CMS-IN 2001/025
July 2, 2001

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

On Calibration, Zero Suppression Algorithms and Data Format for the Silicon Tracker FEDs

I.R. Tomalin
R.A.L.

Abstract

This note proposes various zero suppression algorithms which the Silicon Strip Tracker FEDs could use to reduce the data volume flowing to the DAQ. These algorithms vary in how they subtract common-mode noise and perform cluster finding. Using Monte Carlo simulations, the relative performance of the algorithms is compared, both in terms of cluster finding efficiency and in terms of their effect on the data volume (tracker strip occupancy). The characteristics of the resulting clusters are explored. Also addressed is the question of how the FEDs should format their output data.

1 Introduction

Signals produced by the silicon micro-strip tracker will initially be processed by the front-end APV chips, each of which reads 128 silicon strips. The APVs amplify the signals and store them in a memory pipeline until the Level 1 trigger decision is made. If this decision is positive, the data from multiplexed pairs of APVs are transmitted along analogue optical fibre links to the Front-End Driver (FED) cards in the control room. The FEDs will digitise the data, optionally perform zero suppression and then transmit the data to the Data Acquisition (DAQ) system. A description of the FED can be found in Refs. [1, 2] and a detailed description of its foreseen functionality in Ref. [3].

The main aim of this document is to help define how the FEDs will perform zero suppression on the tracker data. The basic idea, as explained in Ref. [2], is for the FEDs to perform cluster finding and only to output to the DAQ, pulse height information for those strips associated to clusters. The FEDs contain firmware-programmable FPGA chips, which allow algorithms of moderate complexity to be implemented (and changed if necessary). The choice of algorithm is guided by a desire to keep it as simple as possible, whilst at the same time obtaining adequate FED performance in terms of cluster finding efficiency and produced data volume (i.e., strip occupancy). This note therefore studies this performance in detail. It also covers the related issues of how the FED should obtain the necessary calibration constants and how the FED output data should be formatted.

If no zero suppression were performed by the FEDs, the data rates flowing from the tracker to the DAQ would be such that the L1 trigger rate would be limited to about 2 KHz [19]. This would be useful for Pb-Pb collisions and test runs, but not for high luminosity p-p collision rates, where L1 trigger rates of up to 100 KHz are required. In this latter case, zero suppression must be performed. The effectiveness of this zero suppression depends on the strip occupancy in the tracker. A 100 KHz trigger rate remains possible providing that the mean strip occupancy in the tracker is less than 2.5%, assuming that the data rates flowing into the tracker assigned inputs of the DAQ event builder switch can be perfectly balanced (which they can't). If one takes into account the likely imbalance in the data rates flowing to each DAQ input, the mean occupancy must be less than 1.8%. It is desirable to reduce the occupancy below this to obtain some safety margin. This is discussed in detail in Ref. [19].

Section 2 briefly describes the calibration procedure which must be performed before the FEDs can do cluster finding. In Sect. 3, the simulation tools which are used for the studies presented in this note are described. Section 4 shows some typical characteristics of clusters in minimum bias events, simply to give an idea of what the FED is looking for. Section 5 compares the performance of possible clustering algorithms, beginning with a comparison of algorithms used by existing experiments and continuing with studies of cluster finding efficiency and ghost rates. Section 6 studies the related problem of how to estimate and subtract the common-mode noise. Finally, Sect. 7 shows the expected strip occupancy using the best algorithm found, and also examines how the output data should be formatted to minimise its size. Section 8 concludes by proposing the best algorithm on the basis of these studies.

2 Calibration and Monitoring

2.1 Calibration

A calibration run could be taken before each LHC fill to update the constants needed by the FED to do cluster finding. This might require 1000 events to be written without zero suppression to a computer, which would then use them to calculate the following quantities:

- The pedestal of each strip. (i.e., The mean pulse height excluding signals.)
- The rms noise on each strip. (i.e., The average variation of the pulse heights about the pedestal, excluding signals.)
- The rms variation from event to event of the common-mode offset per APV, (where the common-mode offset is defined as the mean pulse height on all strips in an event, excluding those with signals.)
- List of bad strips, defined as those which are either dead or very noisy.

Using 1000 events ensures that the statistical uncertainties on the pedestals and noise would be smaller by a factor of ≈ 30 than the noise itself, and hence small enough not to degrade cluster finding performance.

It is proposed to read out these calibration events via the VME back-plane of each FED and process them in the FED crate controller. An alternative solution would be to read out the events through the fast S-LINK data channel, and process them on the DAQ computer farm. Which of these two solutions is best may depend on the data rates. For a 96 channel FED, which digitises the pulse height information from each strip with a 10 bit ADC, the data volume per event is $96 \times 2 \times 128 \times 10$ bits = 31 Kbytes. As there are likely to be 20 FEDs per crate, this implies a data size of 610 Kbytes/event/crate.

A typical VME crate controller, such as the CES RIO3, has a 1 Gbyte DRAM memory [4], which would allow it to buffer 1600 events in memory. This is more than adequate, since it is likely that each event could be analysed as it arrives, in which case only a small buffer would be needed.

The time required to transfer the data over the VME link to the crate controller is also an issue. Assuming a data transfer rate of 30 Mbytes/s [3], the time required to transfer 1000 events from the crate would be $1000 \times 0.61/30 = 20$ seconds. One should also consider the time required for the processor to determine the calibration constants for each of the $20 \times 96 \times 2 \times 128 = 500$k strips in the crate. Sophisticated, off-line analyses running on present day computers can do this in less than one minute. This could therefore easily be done between LHC fills.

The calibration constants must be available off-line, so need to be merged with the data stream as start-of-run/slow-control records or stored in a calibration database.

2.2 Monitoring

The calibration constants should occasionally (e.g., once per hour) be checked during a fill, to see if they have altered or if any additional bad wires have been identified. This would help ensure data quality. One might also wish to update the constants if they do change.

3 Simulation Tools

Most of the studies presented in this note were performed with the CMS Monte Carlo. This has two parts: CMSIM v120 which simulates the energy deposits of particles in the silicon wafers, and ORCA v4.50 which converts these deposits to a pulse height on each strip, after addition of noise and common-mode noise. ORCA also digitises the signals and simulates the various FED zero suppression algorithms. The tracker geometry used corresponds to the latest all-silicon tracker proposals [5]. Unless otherwise stated, high luminosity p–p collisions were simulated, which implies a mean of 24 superimposed minimum bias events per bunch-crossing.

Estimates of cluster finding efficiency are extremely sensitive to the assumed signal to noise (S/N) ratio. It is expected that, after irradiation, the final tracker will have S/N ratios of 13 and 15 in the inner and outer tracker respectively. However, to be prudent, it is desirable to use smaller S/N values of 10 and 12 respectively for studies of FED performance [6]. Unfortunately, in CMSIM v120, whilst the S/N ratio usually takes a fairly reasonable value of 10.4, it drops to 6 in the inner three end-cap rings and rises to 15 in much of the inner four barrel layers ! (This problem has just been fixed in CMSIM v122.) The cluster finding efficiency also relies on an accurate simulation of Landau fluctuations, which fortunately is provided in ORCA v4.50.

In view of the inappropriate S/N ratios in the simulation, most of the ORCA studies presented in this note do not use the end-caps. Even so, one should treat the results with caution. In particular, those on cluster finding efficiency in the inner barrel are doubtless optimistic.

Motivated by the problems with ORCA, a second simulation tool was developed for studying cluster finding efficiencies. This is the Toy Monte Carlo described in Sect. 3.1, which provides an S/N ratio of 10.4.

Studies of occupancy in the tracker are sensitive to the assumed capacitive couplings between strips, since the larger these are, the wider the reconstructed clusters become. In the ORCA simulations used here, a charge deposited on one strip induces 6% of its value on each of the neighbouring strips. (N.B. This differs from the default capacitive couplings in ORCA, which unrealistically also assume large capacitive couplings to the next-to-neighbouring strips.)

Tracker occupancy is also sensitive to the APV time resolution, since a good resolution suppresses clusters from out-of-time bunch-crossings. In ORCA the time resolution is assumed to be 12.1 ns, when the APV is running in deconvolution mode. This corresponds to recent measurements of APV25 performance [7].

3.1 The Toy Monte Carlo

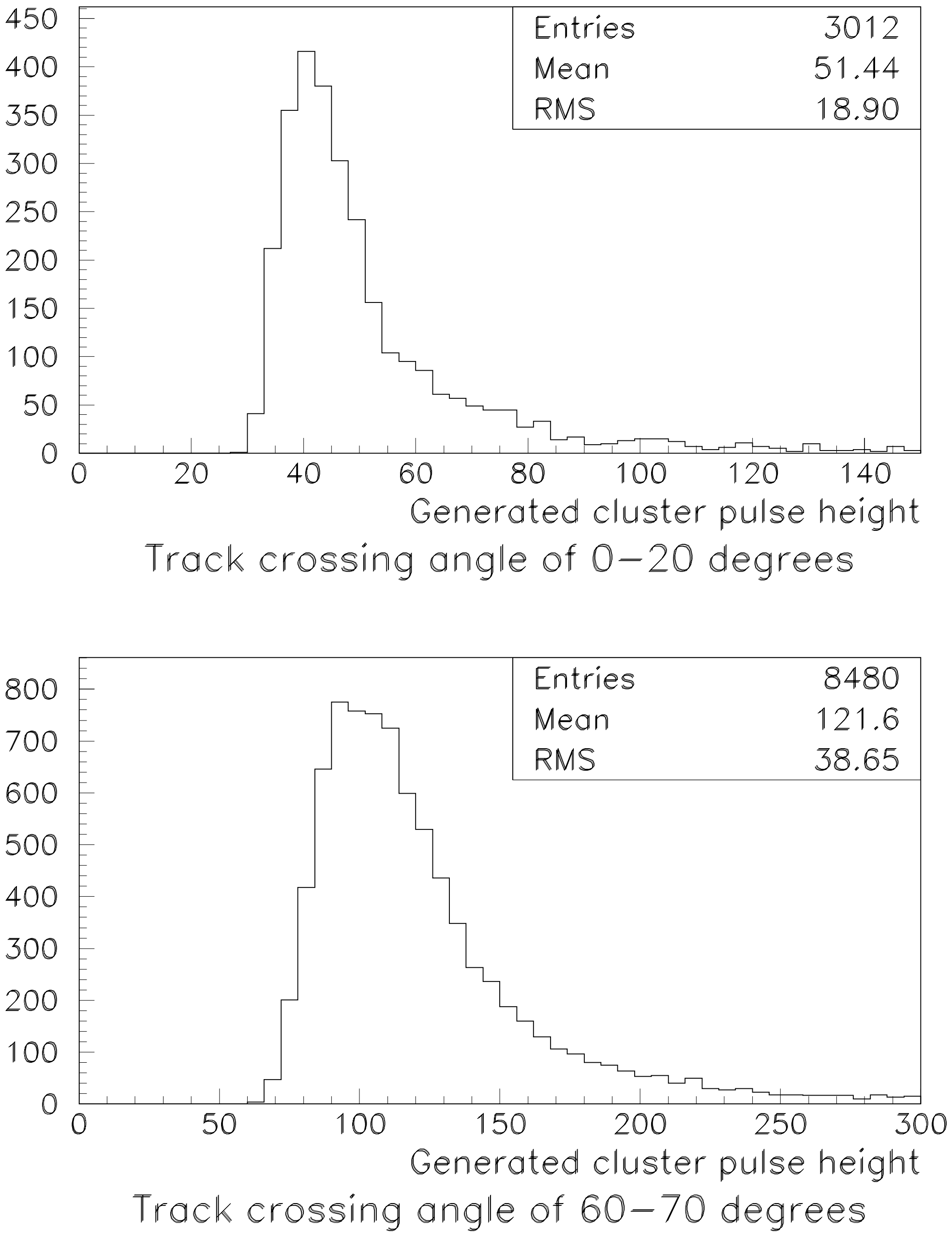
The Toy Monte Carlo program [8] simulates a detector with 128 strips as follows:

- Noise is simulated on each strip according to a Gaussian distribution of width 3.5 ADC counts.
- Common-mode noise is neglected.
- Signal can optionally be added, either for tracks passing perpendicularly through the detector, or for inclined tracks.
  - For perpendicular tracks, which are assumed to deposit their energy on only one strip, the energy deposited is given by $\Delta_E = C[\lambda + K]$ [9], where λ is a random number from a Landau distribution. The parameters C and K are constants, whose values are chosen so as to obtain a signal to noise ratio of 10.4 and a signal distribution of width similar to that seen in test-beams [10].
  - For inclined tracks making an angle θ to the normal to the detector, the energy deposited on each strip across which the track passes, is given by $\Delta_E = C(d/t)[\lambda + K + \log_e(d/t)]$ [9], where C and K are the same constants used for perpendicular tracks, d is the length of the track's path through the silicon above a given strip and t is the thickness of the silicon. The strip pitch p was set to $0.244t$, corresponding to the outer-most barrel layer. This layer has the widest (and hence most difficult to reconstruct) clusters, for a track of given P_t. Fig. 1 illustrates the dependence of the cluster pulse height distribution on track crossing-angle.
- A small fraction (6% in accordance with test-beam data) of the charge on each strip is moved to each of its immediate neighbours, to take into account capacitive coupling.
- Finally, a clustering algorithm is applied to the simulated strips.

4 Typical Cluster Characteristics

This section presents some basic characteristics of reconstructed clusters in minimum bias events, as obtained from the ORCA simulation. The aim of this is just to give a 'feeling' for what the FED is dealing with. High luminosity conditions are assumed, which implies a mean of 24 superimposed events per bunch-crossing. The clusters were reconstructed using the FED clustering algorithm "FED 2" recommended in Section 8.

Figure 2 shows the transverse momentum P_t with respect to the beam-axis of the track which produced each genuine, reconstructed cluster. Most clusters are produced by tracks with $0.05 < P_t < 1$ GeV/c. Tracks with $P_t < 0.65$ GeV/c can spiral inside the tracker and so produce many entries in this figure. A small fraction of clusters are produced by very low momentum ($P_t \approx 0.5$ MeV/c) tracks, which are presumably delta-ray electrons produced in the silicon. The fraction of clusters produced by the more "interesting" tracks, defined as those with $P_t > 1$ GeV/c is only 4%.



Track crossing angle of 0−20 degrees

Track crossing angle of 60−70 degrees

Figure 1: Comparison of simulated total cluster pulse height for two different track crossing-angles.

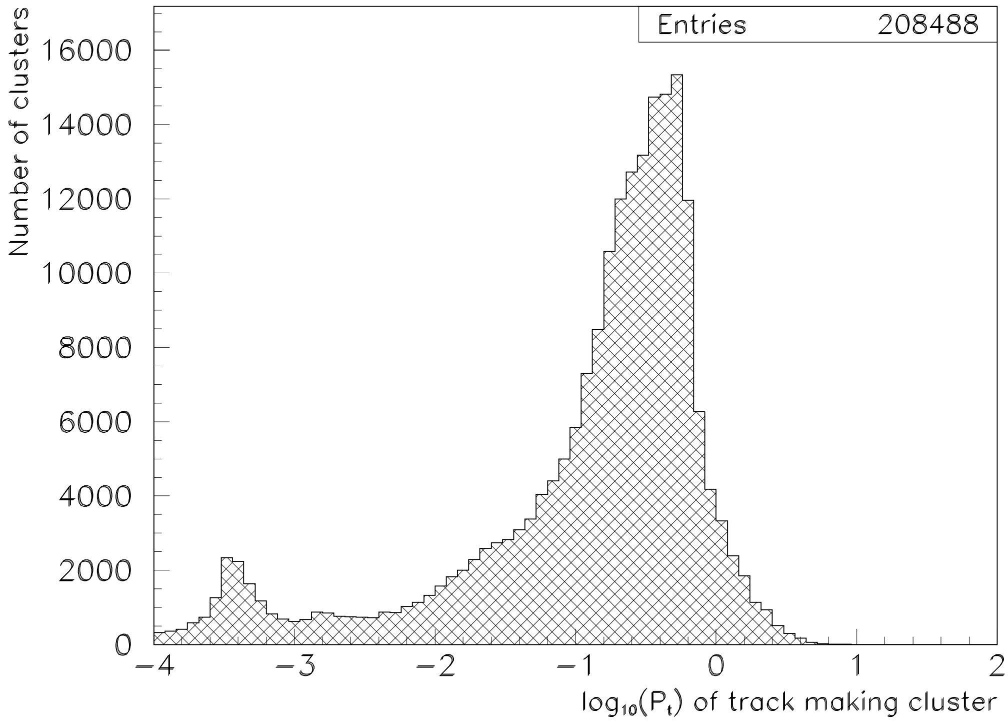


Figure 2: For every genuine, reconstructed cluster in the tracker, this shows the transverse momentum with respect to the beam-axis of the track which produced it.

Figure 3 shows the time at which genuine, reconstructed clusters are produced relative to the nominal readout time. Many clusters are produced near the nominal readout time. However, a significant fraction are produced several nanoseconds late, by low momentum, spiralling tracks. When the APV is operating in deconvolution mode, its good time resolution suppresses clusters from the wrong bunch-crossing: only 23% / 2.9% of genuine, reconstructed clusters are produced in the previous / following bunch-crossings respectively. (The asymmetry in these two numbers arises from clusters produced by spiralling tracks.)

Figure 4 shows the width in strips of reconstructed clusters. It shows *prompt* clusters (i.e., those produced within 5 ns of the nominal beam-crossing time) in two separate transverse momentum ranges: $P_t > 1$ GeV/c (useful for many physics analyses) and $1 > P_t > 0.5$ GeV/c (useful for a few physics analysis [11]). It also shows the other clusters which are produced in the correct beam-crossing but are of no physics interest, since they either have $P_t < 0.5$ GeV/c or are not prompt (i.e., they have spiralled around the tracker). Finally, it shows the widths of clusters reconstructed in the wrong bunch-crossing, and also of fake clusters (i.e., those containing no strips on which a track deposited energy).

Prompt clusters produced by the tracks which are most interesting for physics analysis ($P_t > 1$ GeV/c) usually have a width of 1–3 strips. However, it is desirable for some physics analysis to reconstruct tracks with momenta as low as 0.5 GeV/c, which implies that the FED should be able to reconstruct clusters with a width of 1–5 strips. This statement can be checked by a simple calculation. – The angle θ to the normal of tracks crossing silicon

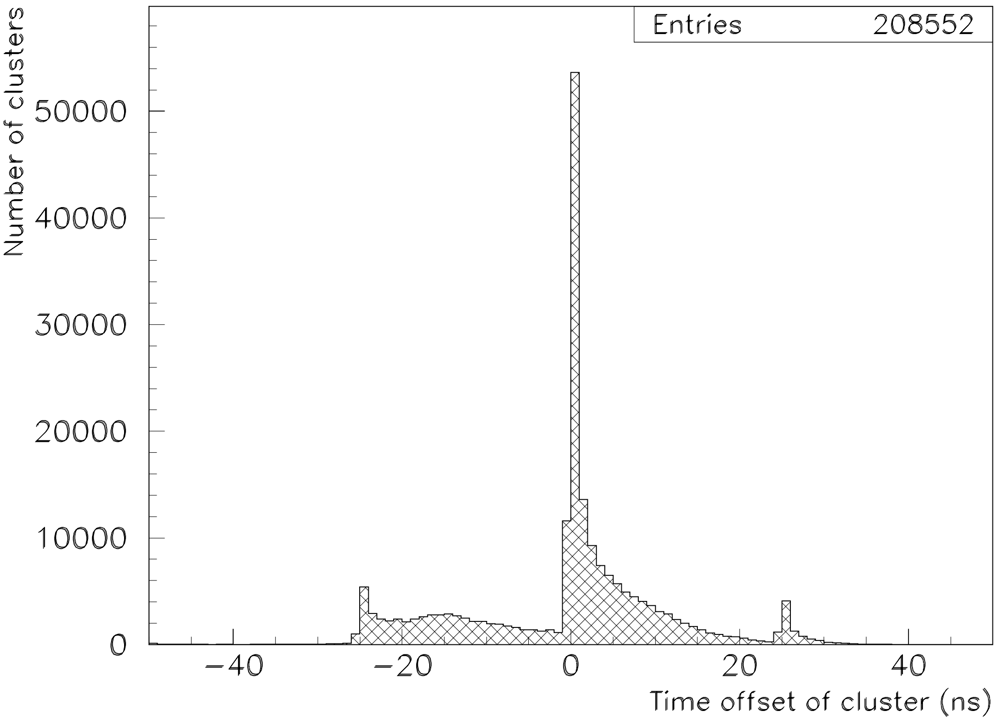


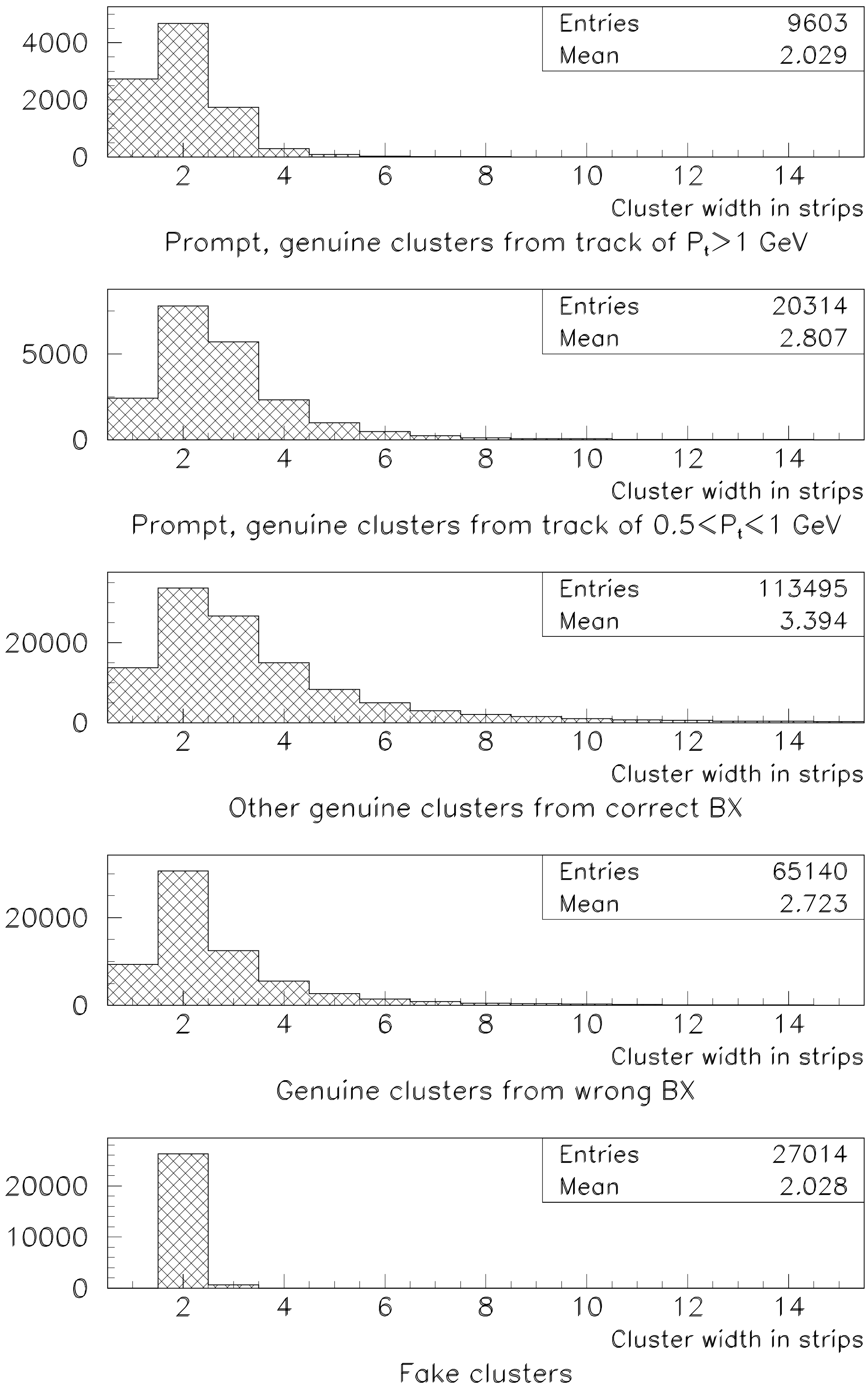
Figure 3: For every genuine, reconstructed cluster this shows the time it was produced relative to the nominal readout time, in high luminosity p–p collisions.

wafers in the barrel is given by $\theta = \arcsin(3RB/20P_t)$, where $B = 4$ T is the magnetic field, R is the distance in meters of the wafer from the beam-axis and P_t is measured in GeV/c. Fig. 5a shows a plot of θ versus P_t for tracks crossing the outer layer of the silicon barrel at $R = 1.08$ m. Knowing that detectors in this layer have a thickness of 500 μm and a pitch of 122 μm, one can calculate the number of strips across which each track passes as a function of its crossing-angle. This is shown in Fig. 5b. From these two figures, it is apparent that a track with $P_t = 1$ GeV/c crosses this layer at an angle of $\theta = 40°$ and can induce signals on up to 4 strips, (even neglecting capacitive coupling to additional strips).

5 Cluster Finding

The aim of this section is to propose a cluster finding algorithm which could be used by the FED. It will be assumed that the noise and pedestal of each strip have previously been determined in a calibration run (discussed in Sect. 2) and that the FED has subtracted the pedestals. It will also be assumed that for each event, the FED has perfectly determined the common-mode offset of each APV (discussed in Sect. 6) and subtracted that too.

Section 5.1 examines cluster finding algorithms used by other experiments and goes on to propose algorithms which might be used in the FED. In Sect. 5.2.1, the efficiency of suggested algorithms is studied using both ORCA and the Toy Monte Carlo. Section 5.2.2



Prompt, genuine clusters from track of P_t>1 GeV

Prompt, genuine clusters from track of 0.5<P_t<1 GeV

Other genuine clusters from correct BX

Genuine clusters from wrong BX

Fake clusters

Figure 4: In high luminosity p–p collisions, the width in strips of reconstructed clusters is shown, divided up into various categories.

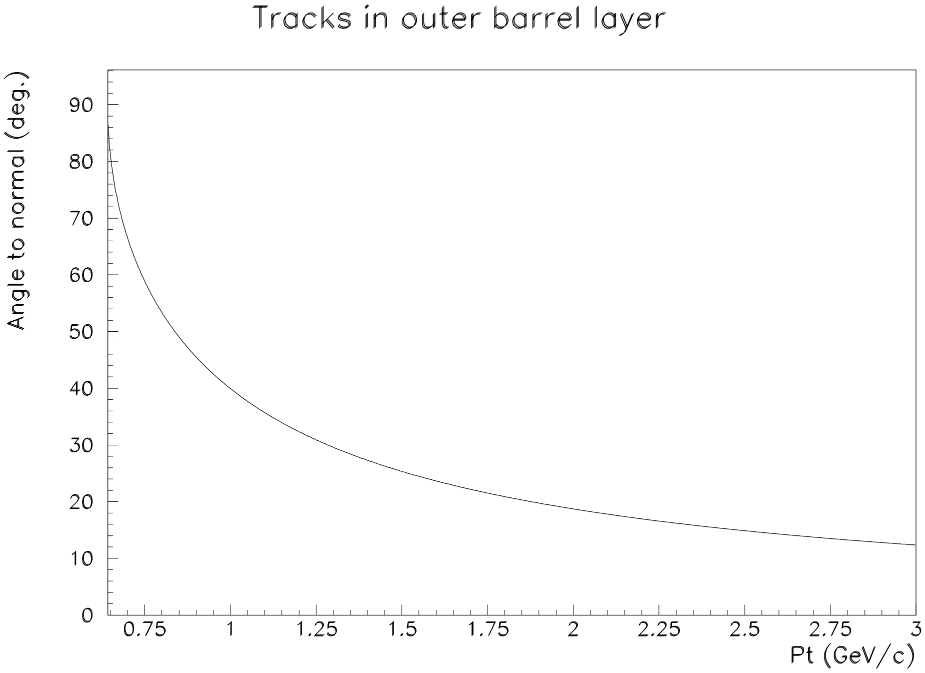


Figure 5a: Angle to normal of tracks crossing outer barrel layer as a function of their transverse momentum to the beam-axis.

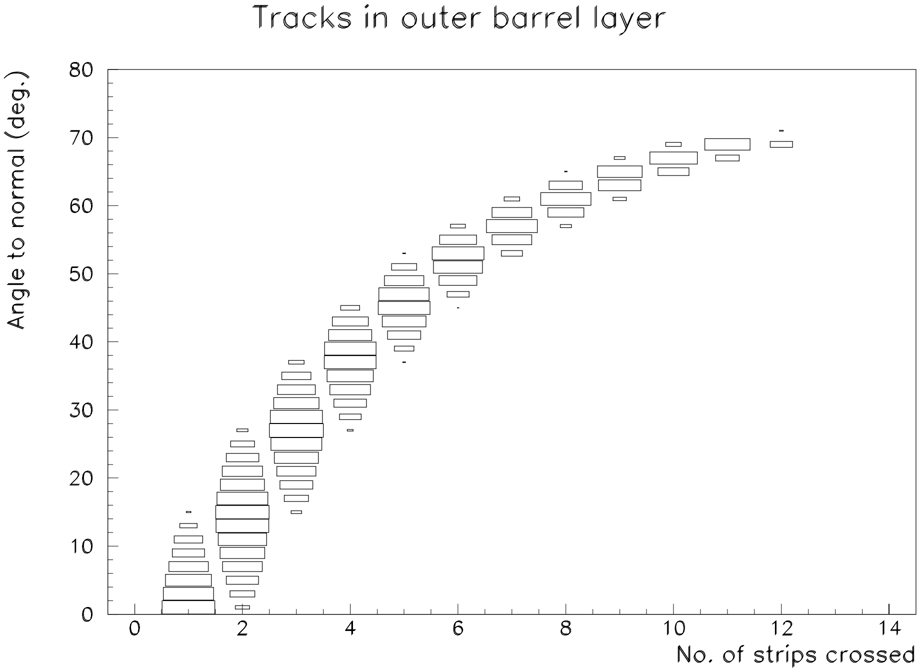


Figure 5b: Angle to normal of tracks crossing outer barrel layer as a function of the number of strips each track crossed in the detector.

examines the ghost rates.

5.1 Consideration of Possible Clustering Algorithms

5.1.1 Algorithms used Online by other Experiments

- **CDF** and **D0** use a simple approach: each strip over threshold is output, together with its immediate neighbour on either side, which improves positional resolution through charge sharing [12, 13].

- **ALEPH** applies the same algorithm as D0, but the low data rates allow it to output seven nearest neighbours on either side of hit strips [14].

- **DELPHI** accepts clusters if they contain n strips and have a total signal to noise ratio (summing over all the strips in the cluster, i.e. $\sum_i S_i/N_i$) exceeding $4\sqrt{n}$. The number of strips n is required to be in the range 1–3. Like CDF, D0 and ALEPH, it also outputs some neighbours of the cluster [15].

- **OPAL** adds the signals on two neighbouring strips, and accepts both strips if the sum exceeds three times the noise [16].

The CDF/D0 algorithm seems feasible for the CMS tracker, but outputting many neighbouring strips as ALEPH does would probably lead to an unacceptably high data volume. The DELPHI algorithm involves floating point division, which renders it less attractive, considering that it must be performed quickly in an FPGA. The OPAL algorithm is possible, but as it only considers pairs of strips, it may produce wider clusters than necessary.

5.1.2 Proposed Algorithms for use in the FED

Three clustering algorithms are proposed here and studied below, together with the off-line TDR algorithm [2] which acts as a reference.

When proposing algorithms for use in the FED, one needs to bear in mind the limitations of the firmware-programmable FPGA chips in which they will be implemented. It is foreseen that any algorithm would simply loop over all the strips in an APV, and accept each one according to whether it and possibly also several of its neighbouring strips pass a set of logical requirements. These requirements should ideally avoid floating point multiplication or division, as these operations are expensive in terms of the numbers of gates required. Furthermore, one should reach a decision about whether to accept or reject a strip in a single pass, to reduce processing time and hence also memory requirements. Memory in the FPGA is limited: about two bytes of memory per silicon strip are available to store clustering thresholds. – The exact amount depending on how much is taken up in implementing other necessary functions in the FPGA.

1. **Algorithm FED 1**: A very simple one, which simply accepts all strips with $S/N > T$, where the threshold T was here set equal to 2.7.

2. **Algorithm FED 2**: Accepts all strips with $S/N > T_l$, but rejects single-strip clusters unless they pass a more severe requirement $S/N > T_h$. Here, $T_l = 2$ and $T_h = 5$. This is motivated by the fact that fake clusters are usually only one strip wide.

3. **Algorithm FED 3**: Similar to that used by CDF/D0: accepts all strips with $S/N > T_h$ and their immediate neighbour on either side if that neighbour has $S/N > T_l$. Here, $T_l = 0$ and $T_h = 3$.

4. **Algorithm TDR**: The Tracker TDR off-line algorithm: requires clusters to contain a seed strip with $S/N > 3$, all other strips in the cluster to have $S/N > 2$ and the cluster as a whole to satisfy $S_{tot}/N_{\mathrm{rms}} > 5$, where S_{tot} is the total charge and N_{rms} is the rms noise of the strips.

When deciding if the pulse height on a given strip exceeds a threshold T, rather than checking if $S/N > T$, which would demand a floating point division, it would be preferable to check if $S > T \times N$, where the product $T \times N$ is passed to the FED as a calibration constant. Since the noise can vary from strip to strip, this would imply that the FED must have one or two registers for each strip it processes, to contain the clustering thresholds. This would also have two additional advantages:

- Different thresholds could be used on the various strips processed by a FED, which may be useful if it reads detectors from different regions of the tracker.
- Noisy strips could be suppressed by setting their thresholds to infinity.

The above algorithms are all capable (at least in principle) of finding very wide clusters. Such clusters may be of little interest for physics, since they will usually only be produced by low P_t tracks in the outermost detector layers. If they were not reconstructed, one could imagine that the effect on physics performance might be small. N.B., This statement has not been proven. With all these algorithms, one could therefore imagine using an additional cut on maximum cluster width to reduce the occupancy (and hence data rate). This possibility is considered in Sect. 7.1.

5.2 Cluster Finding Performance

As cluster finding performance is particularly sensitive to inaccuracies in the simulation, it is studied using both ORCA and the Toy Monte Carlo.

5.2.1 Cluster Finding Efficiency

When studying the efficiency, two alternative definitions have been used for what constitutes a reconstructed hit:

- A loose definition, stating that a hit is reconstructed, if a cluster is found containing at least one of the strips which was crossed by the particle track.

- A strict definition, requiring that all of the strips fully crossed by the track be reconstructed in a single cluster.

According to both ORCA and the Toy Monte Carlo, all three FED algorithms yield a 'loose' cluster finding efficiency in excess of 99.5% for track crossing-angles of up to 60°. (The 'crossing-angle' is measured relative to the normal to the detector module, projected into the plane perpendicular to the strip direction). Hence hits are virtually always at least partially reconstructed. However, unless all strips fully crossed by a track are reconstructed in a single cluster, the reconstructed cluster position will be highly inaccurate. The 'strict' definition of efficiency is therefore most relevant.

Figures. 6a and b show the estimated strict efficiency as a function of crossing-angle using ORCA and the Toy Monte Carlo, respectively. Both plots show results from all four algorithms. The ORCA results were obtained for the outer barrel layer, where the ratio of strip pitch to wafer thickness is the same as that assumed in the Toy Monte Carlo. Within rather large statistical errors, the results from the two simulations seem comparable, which is reassuring. From the Toy Monte Carlo results, it is apparent that the FED 1 algorithm loses efficiency for fairly modest angles, whereas the other algorithms do much better. This would disfavour use of the FED 1 algorithm in the FEDs. The near equality of the TDR and FED 2 algorithms' performance is no coincidence: for clusters consisting either of a single strip or of three or more strips, the two algorithms are mathematically identical. The FED 3 algorithm performs best. At large crossing angles, many of the missed clusters are actually split in two by the clustering algorithms, because a strip with low pulse height in the middle of the cluster produces a 'hole'. Algorithm FED 3 is less sensitive to this effect, as its approach of outputting the neighbours of hit strips means that it can fill in most holes.

Figure. 7 shows three plots of the strict efficiency as a function of barrel layer obtained using ORCA. The plots show the efficiency to reconstruct clusters produced by tracks in three different transverse momentum (P_t) ranges. For $P_t \approx 1.2$ GeV/c, all algorithms give almost perfect performance. For $P_t \approx 0.8$ GeV/c (and to a lesser extent for $P_t \approx 1.2$ GeV/c), a drop in efficiency is apparent in the outer two barrel layers. This is partially due to the fact that the crossing-angle is very large in these layers for such low P_t tracks and partially due to the small ratio of strip pitch to wafer thickness, which results in clusters being spread over several strips. Algorithm FED 1 performs significantly worse than the others in these layers. However, none of the four algorithms performs well there, implying that it may be necessary to use looser cuts in these two layers. For the lowest momentum tracks $P_t \approx 0.5$ GeV/c, the efficiency is virtually zero in the outer two layers, simply because the tracks circle before reaching it. The efficiency is well below unity in several other layers of the outer barrel, particularly for algorithm FED 1. The layers with the lowest efficiency are those with the smallest ratio of strip pitch to wafer thickness. Note that the high efficiency in the innermost four layers should not be taken too seriously: it is strongly biased by the high S/N ratio of 15, assumed by ORCA in much of these layers.

5.2.2 Ghost Rates

Table 1 shows the mean ghost rate per strip obtained with each of the four algorithms. This is defined as the total number of strips associated to fake, reconstructed clusters divided by the

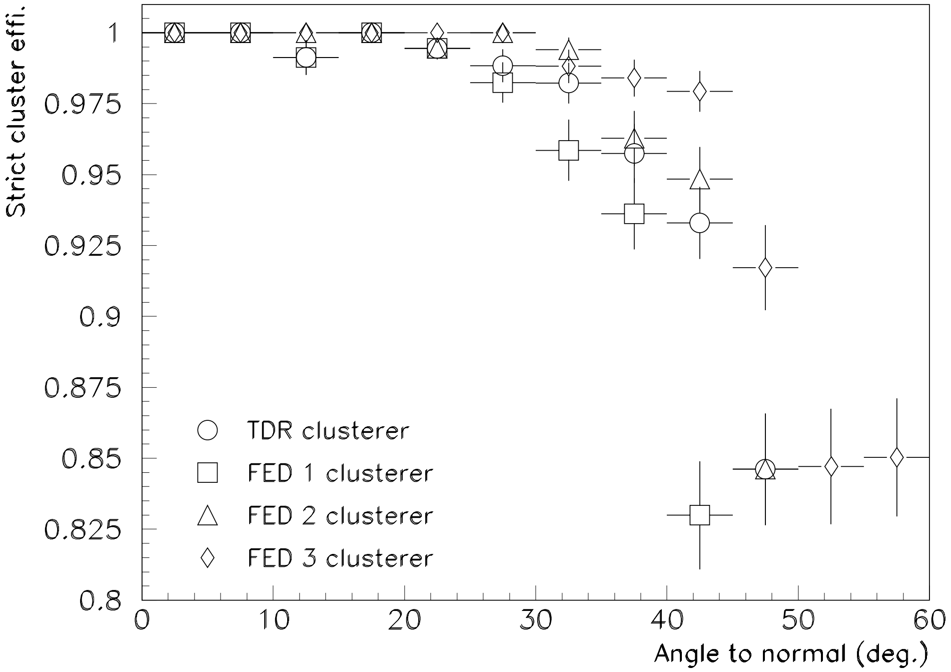


Figure 6a: ORCA prediction of strict cluster finding efficiency vs. crossing-angle for all four algorithms in the outer barrel layer.

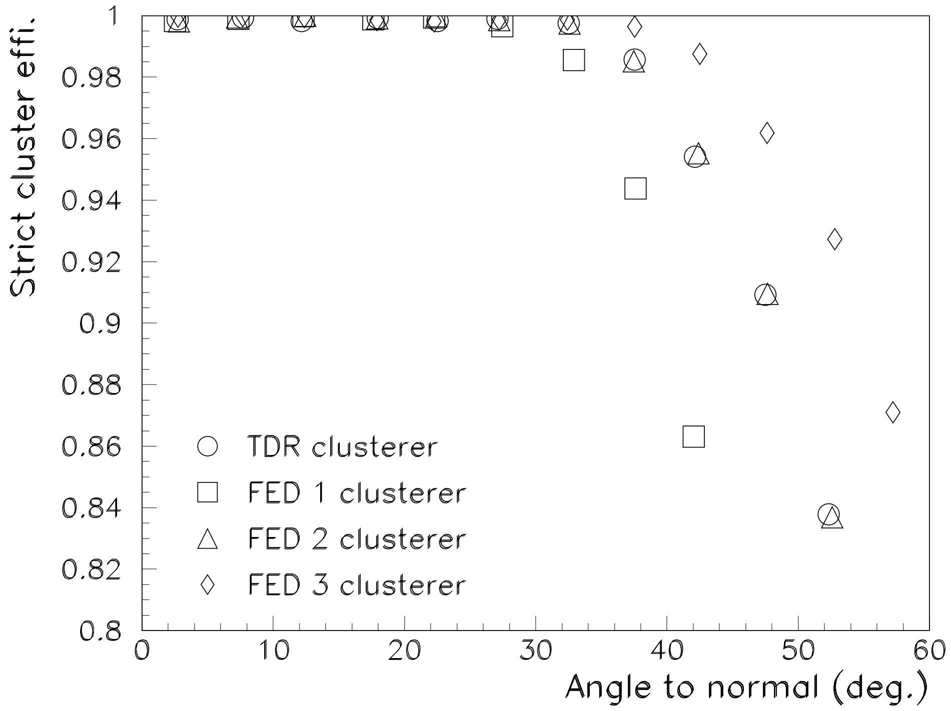
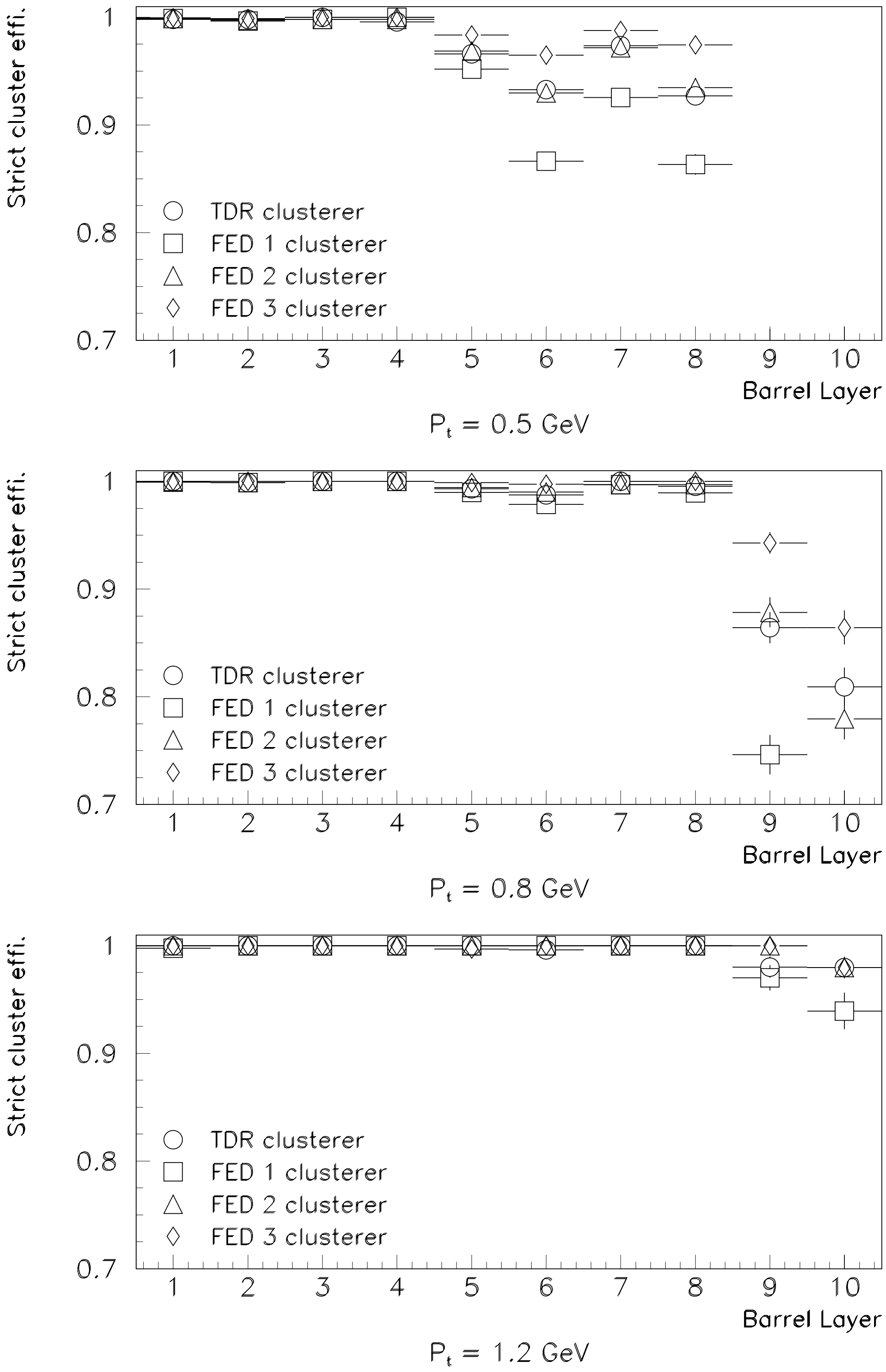


Figure 6b: Toy Monte Carlo prediction of strict cluster finding efficiency vs. crossing-angle for all four algorithms in the outer barrel layer.



P_t = 0.5 GeV

P_t = 0.8 GeV

P_t = 1.2 GeV

Figure 7: ORCA predictions for strict cluster finding efficiency vs. barrel layer in three P_t regions, for all four algorithms.

total number of strips in the detector. Predicted ghost rates from ORCA and the Toy Monte Carlo are in reasonable agreement.

Table 1: The mean ghost rate per strip for each cluster finding algorithm. Results obtained from both ORCA and the Toy Monte Carlo are compared.

Algorithm	Ghost rate (%)	
	ORCA	Toy Monte Carlo
FED 1	0.29	0.36
FED 2	0.09	0.11
FED 3	0.23	0.28
TDR	0.014	0.013

Although the FED algorithms all yield far higher ghost rates than the TDR algorithm, this is of little consequence for the DAQ. All one cares about is that the ghost rate should be small compared with the rate from genuine hits, and all three FED algorithms achieve this. Their higher ghost rates are a direct consequence of the loose cuts, which they use to obtain comparable efficiency to the TDR algorithm with simpler algorithms.

There is some variation in the ghost rate between the three FED algorithms. However, comparison of these values is not an useful way of judging their relative merits. A far more relevant quantity is the total strip occupancy they produce. This will be studied in Sect. 7.

Figure 8 shows the width in strips of fake clusters obtained from the four algorithms, as estimated with the Toy Monte Carlo. The simple FED 1 algorithm is dominated by clusters only one strip wide, which are neatly suppressed by the extra cut in the FED 2 algorithm. This allows the latter to use looser cuts for wide clusters. The FED 3 algorithm can produce rather wide fake clusters, as a result of its tendency to output neighbouring strips.

6 Estimating the Common-Mode Offset

The results on cluster finding performance presented in Sect. 5.2 assumed that the common-mode offset can be perfectly estimated and subtracted, It reality, it is not possible to achieve this using a simple algorithm in the FED. This section considers what might practically be achieved.

6.1 Algorithms used Online by other Experiments

D0 neglects common-mode noise [13], which would represent a gamble. ALEPH uses a rather sophisticated histogramming algorithm to evaluate it [14], which can't easily be implemented in the FED. CDF and OPAL assume that the common-mode offset is equal to the median pulse height, which is less biased upwards by strips containing signals than the mean pulse height would be [12, 16]. This algorithm may work well in the FED.

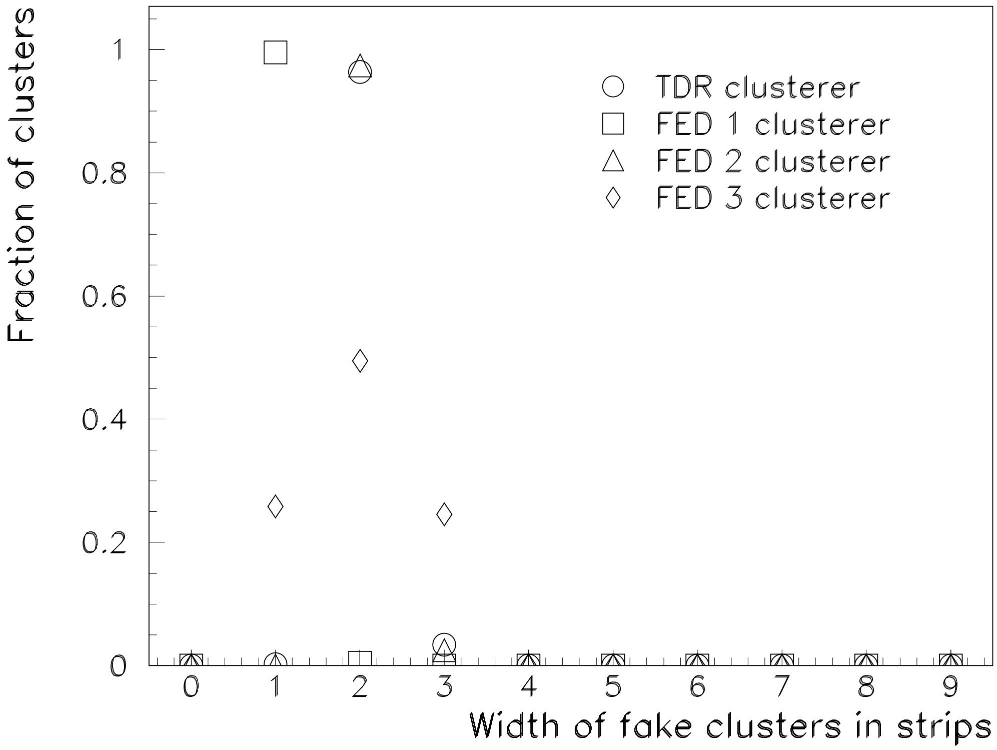


Figure 8: Width of fake clusters in strips for all four algorithms.

6.2 Proposed Algorithms for use in the FED

Two basic algorithms have been considered:

- **The 'mean' algorithm**

  1. Estimate common-mode offset for each APV in the event, by setting it equal to the mean pulse height on its 128 strips.
  2. Flag strips which have $S/N > T_c$, since these are likely have a signal and this will bias upwards the common-mode estimate. For the study presented below, the threshold $T_c = 2$.
  3. Repeat step (1), ignoring flagged strips.

  Performing several iterations of this algorithm will yield more accurate results, but will be prohibitive in terms of the time used. In the study presented here, results are presented for only 1 iteration (i.e., step (1) performed once) and 2 iterations (i.e., step (1) performed twice).

- **The 'median' algorithm** This simply estimates the common-mode offset for each APV in the event, by setting it equal to the median pulse height on its 128 strips. This is less strongly biased by signals or noisy strips than the mean, so no attempt is made to reject these. An FPGA algorithm for calculating the median has been proposed [17], which is similar in complexity to the mean algorithm with two iterations.

Since the FED clustering algorithms presented in Sect. 5.1 use thresholds of a few times the rms noise, it seems likely that if the bias in the common-mode offset (defined as the estimated minus the true common-mode offset) exceeds the rms noise, the cluster finding efficiency will suffer. Figure 9a shows the probability of this happening as a function of barrel layer, when using each of the above algorithms for estimating the common-mode offset. If only one iteration of the mean algorithm is performed, then in the innermost layers, where the occupancy is highest, several percent of the APVs have a significant bias in their estimated common-mode offset. It seems probable that this would result in a loss of cluster finding efficiency. However, performing either two iterations of the mean algorithm or alternatively using the median algorithm both eliminate this problem. In the few cases where the median algorithm goes badly wrong, this is usually a result of delta ray electrons travelling in the plane of the silicon and depositing their energy on many tens of strips.

Figure 9b shows the same thing again but for simulated Pb–Pb collisions. As no heavy ion Monte Carlo was available, this plot was obtained by superimposing 250 minimum bias p–p events. This yields roughly half the tracker strip occupancy conservatively expected in Pb–Pb collisions [18], and therefore represents a rather optimistic (but not implausible) scenario. Figure 9b shows that provided that two iterations of the mean algorithm are performed or the median algorithm is used, it may be possible to estimate the common-mode offset in Pb–Pb events, especially in the outermost layers. However, based on the expected trigger rate during Pb–Pb runs [18], this does not appear to be essential: it should be possible to switch off the FED zero suppression and output pulse height information for all strips. Nonetheless, zero suppression may be used in lighter ion collisions, as these yield occupancies at least a factor three lower than those in Pb–Pb collisions, but demand higher trigger rates.

Although the statistical precision of the ORCA studies is limited, one can obtain some confirmation of these ideas in Fig. 10. This shows, for minimum bias events, the dependence of the strict cluster finding efficiency on barrel layer, in three transverse momentum ranges. The efficiencies are displayed for the FED 2 algorithm, either using one of the above common-mode correction algorithms or simply taking the common-mode offset from its simulated value. The two more complex algorithms significantly improve the efficiency, especially for $P_t < 1$ GeV/c.

7 Strip Occupancy and Data Format

7.1 Strip Occupancy

The strip occupancy in the tracker (defined as the fraction of strips associated to reconstructed clusters) can be directly related to the data rate flowing to the DAQ. It is therefore as important as the cluster finding efficiency, when deciding which clustering algorithm to use in the FEDs.

Figure 11 shows the dependence of the strip occupancy on barrel layer and on end-cap disk. The clusters were reconstructed using the FED 2 algorithm, with the common-mode offset being estimated with the median algorithm. The occupancy reaches 3.0% in the innermost barrel layer.

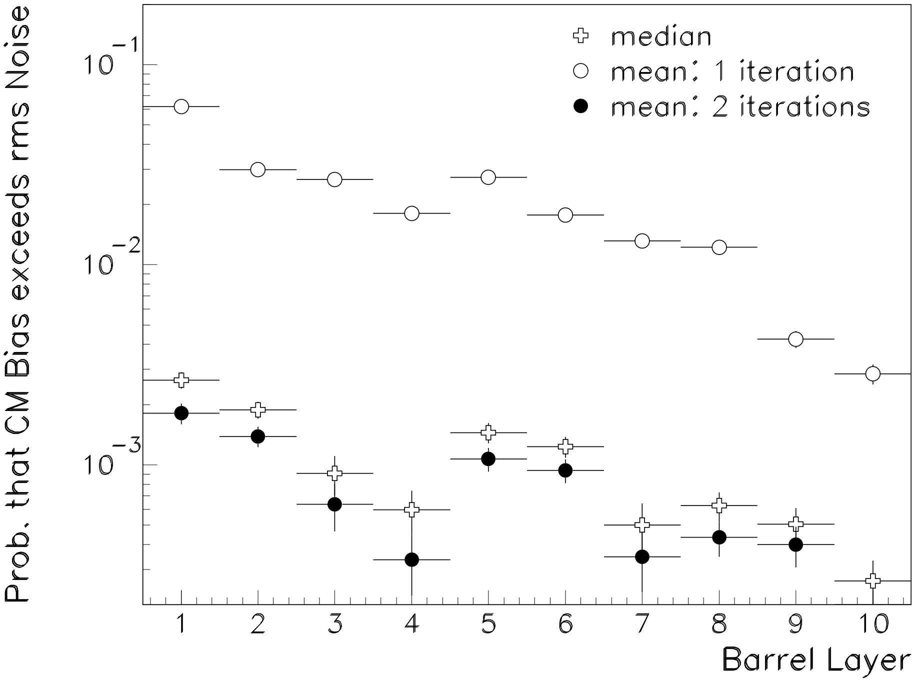


Figure 9a: For high luminosity p–p collisions, the probability that the estimated common-mode offset exceeds its true value by more than the rms noise is shown, as a function of barrel layer, using the various common-mode correction algorithms.

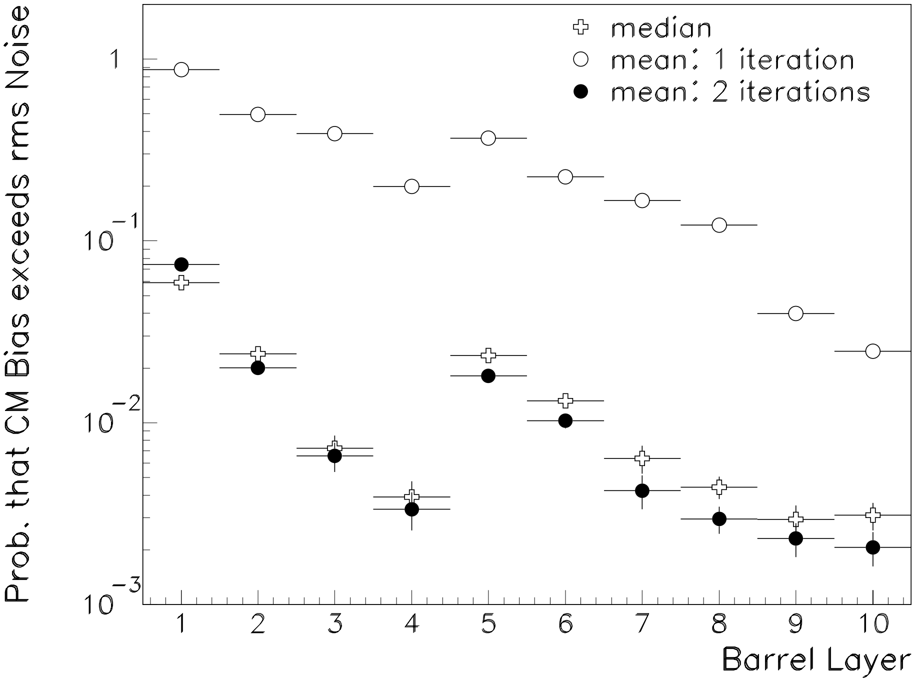
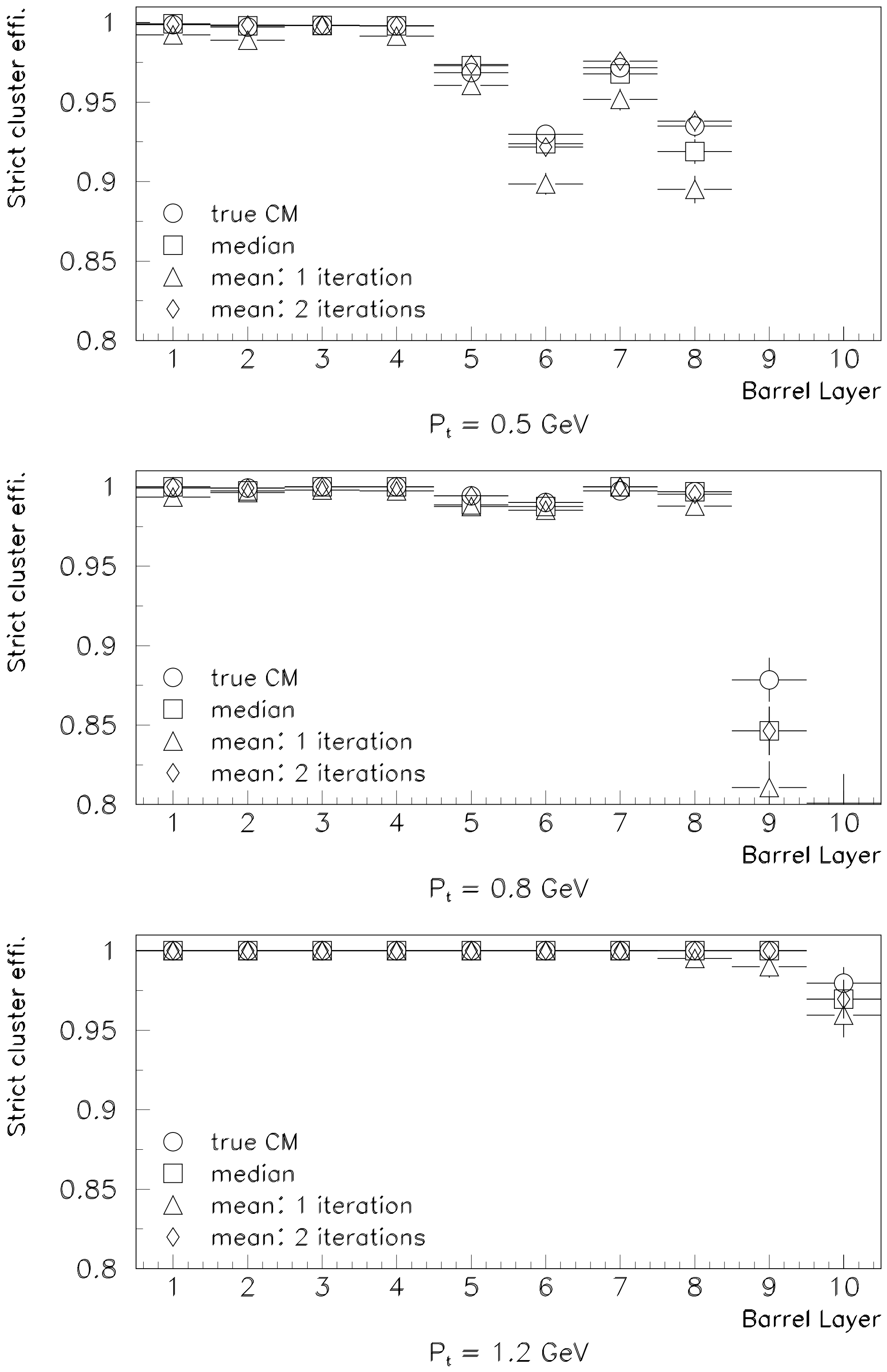


Figure 9b: Same again, but for an optimistic simulation of Pb–Pb events.


P_t = 0.5 GeV

P_t = 0.8 GeV

P_t = 1.2 GeV

Figure 10: ORCA predictions in high luminosity p–p collisions, for strict cluster finding efficiency vs. barrel layer in three P_t regions, for the FED 2 algorithm. Results for each common-mode correction algorithm are given.

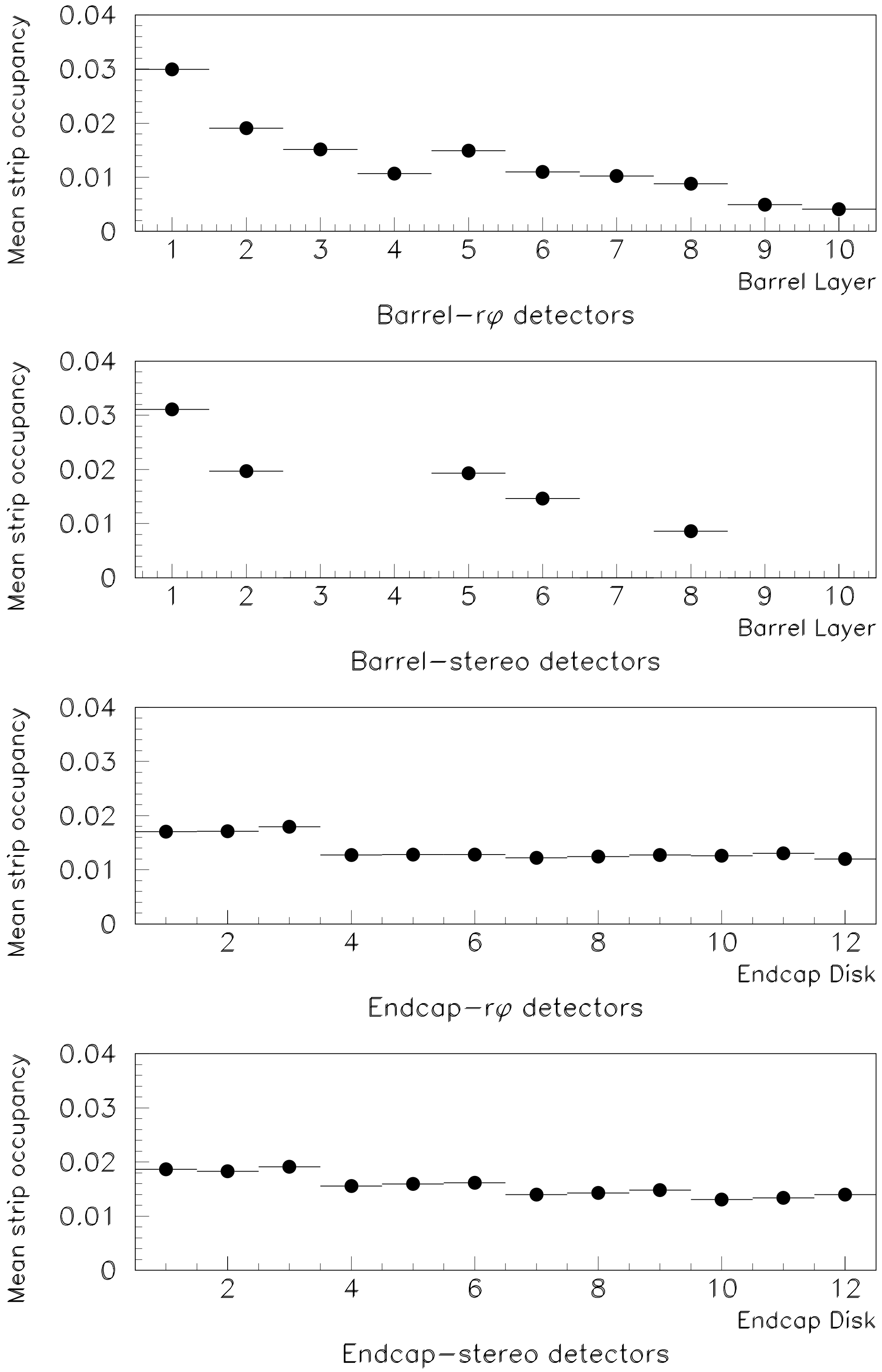


Figure 11: Expected strip occupancy in FED data, from minimum bias events during high luminosity p–p running.

The strip occupancy depends strongly on the clustering algorithm. Table 2 shows its mean value averaged over the barrel for the algorithms considered this note. This table also reveals the relative weight of different contributions to the occupancy. The FED 2 algorithm achieves a slightly lower occupancy than the FED 1 algorithm. The FED 3 algorithm performs much worse, which disfavours its use in the FEDs. The reason for this is that as it outputs the neighbours of hit strips, the FED 3 algorithm produces much wider clusters. This statement is confirmed by Table 3, which shows the mean number of clusters per detector module: the FED 3 algorithm is not guilty of finding more clusters than any other.

It is interesting to note that the strip occupancy is dominated by non-prompt clusters: either from spiralling low momentum tracks or from the wrong bunch-crossing. Such clusters look identical to prompt clusters, except that they have a poor signal to noise ratio due to the time-response of the APV. Unfortunately, this makes it virtually impossible to suppress them, whilst at the same time preserving a good efficiency for prompt clusters in a tracker with a low S/N ratio. The number of such clusters will scale with increasing luminosity. It will also increase if the tracker yields better S/N ratios than those pessimistically assumed for this study. It should therefore be noted that if the tracker has a good S/N ratio, all clustering cuts proposed in Sect 5.1.2 will need to be increased.

Table 2: The mean barrel strip occupancy in high luminosity p–p collisions for various clustering algorithms. The occupancy is subdivided into four separate contributions: (i) prompt clusters (i.e., those produced within 5 ns of the nominal beam-crossing time) from tracks with $P_t > 0.5$ GeV/c, (ii) other (uninteresting) clusters produced in the correct beam-crossing, (iii) clusters from the wrong beam-crossing, (iv) fake clusters.

Clustering Algorithm	Common-Mode Algorithm	Strip Occupancy (%)				
		Prompt $P_t > 0.5$ GeV/c	Other in correct BX	Wrong BX	Fake	Total
FED 1	true CM	0.13	0.70	0.34	0.29	1.46
FED 2	true CM	0.14	0.78	0.35	0.09	1.36
FED 3	true CM	0.21	1.00	0.52	0.23	1.96
TDR	true CM	0.14	0.78	0.32	0.01	1.25
FED 2	mean: 1 iter.	0.13	0.68	0.29	0.07	1.17
FED 2	mean: 2 iter.	0.14	0.75	0.34	0.10	1.33
FED 2	median	0.14	0.74	0.33	0.10	1.31

As mentioned in Sect. 5.1.2, one could reduce the occupancy further by applying a cut on maximum cluster width. This may have a detrimental effect on reconstruction efficiency and P_t resolution of low P_t tracks, although this has not been verified. It may also harm cluster finding efficiency in cases were several clusters have been produced close together, as the result of a particle decay for example, and have been merged by the FED clustering algorithm into a single wide cluster. Such clusters can often be resolved offline, so one would not wish to lose them. Nonetheless, if CMS does have problems with tracker data rates, it is interesting to discover what could be achieved by cutting on the maximum cluster width. This is shown in Table 4. In fact, the true reduction in data rates may be larger than shown, as with a maximum cluster width of 8, it would theoretically be possible to output the cluster width to the DAQ as a 3 bit number, instead of an 8 bit one.

Table 3: The mean number of clusters per barrel detector module in high luminosity p–p collisions found by various clustering algorithms. The number is subdivided into four separate contributions: (i) prompt clusters (i.e., those produced within 5 ns of the nominal beam-crossing time) from tracks with $P_t > 0.5$ GeV/c, (ii) other (uninteresting) clusters produced in the correct beam-crossing, (iii) clusters from the wrong beam-crossing, (iv) fake clusters.

Clustering Algorithm	Common-Mode Algorithm	Number of Clusters / Detector				
		Prompt $P_t > 0.5$ GeV/c	Other in correct BX	Wrong BX	Fake	Total
FED 1	true CM	0.35	1.45	1.21	1.83	4.84
FED 2	true CM	0.34	1.30	0.78	0.29	2.71
FED 3	true CM	0.34	1.28	0.93	0.75	3.30
TDR	true CM	0.34	1.27	0.67	0.04	2.32

Table 4: Effect on mean tracker strip occupancy of rejecting clusters above a certain width. Results are presented for the FED 2 clustering algorithm with common-mode noise subtracted using the median algorithm.

Maximum Cluster Width	Reduction in Occupancy (%)
5	27
8	15

7.2 Data Format

In the Tracker TDR [2] it is proposed that for each strip selected by the FED, both the strip address and the ADC pulse height should be output to the DAQ. This corresponds to a total of $2 \times N_{strip}$ bytes of information per cluster, where N_{strip} is the number of strips in the cluster. An alternative would be to output the address of the first strip in a cluster, the cluster size and the ADC pulse height of each strip in the cluster. This requires $2 + N_{strip}$ bytes. This leads to a larger data volume if most clusters are one strip wide, but a smaller data volume if many clusters are three or more strips wide.

Figure 12 shows the mean data size per APV obtained using these two possible data formats, as a function of barrel layer or end-cap disk. The clusters were found use the FED 2 algorithm, with the common-mode offset being estimated with the median algorithm. The results suggest that using the second data formatting scheme leads to 15% reduction in data volume. This confirms the results of a study by G. Pasztor [20]. One should note however, that the veracity of this statement is dependent on the clustering algorithm and thresholds used.

A data compression algorithm such as “Huffman” might be used after cluster finding to reduce data volume by a further $\approx$ 23%, at the cost of extra complexity [20].

Further information on expected occupancies and data rates, together with an estimate of the total number of FEDs required in the tracker, can be found in Ref. [19]. This includes information on occupancies in heavy ion collisions.

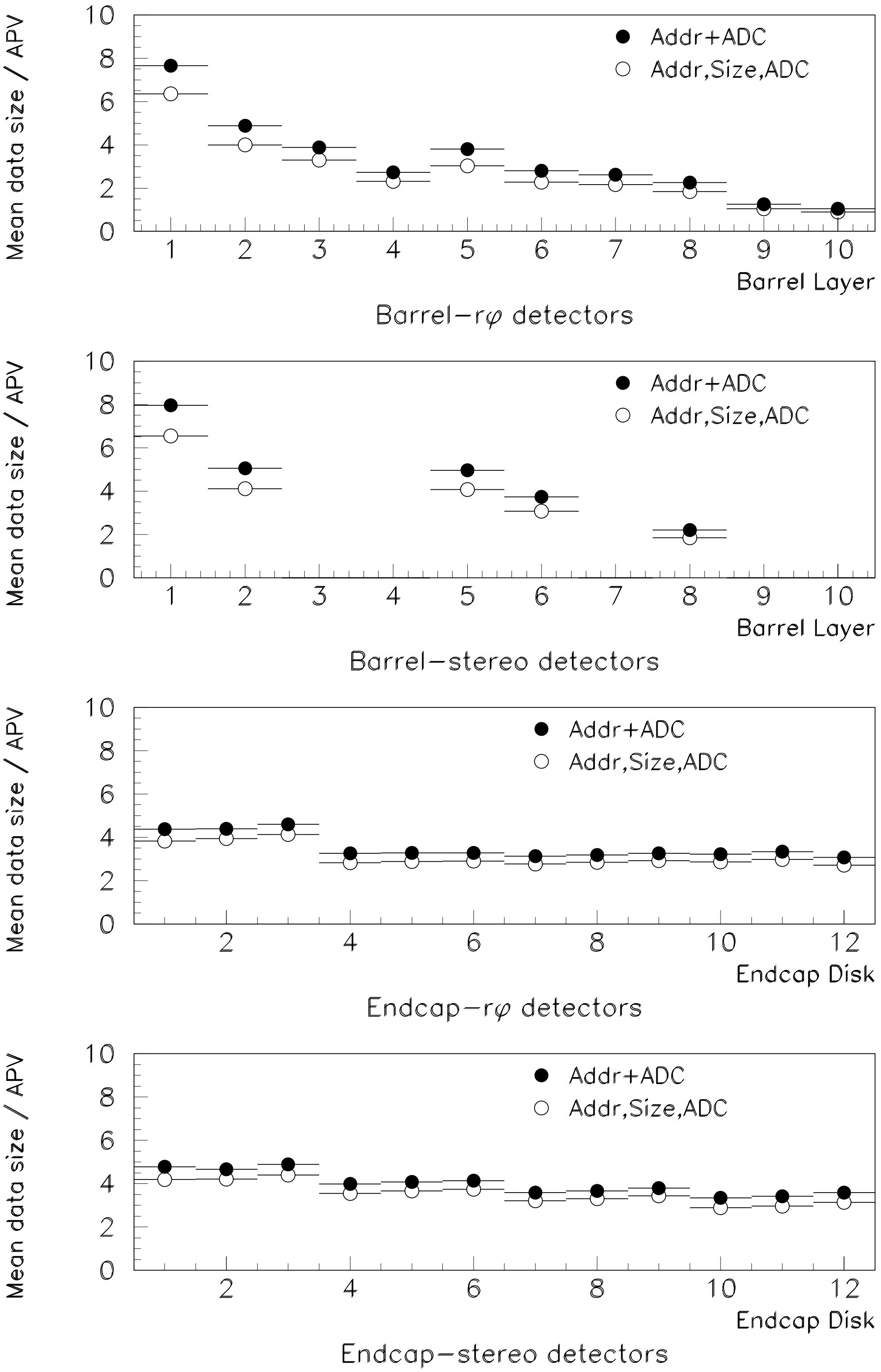


Figure 12: Mean data size per APV using two alternative data formats. Clusters were found using the FED 2 algorithm. High luminosity p–p conditions were assumed.

8 Conclusions

Considering the clustering algorithms proposed in Sect. 5.1.2:

The FED 1 algorithm gives excellent performance for tracks with $P_t > 1.2$ GeV/c. However, at lower momenta, its cluster finding efficiency is inadequate, especially in the outer detector layers. Nonetheless, it is possible that some of this efficiency might be recovered by using looser cuts in the outer layers, and it should be borne in mind that most physics analyses do not need such low momentum tracks. One should also remember that a pessimistic value was assumed for the detector S/N ratio: doubtless, the FED 1 algorithm would give much better performance if the S/N ratio in the final tracker were really 13–15 as seems likely.

Nonetheless, the FED 2 algorithm gives significantly higher efficiency and somewhat lower occupancies. It would therefore be preferable to implement this in the FED if FPGA resources permit this. The clustering algorithm could be implemented in the FED by looping through the strips sequentially, and deciding if a strip should be kept, by checking if it and its immediate neighbours exceed the requested thresholds.

The FED 3 algorithm gives excellent cluster finding efficiencies, but is disfavoured on the basis of the high occupancies it yields.

It should be noted that if the S/N of the tracker is better than that assumed here, all cluster finding thresholds will need to be tightened to reduce occupancy from out-of-time clusters. It should also be noted that better perfomance, than that shown in this note, might be achievable by using different clustering thresholds in different regions of the tracker.

Considering the common-mode noise algorithms proposed in Sect. 6.2:

It is apparent that the median algorithm performs well. Alternatively, two iterations of the mean algorithm may be used, but in this case, it may be necessary to suppress even moderately noisy strips during the common-mode determination, as these add a large statistical uncertainty to the mean. (Offline analyses are insensitive to this effect, as they weight the pulse heights according to the inverse noise squared when estimating the common-mode offset. Unfortunately, this would not be practicable in the FED.) As the common-mode offset is a floating point number, some thought must be given to whether it is truncated.

Regarding the format of the FED data discussed in Sect. 7.2:

It is probably not optimal to output the address and ADC count of each selected strip. Instead, it is better to output the address of the first strip in a cluster, the cluster size and the ADC pulse height of each strip in the cluster. This reduces the data volume by roughly 15%, with the exact reduction being dependent on the cluster algorithm and thresholds used.

Acknowledgements

I'm particularly grateful to Duccio Abbaneo for providing much of the code for the toy Monte Carlo and to Vitaliano Ciulli for providing much needed example software for ORCA tracker analysis.

References

[1] J. Coughlan *et al.*, *“Design of the Front-End Driver Card for CMS Silicon Micro-strip Tracker Readout”*, Proceedings of 6^{th} Workshop on Electronics for LHC Experiments, Krakow, CERN/LHCC/2000-041, p444.

[2] *“CMS Tracker TDR”*, CERN/LHCC 98-6 (1998)

[3] K.W. Bell *et al.*, *“FED User Requirements document”*, http://hepwww.rl.ac.uk/CMS_fed/Final_FED_User_Req/fed_urd.pdf .

[4] R. Halsall, private communication.

[5] *“The Tracker Layout Working Page”*, http://cern.ch/duccio/layout/lay.html .

[6] M. Mannelli, private communication.

[7] M. Raymond *et al.*, *“The CMS Tracker APV25 0.25 μm CMOS Readout Chip”*, Proceedings of 6^{th} Workshop on Electronics for LHC Experiments, Krakow, CERN/LHCC/2000-041, p130.

[8] The toy Monte Carlo was derived from an earlier version developed for MSGC studies by D. Abbaneo.

[9] G. Cowan, *“Statistical Data Analysis”*, Oxford Science Publications (1998), Chapter 2.9.

[10] S. Braibant *et al.*, *“Test-beam results on ⟨100⟩ silicon prototype detectors with APV6 front-end chip readout”*, CMS note 2000/050.

[11] Private communication from CMS b–τ group.

[12] M. Rehn *et al.*, *“Statistical Study of SVX3D Chip Dynamic Pedestal Subtraction Threshold Level”*, CDF/DOC/SEC_VTX/PUBLIC/4852 (1999).

[13] R. Yarema *et al.*, *“A beginners guide to the SVXIIE”*, FERMILAB-TM-1892, page 4 (1996).

[14] H. Seywerd, *“Sirocco programs for the VDET II”*, ALEPH note 97-109, page 3 (1997).

[15] V. Chabaud *et al.*, *“The DELPHI silicon strip microvertex detector with double sided readout”*, CERN-PPE/95-86, pages 9-10 (1995).

[16] P. Allport *et al.*, *“The OPAL silicon microvertex detector”*, Nucl. Instr. Meth. **A324** (1993) 34.

[17] M. French, W. Gannon, private communication.

[18] A. Racz *et al.*, *“Data Acquisition for Heavy Ion Physics”*, CMS IN 2000/027.

[19] A. Caner *et al.* *“On balancing data flow from the silicon tracker”*, (draft note) http://cern.ch/tomalini/readout.pdf (2001).

[20] G. Pasztor, *“Output data format and on-line data compression algorithms for the CMS Tracker FED”*, CMS-IN-2001-??? in preparation.