_M1

¹⁰⁾offline ID $[2,1,2,2,12,0,0]$, serial number 510392, geographical IDs D3A-S03-M1, D3A_B02_S2_M1

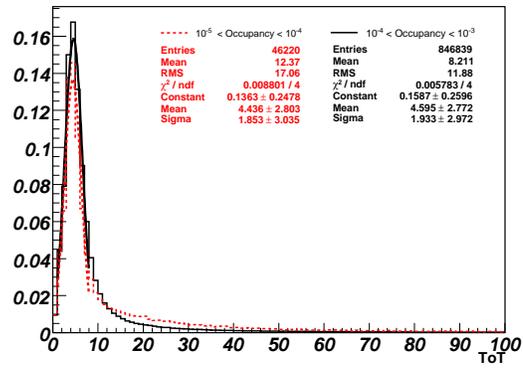


Figure 42: ToT distribution for Pixels with an occupancy between $10^{-4} - 10^{-3}$ in black solid line and $10^{-4} - 10^{-5}$ in red dashed line. All distributions are normalized to 1.

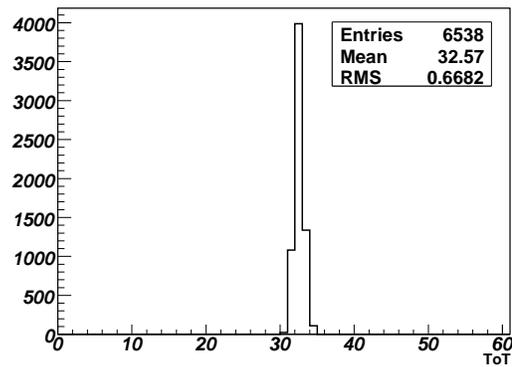


Figure 43: ToT distribution for one of the pixels with a strange behavior that induces a peak in the ToT distribution for all pixels.

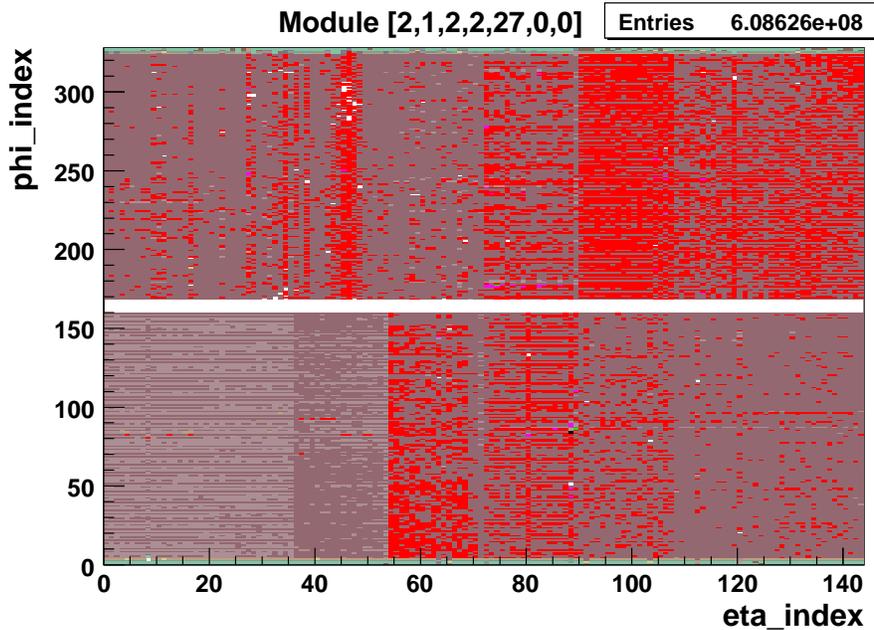


Figure 44: Hit map for module [2,1,2,2,27,0,0] which had a low threshold at TDAC-25 in run 1153.

module¹¹⁾ has a similarly high number of hits but only four front-end chips (0, 6, 8, 15) show a lower number of hits in the first four rows. Hit maps for these four modules with anomalously high noise levels are shown in figures 46 and 47. The other modules in run 1144 exhibit noise patterns that are comparable to those typically found in the runs at default settings except for a higher number of hits in the ganged pixels. Run 1144 contains 16 774 967 events in total taken at a trigger rate increasing from 5 kHz to 13 kHz. A single level 1 accept signal was used.

Several pixels on module [2,1,2,2,27,0,0] show no noise hits in run 1153. The positions of these 82 dead pixels were compared to the positions of the 372 special pixels from the production tests. 61 (74 %) of the 82 dead pixels are special. There are 61 special pixels with the status bits (0,11,12,13,14) set (no useful data, dead with particles, low efficiency with particles, threshold not tunable, ToT not tunable), 57 (92 %) of which are dead. In addition there are 5 special pixels with the status bits (0,1,13,14) set (no useful data, off for data, threshold not tunable, ToT not tunable), 4 of which are dead. These two classes of pixels contain all the pixels with status bit 14 set. None of the 306 special pixels with other combinations of status bits are dead. As this comparison shows, for this module there is an almost one-to-one correspondence between the pixels that show no noise hits and the pixels with status bit 14 set.

Four modules in run 1153 had special TDAC pattern masks applied to them. Modules [2,1,2,1,33,0,0]¹²⁾ and [2,1,2,1,14,0,0]¹³⁾ had the mask shown on the left in figure 48 applied to them. The greyscale corresponds to different TDAC settings, black corresponding to TDAC-25 and white corresponding to TDAC+25. No effect on the number of noise hits was observed for these two modules. Almost all pixels on these two modules show no noise hits.

The mask shown on the right in figure 48 was applied to modules [2,1,2,2,44,0,0]¹⁴⁾ and [2,1,2,0,33,0,0]¹⁵⁾. Maps of the hits for these two modules are shown in figure 49. One can see an increased number of noise hits in the areas with the lowest thresholds, especially for the ganged pixels. Since the TDAC pattern mask was applied using column and row numbers instead of eta_indices and phi_indices, the phi_index direction has to be flipped for module [2,1,2,0,33,0,0] for comparison with the greyscale image.

¹¹⁾offline ID [2,1,2,0,10,0,0], serial number 510418, geographical IDs D1A-S02-M3, D1A_B02_S1_M3

¹²⁾offline ID [2,1,2,1,33,0,0], serial number 512351, geographical IDs D2A-S06-M5, D2A_B04_S1_M5

¹³⁾offline ID [2,1,2,1,14,0,0], serial number 512831, geographical IDs D2A-S03-M2, D2A_B02_S2_M2

¹⁴⁾offline ID [2,1,2,2,44,0,0], serial number 510559, geographical IDs D3A-S08-M2, D3A_B01_S1_M2

¹⁵⁾offline ID [2,1,2,0,33,0,0], serial number 510963, geographical IDs D1A-S06-M5, D1A_B04_S1_M5

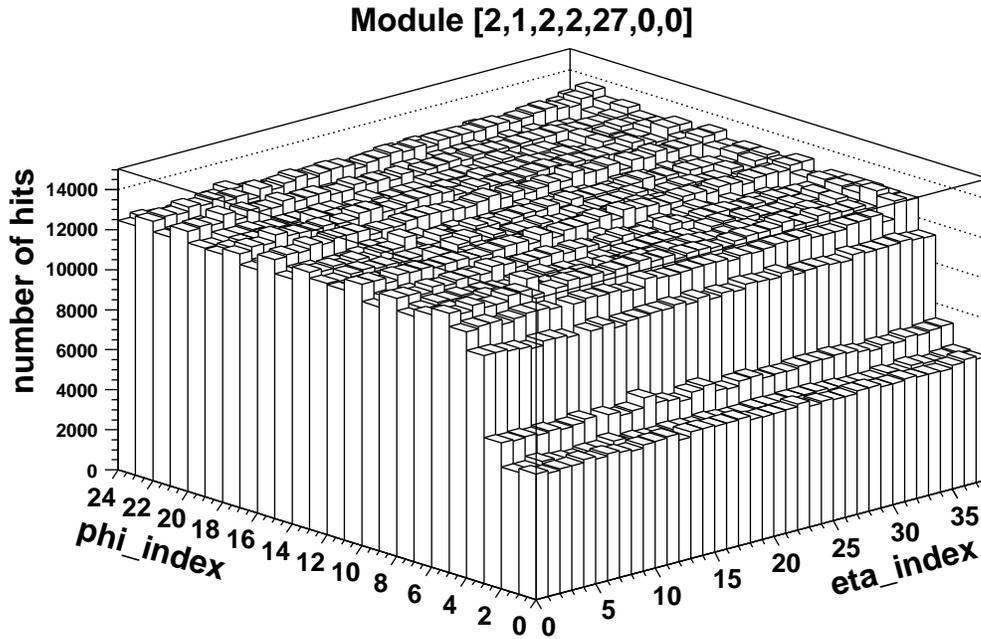


Figure 45: Hit map of a section of module [2,1,2,2,27,0,0] which had a low threshold at TDAC-25 in run 1153.

3.5 Run 2043, 2044, and 2045 with Different Trigger Frequencies

Runs 2043, 2044 and 2045 were taken with external trigger frequencies of 20KHz, 10KHz, 40KHz respectively. Every run contains about 17 M events and the number of modules with a significant amount of hits (>15) is 7. For these reasons a direct comparison of results from these three runs is possible.

3.5.1 Noise vs Readout Time

Figures 50 and 51 show the number of hit in a module as a function of time, obtained by summing counts within a fixed time interval. The number of intervals is chosen dependently by frequency: it is used a number of 25 steps for 20KHz and 40KHz runs and a number of 50 steps for 10KHz run; this allows for the investigation of noise features in the same range of frequencies. The contribution of hot and non hot pixels¹⁶⁾ is investigated by plotting the same quantities before and after hot pixel removal. This permits to underline possible correlations and analogies between noise patterns characterizing hot and non-hot pixels.

Three different noise level behaviours can be observed and these features are clearly reproducible from run to run:

- unresolved trend for both kind of pixels as shown in Figure 50. This is the characteristic of the major part of the module;
- Oscillating pattern with approximately 5 minutes periods¹⁷⁾ for hot pixel and ambiguous trend for the rest (this ambiguous trend is also due to lack of statistics) as in Figure 51. The fitting function used is $[p_0] + [p_1] \sin\left(2\pi \frac{t + [p_3]}{[p_2]}\right)$ where the time is expressed in seconds and the parameter $[p_3]$ represents the oscillation period in seconds¹⁸⁾.

¹⁶⁾The definitions of hot pixel is as in 3.1

¹⁷⁾It is interesting to notice that this oscillating pattern is characterized by a period of approx. (250; 300) s and this fact is independent by the run and the module.

¹⁸⁾It is important to notice that the fitting function is chosen only to extract the oscillation period and it is not intended to be used to entirely model the noise patterns. In particular, the function describes only the dominant modulation effect, completely ignoring all the other possible effects which affect the measurement.

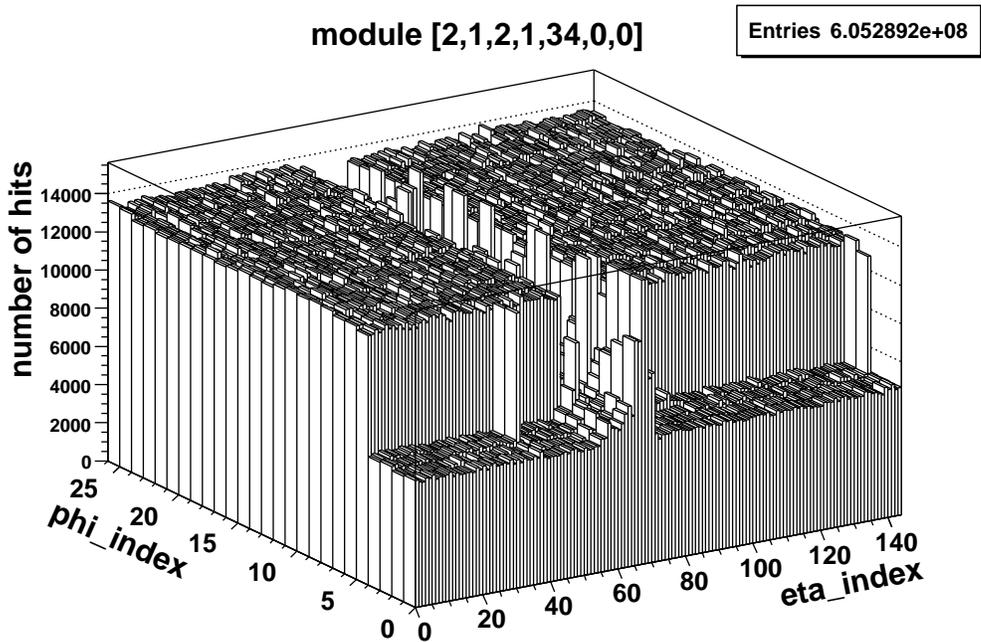
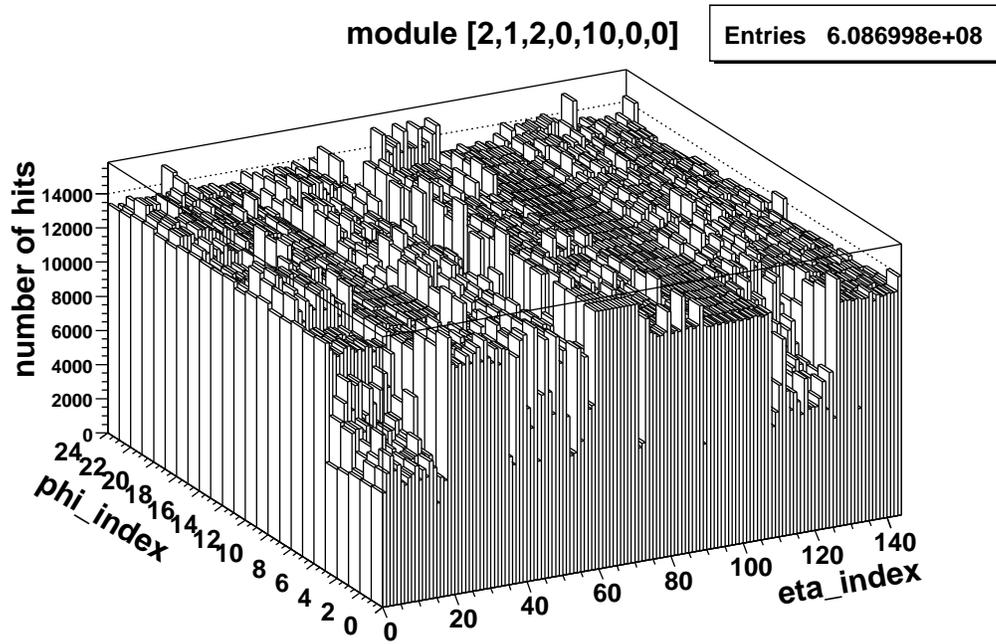


Figure 46: Hit maps of sections of two modules with unusually high noise levels in run 1144, in which all modules had a low threshold at TDAC-20.

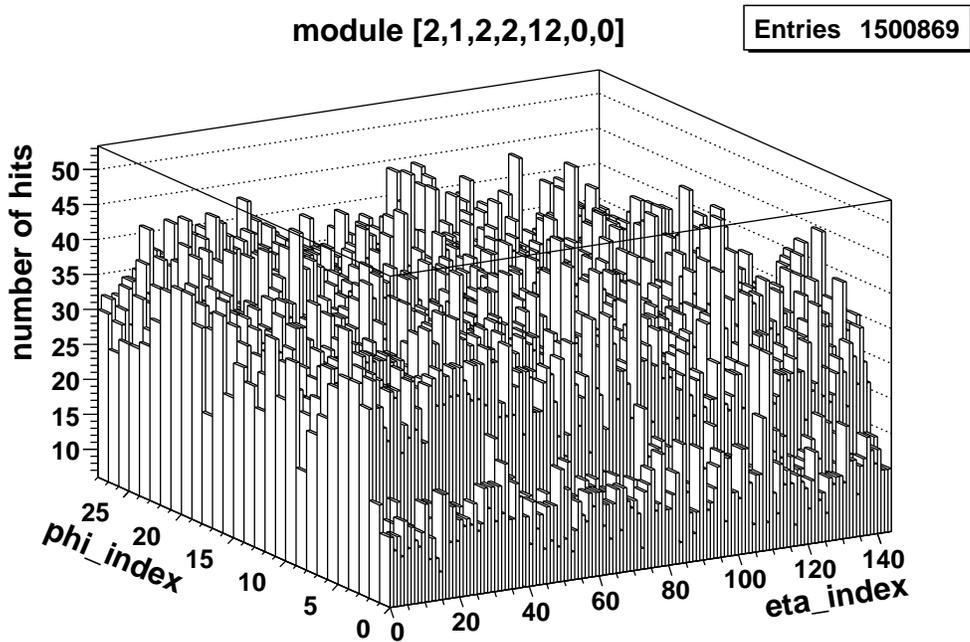
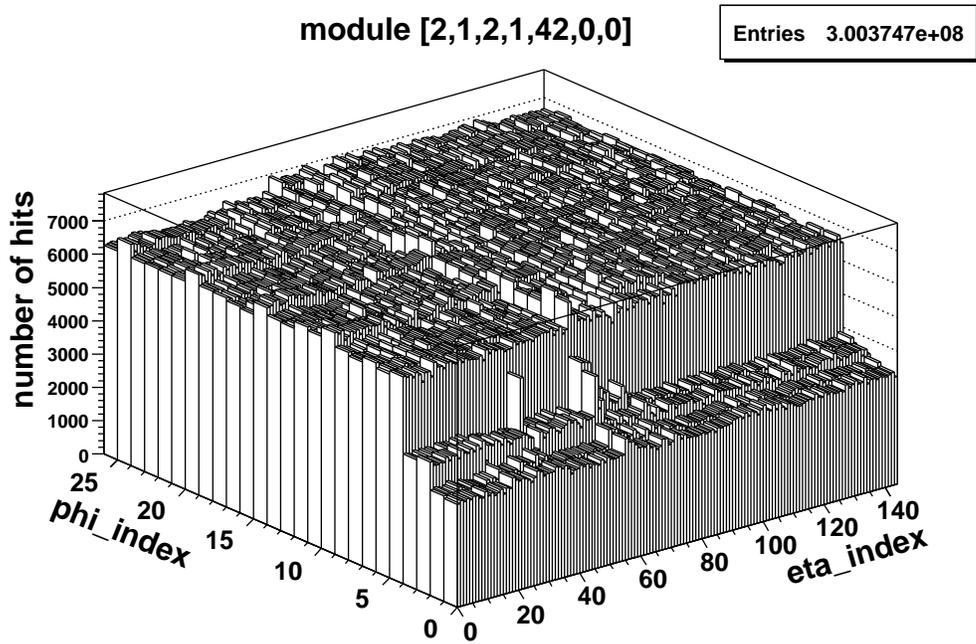


Figure 47: Hit maps of sections of two modules with unusually high noise levels in run 1144, in which all modules had a low threshold at TDAC-20.

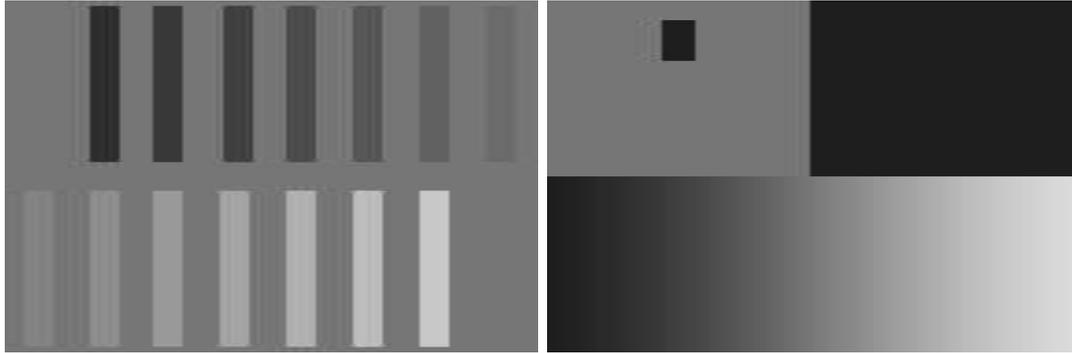


Figure 48: Greyscale images representing TDAC pattern masks that were applied to several modules in run 1153. Black corresponds to TDAC-25, white corresponds to TDAC+25. The mask on the left was applied to modules [2,1,2,1,33,0,0] and [2,1,2,1,14,0,0], the mask on the right was applied to modules [2,1,2,2,44,0,0] and [2,1,2,0,33,0,0].

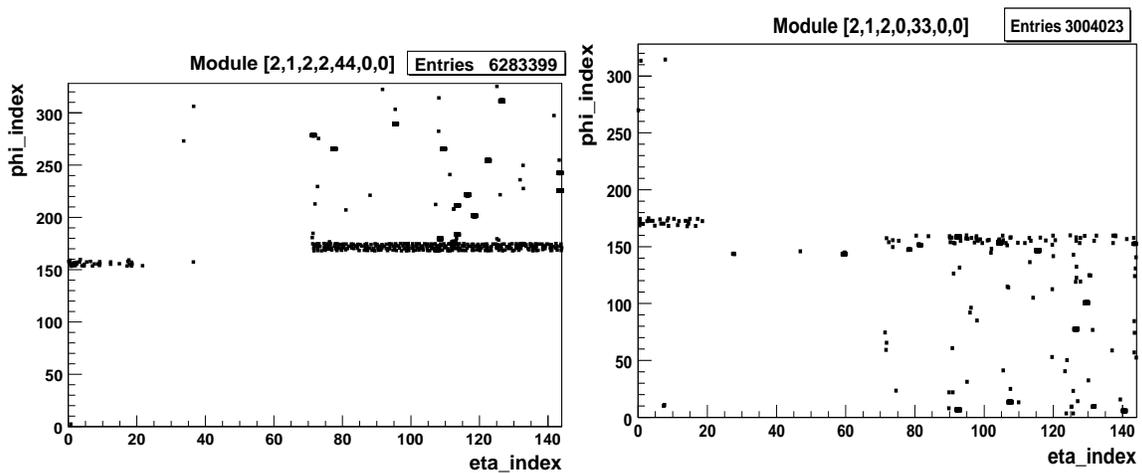


Figure 49: Hit maps for modules [2,1,2,2,44,0,0] and [2,1,2,0,33,0,0] in run 1153 to which the TDAC pattern mask corresponding to the greyscale image shown on the right in figure 48 was applied. Note that the TDAC pattern mask was applied using column and row numbers instead of eta_indices and phi_indices and hence the phi_index direction has to be flipped for module [2,1,2,0,33,0,0].

Apart from this, no net dependence of noise by time is visible and thus the occupancy level of pixels is proved to be stable.

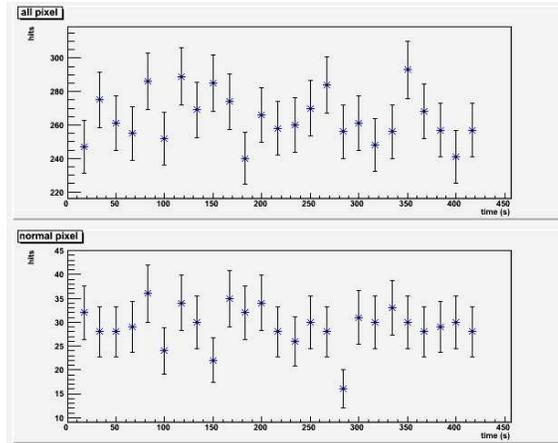


Figure 50: Number of pixel hit as a function of time for all (top) and normal (bottom) pixels on module 18 of disk 0 in Run 2045.

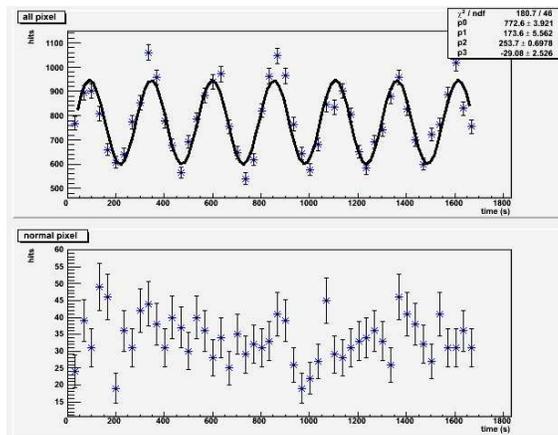


Figure 51: Number of pixel hit as a function of time for all (top) and normal (bottom) pixels on module 21 of disk 0 in Run 2044. The fitting function shows the clear evidence of a modulation of the hot pixel noise.

3.5.2 Noise vs Readout Frequency

Run executed at constant datataking frequency are also aimed to study how the noise level is connected to the rate of data out the detector. Since the number of events in all the runs is approximately the same, it is sufficient to look at the hit in every module in order to compare the occupancy.

The results are summarized in table 8 in which only the meaningful modules are taken into account (the noisy module described in 3.4.1 is not considered). As in the previous analysis, data are considered before and after hot pixel removal. An interesting fact is that, after the removal procedure, there is still a broad range of hits per module. This may suggest that the cut in occupancy which define hot pixels is in some sense arbitrary and can not be so sharp. The number of hits on the same module in different runs is almost the same and the differences are within a few percent. This result suggests that the noise level has a weak correlation with frequency in the range analyzed.

<i>Module</i>	<i>Run 2043 (all hits)</i>	<i>Run 2043 (normal hits)</i>
0_18	6401	782
0_19	141696	–
0_20	6120	1375
0_21	40411	1646
1_18	9045	2
1_20	298123	6233
	<i>Run 2044 (all hits)</i>	<i>Run 2044 (normal hits)</i>
0_18	6271	770
0_19	141374	5
0_20	6194	1355
0_21	39220	1683
1_18	9062	–
1_20	297561	5994
	<i>Run 2045 (all hits)</i>	<i>Run 2045 (normal hits)</i>
0_18	6393	731
0_19	141642	–
0_20	6081	1417
0_21	42576	1597
1_18	9102	–
1_20	302266	6330

Table 8: Number of pixel hit per module for all and non hot pixel in each run. The module number is derived from the following convention: nLayer_nPhi where nLayer represents the disk and nPhi the module’s phi angle.

3.6 Summary

The noise in endcap A of the pixel detector was studied for several exemplary runs with different detector settings. The analysis of noise in the cosmics data run 1125 and 1129 shows that the noise signal, as expected, is uncorrelated with the timing relative to the trigger signal and that the noise is dominated by *fixed pattern* noise, i.e. by hits in a relatively small number of noisy pixels. After removal of the noisy pixels the noise occupancy for the endcap is of the order 10^{-10} . This result for the occupancy is confirmed in an analysis of the noise run 1131 in which a random trigger signal was used. A comparison shows that most of the noisy pixels in this run were found to be *special* during the production tests of the individual modules. An analysis of modules with a low threshold setting in runs 1153 and 1144 shows a moderate increase in the number of noise hits, especially in the ganged pixels. Several modules in these runs show an atypically high level of noise with the unexpected feature that the noise level in the first two to four rows of most front-end chips is only about half as high as in the other rows. Finally, we have checked the noise level in terms of readout time and external trigger frequencies and found that the noise level seems stable over time except an unexpected modulation of the hot pixel noise.

4 Timing study of cosmic trigger

An important feature we managed to check during the cosmic data taking was the synchronization of the modules. The study of LVL1 distribution has been used to measure relative synchronization of modules with a resolution better than 1 ns. In fact different values of delay correspond to different shapes of the bunch crossing distribution. Using simulation we built a function that fit the BCID plot giving an estimation of the delay. First of all we generated the BCID plot for delay from 0 to 400 ns and we calculated the fraction F_N of hits in a given bin N of the BCID plot as a function of the delay. For any bin the fraction can be obtained considering that

$$F_N(t) = F_{N-1}(t + 25) = F_{N-2}(t + 50) = \dots \quad (2)$$

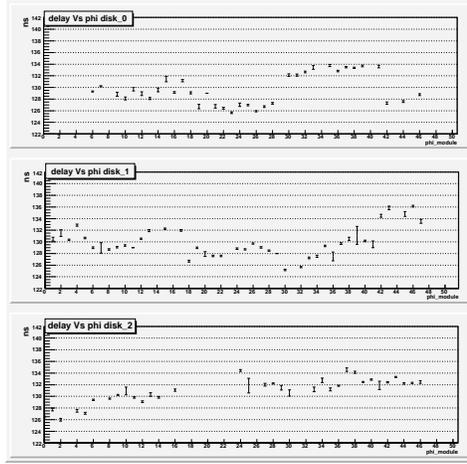


Figure 52: The fitted delay time for each module on three endcap disks obtained from the run 1129.

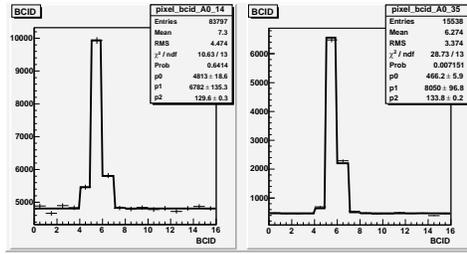


Figure 53: Comparison of fitted BCID distributions for module 14 and 35 of disk 0 that has very different time delays.

and

$$F_N(t) = F_{N+1}(t - 25) = F_{N+2}(t - 50) = \dots \quad (3)$$

The different mean incidence angle affects the charge collection and therefore the timewalk and the hit arrival time so it has been necessary to make the function $F_N(t)$ for each module. Then we provided a function that fits BCID plot:

$$\psi(BCID) = n + h \cdot F_N(d + (N - BCID) \cdot 25) \quad (4)$$

where $BCID$ is the variable, N is a reference $BCID$ used to build F and the parameters are:

- n is the noise probability i.e. number of hits per bunch crossing
- h is the number of hit from a track in the plot
- d is the delay value

Tests of the algorithm fit on monte Carlo data (with no asynchronous modules) give satisfactory results, in fact we obtained a distribution for the parameter d (delay), with sigma less than 1 ns and pull functions RMS of 1.15. The results of the fit for the run 1129 are shown in figure 52. The estimated delay is plotted for each phi module on three endcap disks. The plot shows that some modules are delayed in comparison to the others; the delay we found is not a problem of the fit, as shown in figure 53 where the fitted BCID distributions for two modules with different delays. Most of the modules are well fitted by the function.

5 Cosmic Tracking Studies

After masking off the special pixels from the database, pixel clusters were reconstructed using a simple clustering algorithm where all adjacent hits sharing at least one of sides were clustered together. The

Table 9: The rate of cosmic tracks found in data and various Monte Carlo samples.

Data Sample	Tracking Rate (%)	Overlap Fraction (%)
Data	3.33 ± 0.02	23.7 ± 0.3
Ideal MC	≈ 6	≈ 28
Realistic MC	4.13 ± 0.02	23.2 ± 0.2

cluster position in the local x and y coordinators was computed using a charge weighted centroid with an uncertainty assigned as the pitch divided by $\sqrt{12}$. For cosmic tracking, we use the modified existing CTBSiTracking package [17]. First the algorithm loops over any pair of the pixel clusters from the inner and outer disk and linearly extrapolates to the middle disk. For any pixel cluster within a search window of 1.5 mm, a correspondent cosmic track candidate is reconstructed including a possible hit in the overlap region from the neighboring module. If there are multiple track candidates, the best one is selected based on the number of pixel hits and the fitted chisq ($\chi^2 < 25/\text{ndof}$) in the x-z and y-z plane. There is no ESD or AOD written out, but CBNTs are saved that contain all the information about pixels, clustering, and tracking for data analysis. Initially, there were couple problems spotted in our geometry descripton. But they were fixed after checking some basic quantities, such as tracking efficiency and residual distributions.

In order to better understand the detector performance, we have generated cosmic Monte Carlo with a realistic detector simulation that uses the calibration from the production database and a list of modules that were disabled during the data taking.

Table 9 summaries the tracking rate per event and the fraction of tracks with overlap hits (≥ 4 pixel hits) in the data and the Monte Carlo with different conditions. The overall tracking rate is 3.33% in data, which is lower than realistic MC 4.13%. However, the fraction of overlap hits is consistent between data and Monte Carlo, which indicates data and Monte Carlo have comparable pixel hit efficiencies.

Figure 54 shows the number of pixel clusters, total chisq, phi and theta of the reconstructed cosmic tracks. We also checked the qualities of clusters associated with the reconstructed track, which are shown in Figure 55 on the time over threshold (TOT or charge), the cluster width, beam cross trigger identifier (BCID), and the module occupancy as function of $\text{phi} + 48 * \text{Layer}$ where phi is the module number between 0 to 47 and Layer is the disk number between 0 to 2. Further more, we have made some comparison between the cosmic data and Monte Carlo, which are shown in Figure 56- 59 for track probability, the cluster width and its BCID, the number of pixel rows in the cluster, and the TOT for single, double and triple hits clusters, respectively. Overall, the agreements between data and Monte Carlo are excellent.

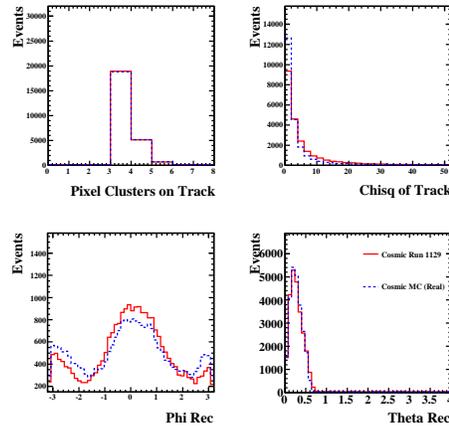


Figure 54: The comparisons of cosmic tracking in terms of the number of pixel hits (top left), the chisq of fit (top right), the phi and theta of the reconstructed cosmic tracks (bottom left and right). The solid curve is data and the dashed one is Monte Carlo.

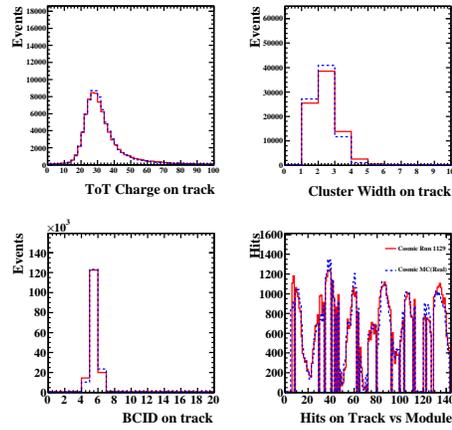


Figure 55: The comparisons of cosmic tracking in terms of the TOT (top left), the cluster width(top right), the beam crossing (BCID) and the module occupancy (bottom left and right). The solid curve is data and the dashed one is Monte Carlo.

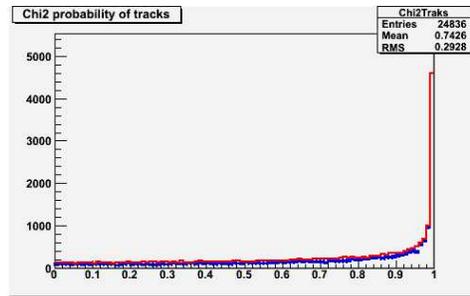


Figure 56: Comparison of track probability distribution between data in blue point and Monte Carlo in red histogram.

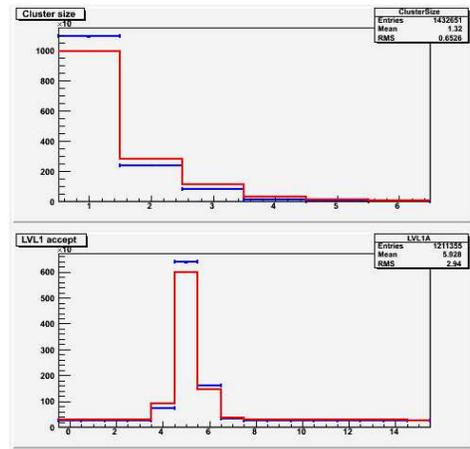


Figure 57: Comparison of the cluster width (top) and its BCID (bottom) between data in blue point and Monte Carlo in red histogram.

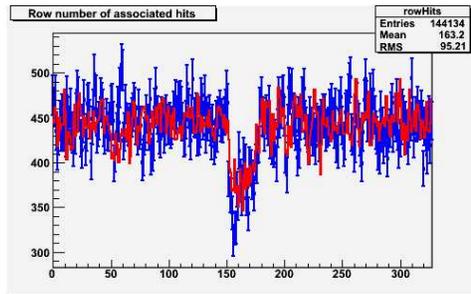


Figure 58: Comparison of the number of pixel rows in the cluster between data in blue point and Monte Carlo in red histogram.

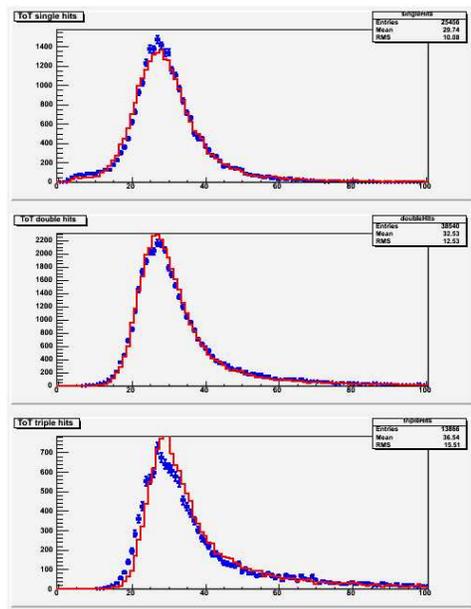


Figure 59: Comparison of the TOT distribution for single, double and triple clusters between data in blue point and Monte Carlo in red histogram.

6 Pixel Clustering and Efficiency Studies

In this section, we present a study of pixel clustering and hit efficiency using the tracks found in the pixel endcapA cosmic data. The quantities we are interested in are the pixel cluster width and its charge, which would increase as the incident angle becomes larger. Figure 62 shows the pixel cluster width in the local x (widthx) direction for all the clusters attached to the tracks as function of the incident angle ($\tan(\theta) \cdot \cos(\phi - \phi_m)$) where θ and ϕ is the reconstructed track parameters and ϕ_m is the ϕ of module m in the global frame. The top left and right plots show the data and the Monte Carlo. The bottom left plot shows the projection of the cluster widthx and the bottom right plot shows the profiling of the cluster widthx vs the incident angle. Figure 63 shows a similar plot, the cluster width in local y (widthy) as function of the incident angle. Since the pixel pitch size in local y is much larger than in local x, the widthy of pixel clusters is expected to be much smaller. The data is in good agreement with Monte Carlo expectation. The cluster charge (TOT) is also plotted as function of the path length through the silicon, which is shown in Figure 64. Overall, the data are in good agreement with the Monte Carlo expectations.

We also checked the cluster ‘‘analogy’’ position, which is defined as $x_{ana} = x_{dig} + k * (\eta - 0.5)$ where η is the ratio of Tot charge in the largest row to the sum of ToT on the largest and the lowest row in the pixel cluster and k is a constant factor, dependent on the track incident angle. Figure 65 and 66 show the scatter plot of δx_{ana} and $\delta \eta$ between the pixel clusters in the left and right overlap regions, respectively. There seems a good agreement between data and Monte Carlo.

Since there is no specific requirement of the overlap hits in the current cosmic tracking, we can use that to measure the pixel hit efficiency properly by checking how often a pair of hits found in the overlap region as expected. We first start with a hit from a track that is on one side of the disk and extrapolate it to the other side of the same disk. If the expected hit is inside a fiducial region of an active module, we then try to match it to a real pixel hit in that module that has a minimum distance ($drmin = \sqrt{(\delta x^2 + \delta y^2)}$). If there is no hit or drmin is larger than 5 mm, we reset drmin to 4.9 mm. Figure 67 shows the drmin distributions from each disk in the data that agree well with the Monte Carlo. The efficiency is found to be close to 99.5% for all three disks by requiring $drmin < 1.0$ mm. Figure 68 shows the efficiency as function of module number (0-143), which indicates most of modules are close to 100% efficient.

As a cross check, we have recomputed the efficiency in overlap regions as function of different geometrical cuts (on x-axis the y-intercept of the oblique line which defines the overlap area), as shown in Figure 60. Results are stable for all choices, even if the error is greater as the overlap area decreases, due to lower number of hits.

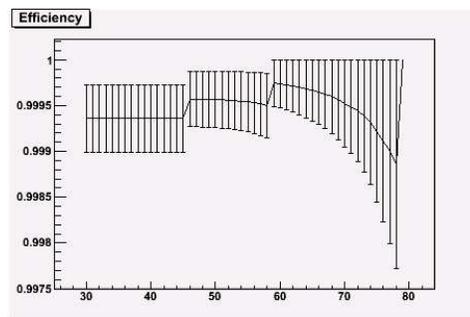


Figure 60: The efficiency in overlap regions as a function of the geometrical cut (on x-axis the y-intercept of the oblique line which defines the overlap area) for simulated sample.

The residual inefficiency is presumably due to the amount of defective pixels in the detector. To check that, we produced samples of simulated data with different values of fraction pixel disabled in the event. Efficiency decreases as the percentage raises, as expected, which is shown in Figure 61. The decrease in efficiency is less than disabled pixels. This fact is still to be understood quantitatively, but it seems due to the size of clusters: even if a pixel is disabled, nearby another pixel can still collect charge to make a hit.

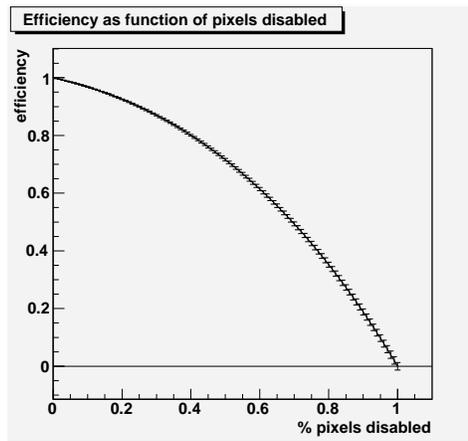


Figure 61: Efficiency between modules of the same disk as a function of the number of disabled pixels.

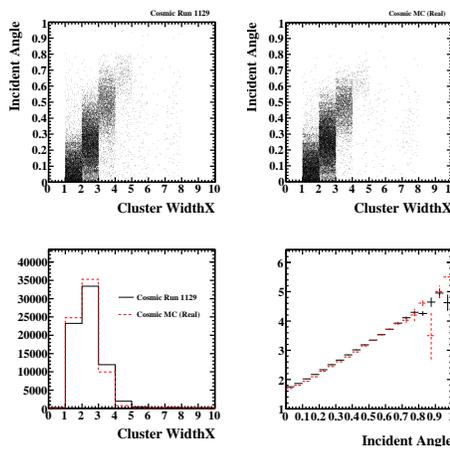


Figure 62: The distributions of pixel widthx for all the clusters found on the track as function of incident angles: top left and right are the data and the MC; bottom left is projection of pixel widthx; bottom right is profiling widthx as function of the incident angle.

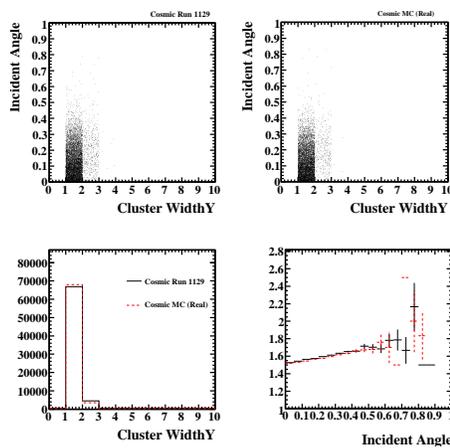


Figure 63: The distributions of pixel widthy for all the clusters found on the track as function of incident angles after noise clean up: top left and right are the data and the MC; bottom left is projection of pixel widthy; bottom right is profiling widthx as function of the incident angle.

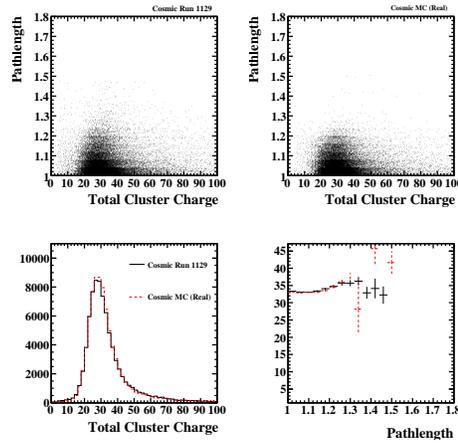


Figure 64: The distributions of Tot for all the clusters found on the track as function of incident angles after noise clean up: top left and right are the data and the MC; bottom left is projection of Tot; bottom right is profiling Tot as function of the incident angle.

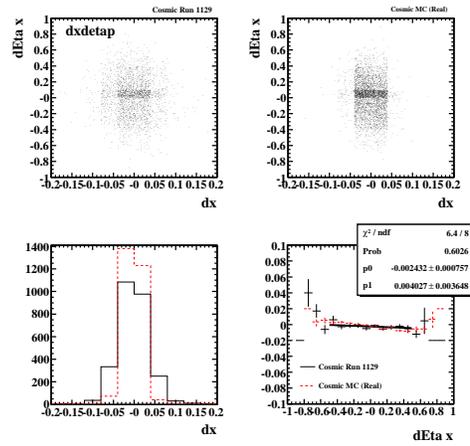


Figure 65: The scatter plot of δx and $\delta \eta$ between the overlap hits in the right overlap region: the top left and right are the data and the Monte Carlo; the bottom left plot is the projection of δx and the right plot is the profiling of δx vs $\delta \eta$. A line of fit gives a slope of $4 \pm 4 \mu\text{m}$ that is consistent with zero.

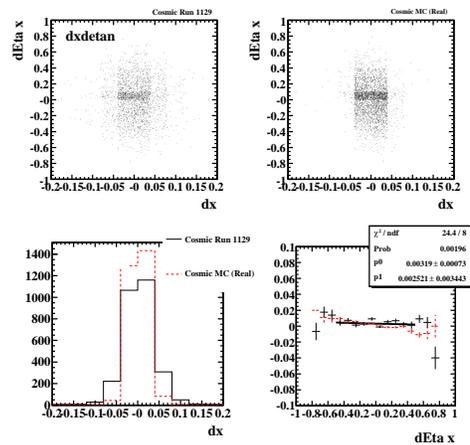


Figure 66: The scatter plot of δx and $\delta \eta$ between the overlap hits in the left overlap region: the top left and right are the data and the Monte Carlo; the bottom left plot is the projection of δx and the right plot is the profiling of δx vs $\delta \eta$. A line of fit gives a slope of $2.5 \pm 3.4 \mu\text{m}$ that is consistent with zero.

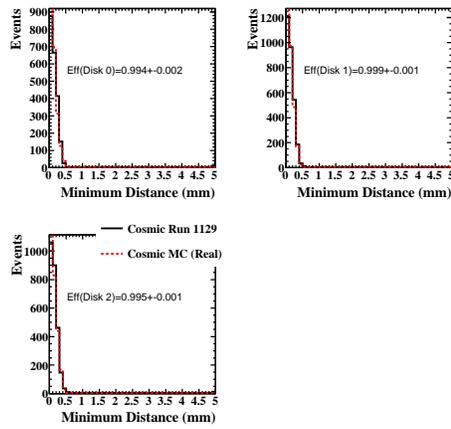


Figure 67: The distributions of the minimum distance between the expected and real pixel hits in the overlap region for three disks in the data (histogram) and in the Monte Carlo (red dash histograms).

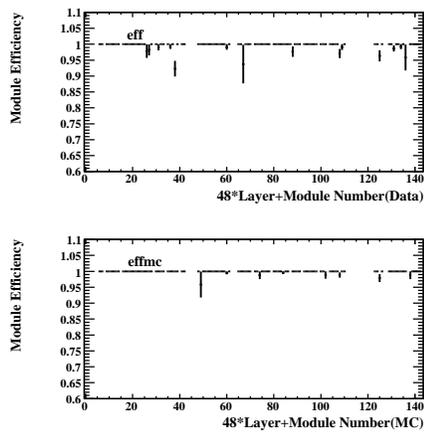


Figure 68: The pixel hit efficiency as function of module number (0-143) in the cosmic data (top) and in the Monte Carlo(bottom).

6.1 Pixel Cluster Width vs Genant4 Cuts in Simulation

In order to understand better the pixel clustering simulation in Monte Carlo, we have made a comparison of the cluster size distribution of clusters associated to a track between data and simulation with different Geant4 cuts. Figure 69 shows the mean cluster size as a function of the incident angle where Red points are the data from run 1129, black ones are the simulation with the default parameters (which are $1000 \mu m$ for the cut on the range of secondary events and no step limitation), blue points refer to the simulation with the cut on the range of secondary events of $100 \mu m$ and the maximum step length of $100 \mu m$, green points refer to a simulation with the cut on the range of secondary events of $50 \mu m$ and the maximum step length of $50 \mu m$.

There seems that a smaller cut on the range of secondary events increases the mean cluster size for small incident angles. On the other side there seems that decreasing the step limitation, introducing Landau fluctuations, decreases the mean cluster size, as if we miss one hit. There is a strange behavior that we still have to understand. For small incident angles, the simulations with non default G4 cuts seem describe the data better but it seems that for bigger incident angle the default simulation is more similar to the data.

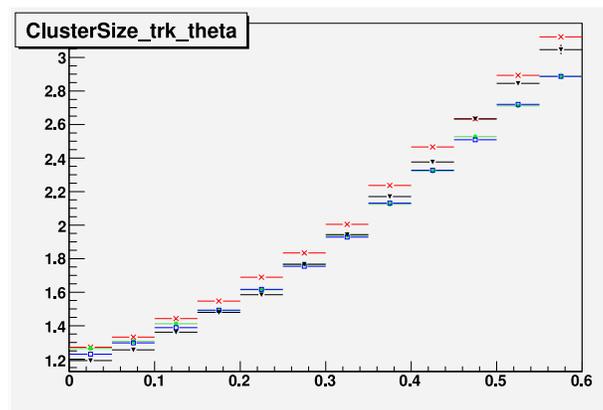


Figure 69: The mean cluster size as a function of the incident angle where Red points are the data from run 1129, black ones are the simulation with the default parameters (which are $1000 \mu m$ for the cut on the range of secondary events and no step limitation), blue points refer to the simulation with the cut on the range of secondary events of $100 \mu m$ and the maximum step length of $100 \mu m$, green points refer to a simulation with the cut on the range of secondary events of $50 \mu m$ and the maximum step length of $50 \mu m$.

7 Track Based Alignment Studies

7.1 Alignment with Overlap Residuals

There are about 24% of tracks containing an overlap hits from neighboring modules, which can be used to check the relative alignment between adjacent modules. Before doing so, we need to make sure the noise contamination of overlap hits are small, as shown in Figure 70 for the overlap hits in terms of the module occupancy, xy scatter plot, track phi and a scatter plot of TOT vs the fitted track chisq. As expected, the tracks with overlap hits seem much more reasonable, and pure compared with the Monte Carlo.

Assuming the module as a rigid body, there are 4 parameters to describe the module position inside the disk:

- Shift X_0 in local X axis along the short pixel direction
- Shift Y_0 in local Y axis along the long pixel direction

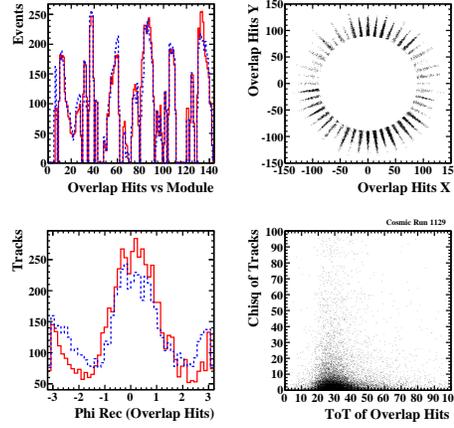


Figure 70: The distributions of the overlap hits in terms of the module occupancy (top left), xy scatter plot (top right), track phi (bottom left) and a scatter plot of TOT vs the fitted track chisq (bottom right). The solid is for the data and the dash is for the Monte Carlo.

- Shift Z_0 in local Z axis perpendicular to the disk
- Rotation α_0 along local Z axis

The relative alignment constants were determined considering the difference Δx and Δy between the positions of the overlap hits in the local frame of the odd module after taking into account the track extrapolation. Figure 71 shows the overlap residuals vs the number of odd modules in the left and right overlap regions. The dX_0 and rotation $d\alpha_0$ were determined by a line fit of Δx vs local Y of the hit in the odd module. The dY_0 were determined from the mean of Δy . The dZ_0 were determined by a line fit of Δx vs $\tan\theta \cdot \cos\phi$ where θ and ϕ are the angles of the reconstructed cosmic track. The overall residuals with nominal geometry before and after alignment correction are shown in Figure 72. The resolution in LocX improves from 23.0 to $16.0 \mu m$ while the resolution in LocY remains the same at $118 \mu m$. This is not far from $14.0 \mu m$ and $115.5 \mu m$ obtained from the Monte Carlo with a perfect geometry as shown in Figure 80. Figure 73 summarizes the relative alignment constants as a function of odd modules in the left and right overlaps regions. Most of them are within $20 \mu m$, which indicates the pixel endcapA is well reconstructed.

We have also checked the residuals with the endcapA as built survey geometry [19], which gives a better resolution of $20.0 \mu m$ than the one with nominal geometry. In order to check the correlation between the alignment constants and the survey data, we compare the relative alignment constants for modules with at least 50 overlap hits as shown in Figure 74, which seems some correlation present between the alignment and survey, but not as strong as we hoped. Another interesting test is to check the relative alignment between adjacent modules in the front and back disk as well as in the same or different sectors. Figure 75 shows the comparison between the alignment and the survey. Again, they seem correlated in some degree. However, it would be difficult to derive absolute alignment constants for the modules within the disk since there are some modules not function during the cosmic test. However, the results are quite interesting for what we have achieved so far with such limited data statistics.

7.1.1 Pixel Cluster Resolution

Figure 76 shows the overlay for the overlap residual in LocX before and after the alignment correction. Since the resolution does depend on the number of pixels in the cluster, we have measured the LocX resolution using the overlap clusters that consist of one or two pixels in LocX direction. For the clusters with a single pixel, the resolution becomes $14.7 \pm 3 \mu m$, which is close to the intrinsic resolution, defined as $50\mu m / \sqrt{12} = 14.4\mu m$. For the clusters with double pixels, the resolution of $13.8 \pm 0.3 \mu m$ is obtained, as shown in Figure 77, which is slight better than the resolution with a single pixel as expected from charge sharing correction.

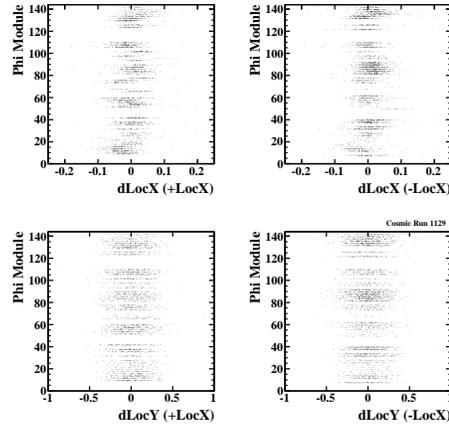


Figure 71: The overlap residual distributions as function of module number (module+48*Layer): residual X in +LocX and -LocX of odd modules (top) and residual Y (bottom).

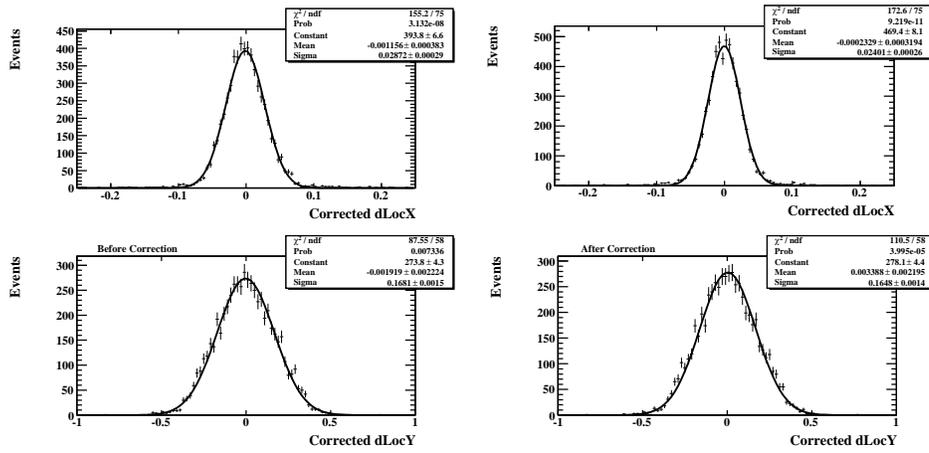


Figure 72: The overlap residual in LocX and LocY with nominal geometry with $dz=4.25$ mm (left) and after alignment correction (right). The resolution is $1/\sqrt{2}$ of the residual shown in the plot.

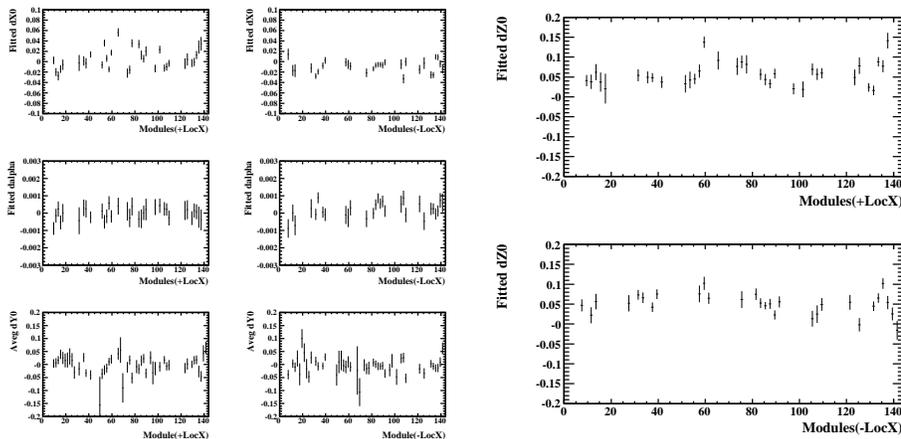


Figure 73: The relative alignment constants (dX_0 , dY_0 , $d\alpha_0$ and dZ_0) derived from the cosmic data as function of odd module in the overlap region with +LocX and -LocX.

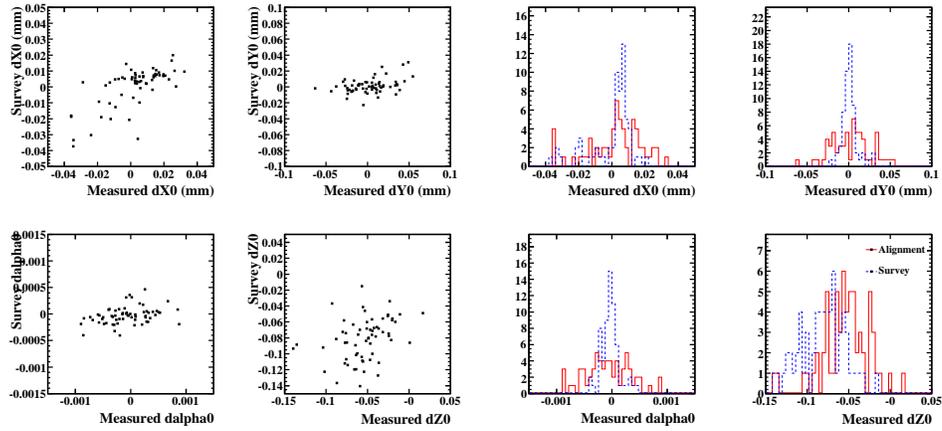


Figure 74: The scatter plot between the measured relative alignment and survey for neighboring modules on the left and the projections are on the right.

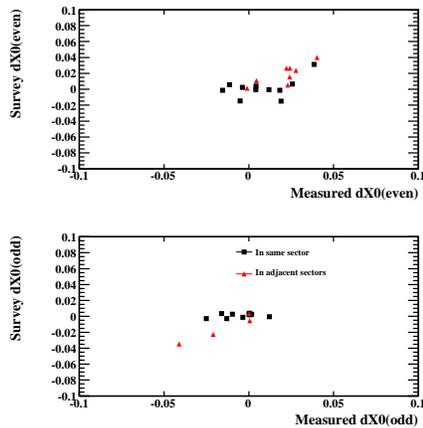


Figure 75: The scatter plot of relative alignment and survey between neighboring modules in the front disk or even modules (top) and in the back disk or odd modules (bottom). The back point is for modules in the same sector and the triangle point is for modules in different sectors.

We also repeated the alignment studies using all tracks without masking off the special pixels as well as an alternative alignment at the pixel hit level that uses the position of each pixel, instead of the cluster centroid [20].

The correlations between these alignment constants as well as with the survey are shown in Figure 78, which seems that alignment with noise removal improves agreement with survey. Figure 79 shows the correlations for the overlap modules in the same and different sectors respectively. The data seem keeping track of the survey well for big misalignment between the modules cross in the adjacent sectors (red star) while not so well for small misalignment for the modules in the same sectors (blue triangle).

7.1.2 Checks with Nominal and CSC Monte Carlo Samples

In order to understand the performance of alignment, we first applied the alignment procedure to the Monte Carlo sample generated and reconstructed with the nominal geometry. Figure 80 shows the overlap residuals before and after alignment correction, which give almost identical resolutions of $14.0 \mu\text{m}$ in LocX and $116 \mu\text{m}$ in LocY. The correspondent alignment constants are consistent with zero as expected as shown in Figure 81. The spread of distribution (RMS) gives the accuracy of alignment constants that expected to achieve with 110 K cosmic Monte Carlo events, which is below $5 \mu\text{m}$ for LocX, $15 \mu\text{m}$ for LocY, 0.2 mrad for α , and $10 \mu\text{m}$ for dZ0 respectively. Second, we applied the same algorithm to the CSC Monte Carlo sample that was generated using the nominal geometry, but reconstructed using a different CSC geometry [21]. Figure 82 shows the overlap residuals before and after alignment correction. The resolution in LocX has improved from $23.6 \mu\text{m}$ to $14.1 \mu\text{m}$, very close to the value obtained with the perfect geometry while the LocY resolution remains the same. There is a strong correlation between the alignment with the CSC constants as shown in Figure 83 as well as the difference between the two. The accuracy of alignment constants obtained this time seems slight worse than with a perfect geometry, which is due to a significant misalignment between disks in the CSC geometry up to mm level.

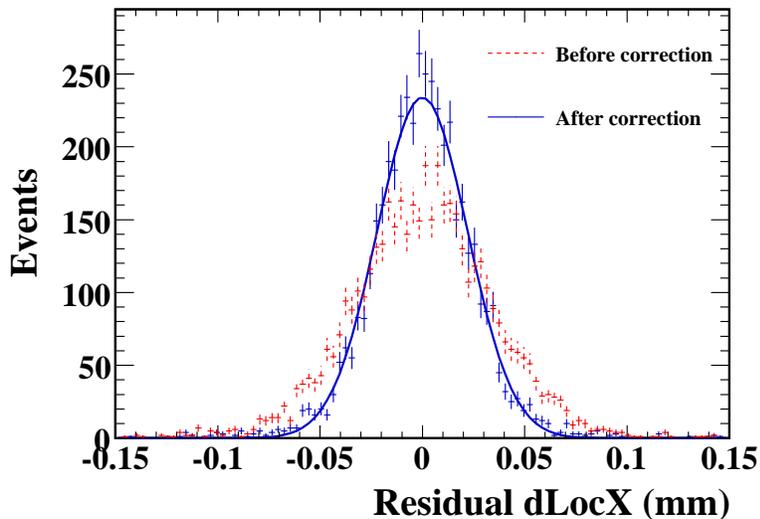


Figure 76: Residuals in LocX before and after alignment correction.

7.2 Alignment at the Pixel Hit Level

7.3 Alignment with the Robust Alignment Algorithm

The Robust Alignment algorithm [22] has been used to align the Pixel Endcap A modules. This algorithm is one of the four official track-based ATLAS alignment algorithms for the inner detector, with a full implementation in ATHENA. It is an iterative method, based on centring track residual and overlap residual distributions, which results in conceptual simplicity and numerical robustness.

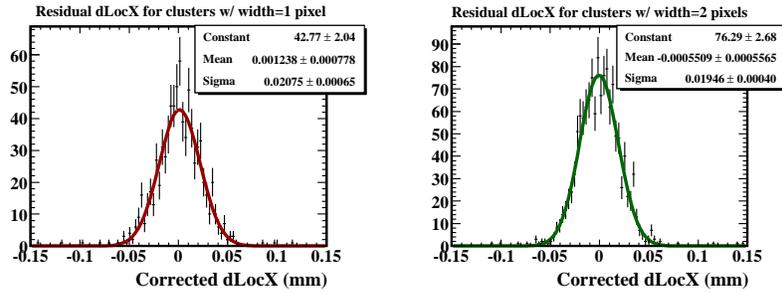


Figure 77: Residuals in LocX after noise removal and alignment correction for 1 pixel clusters(left) and 2 pixel clusters(right).

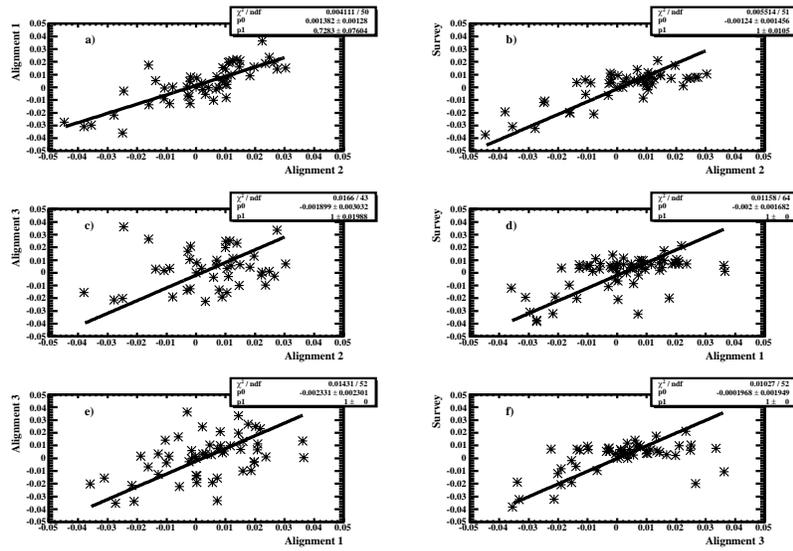


Figure 78: Scatter plots comparing the relative alignment constants between adjacent modules obtained from each study and the survey. Labels: Alignment 1 is at cluster level including noise; Alignment 2 is at cluster level after noise removal; Alignment 3 is at pixel level including noise.

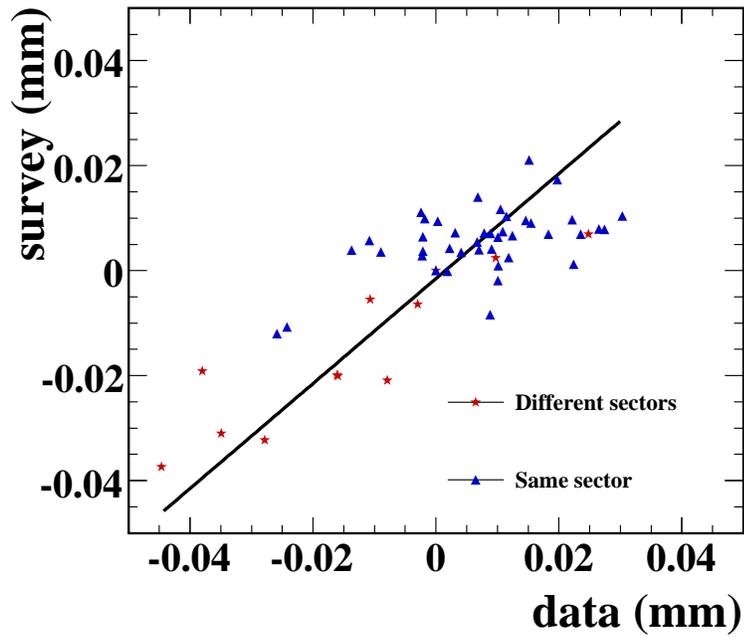


Figure 79: A comparison between the alignment at the cluster level after noise removal and the survey for the overlap modules in the same and different sectors.

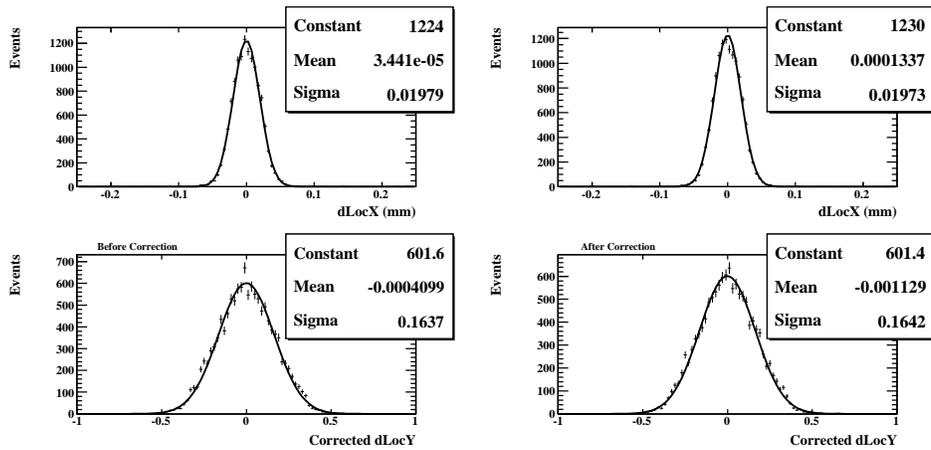


Figure 80: The overlap residual in LocX and LocY in the Monte Carlo with nominal geometry before and after alignment. The resolution is $1/\sqrt{2}$ of the residual shown in the plot.

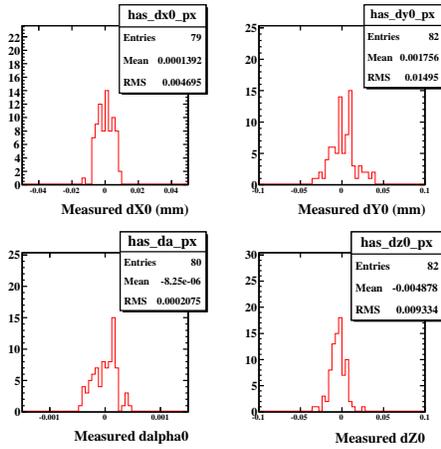


Figure 81: The distributions of alignment constants obtained from a perfect geometry, which show the central values consistent with zero and RMS the expected accuracy.

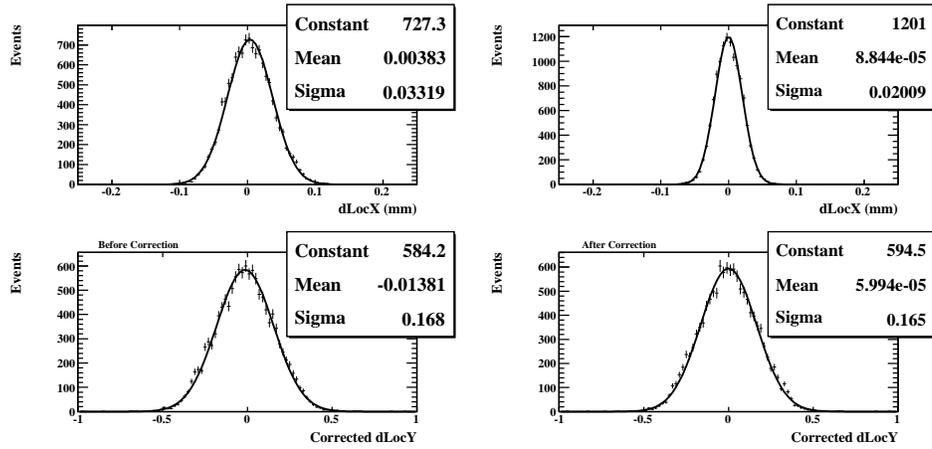


Figure 82: The overlap residual in LocX and LocY in the Monte Carlo with csc misaligned geometry before and after alignment. The resolution is $1/\sqrt{2}$ of the residual shown in the plot.

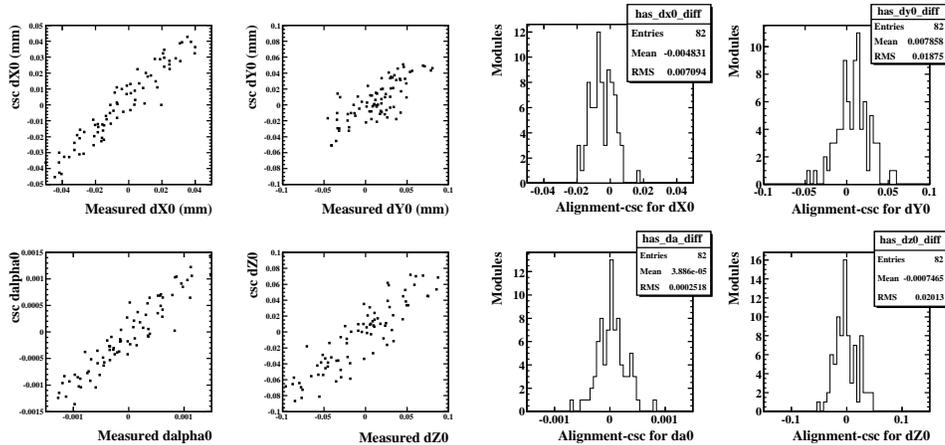


Figure 83: The distributions of alignment constants obtained from csc misaligned geometry, which show the central values consistent with zero and RMS the expected accuracy.

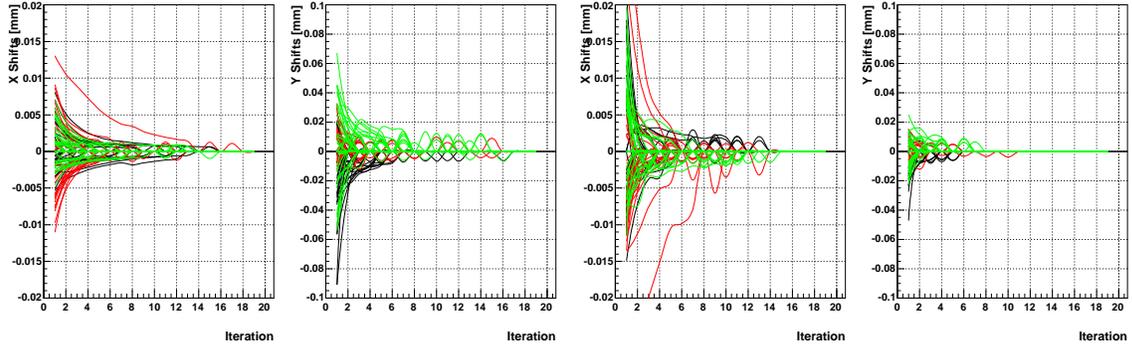


Figure 84: Differential shifts applied to modules per iteration for biased residuals (**left**) and unbiased residuals (**right**). Each line corresponds to a module. The colours red, green and blue stand for layer 0, 1, and 2, respectively.

Similar to alignment with overlap residuals described above, the residual R in local x, y -direction is defined as the distance between the hit and the intersect of the track with a given module projected on the measurement direction. Further, the same definition of the local coordinate frame of the module is used. If there are two hits from the same track in two adjacent modules a, b in the same layer of the detector, an overlap residual can be defined as the difference of the residuals for module a and b : $O(a, b) \equiv R(a) - R(b)$. For the endcaps of the pixel detector, the overlap residual can only be defined for the long side of the detector ('local x ' side, second index), in local x, y directions (first index): O_{xx}, O_{yx} . Two types of (overlap) residuals have been used: biased and unbiased. In the biased case the track is fitted using all of its associated hits, whereas in the unbiased case all hits but the one used for the residual calculation are taken into account. The alignment is performed for 2 major degrees of freedom for each module – translations in local x and local y .

The Robust alignment is based on two assumptions for a perfectly aligned detector:

- residual and overlap residual distributions are centred around 0,
- if a single module is shifted by δr in the direction of residual measurement r , the mean of the unbiased residual distribution is shifted by $-\delta r$.

Correlations between non-adjacent modules are taken into account by means of iterations.

The alignment of the Pixel End Cap-A modules with the Robust Alignment algorithm has been performed in ATHENA release 13.0.20. The default tag of SiRobustAlignTools has been used, whereas a patched 13.0.30 SiRobustAlignAlgs version was employed. Noisy pixels have been removed. Due to the missing magnetic field, the track fit was done with the GlobalChi2Fitter algorithm forced to straight line mode, and accounting for multiple scattering effects and track energy loss have been switched off. Unless indicated otherwise, the alignment was started from the nominal geometry, i.e. without any prior knowledge of alignment.

In Fig. 84, the differential shifts applied to modules are shown for 20 iterations for biased (left) and unbiased (right) residuals. The differences between biased and unbiased residuals will be highlighted later in the text. Each line corresponds to one module, and the colour-coding in red, green and blue stands for Layer 0, 1, and 2, respectively. The alignment results of the Robust Alignment algorithm converge, and the shifts applied to modules decrease with the number of iterations. For most of the modules they go to 0, as the condition for a module to be shifted is not fulfilled any more:

$$\delta r > \frac{1}{N_a} \cdot \frac{1}{\frac{1}{\langle R_r^2(a) \rangle - \langle R_r(a) \rangle^2} + \frac{1}{G \cdot (\langle O_r^2(a) \rangle - \langle O_r(a) \rangle^2)}}, \quad (5)$$

i.e. the shift δr calculated for module a is smaller than the statistical error from its residual $R_r(a)$ and overlap residual $O_r(a)$ distribution. Here, N_a stands for the number of hits collected, and G is a geometrical weighting factor typically in the range of 10-300 defined in [22].

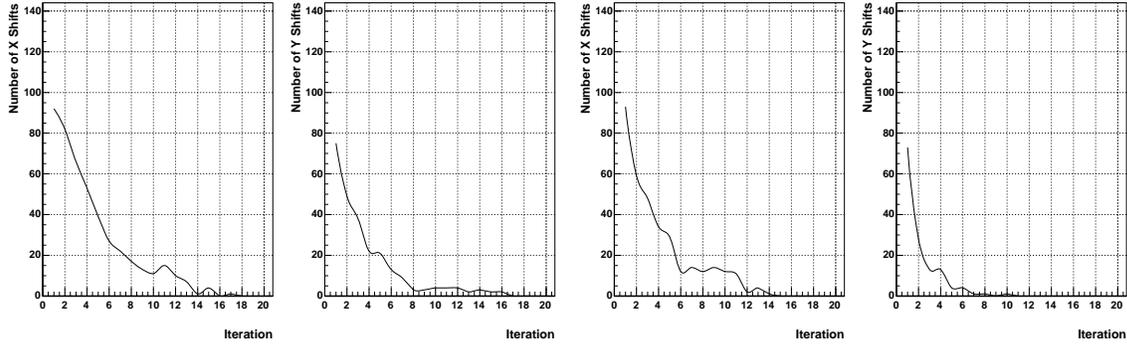


Figure 85: Number of modules shifted per iteration for biased residuals (**left**) and unbiased residuals (**right**). For shift conditions see text.

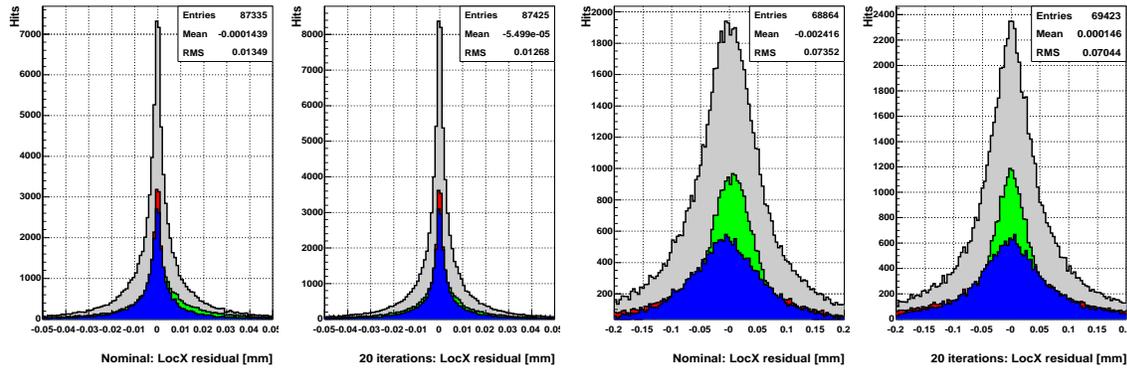


Figure 86: Residual distributions in local x direction for the biased (**left**) and the unbiased (**right**) case for nominal alignment and after 20 iterations of the Robust Alignment algorithm. The colour-coding is red, green, blue for layer 0, 1, 2, and grey for the sum of the residual distributions of the three layers.

Since the alignment of the detector improves with iterations, the number of modules to be shifted should decrease, as evident from the shift condition defined in Eqn. 5. As demonstrated in Fig. 85, this number asymptotically approaches 0. The initial number of modules to be shifted does not correspond to 144 – the total number of Pixel End Cap-A modules. This is due to the fact, that 23 modules were not read out. On top of that, a condition on a minimum number of hits per module in order for it to be considered by the Robust Alignment algorithm was required. It was set to 100 hits for the biased and, as a check, to 25 hits for the unbiased case.

With the improving alignment of the detector, the residual and overlap residual distributions should improve, i.e. their root mean square should decrease and their mean should approach 0. For the local x residual, this is demonstrated in Fig. 86 for the biased (left) and the unbiased case (right). The three layers 0, 1, and 3 are colour-coded as red, green and blue. The grey histogram is the sum of the residual distributions from all three layers. The general difference in the width of the residual distribution for the biased and the unbiased case results from the fact, that some 70% of the tracks considered have 3 hits. Therefore, typically, for unbiased residuals a fit through only 2 points will be made, resulting in larger residuals for the module with the non-fitted hit. Additionally, for biased residuals, one might face a pathological situation, where the χ^2 of the track fit is minimised for a track going through one or even two hits. This will result in entries close to 0 in the residual distribution.

As the alignment of the detector improves, more tracks should be found and reconstructed by the track fitting algorithm, resulting in an increase of the number of residuals, as can be seen from the number of entries in the respective histograms in Fig. 86.

The improvement in the residual distributions discussed above is rather limited. This is mainly due to high multiple scattering effects, since the momentum spectrum of cosmic rays triggered in this setup

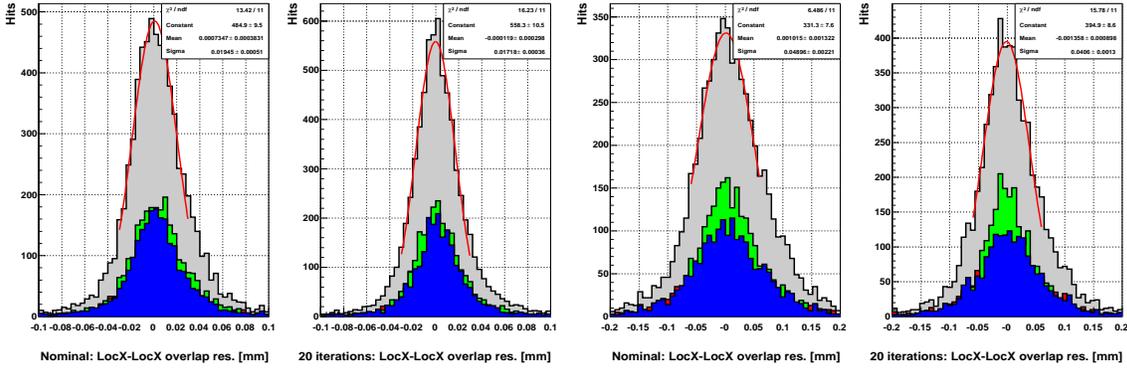


Figure 87: Overlap residual distributions O_{xx} of type x in local x direction for the biased (**left**) and the unbiased (**right**) case for nominal alignment and after 20 iterations of the Robust Alignment algorithm. The colour-coding is as in Fig. 86.

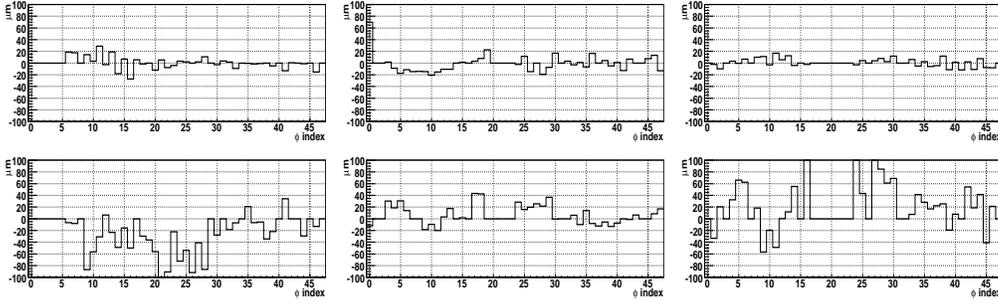


Figure 88: Alignment constants in local x,y after 20 iterations for layer 0 (**left**), layer 1 (**middle**) and layer 2 (**right**) of the Pixel End Cap-A subdetector with respect to nominal geometry, derived with biased residuals.

starts as low as $p^{\min} \simeq 140$ MeV. Therefore, it is favourable to consider overlap residuals, since the multiple scattering effects are smaller. They are assigned a higher weight with the Robust Alignment algorithm compared to regular residuals, being 20 for the biased and 10 for the unbiased case. The O_{xx} overlap residual distribution is shown in Fig. 87 for the biased and the unbiased case on the left or right hand side, respectively. For the same reasons as detailed in the previous paragraph, the unbiased overlap residuals are approximately twice larger than the biased ones. The improvement in the RMS of the overlap residuals is $\sim 12\%$ for the biased and $\sim 17\%$ for the unbiased case, which can be attributed to dominating multiple scattering effects, limited statistics and the high mounting precision of the Pixel End Cap-A modules.

7.3.1 Derivation of Alignment Constants

The alignment constants for the Pixel End Cap-A modules have been derived with respect to the nominal module positions in the local x,y coordinates, using biased and unbiased residuals over 20 iterations. The details of the calculation can be found in [22]. The constants for the biased approach are visualised in Fig. 88 for layer 0, 1, and 2.

The average magnitude of corrections for local x direction is $O(10\mu\text{m})$, with good agreement between the two sets of constants. For local y direction, the corrections are $O(50\mu\text{m})$, and there is limited agreement between the two sets of alignment constants. This can be attributed to the fact, that the measurement in local y is one order of magnitude less precise than in local x . Therefore, the width and the resulting uncertainty of the residual distribution is larger, and so are the corrections. The mean correlation in all 3 layers between the two sets of alignment constants in local x is $\langle\rho\rangle \simeq 84\%$, and $\langle\rho\rangle \simeq 48\%$ in local y , for details by layers see first row of Tab. 10. The conclusion is, that with the limited track

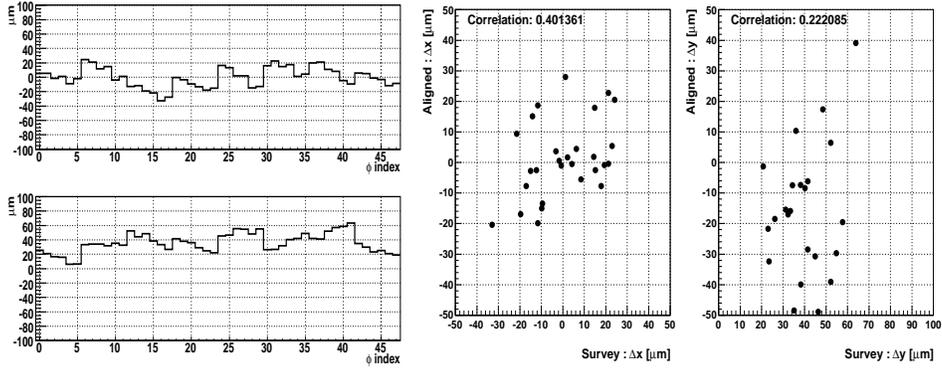


Figure 89: Alignment constants in local x, y for layer 0 of the Pixel End Cap-A subdetector as obtained with the survey without any track-based alignment (**left**). The correlation between alignment constants derived with the Robust Alignment and the survey (**right**).

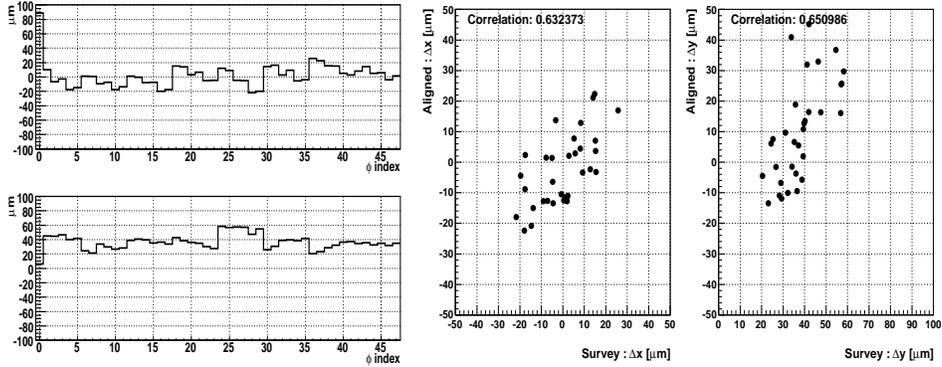


Figure 90: Alignment constants in local x, y for layer 1 of the Pixel End Cap-A subdetector as obtained with the survey without any track-based alignment (**left**). The correlation between alignment constants derived with the Robust Alignment and the survey (**right**).

statistics available and the low momentum scale of the tracks, the local y precision of track-based alignment is lower than the mounting precision. For the local x direction, the Robust Alignment algorithm yields numerically stable and reliable results.

7.3.2 Comparison of the Alignment Constants Derived using Robust Alignment with the Survey

As already detailed in previous subsection, an optical survey of the Pixel End Cap-A modules has been made. The alignment constants determined with the survey and without any track-based alignment are presented for layer 0, 1, and 2 on the left hand side of Fig. 89, Fig. 90, and Fig. 91. The alignment constants obtained with Robust Alignment have been compared with the results of the survey, as shown on the right hand side of the respective figures. The agreement between the results obtained using Robust Alignment and the optical survey is limited for local x ($\langle \rho \rangle \simeq 42\%$, biased). The situation in local y is exacerbated ($\langle \rho \rangle \simeq 30\%$, biased). The correlation of alignment constants is detailed in the second and third row of Tab. 10 by layers. There is a significant positive offset of alignment constants obtained with the survey with respect to Robust Alignment in local y . This is due to the fact that the Pixel End Cap rings have $O(30\mu\text{m})$ larger radii than nominal geometry. With the circle of operating modules on a given layer not closed, such correlated movement of the modules is beyond detection with the Robust Alignment algorithm.

As the agreement between track based alignment and the optical survey is limited, several cross-checks have been made. The Robust Alignment algorithm has been employed to align the subdetector over 20 iterations starting from the survey, rather than nominal geometry. The resulting alignment con-

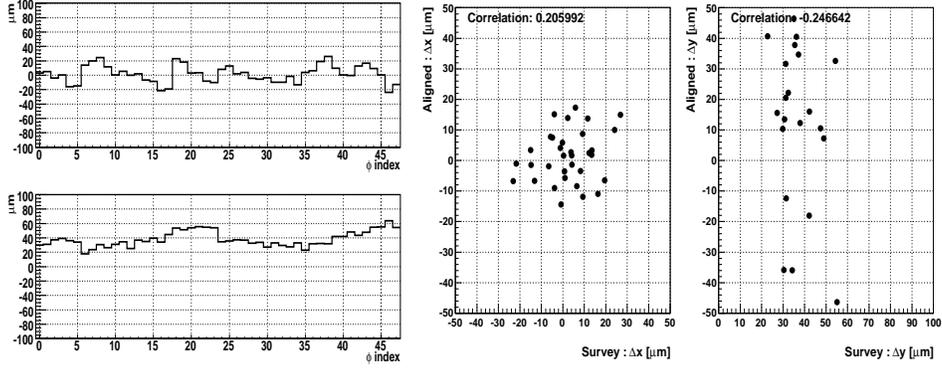


Figure 91: Alignment constants in local x,y for layer 2 of the Pixel End Cap-A subdetector as obtained with the survey without any track-based alignment (**left**). The correlation between alignment constants derived with the Robust Alignment and the survey (**right**).

set 1	set 2	L0 : x	L0 : y	L1 : x	L1 : y	L2 : x	L2 : y
Biased	Unbiased	0.86	0.34	0.90	0.70	0.76	0.41
Biased	Survey	0.36	0.27	0.64	0.65	0.25	-0.01
Unbiased	Survey	0.48	0.38	0.49	0.32	0.22	0.24
Biased from Survey	Survey	0.90	0.27	0.76	0.64	0.92	-0.01
Biased from Survey	Biased	0.52	0.03	0.68	0.65	0.46	-0.06
Biased w. γ from Survey	Survey	0.40	0.22	0.63	0.65	0.21	-0.24
Biased w. γ from Survey	Biased	0.97	0.94	0.99	0.96	0.97	0.94

Table 10: Correlation between various sets of alignment constants in local x and local y . The last two rows are detailed in the text.

stants in local x and the Robust Alignment results obtained starting from nominal geometry are much alike, with a mean correlation of $\langle \rho \rangle \simeq 55\%$. The situation in local y is similar, but less pronounced with a mean correlation of $\langle \rho \rangle \simeq 21\%$. This is shown in fifth row of Tab. 10. To cross-check which role is played by the alignment of rotations around local z axis (commonly referred to as γ -angle misalignment) in this comparison, a set of alignment constants has been derived with the Robust Alignment algorithm over 20 iterations using the nominal geometry for all 6 degrees of freedom per module but the rotations around local z axis. For these, survey results have been used. Again, the resulting alignment constants are very similar to the Robust Alignment results shown above, with a correlation coefficient of $\langle \rho \rangle \simeq 98\%$ in local x , and $\langle \rho \rangle \simeq 95\%$ in local y (last row of Tab. 10). All cross-checks have been made with biased residuals.

8 Lesson learned and future improvements

In the previous sections there have been presented a wide spectrum set of studies that provides significant insight about what to expect when the full detector will be in operation.

First of all, from the noise measurements, it is possible to conclude that the most relevant noise source is *fixed pattern noise*, which in principle can be suppressed almost completely by masking, either on-line or off-line the noisy channels.

Almost the totality of noisy channels was detected as problematic during the module acceptance tests. Unfortunately, it is not possible, by the simple fact a pixels was *special* during these tests, to predict this specific pixel will be noisy. Therefore the number of *special* pixels can be taken as an upper limit to the inefficiencies, including both dead channels and channels masked because of the excessive noise rate. This number is few per mill of the total number of pixels.

Random noise, instead, is at a very low level and can be neglected for most application.

Digitization parameters have been taken from the characterization tests performed during module

production. The simulation produced with these parameters has been compared with the collected data. This proved to be a good validation of the ATLAS pixel detector simulation and makes us confident that, extraction of calibration data on-site, from dedicated runs, will be a reliable source of updated information about the evolution of the detector operating conditions in the LHC running.

The tracking studies, especially the ones related to particles passing in the overlap regions between adjacent modules, have been very useful in spotting problems in our geometry description. In particular, different conventions in the detector description and Reconstruction software have been spotted and properly taken into account. Another observation was that a mismatch between the fabrication drawing and the actual detector assembly, initially observed in the sectors' survey, is confirmed by alignment data. The better agreement of alignment data with the survey than with the nominal drawings, show the survey is a reliable starting point.

This will be of much more relevance for the barrel part, since in that region the disagreement between the survey and nominal position is much worse, at the $200\ \mu\text{m}$ level.

Besides the analysis summarized above this run was a very useful opportunity to finalize many software updates needed for the detector commissioning, in particular a complete revision of the digitization, the implementation of a calibration database in COOL, which can be accessed both by simulation and reconstruction processes, and finally the insertion of the survey information to be used as initial alignment step. The technical aspects of this updates in the pixel software and description are described in an accompanying Computing System Commissioning note [18].

There are some additional studies interesting for the pixel offline analysis and calibration, which include the collection of DCS information, in order to estimate the amount of data that will be collected during normal data taking and smoothing parameters for the PVSS logging and transfer to COOL.

9 Conclusion

This document summarizes the results of the offline analysis for the pixel endcapA system test cosmic data. The setup consists of one pixel endcap of three disks, for a total of 144 modules and 6.6 million channels, about the 8% of the full detector. The endcap is hung vertically and sandwiched between one scintillator at the top and a set of three scintillators at the bottom for trigger. It is completely equipped with services and managed by a initial production of the ATLAS DAQ system components with the goal of exercising the readout system, data taking and testing the full chain of offline reconstruction.

Runs with random trigger allow us to measure the noise rate. The observed noise occupancy per BCID readout is achieved to 10^{-10} after removing the noisy pixels. Comparison with the detector characterization performed during the detector assembly shows that approximately 90% of these noisy pixels were already flagged as "special" during the production test and that the total fraction of special pixels is below 0.1%.

The tracking studies, especially relating to tracks passing in the overlap regions between adjacent modules in the same disk, have been very useful in spotting problems in our geometry description and can be used for the relative alignment between the adjacent modules with the residuals of the overlap hits.

The characteristic of pixel clustering in the data are checked and agree well with the Monte Carlo simulation. The pixel cluster efficiency has been measured to be $> 99\%$ using the hits in the module overlap region. When using the nominal geometry, an initial resolution of $23.0\ \mu\text{m}$ is obtained. After a preliminary alignment this improves to $16.0\ \mu\text{m}$, which is not far from $14.0\ \mu\text{m}$ expected from the Monte Carlo simulation. The relative alignment constants are also cross checked between the data and the survey obtained during the detector assembly, which shows a clear correlation for the modules with enough overlap hits (≥ 50) in the data.

The experience gained in the SR1 running will be also extremely useful for the preparation of the cosmics running within the whole ATLAS setup in the pit and its analysis and understanding.

References

- [1] M. Keil *et al.*, *Pixel Endcap A System Test*, in preparation.

- [2] M. Leyton *et al.*, *Service Tests for the ATLAS Pixel Detector*, ATL-IP-QA-0026 (2007).
- [3] G. Alimonti *et al.*, *Analysis of testbeam data of ATLAS Pixel detector modules with a high intensity pion beam*, ATL-COM-INDET-2005-012, (2005).
- [4] M. Zdražil *et al.*, *Pixel EndCap A cosmic test documentation on ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/PixelCosmic>.
- [5] M. Keil, P. Morettini *et al.*, *Pixel Detector System Test - ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/SystemTest>.
- [6] K. Einsweiler *et al.*, *ATLAS On-detector Electronics Architecture*, version 3.0, Dec 9, 2003. Also see Kevin's presentation at LBNL Instrumentation Colloquium, April 13, 2005, *The Atlas Pixel Detector*, p. 8-18 mainly.
- [7] K. Einsweiler *et al.*, *ATLAS On-detector Electronics Architecture*, version 3.0, Dec 9, 2003. Also see Kevin's presentation at LBNL Instrumentation Colloquium, April 13, 2005, *The Atlas Pixel Detector*, p. 8-18 mainly.
- [8] S. Golfarb and A. Schaffer *et al.*, *Definition of Off-line Readout Identifiers for the ATLAS Detector*, ATL-SOFT-2001-004, (2001).
- [9] SuDong, H. Gray *et al.*, *Pixel Scintillator Trigger Layout documentation - ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/PixelScintillatorTriggerLayout>.
- [10] S. Eidelman *et al.*, *Review of Particle Physics*, Physics Letters B 592, 1 (2004). See <http://pdg.lbl.gov> for Particle Listings and complete reviews, plus a directory of online HEP information.
- [11] A. di Simone *et al.*, *Cosmic simulation package documentation on ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/CosmicSimulation>.
- [12] D. Costanzo, G. Gorfine, *PixelGeoModel package documentation on ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/PixelGeoModel>.
- [13] M. Gallas and Simone Dell'Acqua *et al.*, *GEANT4 ATLAS detector simulation applications*, <http://atlas-computing.web.cern.ch/atlas-computing/packages/simulation/geant4/G4Atlas-Apps/doc/index.html>.
- [14] O. Oye *et al.*, *InDetCosmicSimAlgs simulation package documentation on ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/InDetCosmicSimAlgs>.
- [15] M. Elsing, F. Tegenfeldt *et al.*, *ATLAS Pixel Detector digitization package on ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/PixelDigitization>.
- [16] M. Zdražil *et al.*, *How to run simulation of pixel endcap A cosmic test*, <https://twiki.cern.ch/twiki/bin/view/Atlas/HowToRunPixelEndCapCCosmicSimulation>.
- [17] T. Cornelissen, *CTBtracking: track reconstruction for the testbeam and cosmics*, ATL-INDET-INT-2006-001 (2006).
- [18] *Pixel Software Commissioning*, in preparation
- [19] A. Andreatza, V. Kostyukhin, R. Madaras, *Pixel Survey*, ATL-IP-QC-0035 (in preparation).
- [20] M. Zdražil, *Comparison of (x,y) shifts derived from cosmic data with survey data*, talk at <http://indico.cern.ch/conferenceDisplay.py?confId=16451>.
- [21] G. Gorfine, *Setting Geometry Version documentation in ATLAS Twiki pages*, <https://twiki.cern.ch/twiki/bin/view/Atlas/SettingGeometryVersion>.

- [22] F. Heinemann, “Track Based Alignment of the ATLAS Silicon Detectors with the Robust Alignment Algorithm”, ATL-INDET-PUB-2007-011; F. Heinemann, PhD Thesis, Univ. of Oxford, in preparation for 2007; D. Hindson, PhD Thesis, Univ. of Oxford, 2004.