PHYSIK-DEPARTMENT

Real time data processing and feature extraction of calorimeter data in COMPASS

DIPLOMA THESIS
BY

Robert Konopka

Technische Universität München

2009-08-04

Table of contents

1 Introduction

The COMPASS experiment is already taking physics data for several years. Since the first technical run in 2001 the beam intensity and trigger rate is increasing from year to year for obtaining better statistics than before. All data have not only to be recorded once, but also to be stored several years for physical analysis. This lets the total amount of data growing. At the current stage of COMPASS, with trigger rates of $\approx 30\,\text{kHz}$ and average event sizes of $\approx 40\,\text{KB}$ in the 2009 run, it gets more and more necessary to attenuate the growth. Otherwise the current trigger rate could not be raised anymore.

Cinderella, the COMPASS online filter, [Nag05] provides the right framework for applying a preselection on the physics data to reduce the total amount to be stored. This Thesis will show different strategies how to reduce the amount of data from the detectors with the highest data load at COMPASS, the electromagnetic calorimeters. But basically Cinderella can handle data from every detector of the COMPASS spectrometer.

Cinderella implements its own high-performance decoding and data processing modules. They are not as complex as the offline processing tools, but they are up to a factor of 40 faster. Due to this Cinderella is able to take decisions in real time by applying a first analysis of the events.

Additionally Cinderella is used more and more for real time monitoring of important components of the COMPASS spectrometer. Information about trigger rate and performance is accumulated from the scalers by a Cinderella module and displayed on a screen in the COMPASS control room.

Since the hadron run in 2009 also a monitoring of the photomultiplier gains from the electromagnetic calorimeters is available. A stable operation of the calorimeters is especially required for the Primakoff data taking, where ECAL2 in particular is involved. With the help of the new monitoring capability it is possible to recognise potential problems before they get critical at the offline analysis stage.

But much more is possible, the current features just indicate the potential of the Cinderella framework. With the planned and partially applied DAQ upgrade in 2009, the COMPASS DAQ is ready for further Cinderella features from the CPU power point of view.

2 The COMPASS Experiment 2008/2009

COMPASS (**C**ommon **M**uon and **P**roton **A**pparatus for **S**tructure and **S**pectroscopy) is a fixed target experiment at the SPS (**S**uper **P**roton **S**ynchrotron) of CERN. The COMPASS physics program is split into the muon and the hadron program. In the muon programme a longitudinally polarised muon beam is used to analyse the spin structure of the nucleon in a polarised ^{6}LiD or NH_3 target. For the hadron programme positive and negative pion, kaon and proton beams are used with a target of liquid hydrogen or solid nuclear targets. The hadron programme started in 2008, after a shot pilot run 2004, with the spectroscopy of exotic mesons which do not fit into the standard quark model, like glue-balls, multi-quark systems and hybrids. These measurements will continue in 2009, but additionally to 2008 a Primakoff programme is planed to measure the π polarisability and chiral anomaly.[Ket09] [Col96]

2.1 The experimental setup

The COMPASS spectrometer consists of two parts. The upstream part is called large angle spectrometer (LAS) and the downstream small angle spectrometer (SAS). Each part is equipped with a spectrometer magnet (SM1 and SM2) and detectors for tracking, particle ID and calorimetry (see figure 1).

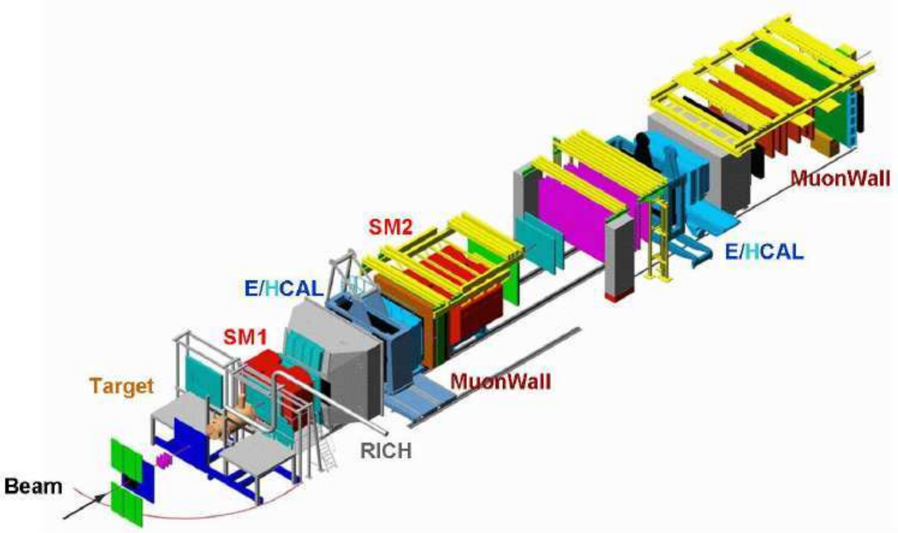


Figure 1. The COMPASS spectrometer

SM2 is the "stronger" magnet with a field integral of 4.4 Tm compared to SM1 with 1.0 Tm. Particles with lower momentum are supposed to be detected in the LAS part while high momentum particles can pass through into the SAS part. For this the "absorbing" detectors of the LAS part, like the calorimeters and the muon wall, have holes in their centres, matched to be covered by the SAS part. This feature is maximising the momentum acceptance of the COMPASS spectrometer. The target in 2008 was a 40 cm tube filled with liquid hydrogen. It was surrounded by the Recoil Proton Detector (RPD), consisting of two rings of scintillators, to detect protons "knocked" out of the target. For Primakoff measurements in 2009 the hydrogen target will be exchanged with a target build of solid lead disks. Incoming particles can be identified by two CEDARs (**CE**renkov **D**ifferential counter with **A**chromatic **R**ing focus) and are tracked by silicon micro-strip and scintillating fibre detectors in front of and after the target. Both magnets are surrounded by various tracking detectors with different resolution followed by the calorimeters and muon walls at the end of each sector. In the LAS sector additionally a RICH (**R**ing **I**maging **CH**erenkov counter) detector for particle identification in the momentum range of 5 – 44 GeV/c is available.

2.2 Physics goals in the Hadron run

The data taking run in 2008 was dedicated to the spectroscopy of exotic mesons produced by diffractive dissociation and central production. In 2009 additionally to the topics of 2008 a Primakoff run is planned.

Diffractive dissociation: Diffraction can be observed in the transfer of momentum if a particle passes the target very close. If the process is inelastic, the particle gets excited, it is called diffractive dissociation.

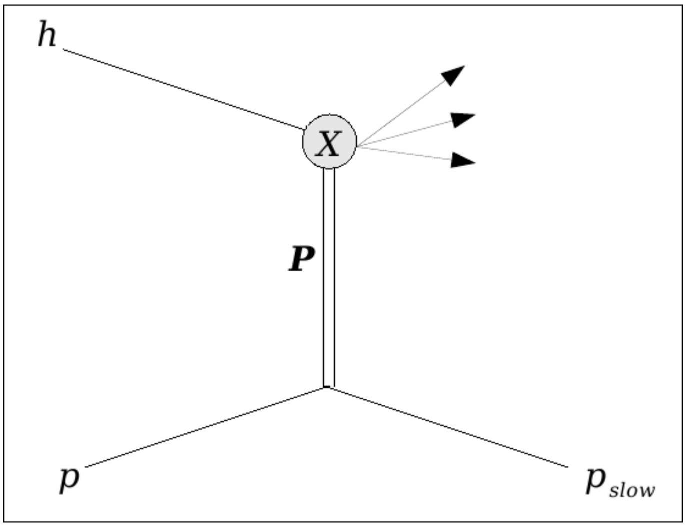


Figure 2. diffractive dissociation: excitation of an incoming hadron (h) by the exchange of a pomeron (P) with a proton (p). The resulting resonance is decaying immediately.

In figure 2 a possible case of diffractive dissociation is showed. An incoming hadron, in COMPASS a $\pi^-/\pi^+/K/p$, is getting excited by the exchange of a Pomeron with a proton from the hydrogen target. The resulting resonance is supposed to be an exotic meson which can be reconstructed by its decay product.

Central production: Another possibility to produce exotic mesons is by central production. It is similar to diffractive dissociation with not exciting the beam particle but inducing a collision between two pomerons.

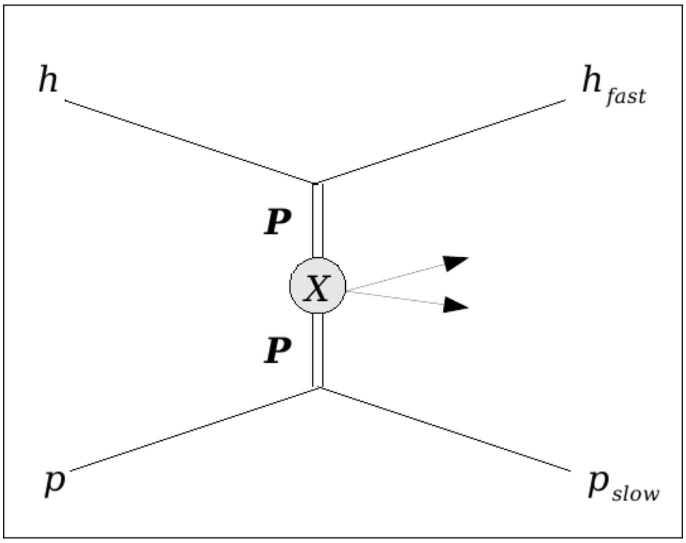


Figure 3. central production: double pomeron exchange (P) between the beam hadron (h) and the target proton (p). Particle X is produced by the collision of the two pomerons.

Exotic Mesons: In the quark model mesons are characterised by their quantum numbers J^{PC}, where J is the total angular momentum, P the parity and C the charge conjugation parity. P and C are given by

$$P = (-1)^{L+1}$$
$$C = (-1)^{L+S}$$

with L, the orbital angular momentum, and S, the total intrinsic spin ($S = 0, 1$). Mesons, consisting of a $q\bar{q}^{'}$ pair, may have in the quark model only specific J^{PC} quantum numbers, e.g. $J^{\mathrm{PC}} = 0^{-+}, 0^{++}, 1^{--}, 1^{++}, 1^{+-}, ...$. Exotic mesons in contrast to the mesons from the quark model, are mesons which either do not consist of one $q\bar{q}^{'}$ pair or have quantum numbers not allowed by the quark model. Since there are no specific assumptions except of the colour neutrality of hadrons, it is possible to compose mesons not only of $q\bar{q}^{'}$ pairs, but also of other colour neutral configurations like gluons, four-quark objects and hybrids, consisting of quarks and gluons, where the gluons contribute to the quantum number of the hadron. These exotic mesons may have quantum numbers not allowed by the quark model like, $J^{\mathrm{PC}} = 0^{--}, 0^{+-}, 1^{-+}, 2^{+-}, 3^{-+}, ...$. Finding particles with one of these “forbidden” quantum numbers would prove the existence of exotic mesons. [Gro08]

Primakoff reactions: In the hadron pilot run 2004 first Primakoff measurements have been made. 2009 the Primakoff programme will be continued with an improved electromagnetic calorimeter, which is the main part of the Primakoff trigger, and a new target, consisting of 16 0.125mm thick lead disks. The aim of the Primakoff measurements is to learn about the chiral anomaly and the electric and magnetic polarisabilities $\bar{\alpha}_{\pi}$ and $\bar{\beta}_{\pi}$ of the pion. With the help of the CEDARs it is possible to distinguish between pions and kaons in the beam. So it is possible to perform the same measurement also for kaons.[Col96]

3 The electromagnetic calorimeters at COMPASS

3.1 General principle of electromagnetic calorimeters and their operation at COMPASS

Electromagnetic CALorimeters (ECALs) are used to measure the energy of particles interacting primarily by the electromagnetic force, like γ or e^-. At COMPASS the ECALs consist of lead glass cells in three different sizes. If a high energy particle, like a γ, hits a lead glass cell it creates a so called electromagnetic shower. The shower is created by pair production, which makes it favourable to have a high density of heavy elements inside the cell. From the original γ an high energy e^- e^+ pair is produced, propagating through the lead glass by emitting bremsstrahlung. The bremsstrahlung is also producing pairs of e^- e^+ and so another instance of the whole process is started. This is going on until the involved particles have an low enough energy to be absorbed by the material.

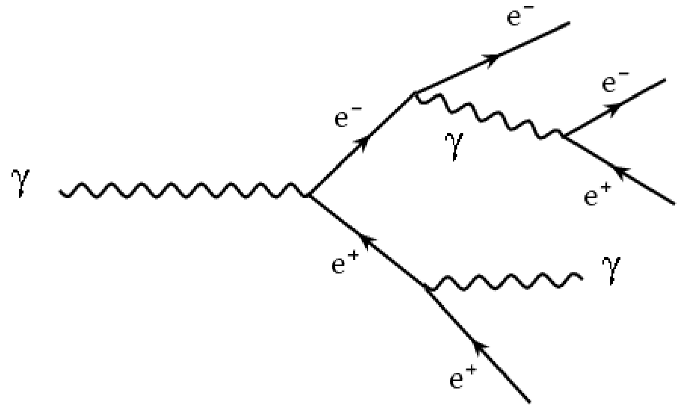


Figure 4. electromagnetic shower process

During the showering process the electrons and positrons are emitting Cerenkov radiation and the resulting photons are detected by photo-multipliers connected to the rear end of the lead glass block. The amount of photons created by this process is proportional to the energy of the initial particle. Together with the position of the cell all parameters needed for reconstructing the momentum four-vector are available.

At COMPASS two electromagnetic calorimeters are placed downstream of the target. They are build of three types of lead glass cells with different sizes and radiation resistance, the very radiation resistant shashlik blocks, radiation hardened lead glass and “usual” lead glass.

ECAL1 consists in total of 1500 channels connected to normal lead glass cells of three different sizes. The GAMS cells with a size of $3.83 \times 3.83\,\mathrm{cm}^2$ are used for the central part, the part above and below the centre is made of MAINZ modules ($7.5 \times 7.5\,\mathrm{cm}^2$) and the outer left and right part of $14.3 \times 14.3\,\mathrm{cm}^2$ OLGA modules. ECAL1 is located 14.1 m downstream of the target and has a size of $3.97 \times 2.86 m^2$ with a central hole of $1.07 \times 0.61 m^2$. Compared to ECAL2 the angular acceptance is larger. [V.K08]

ECAL2 is made of 3068 GAMS lead glass cells ($3.83 \times 3.83\,\text{cm}^2$) and has a size of $2.44 \times 1.83 m^2$ with a central hole of $0.08 \times 0.08 m^2$. 1440 cells are of the normal lead glass type and used for the outer parts of ECAL2, in the central part 864 radiation-hard shashlik cells are used and between these two parts 768 radiation hardened GAMS lead glass cells are used. ECAL2 is located $33.2 m$ downstream of the target and is covering the hole of ECAL1.

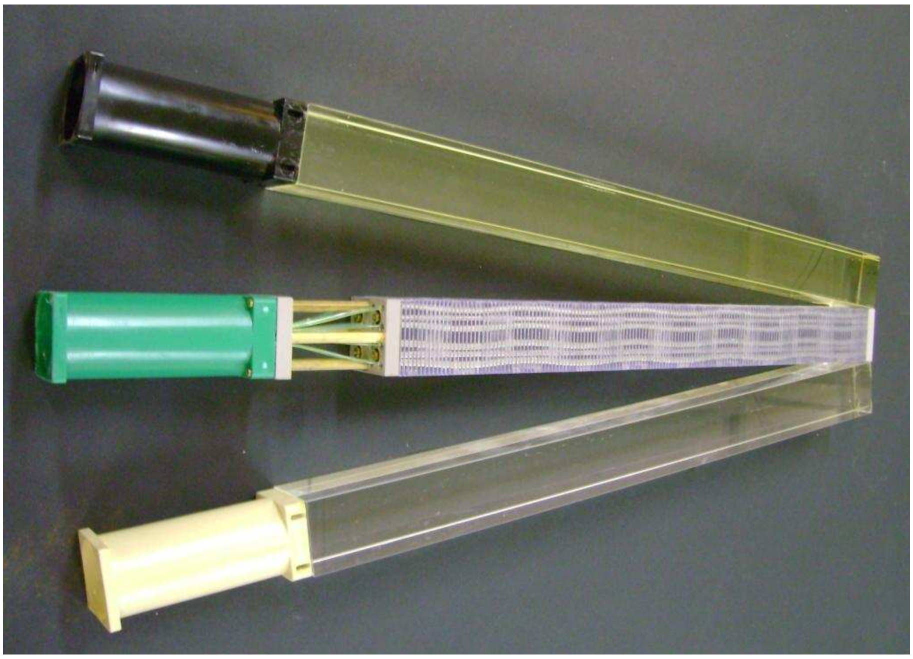


Figure 5. The 3 types of lead glass blocks used at COMPASS: radiation hard GAMS (top), shashlik (middle) and "normal" GAMS (bottom)

A crucial parameter of both ECALs is the gain of the photo-multipliers. Since the emitted Cerenkov light is proportional to the energy of the incoming particle, the gain should be constant for reliable measurements. To monitor and to correct fluctuations of the gain both ECALs are connected to external light sources. During the off-spill period, when no beam is crossing the spectrometer, light pulses are sent into all cells and the signals are readout via so called calibration events. ECAL1 is connected since 2009 to a new laser calibration system while for ECAL2 LEDs are used. In both cases the light is distributed by fibres directly to the cells. When the pulses carry a constant amount of light also the resulting signal amplitudes should be constant if the gain has not changed. Since the beginning of the 2009 run Cinderella is extracting the amplitudes of all ECAL calibration events and storing a 10 spill mean value of the signal amplitudes into the COMPASS database. The DCS system is reading these values and comparing them to previously defined reference values. When the deviation from the reference exceeds a certain threshold (or channels are missing completely) an alarm is triggered. This makes it possible to discover problems, like dead channels, gain fluctuations or even a turned off high voltage at a short timescale. The latest insight of this monitoring is, that the amplitudes of the calibration event signals are strongly needed to correct the calibration coefficients at a spill by spill stage for physical analysis, because the gain of each channel seems to change over time.

3.2 The readout of the electromagnetic calorimeters

The ECALs at COMPASS are read out by so called “Sampling Analog to Digital Converters” (SADCs). SADCs do not just read out a single value but can read out up to 128 samples with 12.5 ns in between. In the ECAL readout the sample number is 32, which is enough to save the whole pulse shape of the signal. Between the SADCs and the photo-multipliers so called shaper cards are installed, they are “slowing down” the signal and having an effect on the resulting pulse shape. Otherwise the signal from the photomultiplier would be too fast for the sampling rate of the SADCs and information could be lost.

In 2008 a new type of SADC, the Mezzanine SADC, has been introduced to the inner part of ECAL2. The difference to the previous version of SADCs is the bigger sample size of 12 bit, compared to 10 bit, which increases the dynamic range by almost a factor of 4. Additionally the MSADCs consist of two sampling ADCs working in an interleaved mode. It allows to reach higher sampling frequencies than with the old SADCs.

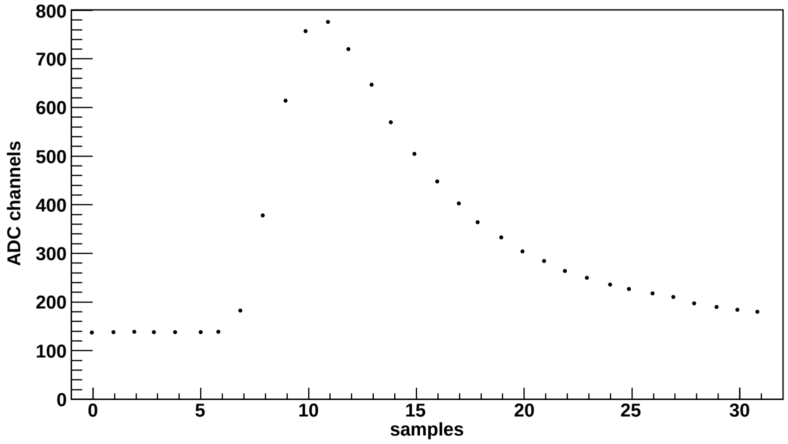


Figure 6. MSADC signal from ECAL2

Every signal coming from an electromagnetic shower looks more or less like shown in figure 6. At the beginning of the signal the first samples are on the baseline level. The baseline level is the signal level which comes from intrinsic characteristics of the readout electronics without a signal from the photomultiplier. In figure 6 the signal begins at sample 7. Main characteristics of a good signal are the fast rise within 4 – 5 samples and the slow exponential decay after the signal peak. In discussions with the expert for SADC readout electronics, Igor Konorov, it turned out the rise time of a signal is determined by the shaper, guaranteeing a rise interval of at least 4 samples. At a later stage it will be explained how to use this feature for distinguishing noise from good signals.

A consequence of the two ADCs inside the MSADC readout module is the presence of two different baselines in every MSADC signal, each belonging to one of the ADCs. The difference between these two baselines can be up to 20 ADC channels and has to be corrected.

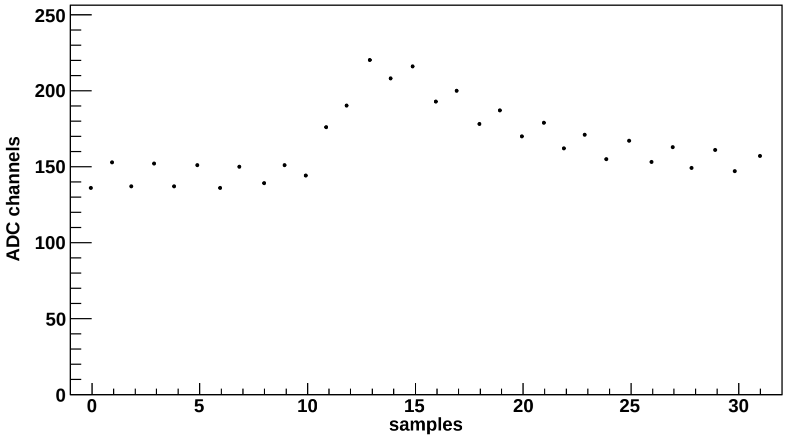


Figure 7. MSADC signal with uncorrected ADC baselines

In 2008 Cinderella was able to correct the baseline difference by calculating the difference between the first two samples. Plans for 2009 are to do this correction inside the readout hardware and additionally to normalise the baseline to a fixed value.

4 The COMPASS DAQ 2008/2009

The COMPASS DAQ in 2008 - overview

COMPASS as an experiment with a high trigger rate needs a reliable and powerful Data AQuisition (DAQ). The DAQ at COMPASS is split into two parts, the so called "front-end electronic" part and the "main DAQ" part. The front-end is the readout electronic chain starting at the detectors including the Analog to Digital Converters (ADCs) and the CATCHes/GeSiCAs (**C**OMPASS **A**ccumulate, **T**ransfer and **C**ontrol **H**ardware / **Ge**m **Si**licon **C**ontrol **A**cquisition). CATCHes and GeSICAs are representing the first out of three points of data concentration. They are accumulating data from multiple ADCs and joining them together. The concentrated data is labelled by a source ID, which is a unique ID for a part of a detectors channels. All CATCHes and GeSICAs are transferring the data to the second concentrator stage, the "Local Data Concentrators" (LDCs), also called "Read Out Buffers" (ROBs), which are already part of the "main DAQ". This transfer is done via slink protocol over a fibre connection. The maximum bandwidth of a slink over fibre connection is 160 MB/s. In 2008 four different slinks cards have been available. The Fibre Channel, the ODIN/double ODIN and the HOLA type. For data transmission of course two slink cards per source ID are needed. One is sitting at the back-plane of a VME crate, where also the CATCHes/GeSICAs are placed, having the role of a "Link Source Card" (LSC). The slink card at the other end has the role of a "Link Destination Card" (LDC) and is sitting on a spill-buffer PCI card in the ROB. Everything before the ROB, including the slink cards and the spill-buffer PCI card, is build from non conventional electronics, while from the ROB stage on everything is standard server hardware, which can be bought at usual server hardware suppliers. In 2008 up to four slink card mounted on four spill-buffers could be build in per ROB. On the ROB the data from all source IDs transmitted via the slink cards are packed into DATE (**D**ata **A**cquisition and **T**est **E**nvironment) sub-events. All sub-events are finally transmitted via Gigabit ethernet to the third concentrator stage represented by the "Global Data Concentrators" (GDCs), also called "EVent Builders" (EVBs). This is the stage where Cinderella is running and the final COMPASS events are assembled. Every event is stored in raw data files. The data files itself have a size of 1 Gigabyte and are called "chunks", because they are a fraction of a whole runs data. They are finally sent to the CASTOR (CERN Advanced STORage manager) storage system via a 10 GBit fibre ethernet connection.

Theoretical performance of the setup

The general performance of the "main DAQ" part, which will be just called DAQ in the rest of this chapter, depends on many parameters. Especially the role of the used software packages is not negligible. But in the following calculation only the role of the hardware will be considered, to get an idea what could be possible in optimal conditions.

One network connection in the DAQ has a limit of approximately 100 MB/s. A 1 GBit network usually has a maximum bandwidth of 120 MB/s, but this value is dependant of the block sizes and the load in the network. Performance tests before the 2008 run have shown, 100 MB/s is a good mean value describing the performance of the COMPASS network. So the maximum speed for data transmission from the ROBs to the EVBs is 100 MB/s. In a 48s cycle is

$$\mathbf{48}s\cdot\mathbf{100\,MB}/s=\mathbf{4.8\,GB}$$

In 2008 12 event-builders have been available, the total amount of data per cycle increases by a factor of 12:

$$\mathbf{4.8\,GB\cdot 12=57.6\ GB}$$

Since the beam time per cycle is only 9.6s, the whole 57 GB has to be accumulated in this period

$$\mathbf{57.6\,GB\,/9.6}s\;=\;\mathbf{6\ GB/s}$$

With an average event size of 40 KB per event, the 6 GB/s correspond to a trigger rate of

$$\frac{\mathbf{6\,GB}/s}{\mathbf{40\,kB}}=\mathbf{150000}\,\frac{\mathbf{1}}{s}$$

So the theoretical limit is a trigger rate of 150000 per second. Of course many idealistic assumptions have been made for this simple calculation. For example it has been assumed the data load is distributed equally on all ROBs, which is not possible in reality, because of the different data load from detector to detector. Also inefficiency of all used software packages has not been taken into account, as already mentioned above.

In 2008 some performance tests have been done and the maximum trigger rate (random trigger with beam) which has been reached was 45000 events per second. This is about a factor of three below the theoretical limit.

For real data taking also the front-end part has to be taken into this calculation, especially the dead time of the readout electronics. But this calculation and rate tests at CERN have shown, the COMPASS DAQ can be made ready for data taking at 50000 events per second with moderate effort. The limiting factor in the main part of the DAQ is the number of EVBs and a few overloaded ROBs at the moment. This can be solved by replacing the overloaded ROBs by either more or more powerful machines. The number of EVBs has been extended already in 2009.

Upcoming problem with the PCI spill-buffer recognised in 2008

The trigger rate in 2008 has been around 18 kHz, which was no challenge for the COMPASS DAQ. An upcoming problem has been recognised with the PCI spill-buffers. The PCI standard is already many years old and dying out at the moment. It will be replaced by the PCI-express (PCIe) standard and many mother board suppliers are not offering boards with 4 PCI slots anymore. From the suppliers still offering boards with enough PCI slots, many are not supporting the PCI 2.2 standard anymore which is absolutely required for the operation of the PCI spill-buffers. Since from time to time new ROB hardware is needed, for replacement of broken ROBs, extensions for new equipments or as spares, it is getting difficult to find suitable hardware.

New PCIe spill-buffer development

The only possibility to avoid the problem of not being able to buy suitable ROB hardware in the future, is to develop spill-buffers cards, which are using the new PCIe standard. This is by the way also a possibility to increase the efficiency of the COMPASS DAQ. So it is planned to build a new PCIe based spill-buffer card with connectors for up to 4 slinks. To keep compatibility to the current spill-buffer driver (with only a few modifications) the 4 slinks will be multiplexed inside the spill-buffer. A sub-event-building capability on hardware level is required for this, which is working similar to the multiplexer cards used for CATCHes. This will decrease the number of needed ROBs and spill-buffer cards, helping to lower the hardware costs. With being compatible to a modified version of the current spill-buffer driver and the current HOLA slink LSC cards, also the development effort is manageable and already existing components might be reused. The main part of the development is the hardware for the LDC part and the modification of the current spill-buffer driver. To be ready for future DAQ upgrades and data taking with increased data rate the maximum amount of buffer memory will be increased up to 2 – 4 GB per slink connection compared to 512 MB at the moment.

The road map of the project is to finish the development of a first evaluation card with two slink connectors during 2009 and supply the modified driver. At this step the new and old spill-buffers can be used together in one machine, which makes the migration to the new hardware easy. After finishing this step the final card with 4 slink connectors will be developed.

Upgrades finished in 2009

To be able to increase the trigger rate in the future the computing capability on the EVB side has to be increased for Cinderella. Otherwise this would lead to problems with storing all this data. Already at the trigger rate from 2008 the whole CASTOR space for COMPASS, which is around 1 PB, would have been filled up completely if the data taking period would not have ended earlier due to the LHC incident. In the following chapters different strategies for data reduction will be presented which are profiting from more computing power. It is also a good idea to have some resources for upcoming development of Cinderella. Therefore 8 new EVBs machines have been bought with 8 CPU cores each. With the additional 64 CPU cores, compared to the 24 existing, there is enough computing power for more sophisticated Cinderella usage than until now. The current DAQ should now be able to apply all new Cinderella features described in the following chapters.

5 Data Reduction

5.1 Recorded Data 2008

The trigger rate at the COMPASS Experiment during 2008 data taking was around 18 kHz. The average size of an event was around 40 KB. This is not very much compared to LHC Experiments like ALICE [HHC02] with its up to 75 MB/Event, but together with the trigger rate the data rate of COMPASS is quite high.

To get an idea of the data rate during the 2008 run, the spill length and the SPS cycle length have to be taken into account. A “spill”, also known as “burst”, is the time period during which the beam is extracted from the SPS. In this period most of our data (all physics data) is recorded and its length during 2008 was constantly 9.6s. The cycle length, in which the spill period is included, is the total time between the beginning of 2 sequential spills. This period was varying from time to time, depending of the state of other experiments.

CNGS	18.0s
LHC	7.2s
fix. target	16.8s
MD	6.0s
total	**48s**

Table 1. beam extraction distribution over all CERN experiments

COMPASS belongs to the group of fixed target experiments and gets a beam extraction time of 16.8s. One part of this period is for injection and acceleration of the particles (7.2s) and the other part (9.6s) is extraction time. Most of the time the cycle length was as shown above at 48s, with 9.6s “on spill” (beam extraction) followed by a 38,4s “off spill” period with no beam. Sometimes the cycle changed due to an unavailability of an other experiment (like the LHC unavailability due to the magnet accident on 2008-09-19). The “on spill” period stayed at 9.6s but the “off spill” period decreased by the extraction interval of the unavailable experiment.

Using these values the on spill data rate at COMPASS can be calculated as following:

$$\mathbf{data\,rate = trigger\,rate \cdot event\,size = 18kHz \cdot 40\,KB = 720\,MB/}\boldsymbol{s}$$

This is the data rate which is produced per second during “on spill” time. But it is not the sustained data rate to be recorded in the end. To get the sustained data rate, the quite long “off spill” period has to be taken into account. “On spill”

$$\mathbf{720\,MB/}\boldsymbol{s}\cdot\mathbf{9.6}\boldsymbol{s}\approx\mathbf{6.9\,GB}$$

of data is produced. In the remaining 38.4s no data is coming anymore from the spectrometer and the DAQ system can process the data in its buffers. So there is 6.9 GB in 48s which gives us a sustained data rate of

$$\frac{\mathbf{6.9\,GB}}{\mathbf{48}\boldsymbol{s}}\approx\mathbf{144\,MB/}\boldsymbol{s}$$

(this is slightly more than a single Gigabit Ethernet interface could handle).

Future plans for COMPASS propose to take data at twice the rate of 2008. This would mean 13.8 GB per spill or 288 MB/s sustained. In less than 965 hours (≈ 41 days) of data taking 1 Peta-Byte of raw data would have been recorded, this is more or less what COMPASS can use on CASTOR per year at the moment. That’s why it is important to develop strategies for data reduction in the future otherwise the COMPASS collaboration has to pay more and more for tape costs or is not able to increase the statistics.

The original plan to reduce data was by using Cinderella online filter. Cinderella is basically able to reduce the amount of data by 2 different ways.

1. event number reduction

This is the original planed operation mode of Cinderella. The strategy here is to identify complete events which are not useful for physics analysis and reject them before they will be recorded (instead just cutting them out at the analysis stage anyway). The advantage of this method is the big impact on the recorded data amount because the effective trigger rate will be reduced. The complicated part is the development of conditions when and at which circumstances to reject an event and when not. This decision has to be made in coordination with the analysis groups and the data quality responsible.

2. event size reduction

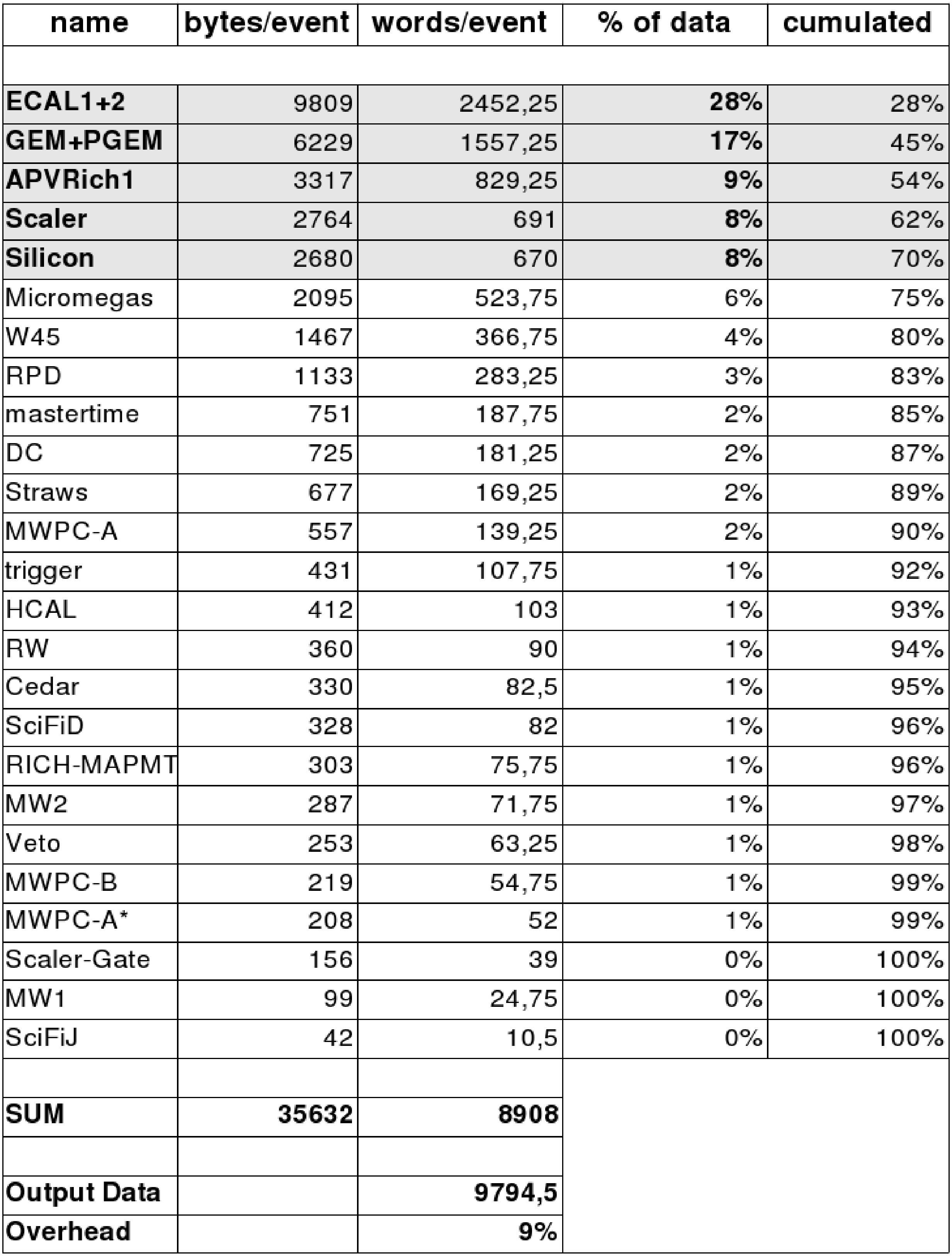
2.1. suppression of noisy channels

From the physics point of view this approach is much simpler than rejecting complete events. Every detector has some noise in its data, one has more another has less. Suppressing noisy channels will reduce the event size. The total achievable reduction rate is limited, because the good part of the event has to be kept. Probably the reduction rate will not be higher than $\approx 50\%$. Combined with the first method the data reduction factors will multiply and increase the efficiency of total data reduction. This method can be applied for every equipment at the COMPASS spectrometer but it requires the support of the corresponding detector groups.

2.2. optimisation of data encoding

For every equipment which has multiple values to be read out, an optimised encoding of the samples can save a lot of storage space (like shown later for the M/SADCs) in comparison to a generic encoding. But this method is not appropriate for equipment with very few values to be read out (or with just one value) if these values fit into one data word, because one data word is the smallest possible unit in the current data format. The advantage of this method is, there are no cuts into the data introduced. Thus the only condition which has to be fulfilled in this method is data integrity and safety. Usually no other group has to be involved which keeps the development process quite simple.

Cinderella is basically able to apply all of the described reduction mechanisms to every equipment at COMPASS. “Optimisation of data encoding” seems to be the most simple one to apply, because no cuts into the physics data will be introduced and thus the verification process is simple. To find the most suitable equipment for optimising the data encoding it might be helpful to get an overview about the data distribution of the COMPASS spectrometer in 2008 and identify the biggest source of data. The Cinderella tool “cat_date” is predestinated for this task. It is able to print out statistics indicating the amount of data of every single source ID. Every source ID belongs to a part of a detector from the spectrometer, so the total data amount from one detector can be calculated just by summing up the corresponding source IDs. For this investigation it also makes sense to group equipments of the same type and data format together. The result of this investigation shows, 70% of the data is coming from 5 of 25 equipments.



name	bytes/event	words/event	% of data	cumulated
ECAL1+2	9809	2452,25	**28%**	28%
GEM+PGEM	6229	1557,25	**17%**	45%
APVRich1	3317	829,25	**9%**	54%
Scaler	2764	691	**8%**	62%
Silicon	2680	670	**8%**	70%
Micromegas	2095	523,75	6%	75%
W45	1467	366,75	4%	80%
RPD	1133	283,25	3%	83%
mastertime	751	187,75	2%	85%
DC	725	181,25	2%	87%
Straws	677	169,25	2%	89%
MWPC-A	557	139,25	2%	90%
trigger	431	107,75	1%	92%
HCAL	412	103	1%	93%
RW	360	90	1%	94%
Cedar	330	82,5	1%	95%
SciFiD	328	82	1%	96%
RICH-MAPMT	303	75,75	1%	96%
MW2	287	71,75	1%	97%
Veto	253	63,25	1%	98%
MWPC-B	219	54,75	1%	99%
MWPC-A*	208	52	1%	99%
Scaler-Gate	156	39	0%	100%
MW1	99	24,75	0%	100%
SciFiJ	42	10,5	0%	100%
SUM	**35632**	**8908**		
Output Data		**9794,5**		
Overhead		**9%**		

Figure 8. contribution of all equipments to total recorded data from run 70502

Figure 8 is showing the data distribution of all detectors from the COMPASS spectrometer. All data from source IDs belonging to one detector type have been summed up event-wise to obtain the total data load per event. To determine the overhead the sum of all detectors has been compared to the total data amount being recorded in the end.

The equipment with the biggest event size, standing out in the table above, are the ElectromagneticCALorimeters (ECALs). They produce almost 30% of the total data. It seems this equipment would be ideal for reducing the event size by reducing noisy channels and optimising the sample encoding. For reducing noisy channels it has to be looked inside the raw data coming from this equipment to see if there is something recorded to be cut out safely. This causes much more effort than just optimising the data encoding and will be done at a later stage.

5.2 Encoding optimisation for ECAL data

The ECALs at COMPASS are read out by so called “Sampling Analog to Digital Converter” (SADC) configured to read out 32 samples wit 12,5 ns in between. There are 2 different kind of SADCs. The so called “SADC” type, which reads 10bit samples and the “MSADC” (Mezzanine SADC), with 12 bit samples. The data format for both is very similar, it just differs in the header and the data word layout. The amount of data coming from the (M)SADCs is quite high because of the 32 read out samples. A 32 bit data word contains three 10 bit SADC samples or two 12 bit MSADC samples with the current data formatting scheme.

In the MSADC data word there are 8 unused bits, although 2 of them are used for data integrity checks, because only two of the 12 bit samples values fit completely into one data word.

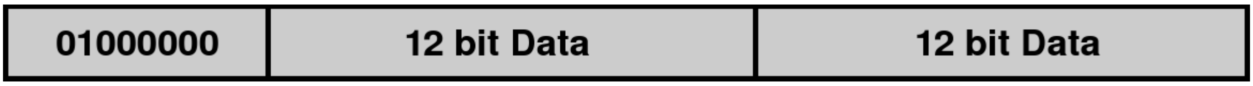


Figure 9. MSADC data-word

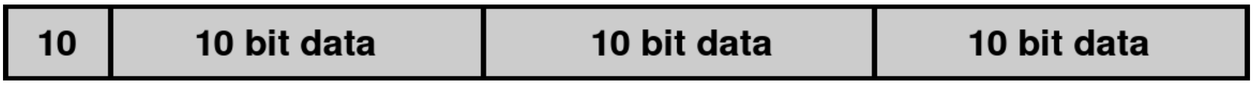


Figure 10. SADC data-word

On the SADC side its more efficient with its three 10 bit sample values, but many of the stored samples are from the same value.

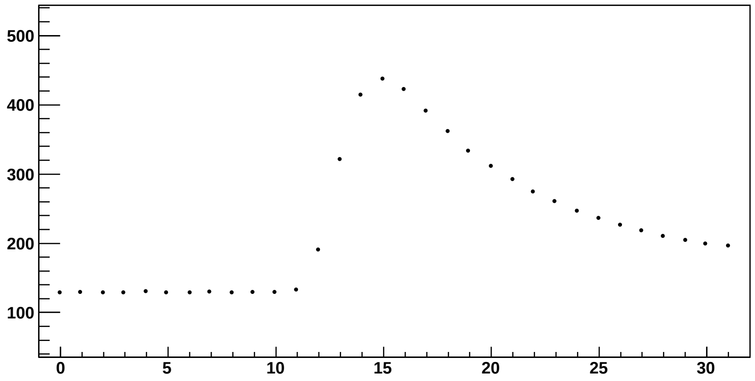


Figure 11. MSADC signal pulse shape

Before the "real" signal with physics information begins, there is the baseline of the ADC chips which contains the first 5 - 7 samples and does not fluctuate for more than 3 - 4 ADC channels. This means 10 respectively 12 bit of memory are used for storing the same values again and again and loosing always a gap of 2 or 8 bit because of storing just "complete" samples (no overlap between data-words). Considering all this one can think of 2 techniques to reduce data amount by optimising the encoding of the data:

1. **more dense packing of the samples**
This strategy is obvious. Using every single bit from the data word by storing one part of a sample in the remaining bits of the current data word and the rest in the next one, can reduce data of the MSADC by 25%. For the SADC of course the gain is much lower (6.25%).

2. **optimisation of data encoding**

2.1. Huffman encoding

The principle of Huffman encoding is the deviation from the principle of constant size encoding. David A. Huffman proved in 1952 [Huf52], that its possible to obtain an optimum encoding of values for minimising the encoding length of a message.

For reducing the amount of data, Huffman encoding applies a shorter encoding to the more frequent values and a longer encoding to the less frequent values. The result is a shorter encoding of the complete sample set if the gain from the shorter encoded values is bigger than the loss of the longer encoded values, this is always the case for an optimal tree.

To achieve this it is important to know the frequency of appearance of every possible value. Cinderella is able to dump a so called "Huffman statistics" file, which just tells how often which value appeared in the currently processed data. From this Huffman statistics file, which is in binary format, the Cinderella tool 'mkhufftree' can generate a binary tree and print it out in text format like it is needed for the mapping file.

To obtain an optimal Huffman tree all values and their frequency information, obtained from the statistics file, are stored in sub-trees. Then the 2 sub-trees with the lowest frequencies are merged together, so that the values represent the leafes. This procedure is repeated until only one big tree is left.

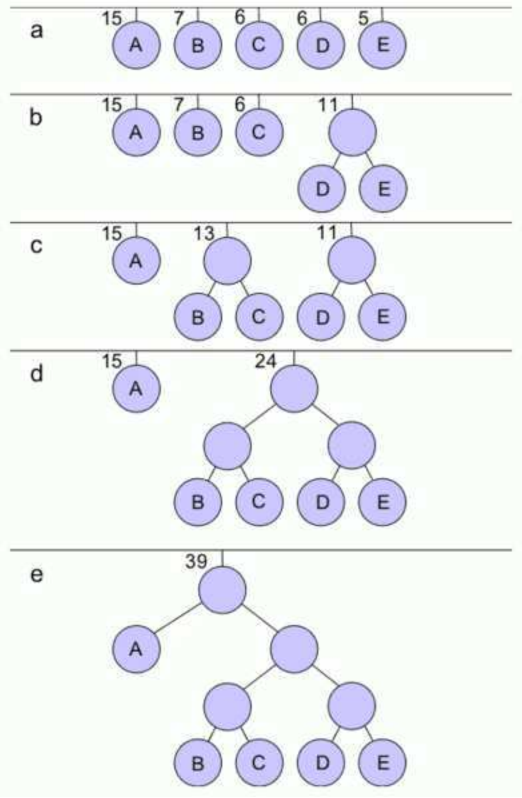


Figure 12. how an optimal Huffman tree is created (Source: Wikipedia Commons author: Andreas.Roever)

The characteristic of the resulting binary tree is, that the path to the most frequent elements is much shorter than the path to very seldom elements. The de-/encoding results from a common convention in which the left child means ’0’ and the right one means ’1’. For reasons of performance Cinderella is using a look-up table for encoding which is a one dimensional array. The index of the array represents the value to be encoded. This increases the encoding performance because no look-up of the value is needed.

The Huffman algorithm is profiting very much from an extremely nonuniform distribution of the encoded values. That’s why storing just the differences between samples can boost up the efficiency of the Huffman algorithm. Figure 13 is showing the distribution of baseline values, represented by the first sample of a channel, and differences between samples.

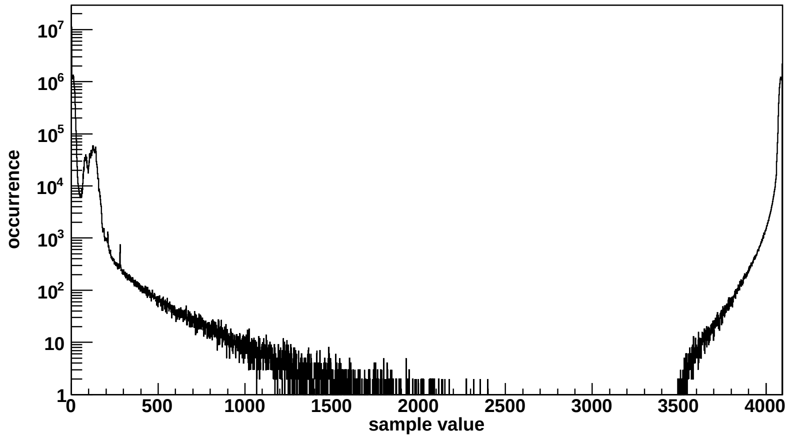


Figure 13. occurrence of (M)SADC baseline values and differences in run 70502

2.2. storing only differences between samples

The baseline values are varying from channel to channel. This means for the Huffman algorithm it has to deal with many different values, which is not an optimal condition. It is possible think about normalising all baselines to a common value, this would improve the efficiency of the Huffman algorithm. But this is not a trivial task for Cinderella to normalise channels where it is not difficult to detect the right baseline (see chapter about baseline detection). The more elegant possibility is just to store the first value of every sample set and for the next one just to store the difference to the previous one. For the Huffman algorithm this technique provides a big improvement, because especially in the baseline region the differences between the samples are mostly in the interval between [-4;4] (just 9 values of 4096). The encoding length also gets independent from signal amplitude and baseline level and the length of the most frequent values “0” and “-1” (4095) is just 2 bit (from 10 respectively 12 bit before).

Offline tests with real data from 2008 combining these 3 techniques (dense packing, Huffman encoding and difference storing) show quite impressive results. The average reduction rate for ECAL data is around 80% using different runs which reduces the total data by $\approx 20\%$.

6 ECAL in Cinderella

6.1 Treatment of ECAL in Cinderella overview

Various modules are available to decode, perform feature extraction and make decisions on ECAL data. All modules have been originally designed for ECAL2 but can be used for ECAL1 as well. The modules depend on each other beginning with the SADC decoding module. The following part gives an overview about all ECAL related modules, which will be described in detail later.

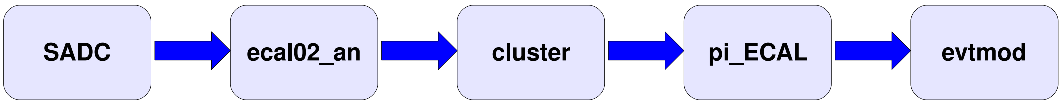


Figure 14. module hierarchy of ECAL related modules

The purpose of the SADC module is to decode channels read out by the SADC or MSADC electronics. It can distinguish between both types based on the “option” attribute in the XML mapping file. Data from all relevant source IDs are read by the SADC module and after some data integrity checks on all header, data and integral words, the data samples from the data words are stored together with the mapping information in a result array for further processing.

After decoding one event, containing one or more channels, the decoded data is passed to the “ecal02_an” module, where the main part of feature extraction is done. This module tries to find the baseline, distinguish noisy from good signals, extract the energy and calculate the position of the channels in COMPASS coordinates. Additionally it can dump channel amplitudes from calibration events into the database for monitoring purposes. At this point the data is ready to be passed to the “cluster” module.

In the “cluster” module neighbouring hits are grouped together into clusters. The energy and the centre of mass position of every cluster is calculated and prepared to be processed by the “pi-ECAL” module.

At this stage every cluster should correspond to one particle which has hit the ECAL. Combining the 4-vectors of 2 particles and calculating the invariant mass should show a $\boldsymbol{\pi^0}$ peak in the invariant mass spectrum.

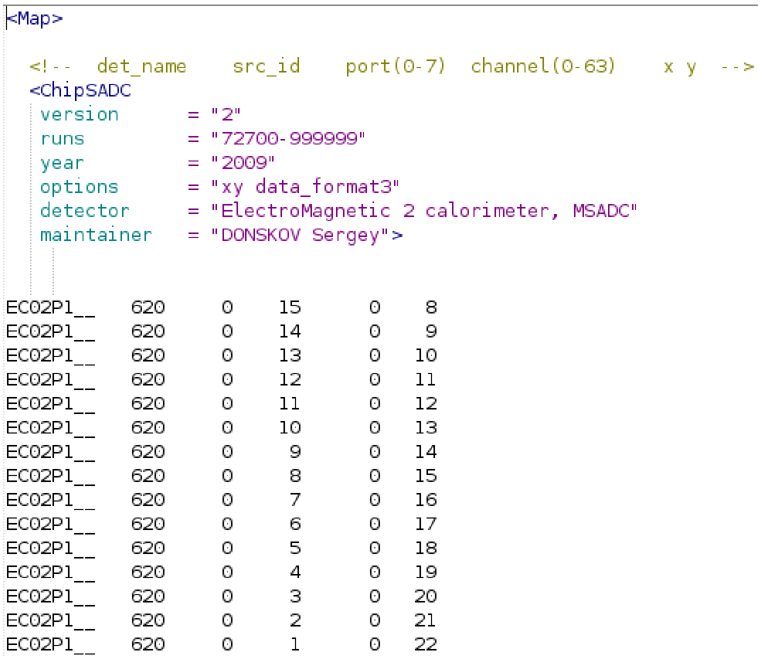
The last module of this chain is the “evtmod”, the event modification, module. Here the event can be modified by cutting out noisy channels, based on decisions of the previous modules, or by applying Huffman encoding to the (M)SADC channels.

Beside of that, these modules have the option to create a big variety of histograms for debugging and quality checks.

6.2 Decoding

The two different types of Sampling ADCs at COMPASS, the SADCs and MSADCs, have a slightly different data format. It is not possible to distinguish between these two data formats during decoding in the SADC module, that's why it has to be known which source ID is delivering which SADC data format before the decoding process starts. This information can be obtained from the mapping files.

Every channel of the ECAL is itemised in one of the corresponding mapping files. The COMPASS mapping files are in XML format and contain information about every channel, like its position inside the detector and the connection to the readout electronics. In the header of the mapping file "XML attributes',' containing some information about the file and the corresponding equipment, can be found. These attributes specify for instance the range of run numbers, for which the entries are valid, the maintainer of the file or some custom entries depending on the equipment.



```
<Map>

  <!--  det_name    src_id    port(0-7)  channel(0-63)    x y  -->
  <ChipSADC
   version      = "2"
   runs         = "72700-999999"
   year         = "2009"
   options      = "xy data_format3"
   detector     = "ElectroMagnetic 2 calorimeter, MSADC"
   maintainer   = "DONSKOV Sergey">


EC02P1__    620    0    15    0    8
EC02P1__    620    0    14    0    9
EC02P1__    620    0    13    0   10
EC02P1__    620    0    12    0   11
EC02P1__    620    0    11    0   12
EC02P1__    620    0    10    0   13
EC02P1__    620    0     9    0   14
EC02P1__    620    0     8    0   15
EC02P1__    620    0     7    0   16
EC02P1__    620    0     6    0   17
EC02P1__    620    0     5    0   18
EC02P1__    620    0     4    0   19
EC02P1__    620    0     3    0   20
EC02P1__    620    0     2    0   21
EC02P1__    620    0     1    0   22
```

Figure 15. mapping file entries for ECAL2

For the ECAL detectors, read out by (M)SADCs, there is an attribute called "options" specifying the data format of the contained channels. In 2008 there have been just two different data formats in use:

"xy data_format2"	SADC
"xy data_format3"	MSADC

Table 2. the two different SADC data formats

The mapping files are parsed at the initialisation phase of the SADC decoding module, which happens before the data taking starts and the SADC module can distinguish during the run between these two data formats based on the source ID.

During this initialisation phase the rest of the mapping files content is used to create a look-up table for the mapping information itself. With the help of this look-up table the X-Y position of every decoded channel within the detector (detector coordinates) can be determined based on its source ID, ADC ID and channel number.

In the decoding process all header, data and integral words are checked not to be corrupted and if all information inside is consistent. By the way also the position of the signals maximum is determined, which is needed at a later stage to extract the energy from the signal.

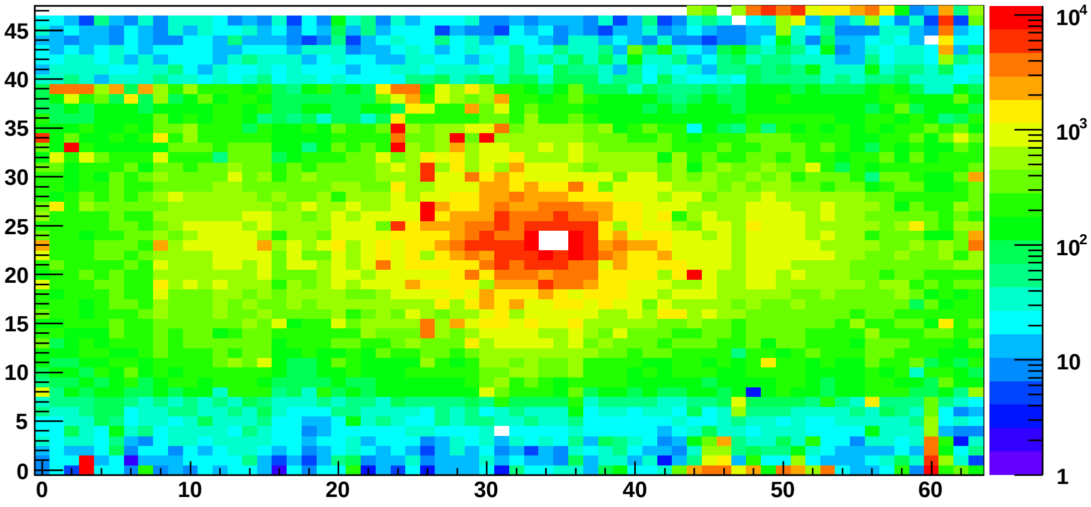


Figure 16. hitmap in "detector coordinates" of ECAL2 after decoding

6.3 Feature extraction

The first step of feature extraction is to detect the baseline of the corresponding channel, which is the signal level of the (M)SADC electronics without a signal from the photomultiplier. If a signal from the photomultiplier arrives, it is sitting on top of the baseline exaggerating its real value. To extract the physical relevant information of a signal it is important to know the baseline level of each channel and subtract it from the signal values. The baseline level may be different for every channel in the ECAL detectors and it may change over time. That's why its recommended to detect the baseline at least every spill or even at every event like it is done in Cinderella.

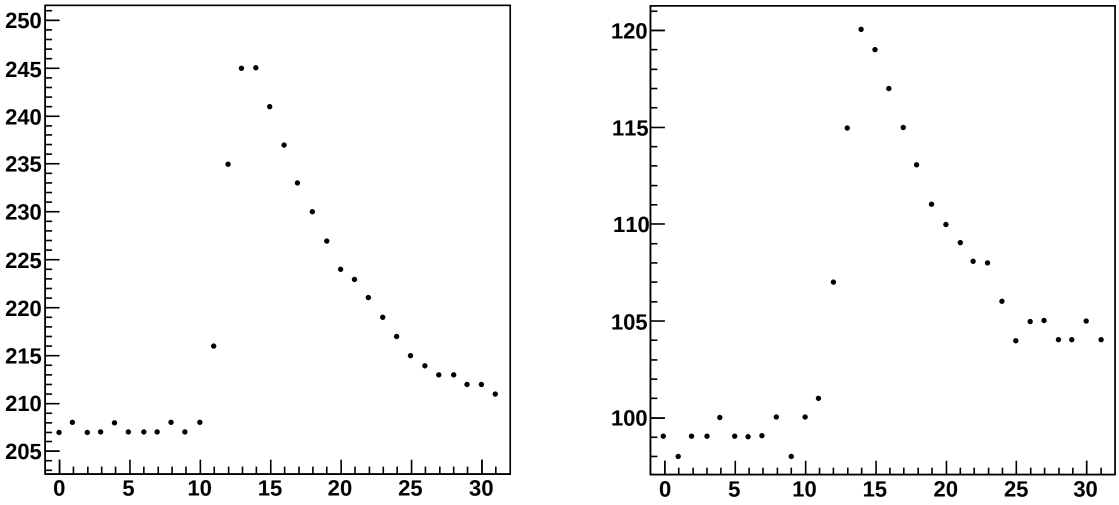


Figure 17. two signals from the SADC readout of ECAL2

Every good signal, coming from an ECAL shower, looks more or less similar. The first 5 – 10 samples are at the baseline level, then the real signal begins with a fast rise and a slow decay after the signal peak. The idea Cinderella is following is to detect the begin of a shower signal by finding the characteristic fast rise. For this the "ecal02_an" module is calculating the slopes between all data samples and is trying to find a region of $\boldsymbol{N}$ consecutive slopes with no negative slope and the total sum of slopes in this region above a threshold $\boldsymbol{S}$. If one interval fulfilling this condition is found, the first sample of this interval is assumed as the beginning of the signal and the end of the baseline region. Now the baseline is retrieved by calculating the mean value of all samples until the end of the baseline region.

The threshold $\boldsymbol{S}$ corresponds to an amplitude cut in the rise region of the signal:

$$\boldsymbol{S} = \sum_{n=x}^{x+N-1} (t_{n+1} - t_n) = t_{x+1} - t_x + \ldots. + t_{x+N} - t_{x+N-1} = \boldsymbol{t_{x+N} - t_x}$$

$\boldsymbol{N}$ is the number of slopes taken into account and t_x is the sample value at position “x”. $(t_{x+1} - t_x)$ is the slope between two following samples beginning from the “x-th” sample. The thresholds $\boldsymbol{N}$ and $\boldsymbol{S}$ are configurable and have to be fine tuned. During offline tests it turned out a reasonable value is **N=3** but for **S** it is not so simple to find a common threshold for all channels. Due to slightly different high voltage values for every channel and different high voltage settings for the inner- and outer part of the ECALs, the responsiveness of every channel is different. This makes an amplitude cut with a fixed amplitude not very reasonable, because this would mean a different cut in energy for every channel. The energy deployed in a channel is defined as the product of signal amplitude and calibration coefficient from the calibration file.

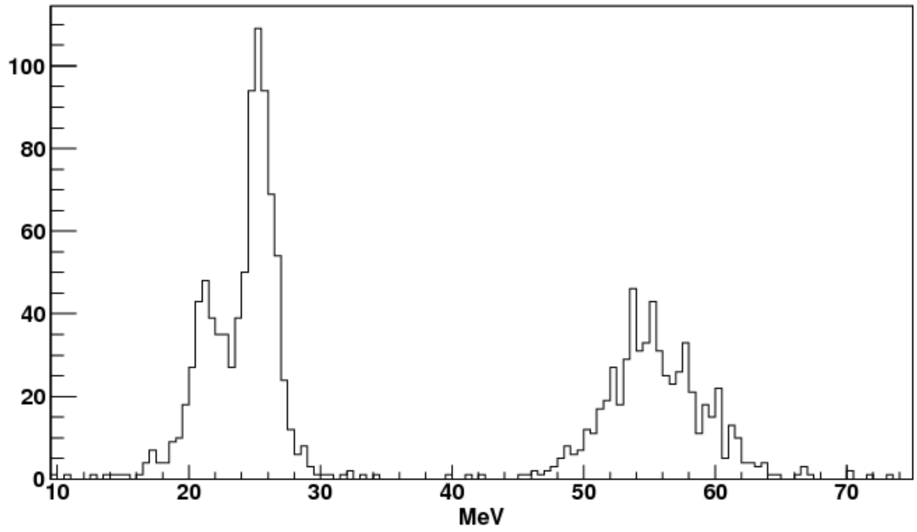


Figure 18. ECAL1 calibration coefficients 2008

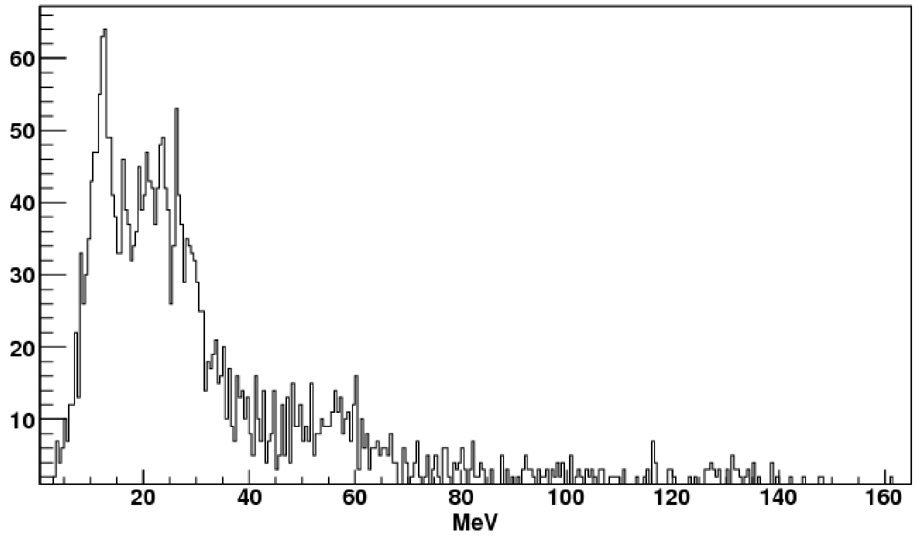


Figure 19. ECAL2 calibration coefficients 2008

A threshold of 5 ADC channels could mean for some ECAL2 channels a cut somewhere between 25 – 500 MeV. With an amplitude corresponding to 500 MeV even good signals could be cut out by this algorithm which correspond to an energy below 500 MeV. That's why Cinderella is able to calculate individual amplitude thresholds for every channel based on the coefficients from the calibration file provided by the ECAL group. After setting the minimal energy which should not be cut out, a minimal amplitude for signal detection is calculated and applied in the baseline detection algorithm.

The baseline detection algorithm is dependant on the rise time of the signal to be not shorter than the chosen slope interval **N**. In discussions with the expert for the (M)SADC readout electronics, Igor Konorov, it turned out, the rise time of all signals are determined by the shaper, who is delaying the signal, and it is guaranteed not to be shorter than 4 samples for good signals. That's why **N=3** should be a safe choice.

Remark 1. *In 2008 new MSADCs have been introduced to the ECAL2 readout. As a special feature they have two ADCs instead of just one, sampling the values in an interleaved mode. Each of the ADCs has its own baseline, which can differ up to 20 ADC channels form the other one. Its planned for 2009 to adapt these two baselines and normalise them to a predefined value, but for 2008 a workaround was needed to make Cinderella work with these different baselines inside one sample set. The introduced workaround measures the difference of the first two sample points and assumes this as the baseline difference of the two ADCs. During 2009 this workaround will not be needed anymore.*

In every event there are some channels, where the baseline cannot be detected. This means the thresholds of the algorithm are not fulfilled. Without knowing the baseline further processing of the affected channels, like extracting energy, is difficult.

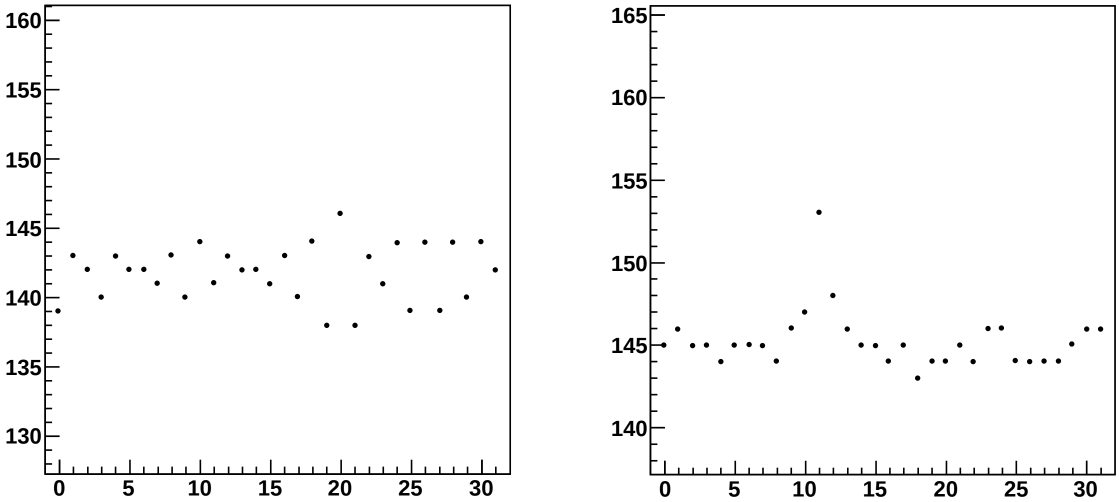


Figure 20. noise from ECAL2

Looking at these signals shows their shape is some kind of strange and looks more as noise than an ECAL shower signal. These signals are candidates to be thrown away but before doing this they have to be proven to be not useful for physics analysis.

For crosschecking and making sure not to throw away possible good signals, another signal detection algorithm is provided. It works by summing up all samples from two independent intervals. The first interval is starting at the first sample of a channel and the second on the sample number 10, where every good signal is expected. The length of the intervals is set to 5 samples. After summing up the samples of these two intervals the difference between them is calculated and if this difference is above a given threshold the corresponding channel is assumed to carry a signal. Effectively this algorithm also corresponds to an amplitude cut. In this case the cut is performed on the average amplitude of the second interval, because the first interval should only contain the baseline.

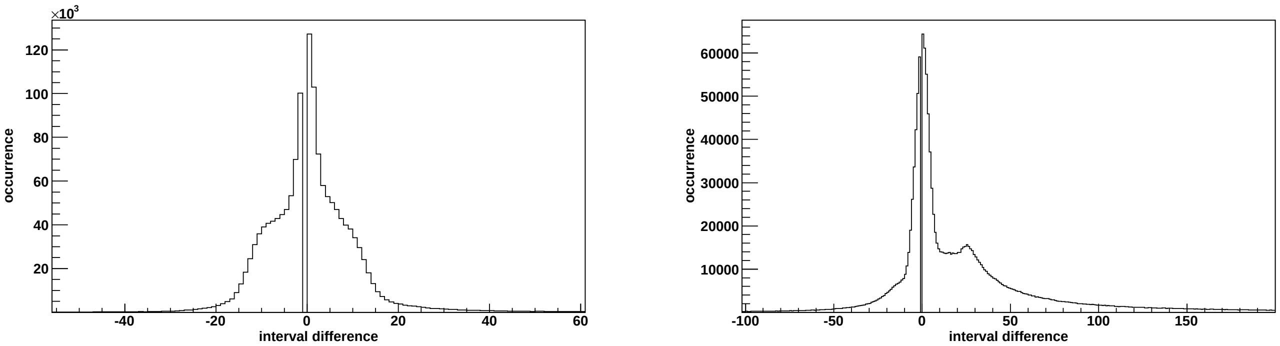


Figure 21. differences of two independent intervals in ECAL1 (left) and ECAL2 (right) data

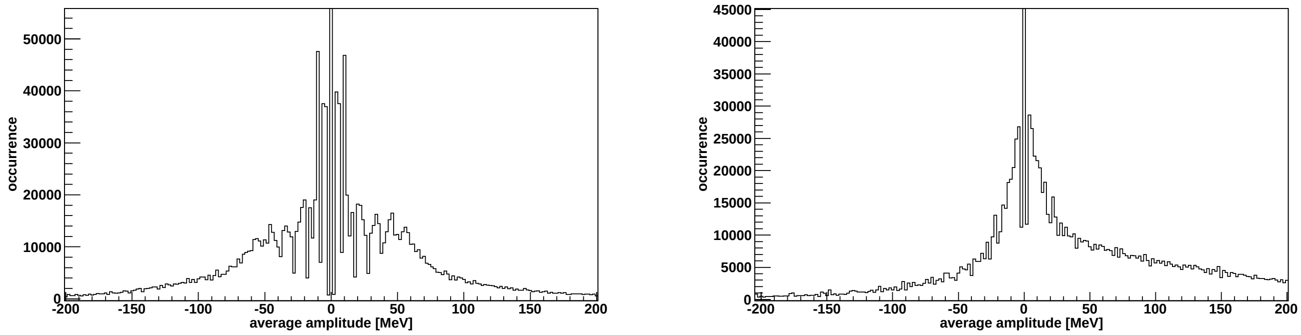


Figure 22. average amplitudes in MeV for ECAL1 (left) and ECAL2 (right)

Like in the case of baseline detection it is necessary to translate the average amplitude into an energy to consider the different sensitivities of the ECAL channels. Unlike to the baseline detection this energy has nothing to do with the real physical energy of the signal. It is just a representation of the average amplitude contained in the second interval.

In figure 21 the distribution of differences between the two intervals are shown for ECAL1 and ECAL2. Figure 22 is showing the same after average amplitudes have been calculated and translated into energy. The details about this algorithm will be discussed in the next section.

Based on the results of the two algorithms, the corresponding channel inside this event is tagged either for rejection or as good. The channel will not be automatically rejected due to this tag. The final decision is postponed after the clustering where all channels, being member of clusters with enough good channels, will be kept regardless if they have been tagged as noisy or not. The decision taking itself is a non trivial task, where many parameters have to be fine tuned and the results have to be understood.

After being sure to have a signal and knowing the baseline, the energy of the signals can be extracted. For this the amplitude of the signal is needed, which corresponds to the highest value in the sample set. This amplitude is multiplied with the respective calibration coefficient for the channel. The position of the maximum value has already been determined by the SADC decoding module and can be used now. Cinderella also calculates the energy for channels with an unknown baseline. in this case the first sample value is subtracted from the maximum sample value and this is taken as the amplitude.

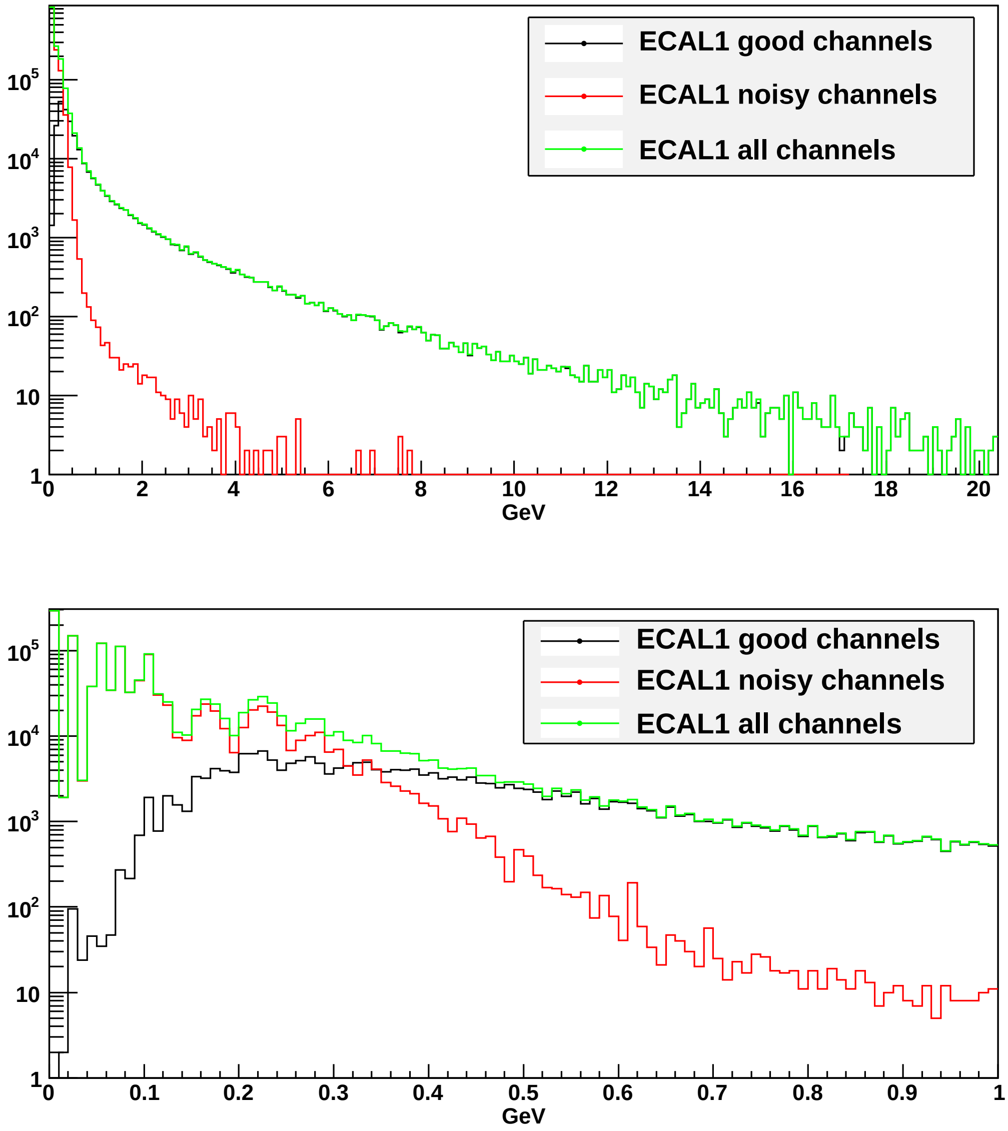


Figure 23. ECAL1 energy spectrum and zoom into low energy region

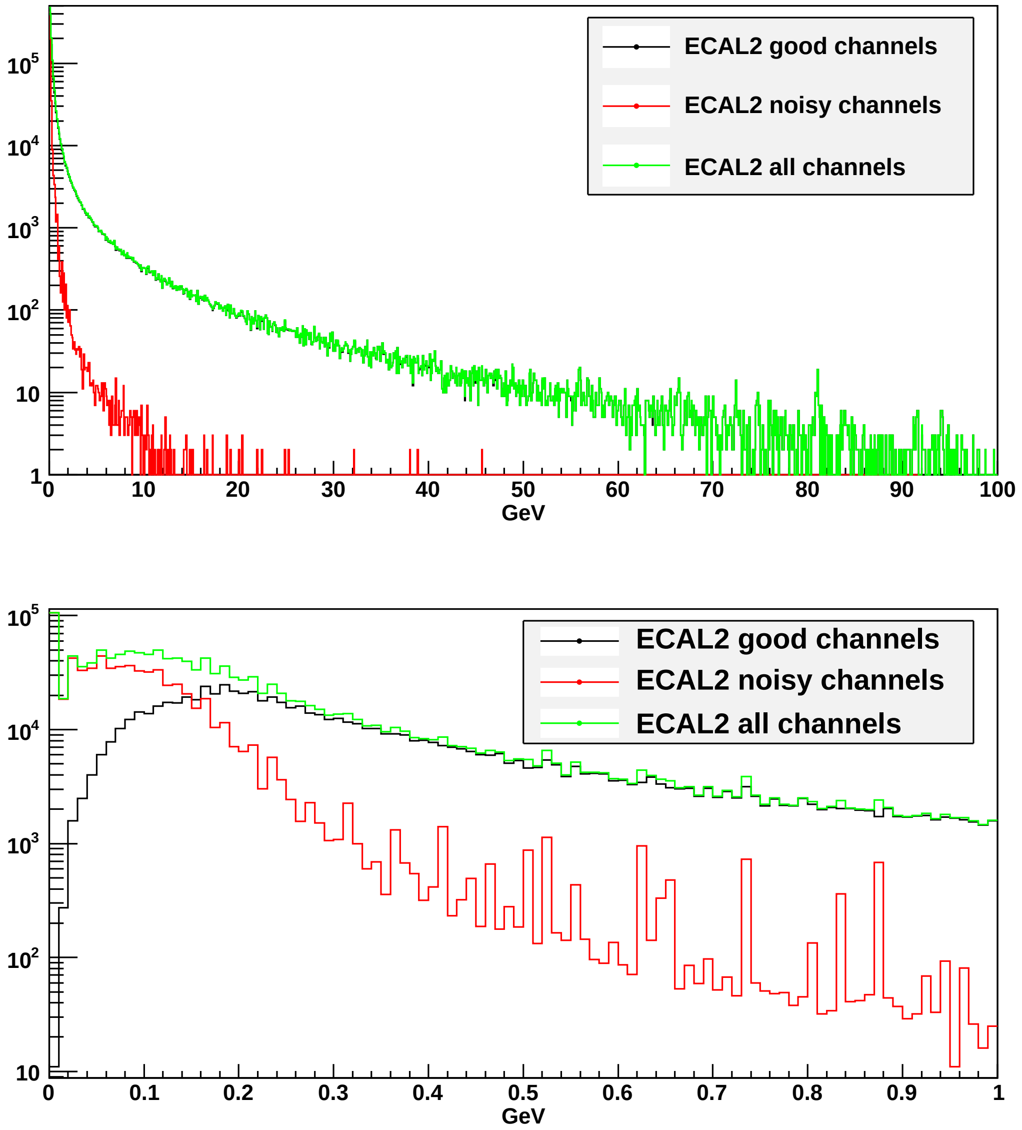


Figure 24. ECAL2 energy spectrum and zoom into low energy region

In the figures 23 and 24 the channel energy spectra of ECAL1 and ECAL2 are shown containing all channels, channels detected to carry a good signal and channels tagged as noisy. The algorithms for signal and noise detection have not been fine-tuned at this point. Also the decision making if a signal is noisy has been a simple “AND” between both algorithms (which is a very conservative setting that will be changed after fine-tuning). What can be already seen, most of the noisy channels in ECAL1 correspond to an energy below $\approx$ 350 MeV and in ECAL2 below $\approx$ 250 MeV. Above these energies the amount of noise seems to be negligible. The dynamic threshold amplitude for baseline detection has been set to the default value of 150 MeV and the average amplitude for alternative signal detection has been 5 ADC channels and was not translated to an energy at this point. With these thresholds the amount of channels tagged as noisy for ECAL1 was $\approx$80% and for ECAL2 $\approx$40%.

Beside the energy of the channels also their position in COMPASS coordinates is an important feature to be extracted. From the SADC decoding module the position in detector coordinates is known. The detector coordinates are integer values and indicate in which column and row the corresponding channel is located. To get the position within the COMPASS spectrometer in cm, the position of the "first" channel with detector coordinates (0,0), the size of every cell/channel and the distance between two channels is needed. This information is available from the so called "detectors.dat" file, where all geometric information concerning the COMPASS spectrometer can be found. In ECAL2 where all cells have the same size, same distance and no gaps in between, its just the multiplication of the corresponding mapping coordinate with the next cell distance added to the position of the first cell at (0,0). For ECAL1 it is more complicated, because its divided into three different parts with independent mapping coordinates, gaps inside two of the three parts and different cells sizes and distances. To be able to extract the right coordinates, Cinderella needs to know the first cell positions of all three independent parts and the location and distance of the gaps inside the two of three parts. Unfortunately the information which part of the mapping file belongs to which part of the "detectors.dat" geometry file is not available to be parsed automatically. It has to be specified in the configuration part for the SADC decoding module and is used by the ecal02_an module as a workaround. In the figures 25 and 26 the hitmaps of ECAL1 and ECAL2 in COMPASS coordinates are shown for all channels and for channels not tagged as noisy (good signals). Though the noise and signal detection has not been fine-tuned the improvement of the hitmaps is very clear. The different geometry of ECAL1 compared to ECAL2 can also be observed very nicely in these plots.

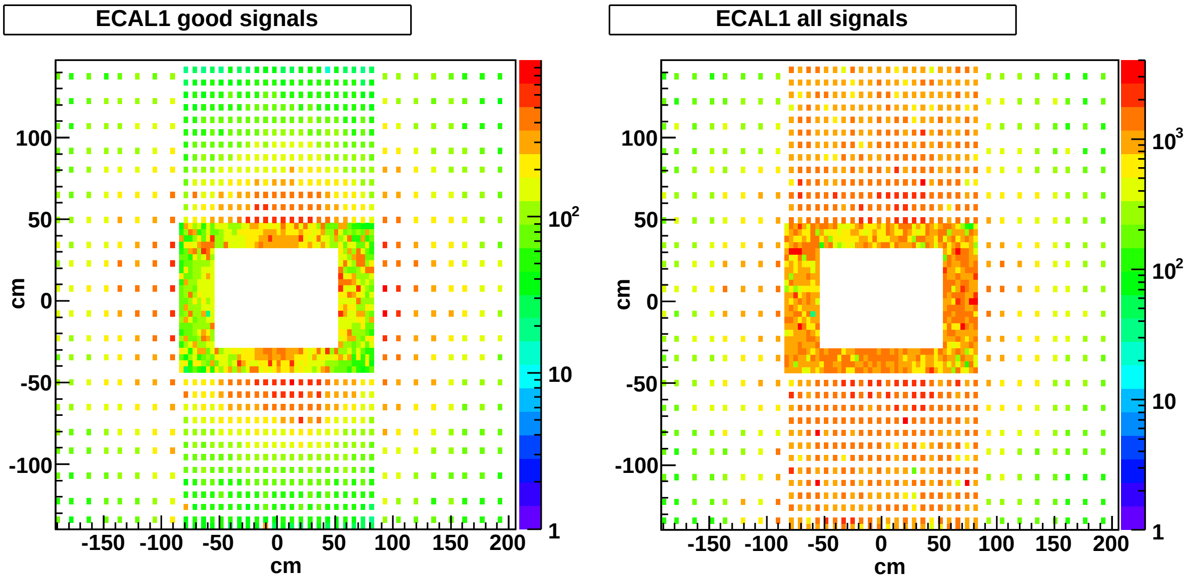


Figure 25. ECAL1 hitmaps for good (left) and all (right)

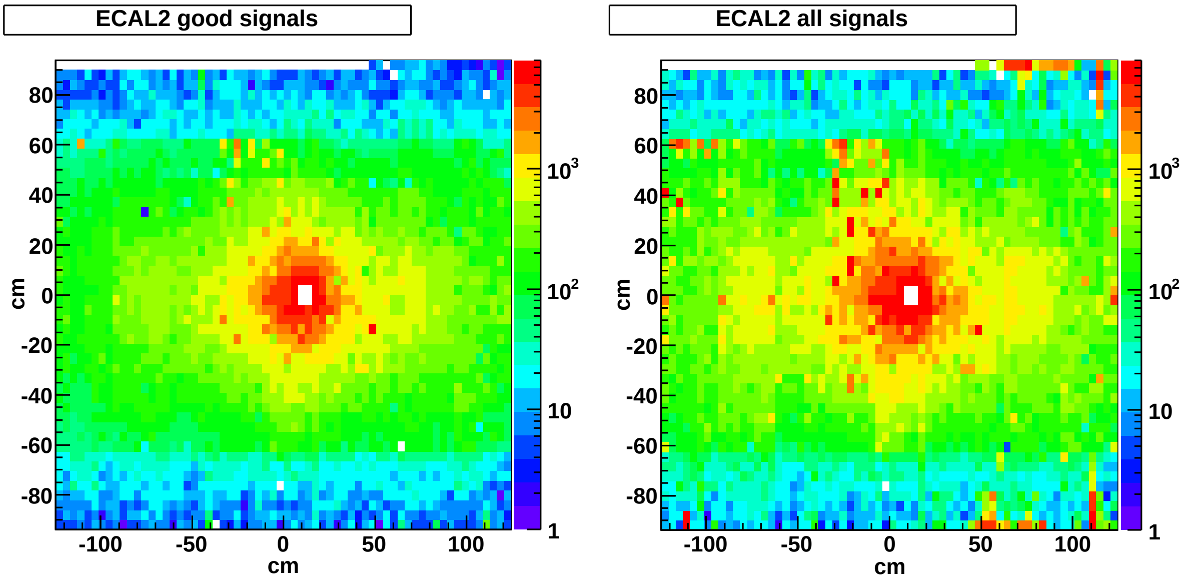


Figure 26. ECAL2 hitmaps for good (left) an all signals (right)

The ecal02_an module is also distinguishing between physics- and calibration events. Calibration events are arriving during off spill time triggered by Laser or LED pulses sent into the ECALs. This is useful for finding dead channels and checking the stability of the high voltage. Cinderella is providing this monitoring information by collecting data from calibration events over a predefined range of spills (10 spills per default) and writing the mean amplitude value of this period into a database. From this collected history deviations in high voltage, dead channels or even a turned off high voltage can be discovered by the DCS system.

At this stage all relevant information has been extracted from the available data and the “cluster” module can go on reconstructing clusters. Up to now all channels have been treated independently from each other, which will change in the next module.

6.4 Finding ECAL clusters

If a particle, like a γ, hits a cell in the ECAL, it creates an electro-magnetic shower which is propagating through this cell and also spreading into neighbouring cells. This causes firing of many cells per hit and all neighbouring cells grouped together are so called “clusters” representing one particle. Thus finding clusters inside the ECAL may be useful for analysing events.

The “cluster” module from Cinderella is able to find clusters among all channels from the ECALs by grouping all neighbouring cells together. Single cells are also considered to be a cluster, with a size of 1.

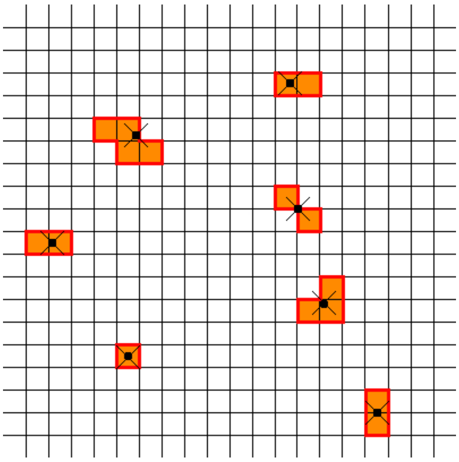


Figure 27. the “cluster” module is grouping all neighbouring cells together and calculating the centre of mass position

The energy of all cells inside one cluster is summed up and representing the cluster energy. As the position of a cluster, the position weighted with the energy from all cluster elements is taken.

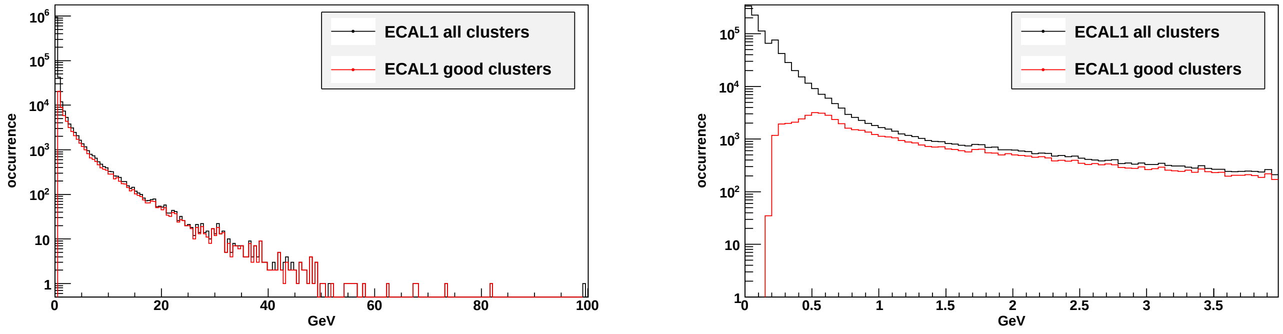


Figure 28. energy spectrum of clusters from ECAL1 (right) and a zoom into low energy area (left)

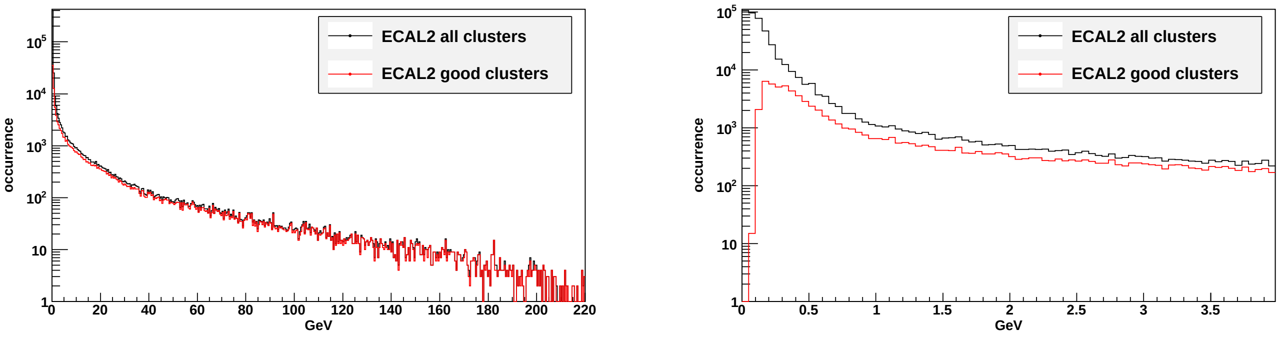


Figure 29. energy spectrum of clusters from ECAL2 (right) and a zoom into low energy area (left)

While creating the cluster information the “cluster” module is also counting the amount of good and bad channels and the energy contributed by these. Based on this value the whole cluster is tagged. If there are no good channels inside the cluster, the whole cluster is assumed as noisy. As long as there is one good channel with an energy > 0 GeV inside the cluster, the ratio of bad to good cells is taken into account and based on this all bad cells can be marked to be kept by the “evtmod” module. It also possible to take the ratio of good/bad energy for this decision, but the characteristics of noise are not well enough understood at the moment for this kind of decision.

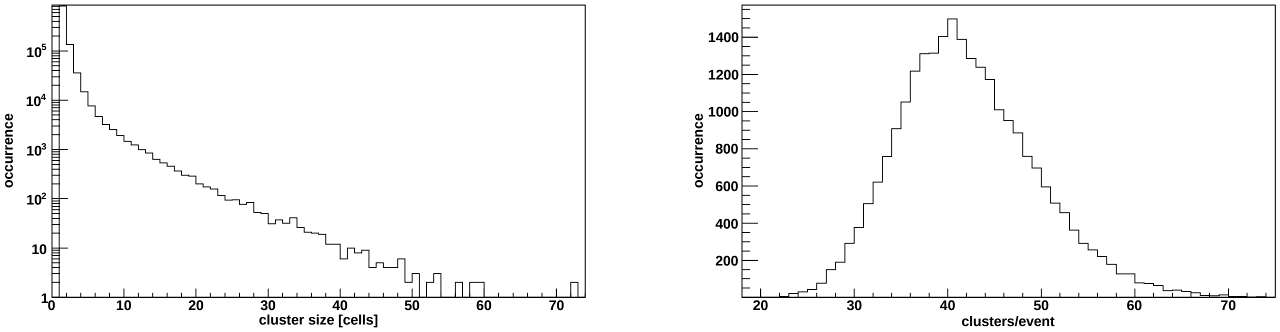


Figure 30. cluster size distribution and number of clusters per event for ECAL1

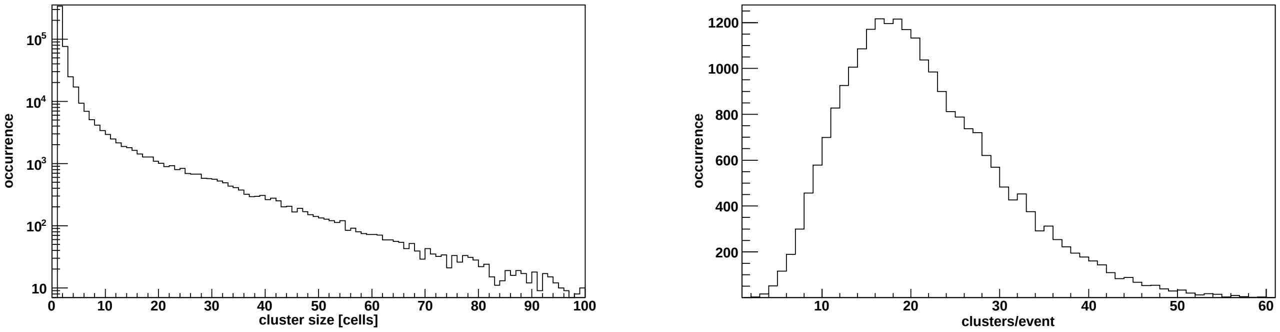


Figure 31. cluster size distribution and number of clusters per event for ECAL2

As mentioned above also single channels, not neighboured to other firing channels, are treated as clusters but they are recognised as “small” clusters. In figures 30 and 31 on the left side the cluster size distribution shows a big amount of clusters with a size of one cell. These “small” clusters are candidates for noise. They are represented by the single red spots in the good − bad distribution plots in figure 32. What is also remarkable is the comparison of the clusters/event plots from ECAL1 and ECAL2. It is showing almost twice as many clusters per event than ECAL2 by having only twice as less channels than ECAL2. Also the amount of “small” clusters is higher by a factor of 2 in ECAL1. This indicates that ECAL1 is more noisy than ECAL2.

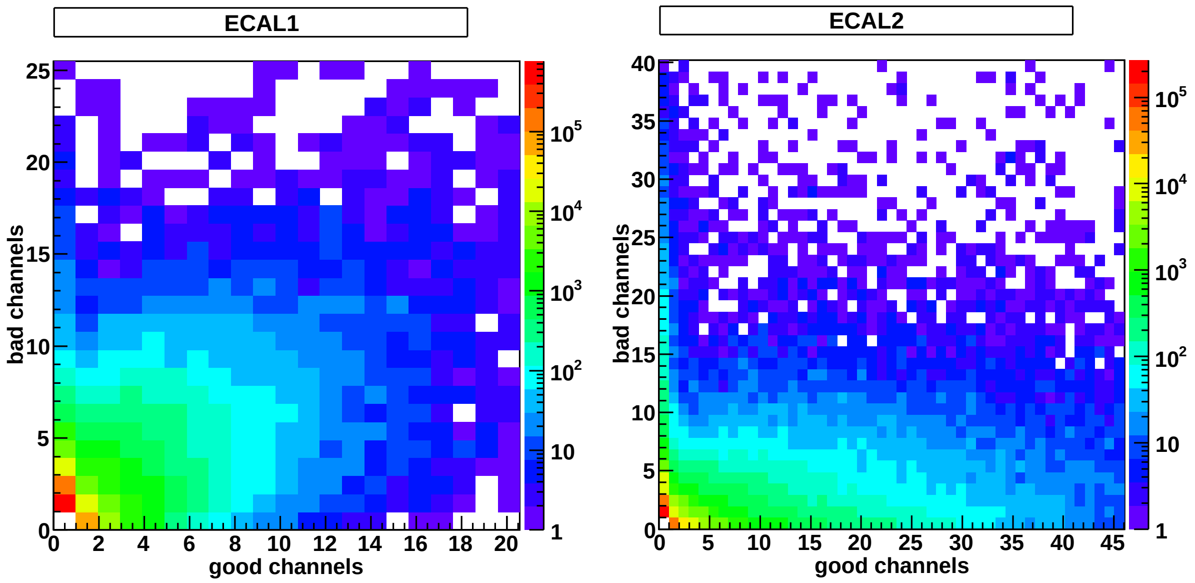


Figure 32. distribution of good and bad signals inside clusters form ECAL1 and ECAL2

In ECAL1 $\approx 50\%$ and in ECAL2 $\approx 20\%$ of all channels can not be connected to any cluster. A look at the noise tag from the ecal02_an module shows, $\approx$ 90%/80% (ECAL1/ECAL2) of these single channels are already tagged as noise. This is a strong indication, these channels do not belong to an ECAL shower from a particle hit. It can also happen two or more of these single cells are located to each other, so that they create a cluster with a size bigger than 1. These kind of clusters form a sharp edge in good/bad channels distribution plot on the “bad channels” side of ECAL2. In ECAL1 this edge is not as sharp as in ECAL2.

6.5 Invariant Mass reconstruction

After reconstruction of ECAL clusters, Cinderella obtained enough information for calculation of the four-vector of momentum for every particle which has hit the ECALs. To calculate the four-vector of momentum the total energy of the particle and its movement direction are needed. The total energy is already known from the cluster energy and the movement direction can be obtained from the connection of target position, which is available from the “detectros.dat” geometry file, and the position of the cluster. With these two parameters all momentum four-vectors can be constructed as following ($\vec{r}$: movement direction, E: cluster energy, $c = 1$) :

$$p^{\mu} = \left(\begin{array}{c} \frac{E}{c} \\ \frac{E}{c} \cdot r_x \\ \frac{E}{c} \cdot r_y \\ \frac{E}{c} \cdot r_z \end{array} \right) = \left(\begin{array}{c} E \\ E \cdot r_x \\ E \cdot r_y \\ E \cdot r_z \end{array} \right)$$

All four-vectors can be combined with each other which leads to $\binom{n}{2}$ combined four-vectors starting from n clusters in the beginning. The square of the invariant mass m is then the squared norm of the recombined four-vector.

$$m^2 = (p^{\mu})^2 = (p^0)^2 - (\vec{p}\,)^2$$

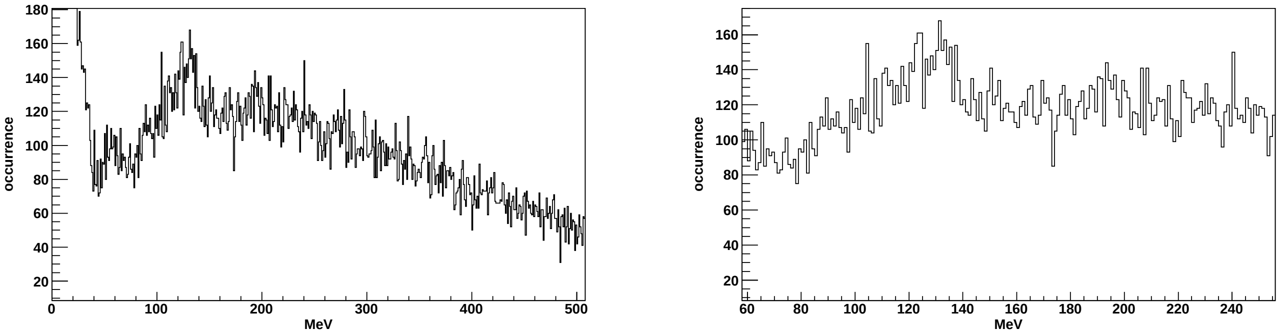


Figure 33. $\gamma\gamma$ invariant mass spectrum

Of course there is quite a lot of background in the invariant mass distribution of Cinderella. At the current stage Cinderella is not able to distinguish neutral from charged tracks. Every background cluster leads to **n-1** background entries in the histogram, because of combining it with all other clusters. But as a quick quality check it is completely sufficient.

7 Noise and Signals in ECAL 1+2

7.1 Performance and tuning of baseline and signal detection

In this chapter the performance and characteristics of the baseline and signal detection algorithms will be analysed. Both algorithms are dependent of the threshold parameters like the energy threshold for the “**B**ase**L**ine **D**etection” (**bld**) and the average amplitude for the “**S**ample su**M** **D**ifference “ (**smd**) algorithm, which can be also translated into energy. A quick feedback after changing these parameters can be retrieved from the hitmaps of the detectors, in which very noisy channels can be identified immediately. For better understanding also a look directly on the pulse shapes is needed, to see what kind of signals will be tagged as noisy an where the limits of the used algorithms are.

The unfiltered hitmaps of ECAL1 and ECAL2 indicate the presence of noise in these detectors. In ECAL2 there are mostly hot cells seen in the hitmap. But the beam profile around the hole can be recognised clearly and beside some artifacts in the region left and right of the hole and the sharp edge between SADC and MSADC channels the distribution looks more or less like expected.

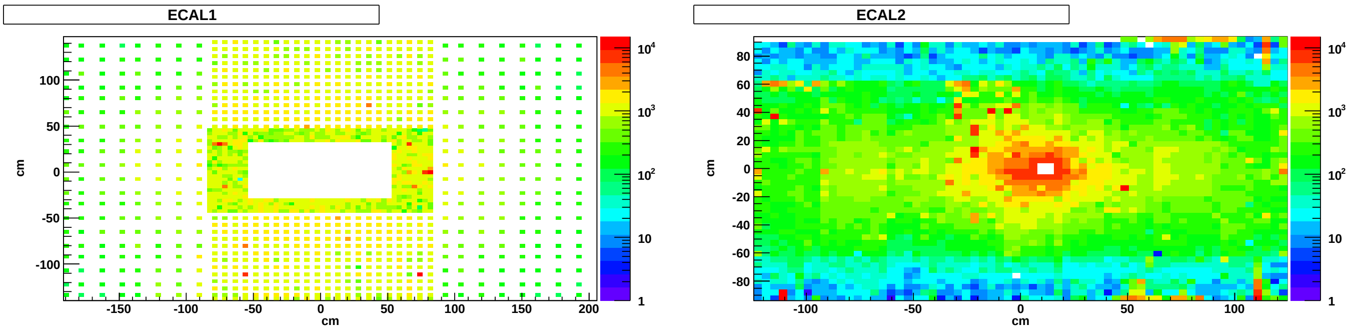


Figure 34. unfiltered hitmaps for ECAL1 and ECAL2

The situation for ECAL1 looks different. The beam profile can only be seen in the outer “OLGA” part, left and right of the centre. But in the central part the hits seem to be distributed more or less equally, the beam profile is not obvious.

7.1.1 Tuning the BaseLine Detection (bld) algorithm

The main task of the **bld** algorithm is, as mentioned in the chapter about feature extraction, the identification of the signals starting point and to calculate the baseline value from the samples before. It was shown the signal detection in this algorithm corresponds to a cut on the signal shape and to an amplitude cut, which can be translated to an energy in MeV. This value means all signals showing the requested characteristics and matching an energy higher or equal than the threshold, will be recognised. If the threshold is too high, the probability not to recognise a good signal (false negative) will rise, if it is too low more noisy signals will be detected as good (false positive). Hitmaps, produced with different thresholds, can give an idea about the region in which the threshold should be. In the following hitmaps one data chunk of the run number 70502 ($\approx$ 25000 events) from 2008 was processed at different thresholds and all channels the

bld algorithm was not able to find a good signal have been filtered out.

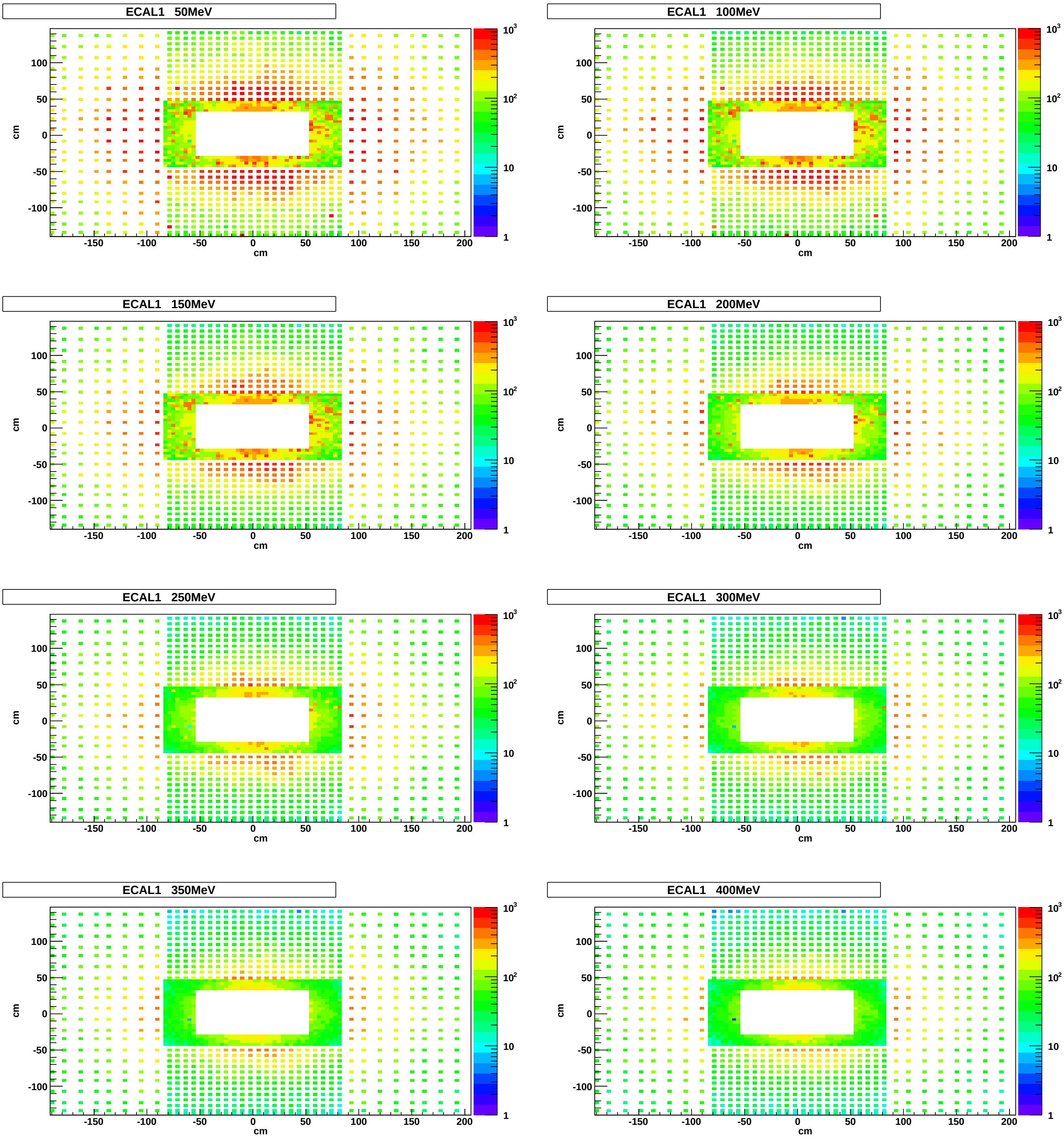


Figure 35. hitmaps for ECAL1 at different bld energy threshold

Compared to the unfiltered hitmap even the hitmap with 50 MeV threshold shows a big improvement. The beam profile gets visible in the central part and there are just some hot cells left. These hot cell disappear almost completely at 250 MeV. Since the final decision for noise tagging will be made together with the smd signal detection and the clustering information, the bld threshold can be set to 200 MeV or even 250 MeV. A higher threshold makes no sense, because above these values no artifacts/hot cells are left anymore and the growth of the reduction factor is decreasing.

threshold (MeV)	good channels	noisy channels	reduction factor (%)
0	1506857	0	0.00%
10	353422	1153435	76.6%
50	351861	1154996	76.7%
100	288786	1218071	80.8%
150	242350	1264507	83.9%
200	190651	1316206	87.4%
250	156579	1350278	89.6%
300	133731	1373126	91.1%
350	118895	1387962	92.1%
400	105813	1401044	93.0%
450	96151	1410706	93.6%
500	87902	1418955	94.1%

Table 3. results of the bld algorithm threshold scan for ECAL1

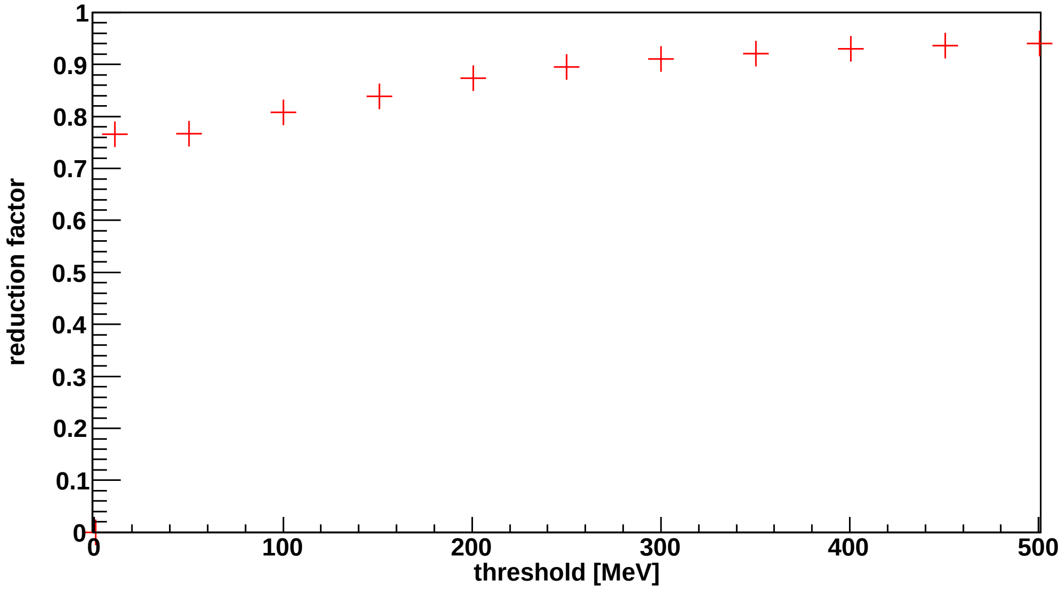


Figure 36. ECAL1 reduction factor of the bld algorithm at different thresholds

Table 3 shows the reduction ratios for different thresholds on ECAL1 data. The biggest change in reduction is in the 50 – 250 MeV interval, above 250 MeV the amount of additionally reduced data is negligible. To get an idea about the performance of the algorithm at thresholds of 200 MeV and 250 MeV a control sample of 100 signals detected as good and 100 signals detected as noisy was taken and the number of false negatives, false positives and uncertain cases (like channels with very low amplitudes, which may carry a signal but could also be noise) will be counted. For this every signal has been printed out and checked manually. The signals have been chosen randomly for both thresholds from one date chunk. Since the bld algorithm is not time sensitive it was supposed detect all signals regardless if they are out of time. All these results have been crosschecked by the (M)SADC expert, Igor Konorov.

threshold	200 MeV	250 MeV
false negatives	3	1
false positives	1	2
uncertain cases	7	7

Table 4. results of the control sample for ECAL1

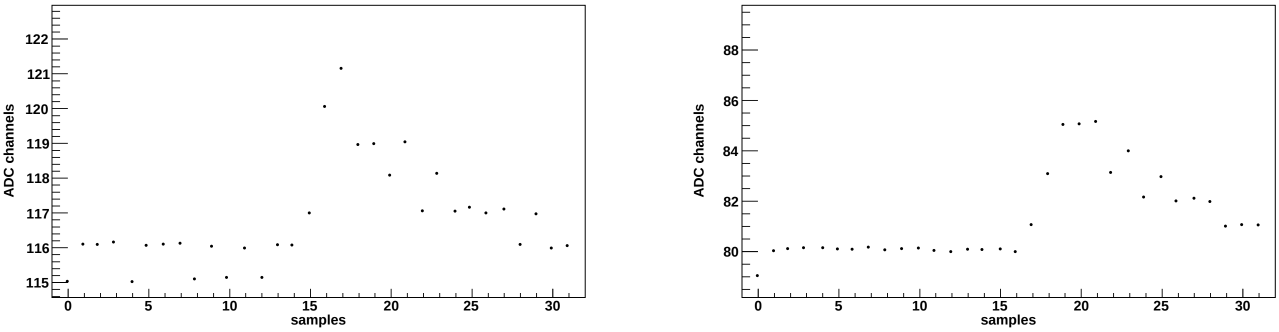


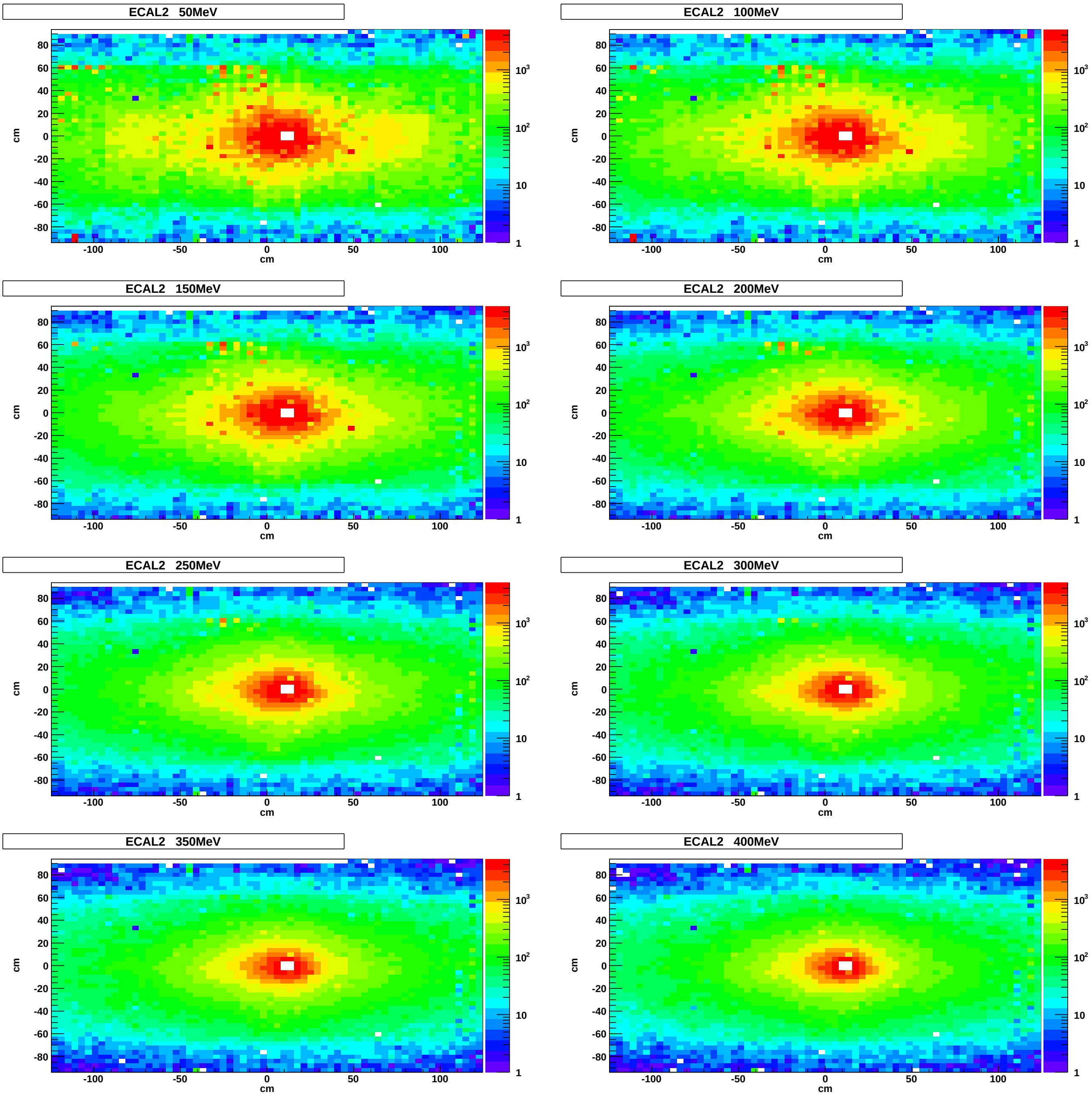
Figure 37. uncertain cases

In table 4 the results of the control sample for the two different thresholds can be compared. All values are in the same order of magnitude for both thresholds and the number of uncertain cases is bigger than the clear false cases. For physical analysis with the ECALs the number of false negatives is critical. A loss of good channels means a loss of cluster energy and this can lead to wrong reconstruction which can make the complete event useless. As Cinderella is operating online and everything rejected is lost forever, the number of false negatives should be reduced to zero for optimal conditions. All false negatives in this control sample have been good signals with an energy below the energy threshold. To save these kind of signals a possibility is to use an additional algorithm, which is working in a different way like the smd algorithm. Beside of the smd algorithm there is a second feature for saving good channels in Cinderella. Low energy channels usually belong to a cluster consisting of other channels with a higher amplitude. As already mentioned in 6.4, all channels belonging to clusters with enough good channels of high energy will be kept regardless of their noise tag. The decision which method will save these small signals will be made later during the fine tuning of the algorithms.

False positives are not this critical, because they still can be rejected at the offline stage, if they really harm. False positives can happen with the bld algorithm if there is low frequent noise with an high enough amplitude which corresponds to an energy higher than the threshold.

About the uncertain cases it is difficult to judge. From the signal shape it is not 100% clear if their origin is really an electro magnetic shower or its something else which is not of physical interest. In this control sample all uncertain cases have been classified as noise. If they are good signals with a very low amplitude they should be part of a cluster with definitely good channels of significantly higher amplitude. In this case the good/bad ratio safety check, in the "cluster" module of Cinderella, will safe them from being rejected like in the case of false negatives.

All this checks can also be performed for ECAL2. As already mentioned the unfiltered ECAL2 hitmap looks better than the ECAL1 hitmap, so its assumed to have a lower capability of data reduction for ECAL2.



ECAL2 50MeV

ECAL2 100MeV

ECAL2 150MeV

ECAL2 200MeV

ECAL2 250MeV

ECAL2 300MeV

ECAL2 350MeV

ECAL2 400MeV

Figure 38. hitmaps for ECAL2 at different bld energy threshold

Looking at the hitmaps in figure 38 a similar situation can be observed like for ECAL1. The hit distribution is getting smoother and the beam profile in the centre is improving with higher thresholds. At 250 MeV almost all hot cells and the artifacts left and right of the hole have disappeared. But already at 150 MeV most of the detectors channels seem to behave well except a small fraction of single cells. It looks sensible to set the threshold between 150 and 200 MeV to detect as many bad signals as possible and avoid a too big amount of the critical false negatives.

threshold (MeV)	good channels	noisy channels	reduction factor (%)
0	1626956	0	0.00%
10	1066546	560410	34.5%
50	1038149	588807	36.2%
100	916247	710709	43.7%
150	752384	874572	53.8%
200	624034	1002922	61.6%
250	532691	1094265	67.3%
300	458629	1168327	71.8%
350	410261	1216695	74.8%
400	369042	1257914	77.3%
450	337803	1289153	79.2%
500	313080	1313876	80.8%

Table 5. results of the bld algorithm threshold scan for ECAL2

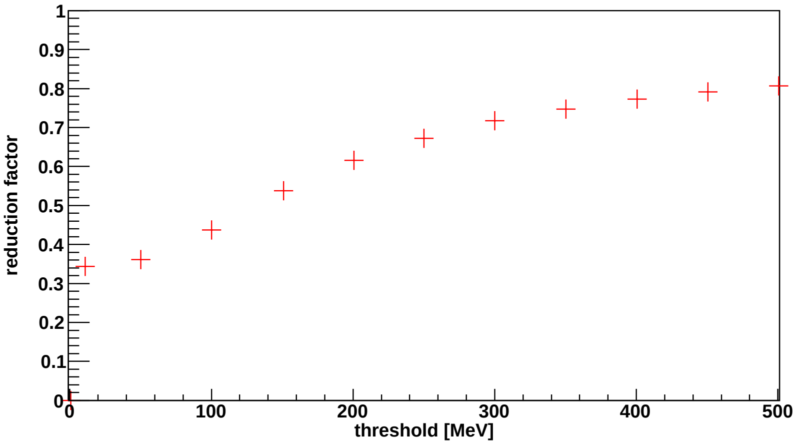


Figure 39. ECAL2 reduction factor of the bld algorithm at different thresholds

Table 5 shows the reduction ratios for ECAL2 at different thresholds. They are, as expected, lower than for ECAL1 but the region of the biggest increasing of the the reduction factor is between 50 − 300 MeV which is similar to ECAL1. The control sample gave the following results:

threshold	150 MeV	200 MeV
false negatives	10	19
false positives	3	2
uncertain cases	6	3

Table 6. results of the control sample for ECAL2

The assumption that ECAL2 is mostly working well between 150 – 250 MeV thresholds seems to be true. At least the results of the control sample can be interpreted like this by looking at the numbers of false negatives. Their number is almost twice as high at 200 MeV than at 150 MeV which indicates the number of good signals at an energy lower than 200 MeV is not negligible. Even at 150 MeV the number of well working channels is significant. Taking this into account the recommended bld threshold for ECAL2 should not be higher than 150 MeV otherwise it could get difficult to save the good signals after the clustering.

The above results, especially for ECAL2, are showing a non homogeneous picture of performance and efficiency of the ECALs. At ECAL1 most of the channels seem to work well at 200 MeV while the last hot cell disappears between 300 – 350 MeV. For ECAL2 it is a similar case, most channels are working well at a signal level of 100 MeV while the last hot cell can be observed at 250 MeV.

The conclusion is there can be no 100% efficient noise algorithm for the ECALs which is operating with the same threshold on all channels. At the first view this may sound very bad, because applying individual thresholds on every channel is not trivial. But an inefficiency of an algorithm is not crucial as long it can be corrected by other algorithms.

7.1.2 Tuning the smd algorithm

Compared to the bld algorithm, smd is working in a different way. Instead of looking for the rise of the signal at any point of the sample set, it is just calculating the difference of two intervals at fixed positions. The main purpose of the smd algorithm is not to find the baseline of the channel but to detect if the channel is carrying a signal. One interval is fixed at sample position 2 the second one at position 10, both have a length of 5 samples. According to Igor Konorov, every signal which is in time should start at least between sample 10 and 15. This makes the smd signal detection time sensitive, because all signals starting later than sample 15 are definitely too late and signals starting before sample 10 have still an amplitude unequal to 0 after sample 10.

Remark 2. The second interval can be configured to scan a certain range of samples to accept a larger time window of signals. In the basic tuning procedure it is not necessary to turn this feature on. This will be done later at the fine tuning stage to keep it simple at the moment. The only difference between fixed position and scanning is the number of detected out of time signals.

By dividing the difference of the two intervals by their length the average amplitude of the signal can be obtained. This average amplitude is the threshold for signal detection which has to be tuned. This average amplitude can also be translated into MeV with the help of the calibration coefficient. To find the right starting point for threshold tuning, the ECAL hitmaps and the average amplitude distribution may help to find a starting point.

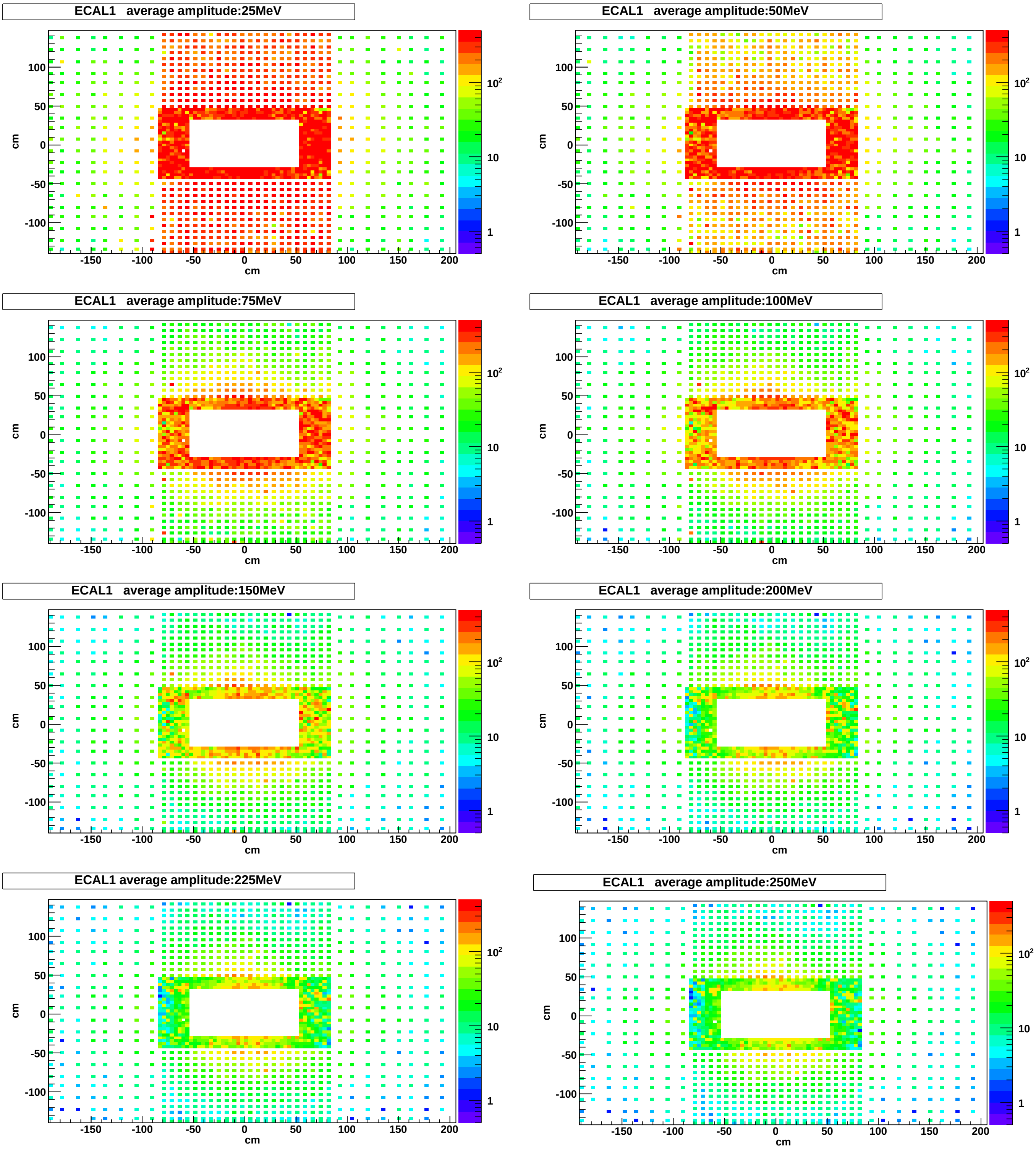


Figure 40. ECAL1 hitmaps for different average amplitudes

In figure 40 the different noise level of the three ECAL1 parts can be observed. The outer OLGA part looks fine for average amplitudes around 50 MeV. The middle top and bottom MAINZ part improves between 75 – 100 MeV while the central GAMS area is not even at 250 MeV completely cleaned up from hot cells. This makes choosing the proper threshold difficult. Since the bld algorithm was more effective in the central region, the bld and smd algorithms should be combined in a smart way with rather lower than too high thresholds. So the smd threshold should be optimised for the OLGA and MAINZ part and the inefficiency in the inner part will be cleared by the bld.

average amplitude (MeV)	good channels	noisy channels	reduction factor (%)
0	1506857	0	0.00%
25	484335	1022522	67.9%
50	327260	1179597	78.3%
75	201399	1305458	86.6%
100	142541	1364316	90.5%
125	107028	1399829	92.9%
150	83116	1423741	94.5%
175	66967	1439890	95.6%
200	55783	1451074	96.3%
225	47793	1459064	96.8%
250	41644	1465213	97.2%

Table 7. results of the smd algorithm threshold scan for ECAL1

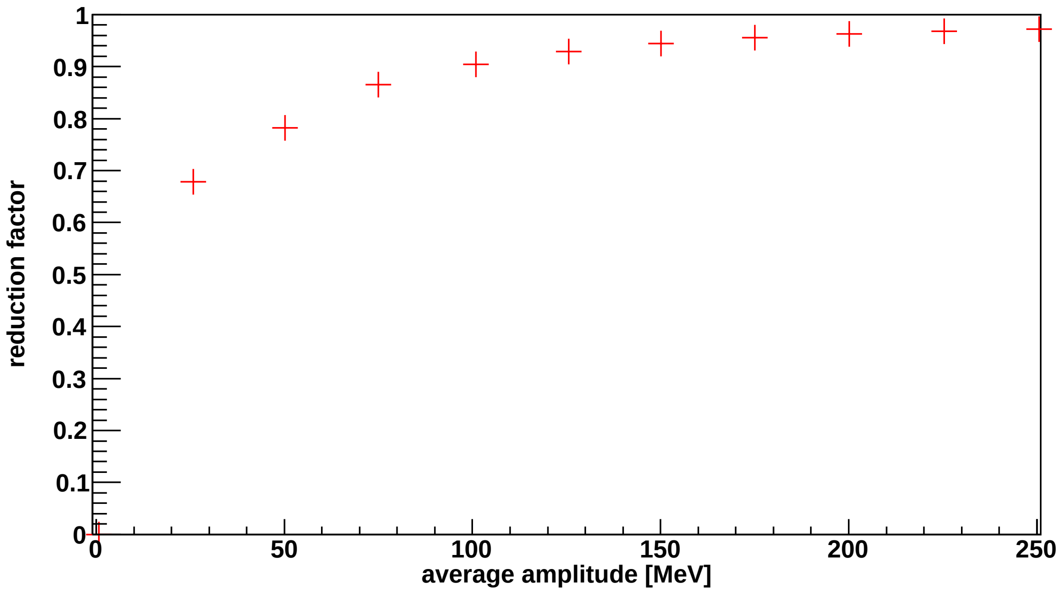


Figure 41. ECAL1 reduction factor of the smd algorithm at different thresholds

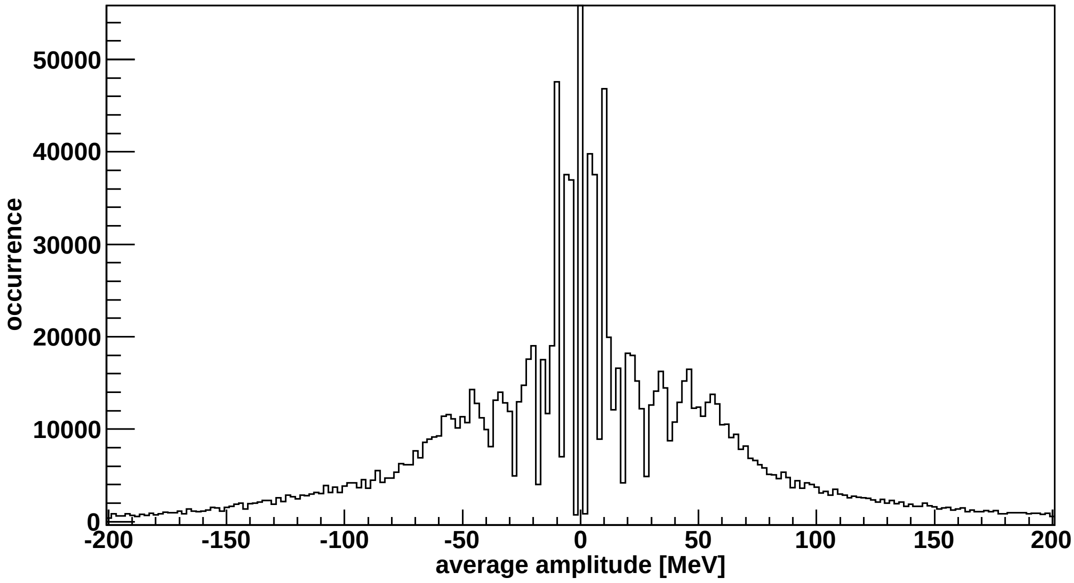


Figure 42. spectrum of average amplitudes for ECAL1

Figure 42 can give a useful hint for choosing the threshold. In the area from $-$ 60 to 60 MeV symmetric peaks can be observed. This effect appears when the difference between the two intervals is very low (e.g. like from -5 to $+5$). To get the average amplitude the difference has to be divided by the interval length, which was 5 in this case, and for low differences the result is something like $\pm 0.2, \pm 0.4, \pm 0.6, \pm 0.8$. This is multiplied with the calibration coefficients which are mainly distributed over 3 regions: 20 – 24, 25 – 30 and 50 – 60 MeV/ADC count (see figure 18 at page 25). It can be checked easily that the result of this multiplication causes the creation of these symmetric peaks in this range. Good signals can not be found in this region, which is feasible due to the low average amplitude. This structure disappears after 70 MeV and considering the improvement of the MAINZ part at 75 MeV, the region from 75 MeV seems to be a good threshold candidate.

The control sample was taken for the thresholds 75 and 100 MeV with the following results:

threshold	75 MeV	100 MeV
false negatives	0	0
false positives	51	54
uncertain cases	1	1

Table 8. results of the control sample for ECAL1

These results are showing a big contrast to the control sample of the bld algorithm. Benefits of the smd algorithm are the time sensitivity, making it able to reject out of time signals, and the calculation of the average amplitude, which is a good protection against false negatives. The disadvantage is the great number of false positives, which are all of the same kind of signal. It looks like a moving baseline with a change high enough to make average amplitude higher than the threshold.

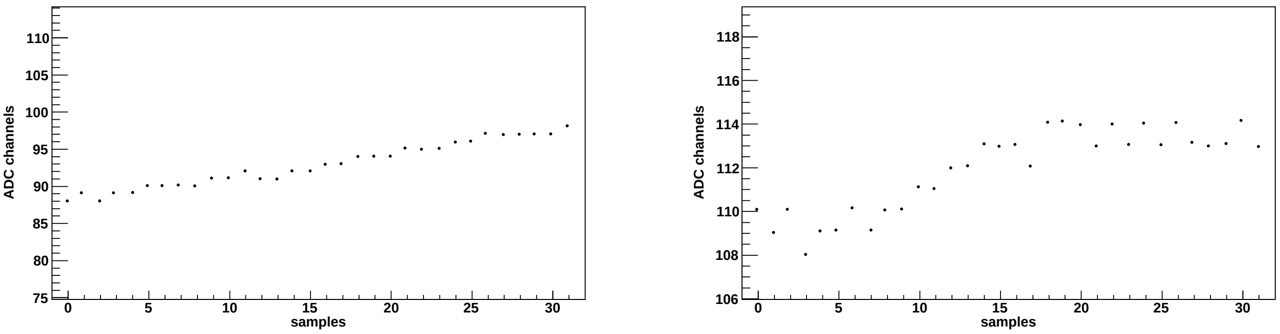


Figure 43. false positives of the smd algorithm for ECAL1

Figure 43 is showing two examples of these signals. All other false positives look more or less the same. For the smd algorithm these kind of signals are very difficult to distinguish from good shower signals but they represent no challenge for the bld baseline detection. A good idea could be to let the two algorithms work together in a way balancing out each others disadvantages. For detecting noise the smd seems to be a good candidate and the bld may be used as a “cleaner” of the good signals detected by smd.

But it might be interesting to see how smd performs at ECAL2 first. The hitmaps in figure 44 indicate a good performance of most of the ECAL2 channels already at a threshold of 75 MeV average amplitude.

The reduction factor at the 75 MeV threshold is similar to the reduction factor of the bld algorithm at a threshold of 150 MeV. Many hot cells and the artifacts left and right of the central hole have disappeared. The hot cells which are still left seem to be difficult to remove by an uniform threshold for all channels. Contrary to ECAL1 the average amplitude spectrum in figure 46 can not give a very good hint for choosing the threshold, because the calibration coefficients of ECAL2 are distributed over a wide range (figure 19 at page 25), which prevents such an obvious structure like for ECAL1. The first and maybe second order of the “low difference” peaks can be recognised between -20 to 20 MeV but this is far away from 75 MeV.

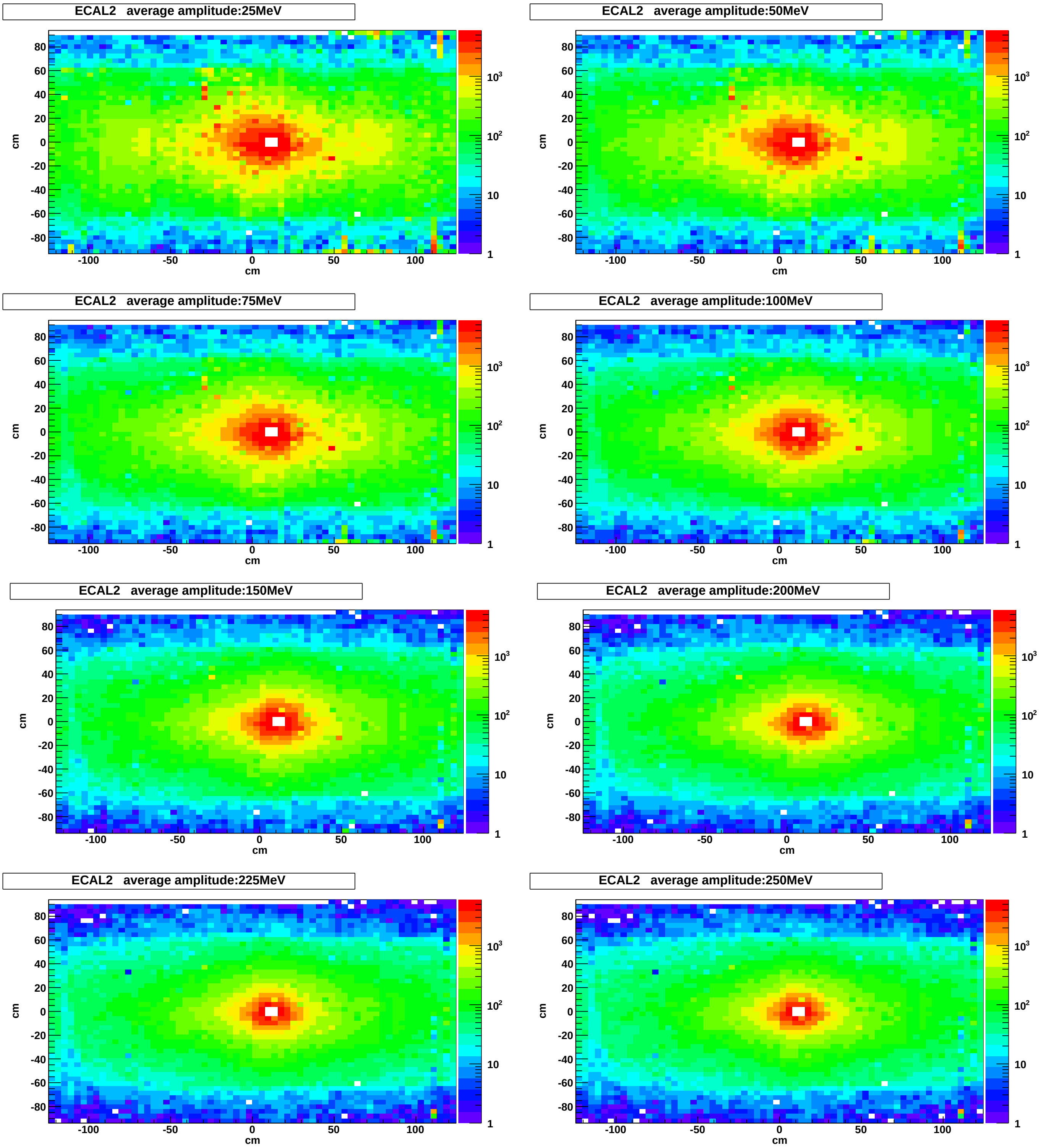


Figure 44. ECAL2 hitmaps for different average amplitudes

average amplitude (MeV)	good channels	noisy channels	reduction factor (%)
0	1626956	0	0.00%
25	914488	712468	43.8%
50	792790	834166	51.3%
75	695763	931193	57.2%
100	615117	1011839	62.2%
125	547457	1079499	66.4%
150	488983	1137973	69.9%
175	441372	1185584	72.9%
200	402044	1224912	75.3%
225	370364	1256592	77.2%
250	343779	1283177	78.9%

Table 9. results of the smd algorithm threshold scan for ECAL2

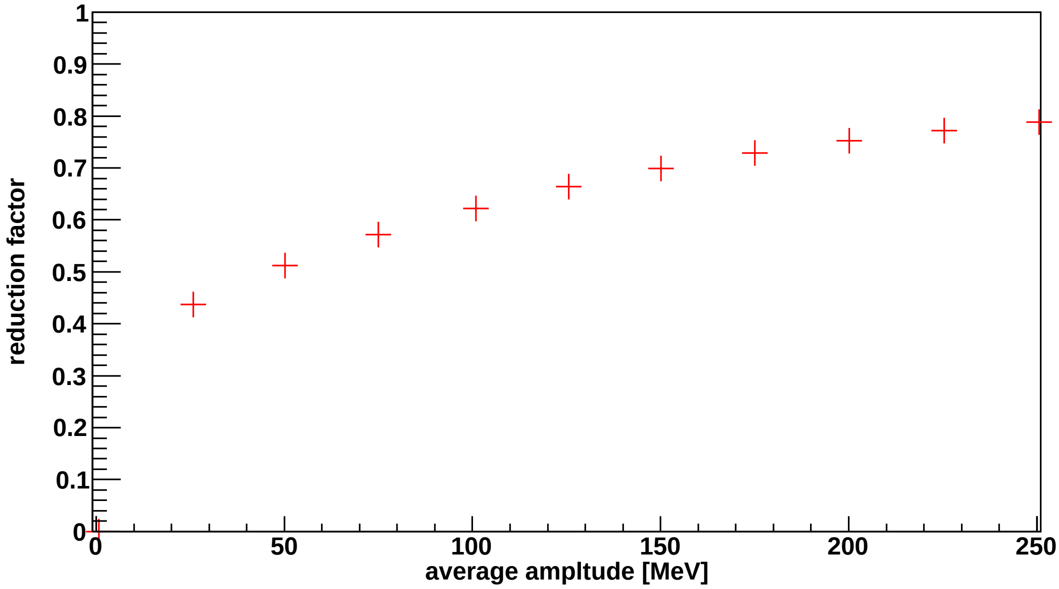


Figure 45. ECAL2 reduction factor of the smd algorithm at different thresholds

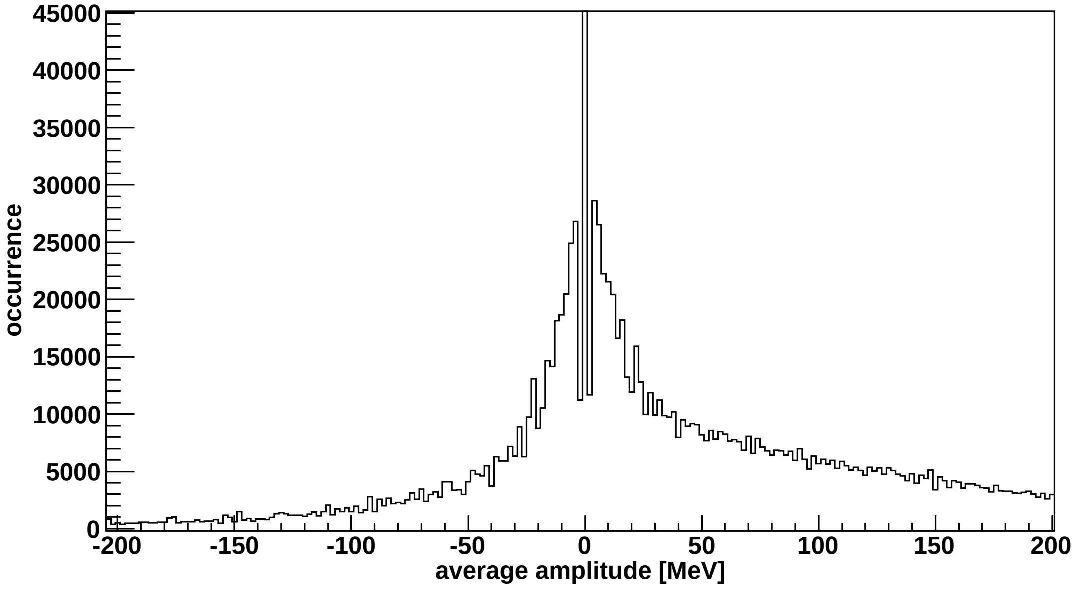


Figure 46. spectrum of average amplitudes for ECAL2

The control sample was made with thresholds at 75 MeV and 100 MeV, like for ECAL1.

threshold	75 MeV	100 MeV
false negatives	3	3
false positives	12	12
uncertain cases	4	13

Table 10. results of the control sample for ECAL2

The false negatives are mostly signals which are beginning late in the second interval, like at sample 13 or 14. Since the timing can be different from channel to channel these kind of signals should be kept in any case. Due to the late beginning only a part of the amplitude is included which leads to a too small difference in respect to the first interval. This can be avoided if the second interval is not fixed at one position but scanning a range of samples in an interval where in time signals are expected. This interval should, according to Igor Konorov, include all samples from sample 10 to 20. But the number of false negatives is, like in ECAL1, much smaller than for the bld algorithm, which confirms the observations about advantages and disadvantages of the two algorithm from the ECAL1 part.

The conclusion of the threshold tuning is, bld has better performance in detecting good signals with a lower false positive rate than smd, which has a lower false negative rate in detecting noise and is able to detect out of time signals. There are various possibilities to let these two algorithms work together.

7.2 Comparing bld and smd

Now there are two different working algorithms available, each with its own advantages and disadvantages. The interesting question is now: How good do both coincide?

To get an idea about this, the same data chunk like in chapter 7.1 will be processed, while counting the cases in which both algorithms coincide and in which their decision differs. The used thresholds for each algorithm and detector are listed in table 11. The only difference is, that the scanning ability of the smd algorithm was turned on to classify also late beginning signals, which begin around samples 14 or 15, as good. The second interval is now scanning the samples from 10 to 16 with a step width of 2 samples and choosing the position with the biggest difference to the first interval. Due to the interval length of 5 samples the sum of the second interval is covering all samples from 10 to 20. The result of this feature is a lower number of rejected channels by smd because some signals considered to be out of time before are now detected as in time.

detector	bld (MeV)	smd (MeV)
ECAL1	200	100
ECAL2	150	75

Table 11. threshold overview

ECAL1	rejected channels	% of all channels
bld	1316206	87.4%
smd	1150697	76.4%
agreement	1105522	73.4%
conflict	255859	17.0%

Table 12. coincidence of bld and smd algorithm in ECAL1

ECAL2	rejected channels	% of all channels
bld	874588	53.8%
smd	775722	47.7%
agreement	641750	39.4%
conflict	366810	22.5%

Table 13. coincidence of bld and smd algorithm in ECAL2

Table 12 and 13 are showing that the bld and smd algorithm are getting the same results for most of the channels. These channels can be assumed to be noisy. But the channels where the results of both algorithms are differing have to be investigated to be able to decide what should happen with them.

In most of the cases when smd detects noise and bld a signal there was a good signal but it is clearly out of time. The rest of these cases are noisy with a very high amplitude corresponding to more energy than the bld threshold. This picture can be observed for both ECALs, so it seems to be safe when tagging a signal for rejection in this case.

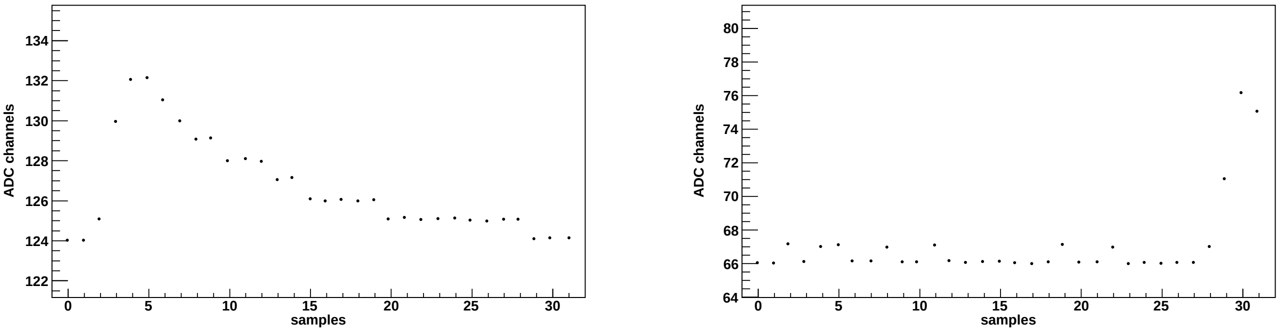


Figure 47. out of time signals from ECAL1

To detect out of time signals the time window of the smd algorithm has been set from sample 10 to 20. If it turns out there can be in time signals outside this time window, it can be adjusted easily.

The other case is if bld detects noise and smd a good signal. Most of these signals are false positives from the smd algorithm as shown in section 7.1. But there are also signals with a small amplitude corresponding to an energy below the bld threshold of 200 MeV for ECAL1 and 150 MeV for ECAL2 (see false negatives from bld algorithm in section 7.1). Since their energy is very low, they should be member of a cluster with signals of higher amplitude/energy and can be "rescued" in the cluster module of Cinderella.

A look at the figures 48 and 49 shows the energy spectra for good and rejected channels of both algorithms. In ECAL1 almost all channels with an energy below 100 MeV are rejected by both algorithms. What seems to be strange is that not all channels are rejected from the bld algorithm below 200 MeV, which was the threshold. A look on the pulse shape at a later stage will give an answer on that. The region from 200 to 500 MeV is dominated by the rejection of the bld algorithm. From this interval most of the false positives, like shown in figure 43, of the smd algorithm are coming from. At higher energies the rejection factor of smd is of course bigger, because its also rejecting out of time signals with significant energy.

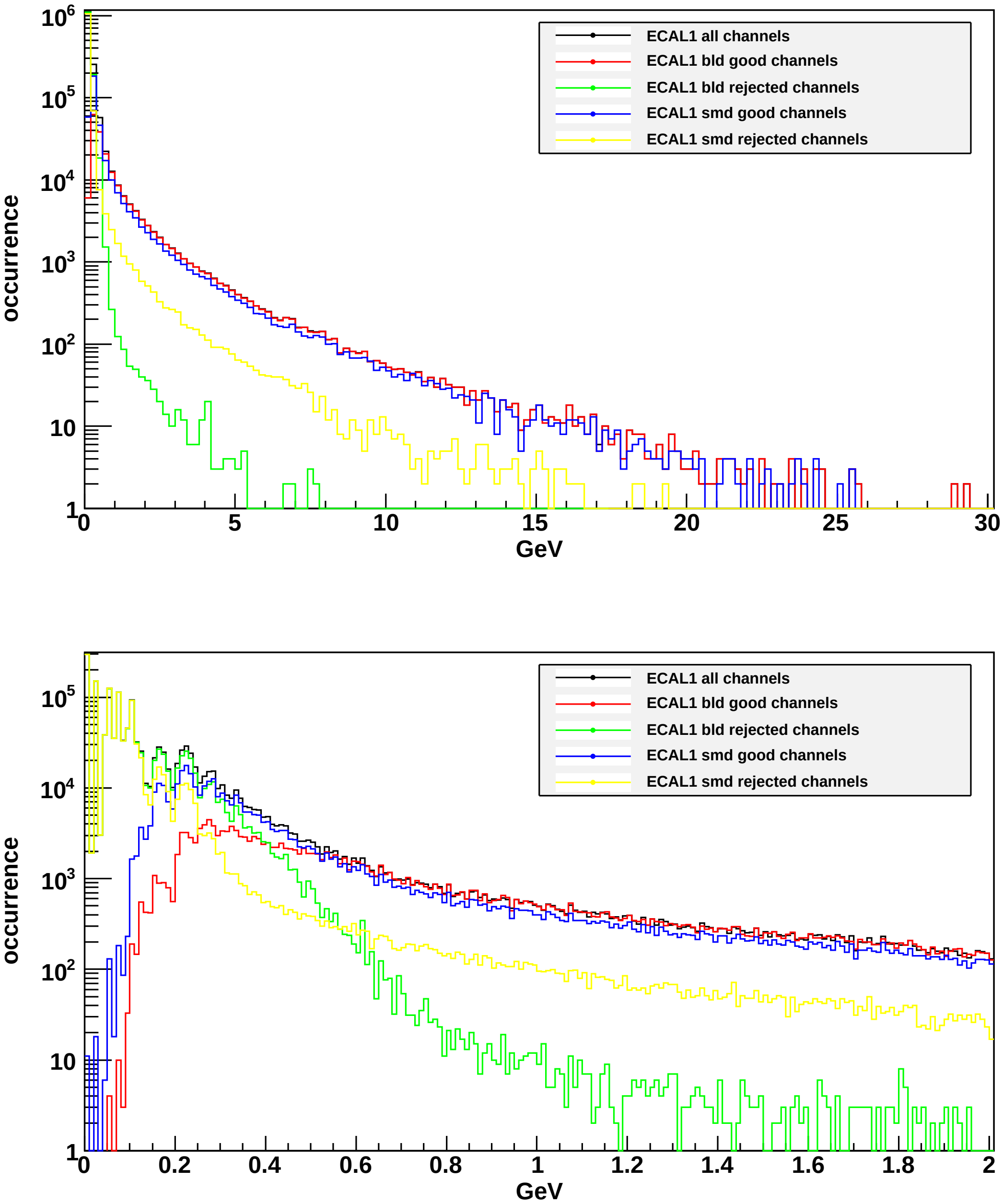


Figure 48. ECAL1 energy spectrum separated by algorithm

Compared to the ECAL1 energy spectrum the ECAL2 spectrum is very similar. Almost no good channels are found below 50 MeV but a significant amount of good channels detected by the bld algorithm below its threshold of 150 MeV. From 100 to 250 MeV bld is rejecting more than smd. One reason are the false negatives of the bld algorithm right below the threshold of 150 MeV and above the significant amount of false positives from the smd algorithm is the reason. At higher energies smd is rejecting more channels like in ECAL1.

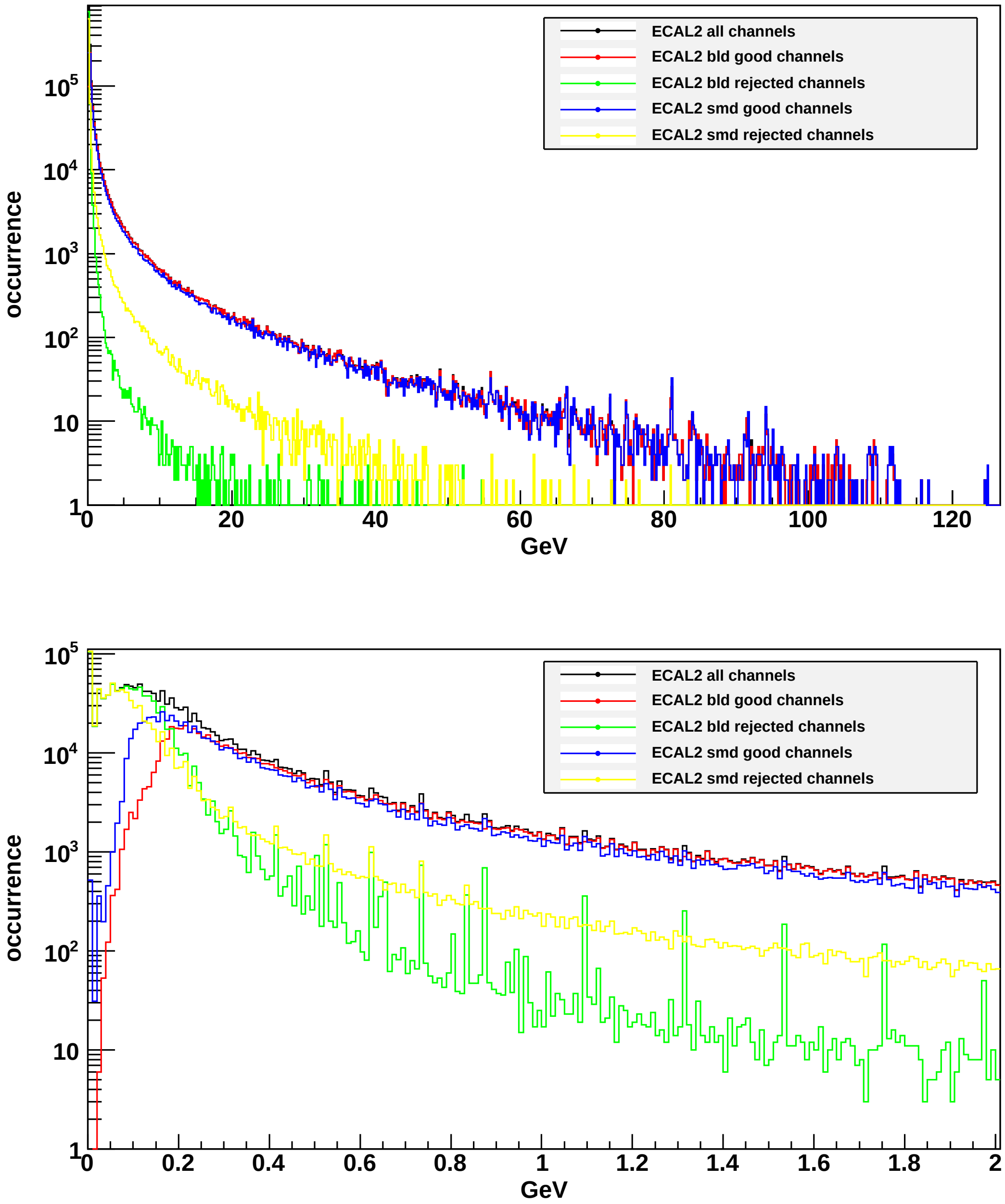


Figure 49. ECAL2 energy spectrum separated by algorithm

The remaining question is, how can bld accept signals below its threshold? A look at the pulse shape can explain this case.

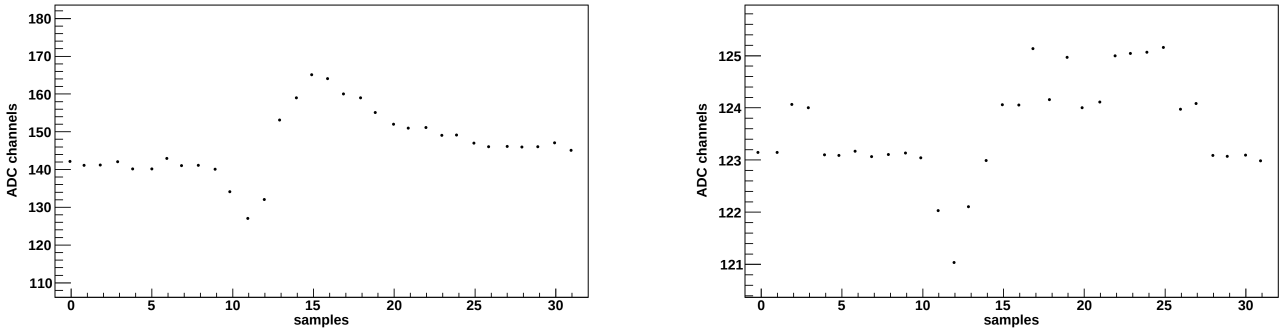


Figure 50. signals passing the bld algorithm with an energy lower than the bld threshold

In figure 50 two examples of these signals are shown. The baseline of this signal makes a significant movement downwards before it goes up again. Since the bld algorithm is looking from slope to slope, it is seeing the characteristic rise, which is big enough due to the previous movement down, and takes the mean value of all samples before as the baseline. This mean value is more or less at the baseline level, because only a few values are below, and the effective energy is lower than seen by the bld algorithm. The reason for this moving down of the baseline is suspected to be crosstalk between channels, by Igor Konorov.

Now its time to define the decision finding with the two algorithms together. If both algorithm are coinciding the decision is clear. Since the false negative rate of the smd algorithm is very low it seems to be safe to tag all channels for rejection which are adjudged as bad by it. But the big amount of false positives from the smd algorithm is disturbing. A possibility is here to use, at least for ECAL2, bld at a slightly lower threshold for cleaning up false positives from smd. At ECAL1 the false negatives from bld have not been very frequent, so that this setting will stay.

detector	bld (MeV)	smd (MeV)
ECAL1	200	100
ECAL2	125	75

Table 14. final threshold setting

	rejected channels	% of all channels
ECAL1	1361381	90.35%
ECAL2	954594	58.67%

Table 15. coincidence of bld and smd algorithm in ECAL2

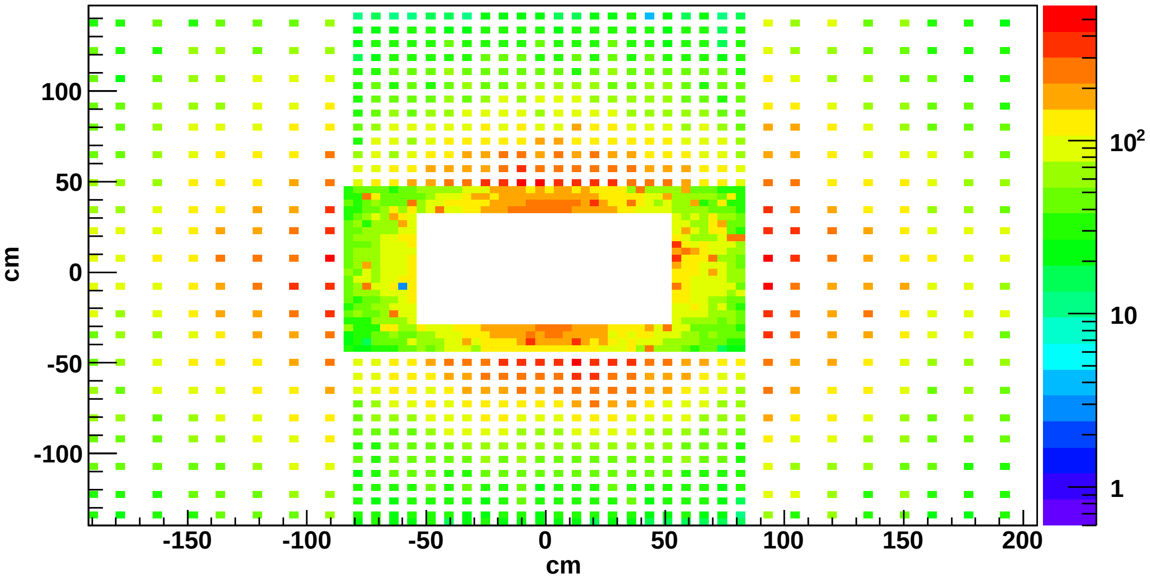


Figure 51. ECAL1 hitmap after applying bld and smd algorithms with the final thresholds

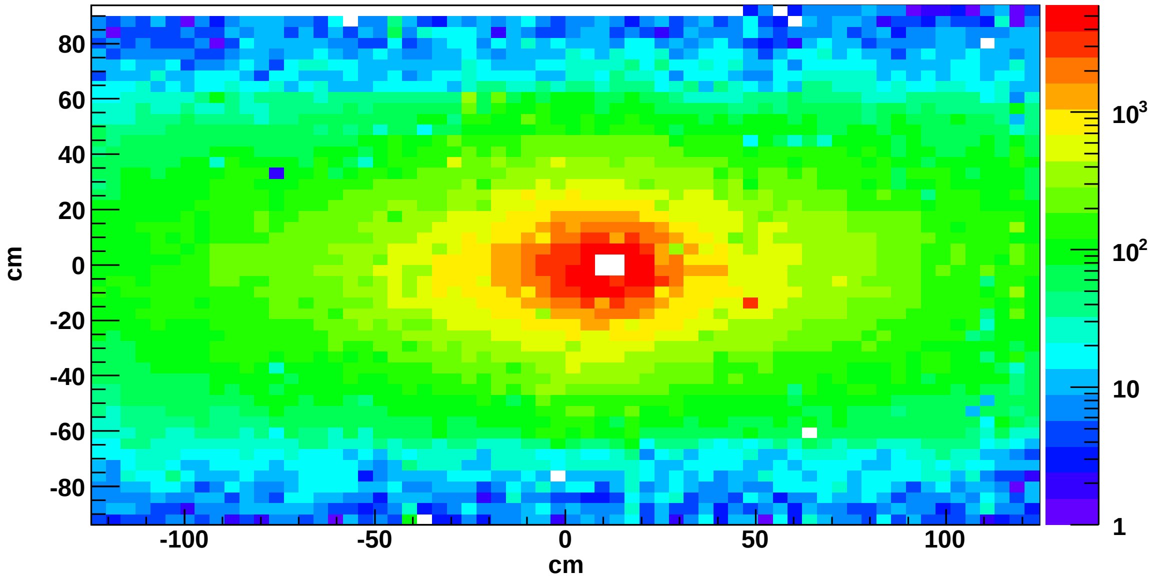


Figure 52. ECAL2 hitmap after applying bld and smd algorithms with the final thresholds

Running with these parameters shows a good improvement of both hitmaps in figures 51 (ECAL1) and 52 (ECAL2). In ECAL1 there are some hot cells left in the central part, their noise is at a very high level which makes it impossible to remove it without using a individual threshold for them. Otherwise all other channels, showing good performance at this threshold would be badly affected. But beside of that ECAL1 looks cleaned up pretty well.

ECAL2 hit map looks very smooth, only one hot cell at the lower right of the beam spot is left. The situation is the same like for the hot cells at ECAL1, an individual threshold is need to remove it safely.

The rejection factors for both ECALs are impressing. Around 90% of all ECAL1 channels and around 60% of ECAL2 have been tagged as rejected. If this can be true has to be checked again by a control sample of 100 random good and 100 random rejected channels. Table 16 shows, the performance in ECAL1 and ECAL2 is now comparable. The number if false negatives is very low. It is unclear if it makes sense trying to reduce this number further by reducing the thresholds. For sure the number of false positives will increase and this can affect the safety algorithm of the cluster module in a bad way. A small number of false negatives is not dangerous, because they will be saved in the next step by the cluster module.

detector	ECAL1	ECAL2
false negatives	2	3
false positives	7	8
uncertain cases	6	10

Table 16. results of the control sample for both ECALs

This result is approving, there is a big amount of noisy and out of time channels in both ECALs, but especially in ECAL1. Before all these channels can be rejected it has to be checked if potential false negatives can be saved by using cluster information.

7.3 Saving potential good channels after clustering

It was already mentioned that channels with a very low signal amplitude and very low energy usually belong to a cluster consisting of channels with higher energy. The channel with the most energy belongs of course to the cell which was hit directly by the particle. After hitting the cell the particle is stopped by creating an electro magnetic shower which ends up in photons detected by the photomultiplier of the ECALs. The intensity of the detected light is dependant to the particles energy. Surrounding cells can also “see” a part of this shower which has a lower intensity than in the starting cell. Cinderella’s noise detection may have problems distinguishing noise from signals induced by the rest of a shower at the outer part of a cluster. To avoid throwing away good data the cluster information can be used to keep all channels belonging to a cluster of significant energy.

The simplest and safest approach is to keep all channels of a cluster as soon there has been one good signal detected. Of course this will also keep all noisy cells connected by chance to good clusters. For a more sophisticated approach more investigation and understanding of the noisy signals and of the characteristics of good clusters is needed. The current progress of these investigations is not advanced enough to come to a final decision , but Cinderella is already providing a lot of possibilities for doing this job.

A look on the composition of clusters in figures 53 and 54 can give a hint how to set the decision criterion. The left plot in the figures 53 (ECAL1) and 54 (ECAL2) shows the amount of good/rejected channels per cluster and the right one the energy contribution of good/rejected channels.

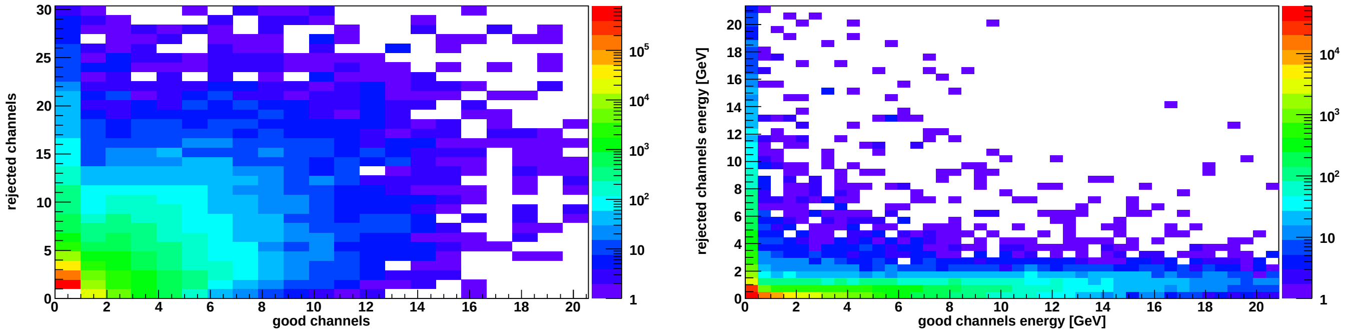


Figure 53. composition of clusters at ECAL1

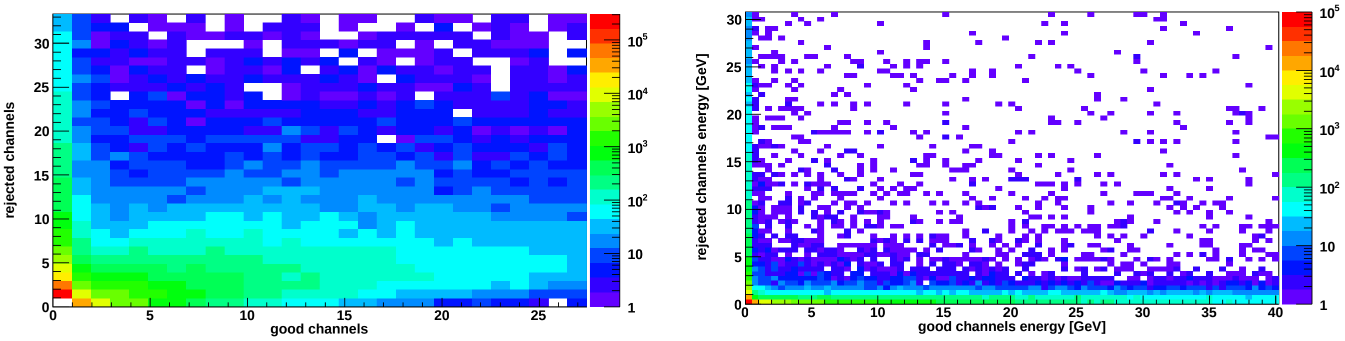


Figure 54. composition of clusters at ECAL2

In both plots a sharp separated band parallel to the y-axis can be observed. In the left plot the domination of single cells contributing to this band can be seen. This is an evidence for single noisy channels, which are sometimes located abreast. If they are located abreast they are forming a "bad" cluster with significant energy. In the right plot these channels can be observed in the region of 0 GeV good channel energy.

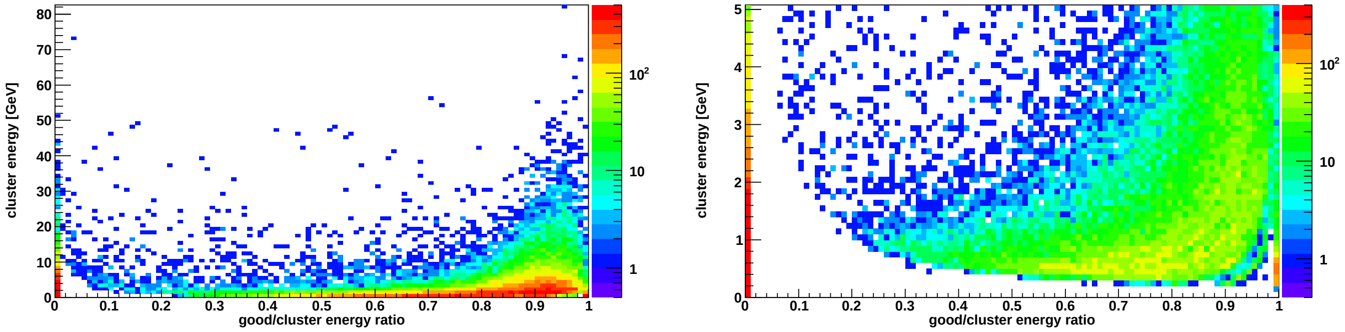


Figure 55. good/cluster energy plotted over cluster energy for ECAL1

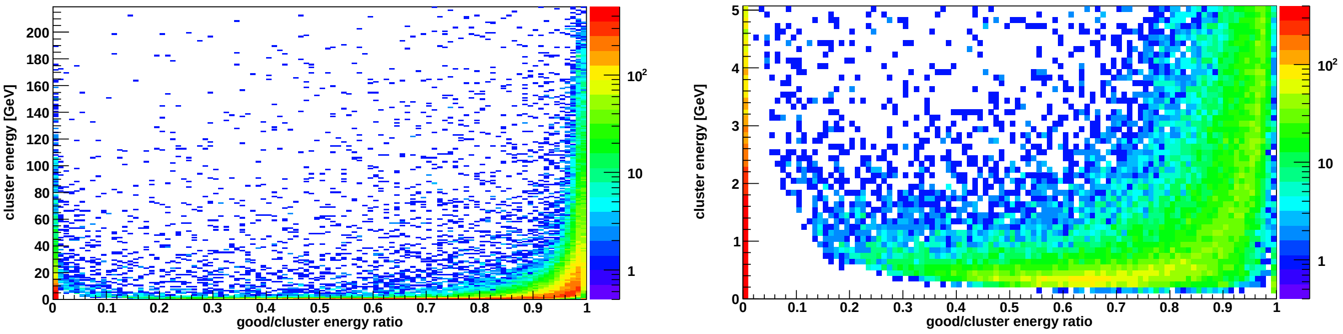


Figure 56. good/cluster energy plotted over cluster energy for ECAL1

Figures 55 and 56 are showing the ratio of energy from good signals and the cluster energy plotted over the cluster energy. Again the sharp band at an energy near 0 GeV is visible and containing most of the channels. Beside of this band it can be said, that the higher the cluster energy the higher the energy contribution of good signals. In comparison of ECAL1 to ECAL2 the amount of good channels seems to be higher at ECAL2, which corresponds to the higher reduction factors observed for ECAL1. But its not obvious how to set a cut on this ratio to eliminate noisy channels connected to good clusters without cutting out potential good signals.

At the current status of Cinderella its recommended to use the “safe” mode, which is rejecting just clusters without a good signal inside. But even in this mode the reduction factors are impressive:

	rejected channels	% of all channels
ECAL1	1201276	79.7%
ECAL2	655274	40.3%

Table 17. data reduction after cluster correction

Like it is written in table 17 up to 80% of ECAL1 and 40% of ECAL2 channels are not inside a cluster with one good signal at all. This means they can be safely rejected and no false negatives are expected anymore. To verify this statement a final control sample was taken. 100 random signals accepted by Cinderella and 100 rejected have been printed out to check the results. The printed out pulse shapes of this final control sample can be found in appendix A.

detector	ECAL1	ECAL2
false negatives	0	0
false positives	28	16

Table 18. final control sample for both ECALs

The results in table 18 look very promising. No false negative has been found, which was identified as the crucial point at the beginning. The increased number of false positives has been expected, because no additional cuts have been applied. All noisy channels attached to a good cluster have been kept. The ratio of the false positive value from ECAL1 and ECAL2 is equivalent to the ratio of the noise percentage from table 17. Also a look at the signals of the control sample in appendix A shows that all false positives are either strange signals which are surely not shower signals or crosstalk signals like in figure 50. The number of uncertain cases has not been counted, because the previous procedure has been done to decide where the uncertain cases belong to.

The conclusion from this investigation is, there is a significant amount of noise in both ECALs. ECAL1 seems to be twice as noisy as ECAL2, with 80% of all channels which can be rejected. But these investigations can only be a first step. To get to a final decision what to do with these “bad” channels, the impact of filtering these channels on the physics analysis has to be checked by the hadron offline analysis group. This can be done by looking on the π^0 mass spectrum or an exclusive physics channel with filtered and unfiltered data, to be able to compare the difference.

First checks on the π^0 mass have been started by Frank Nerling by comparing the mass spectrum of filtered and unfiltered data from 2008.

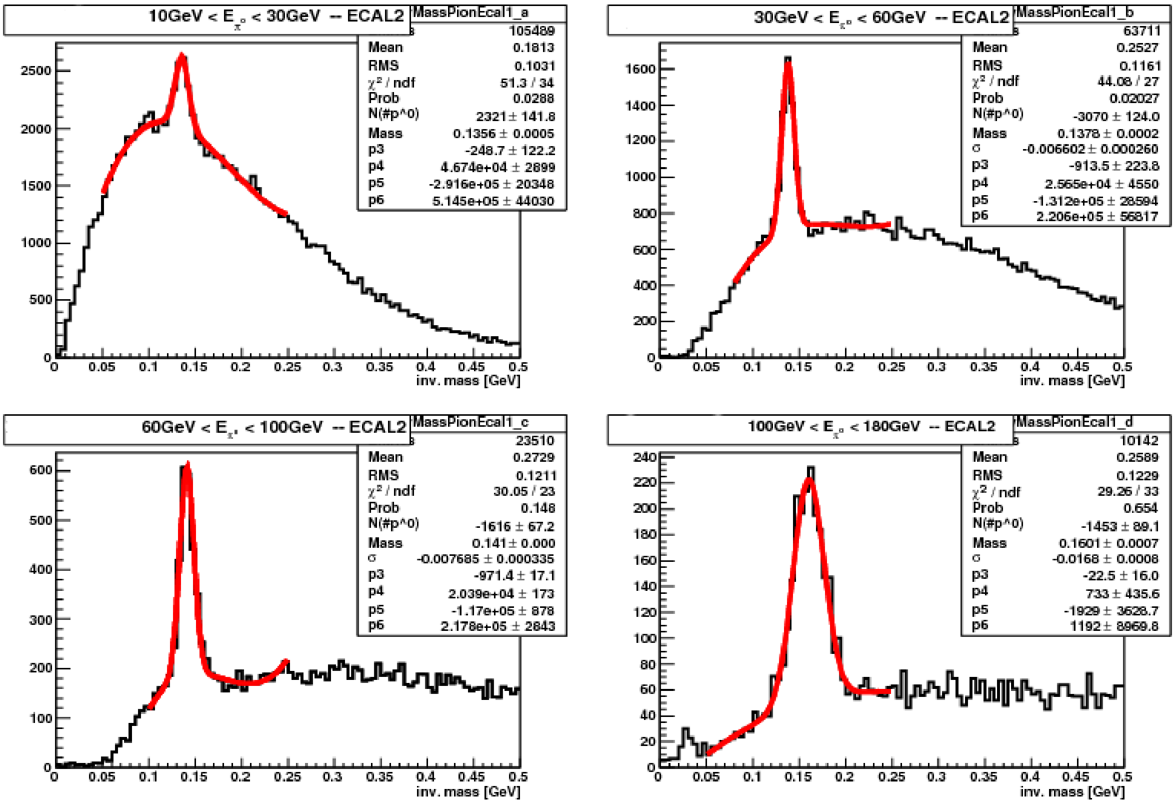


Figure 57. π^0 mass spectrum from unfiltered ECAL2 data (courtesy of Frank Nerling)

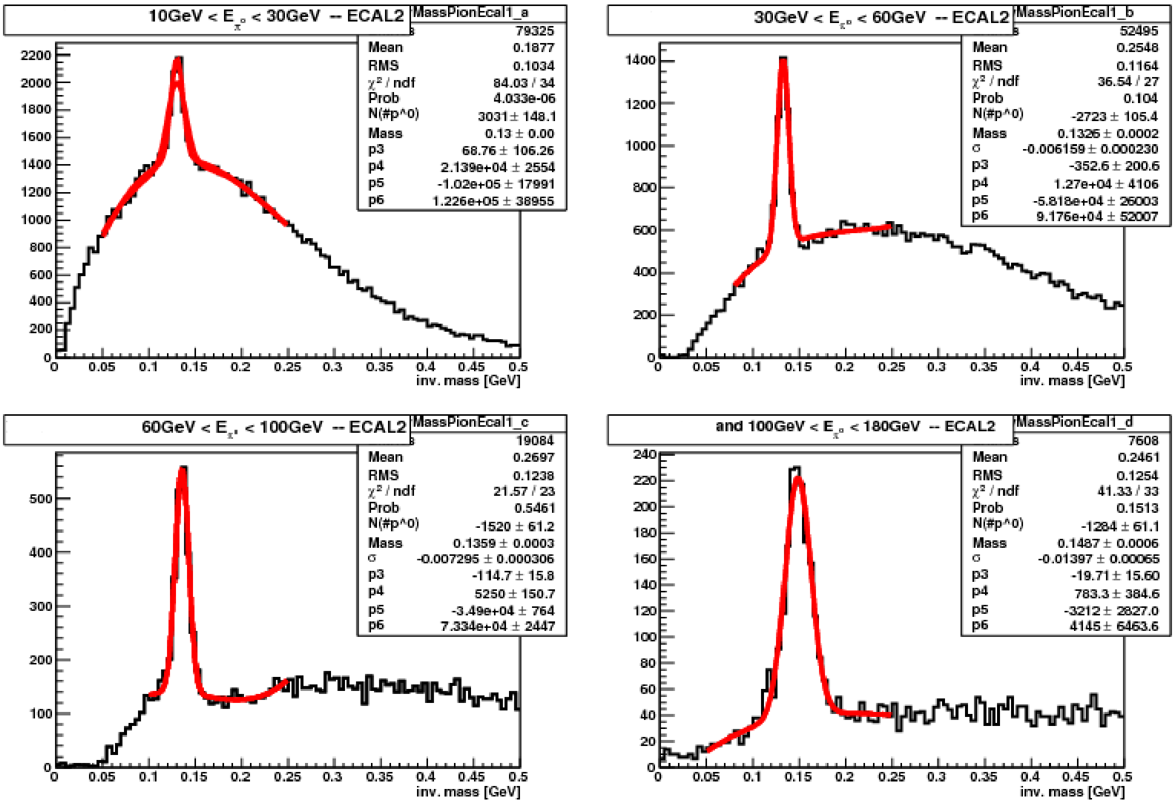


Figure 58. π^0 mass spectrum from filtered ECAL2 data by the bld algorithm (courtesy of Frank Nerling)

Figures 57 (unfiltered) and 58 (filtered) are showing a comparison of the π^0 mass spectrum from 2008 data for ECAL2. What can be observed is the total number of detected π^0 is in both cases the same. The interesting effect is here the increased number of π^0 at low energies 10 GeV $< E_{\pi^0} <$ 30 GeV and a decrease at all other energy ranges. Also the π^0 mass is shifted to lower masses at all energies. This improves the mass at higher energies 60 GeV $< E_{\pi^0} <$ 180 GeV, which have been too high at the unfiltered case but under estimates the π^0 mass at lower energies 10 GeV $< E_{\pi^0} <$ 60 GeV. This indicates that the calibration coefficients have been calculated including noisy cells and obviously the calibration coefficients can not be used for all energy ranges without a correction factor, but this is currently under investigation by the ECAL group.

These first checks have been done by using only the bld algorithm, since the smd algorithm was not ready at that point. It is recommended to repeat this check by using both algorithms together with the threshold obtained from this chapter.

8 Conclusion and outlook

It was shown in the previous chapters, that Cinderella is offering powerful possibilities for real time data reduction and monitoring. By applying Huffman encoding and rejection of noisy channels the data of both ECALs can be reduced by up to 90%. The ECALs will be “degraded” from a high rate equipment with $\approx 28\%$ of the whole data to a low rate equipment with only $\approx 3\%$.

Event number reduction based on ECAL information was not applied at the moment. Taking a decision to reject a complete event is not commendable without respecting data from other equipments like tracking detectors. But the current status is a good basis for further development of Cinderella. With the high trigger- , and data rates, there will be no other possibility than preselecting data for recording if the statistics of the measured physic have to be increased in the future.

All described features of Cinderella are fully implemented and tested. Appendix B gives an overview about all configuration parameters of the relevant modules needed for ECAL data processing. Information about the core framework of Cinderella can be found in [Kuh07] and [Nag05]. First operating experience of Cinderella from the 2004 run can also be found in [Nag05].

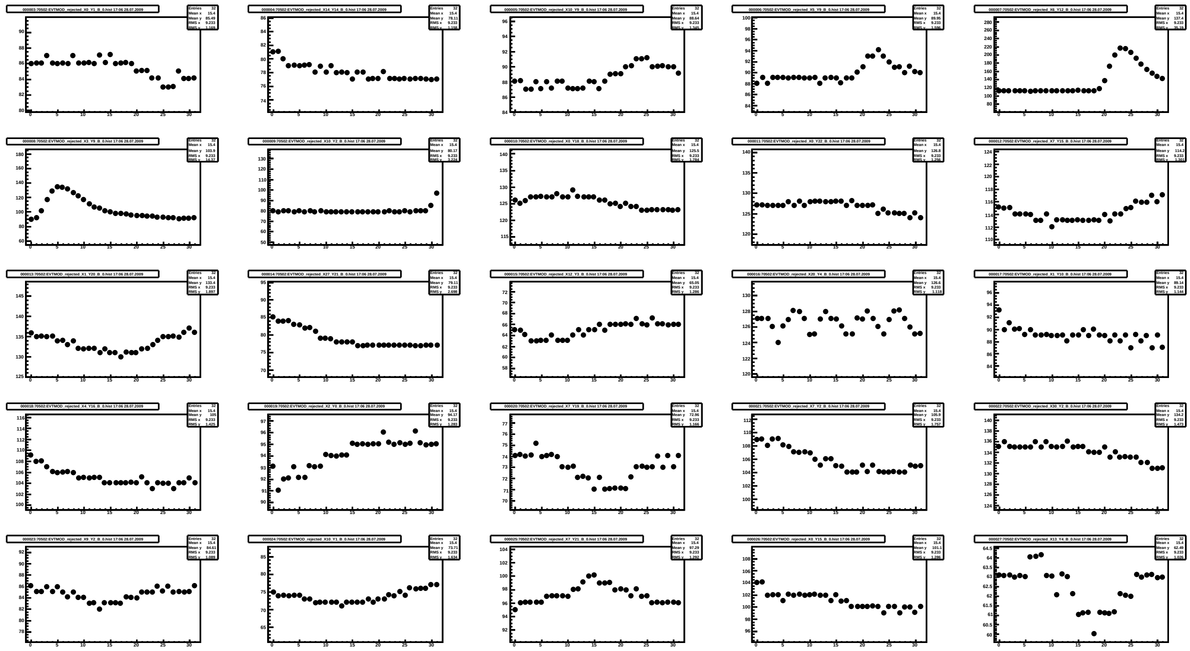
Due to the real time monitoring capabilities Cinderella already has a key role in the COMPASS data taking. With increasing data rate in the future it will become even more important. All new experiments at CERN, like ATLAS or CMS are already planned with a software based second and even third level trigger. This indicates, high rate experiments in particle physics will not be able to exist without a powerful filtering strategy, otherwise they will drown in their own data. That’s why further development and evaluation of Cinderella is recommended for successful data taking in the future.

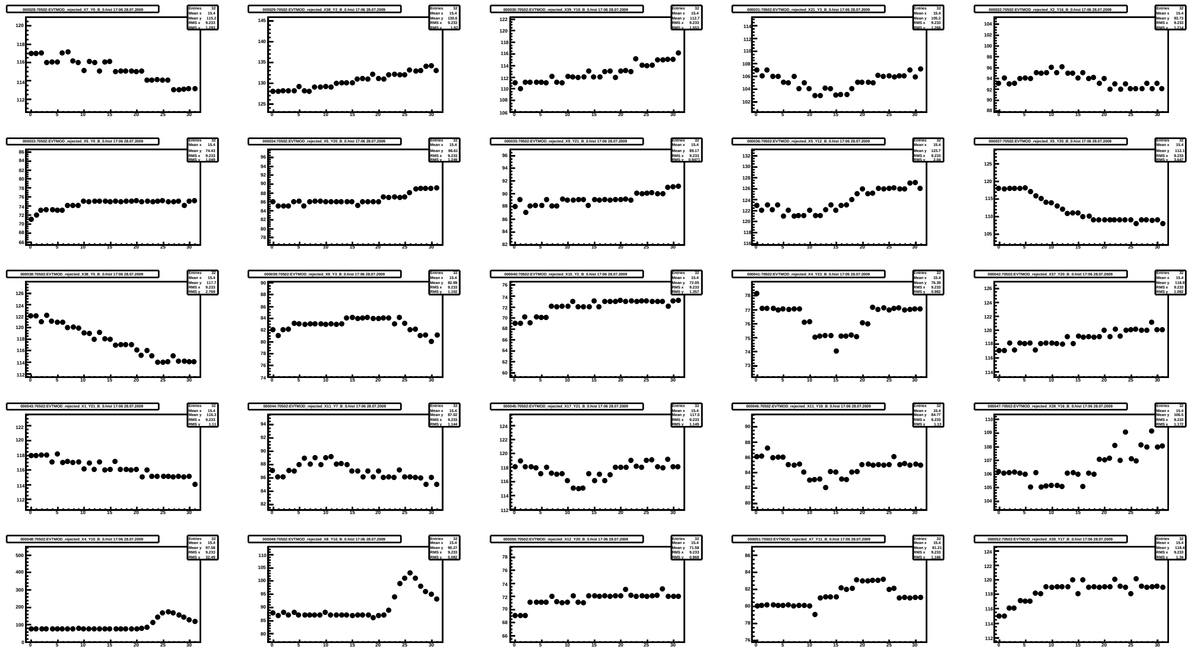
Appendix A

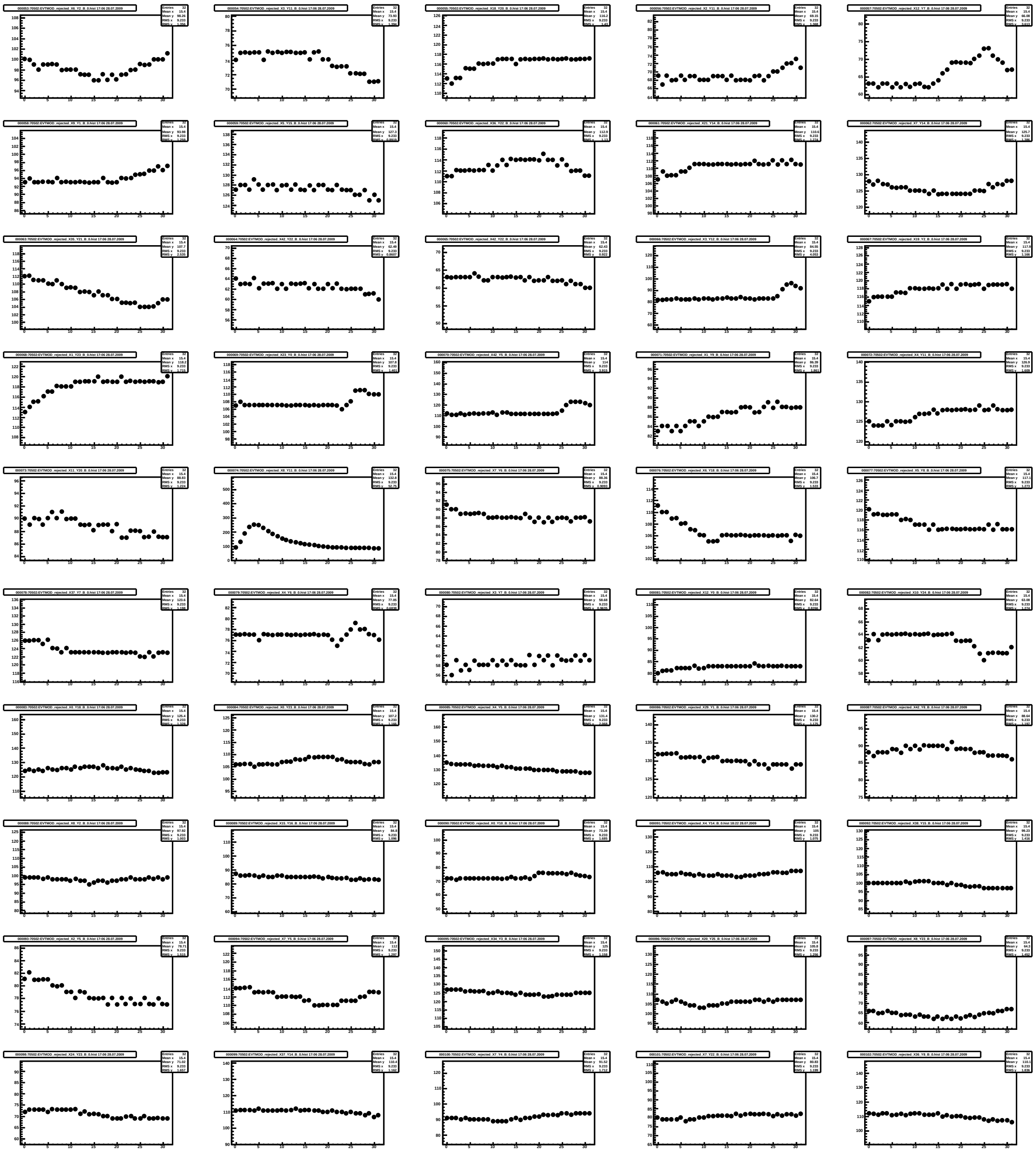
Final control sample of ECAL noise filtering

For verifying the efficiency of the noise filtering algorithms described in this thesis a control sample of 100 random good and 100 random bad signals has been made to see the results of the applied algorithms.

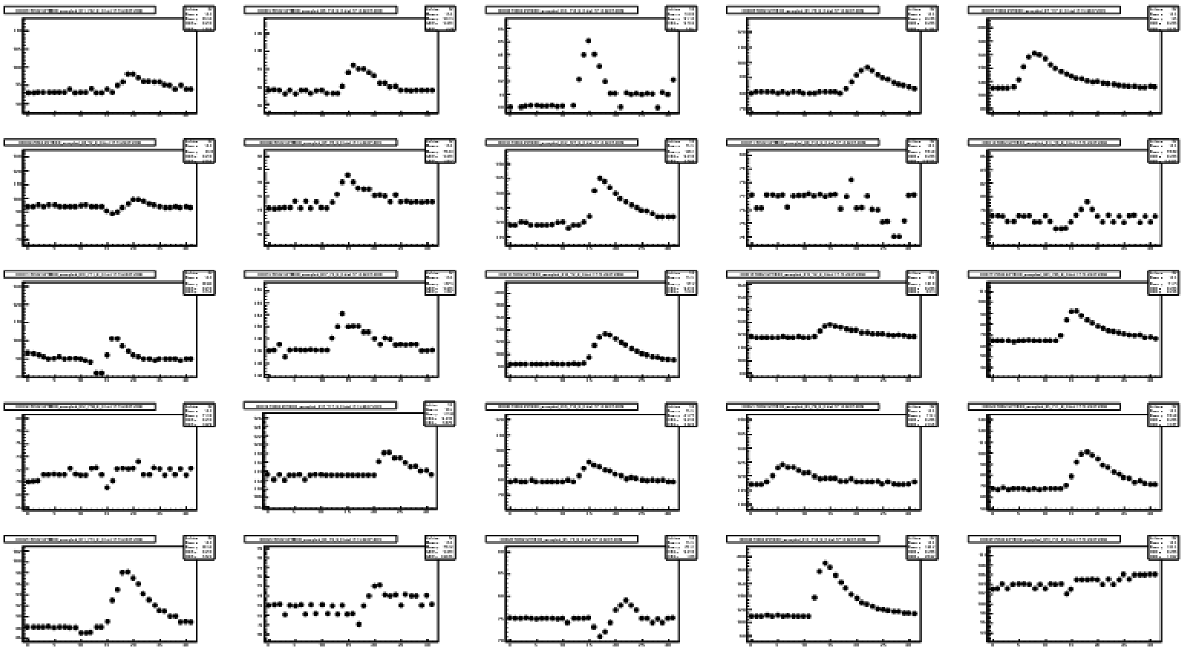
ECAL1 rejected signals

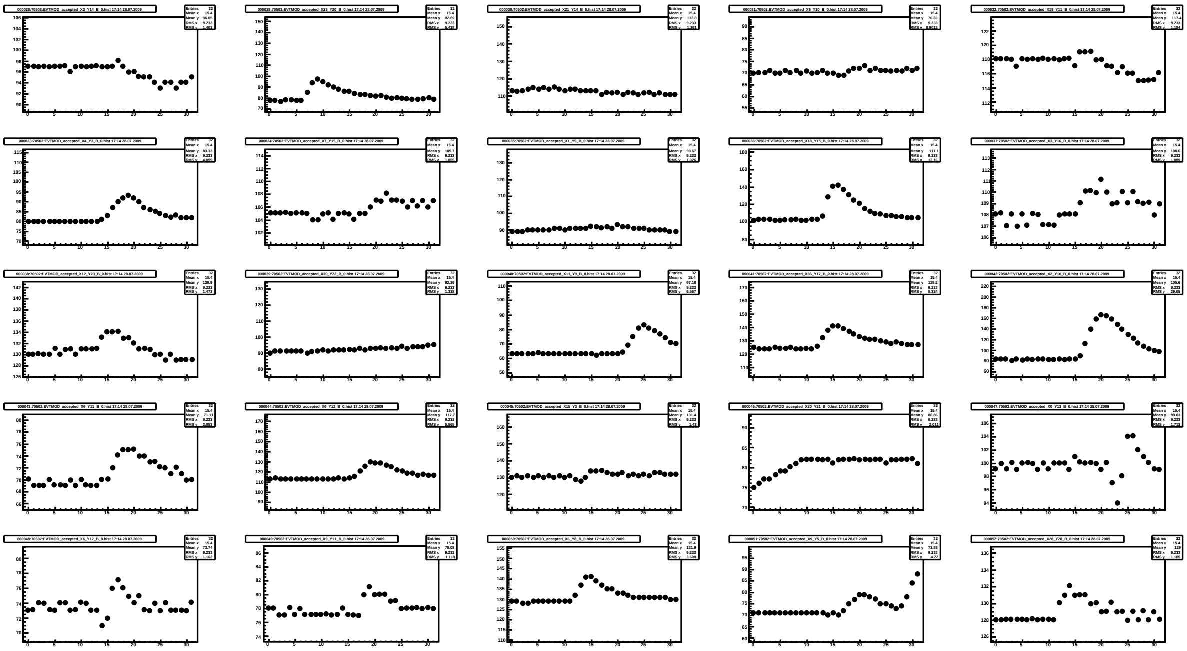


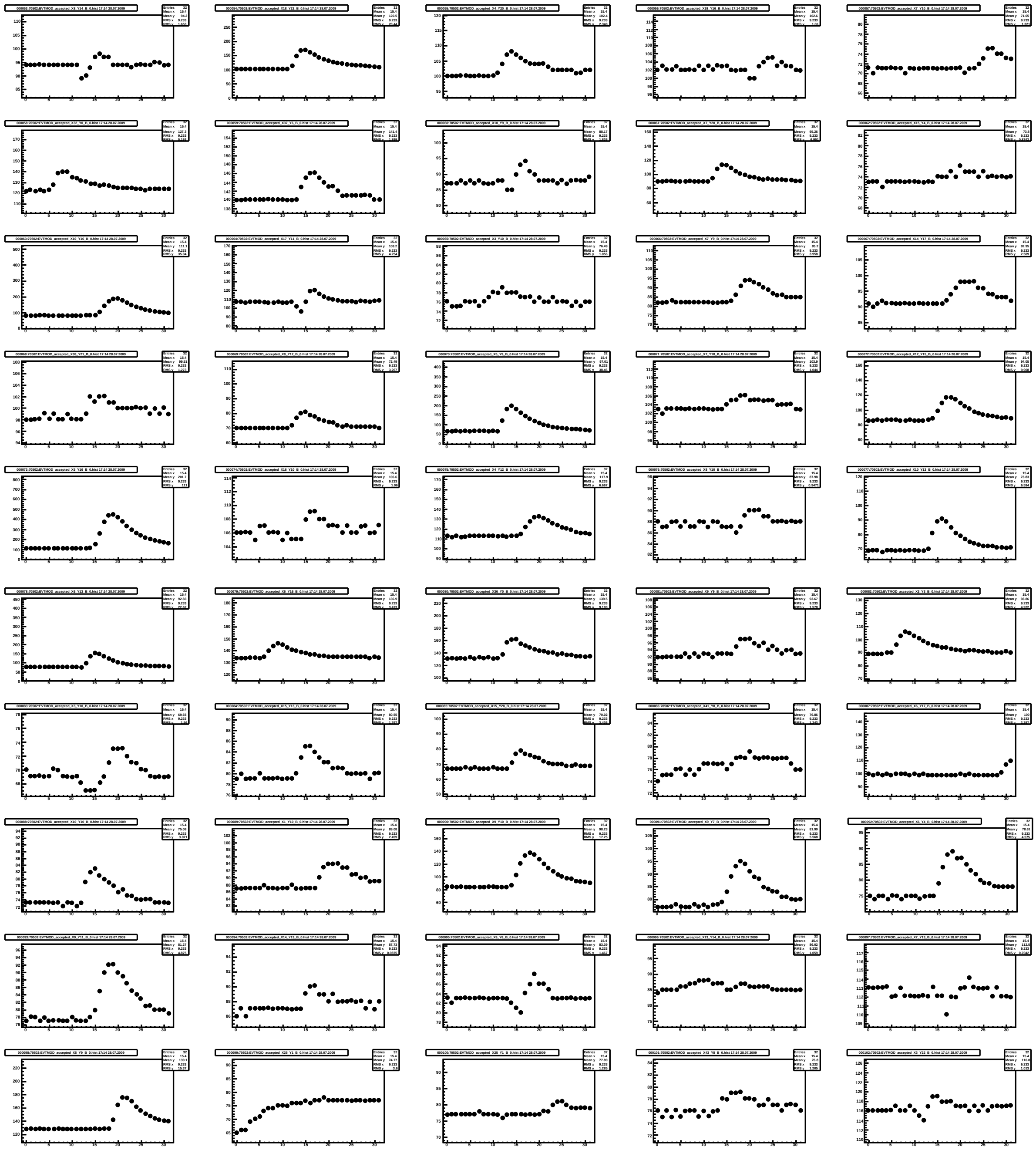




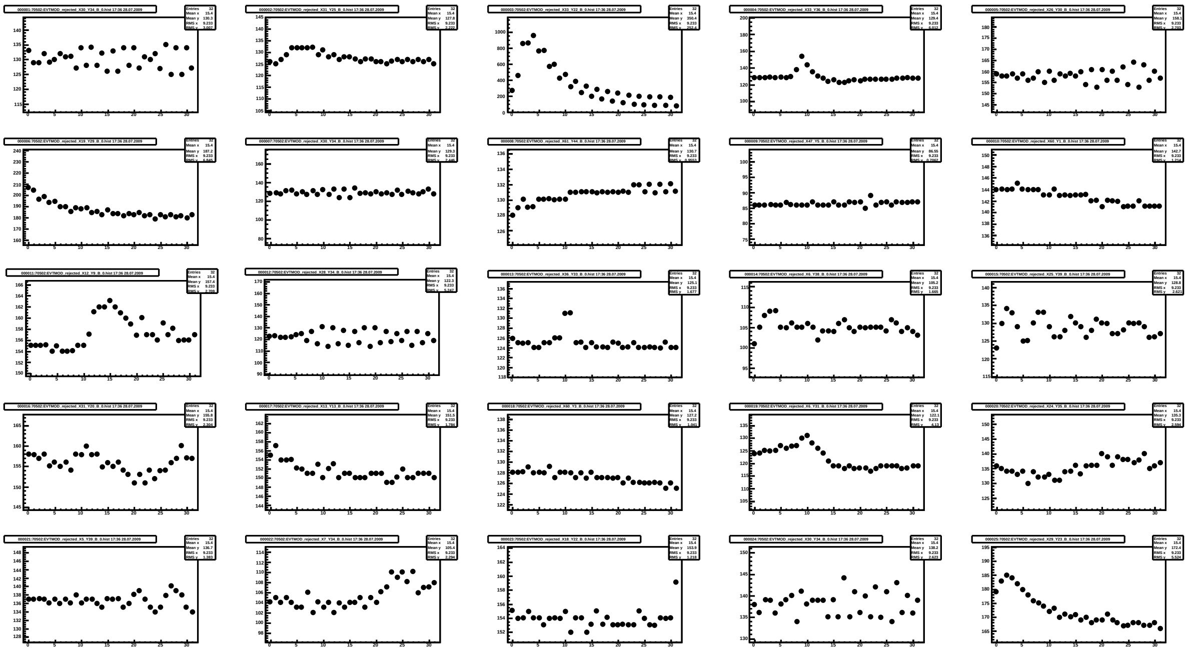
ECAL1 accepted signals

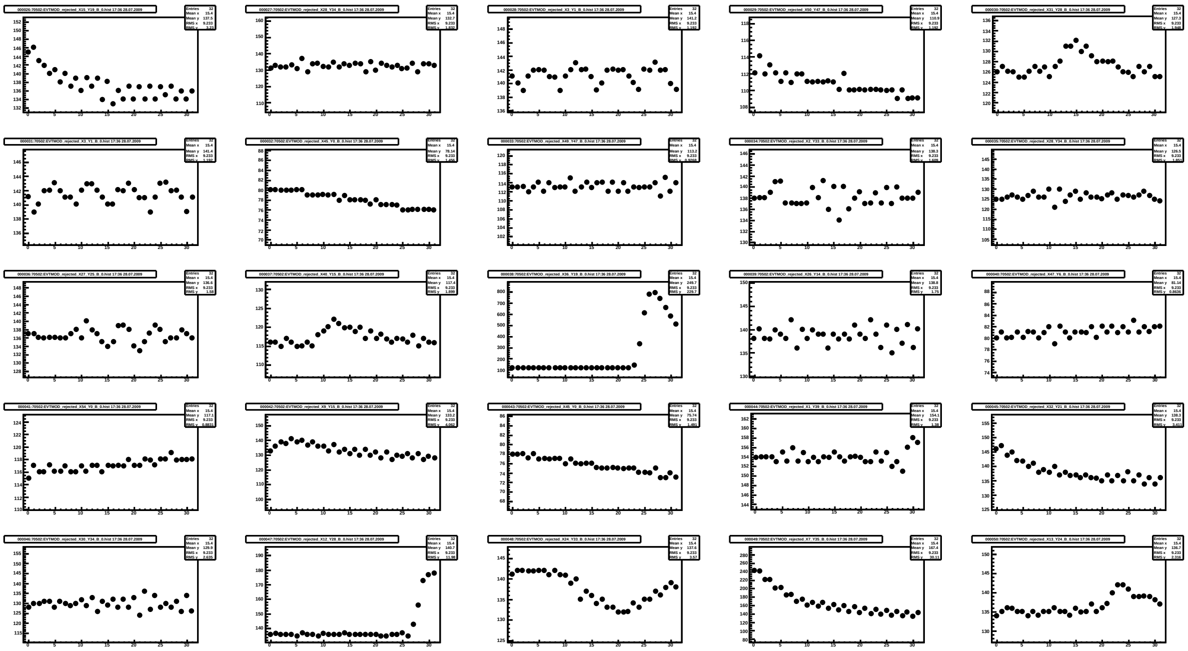


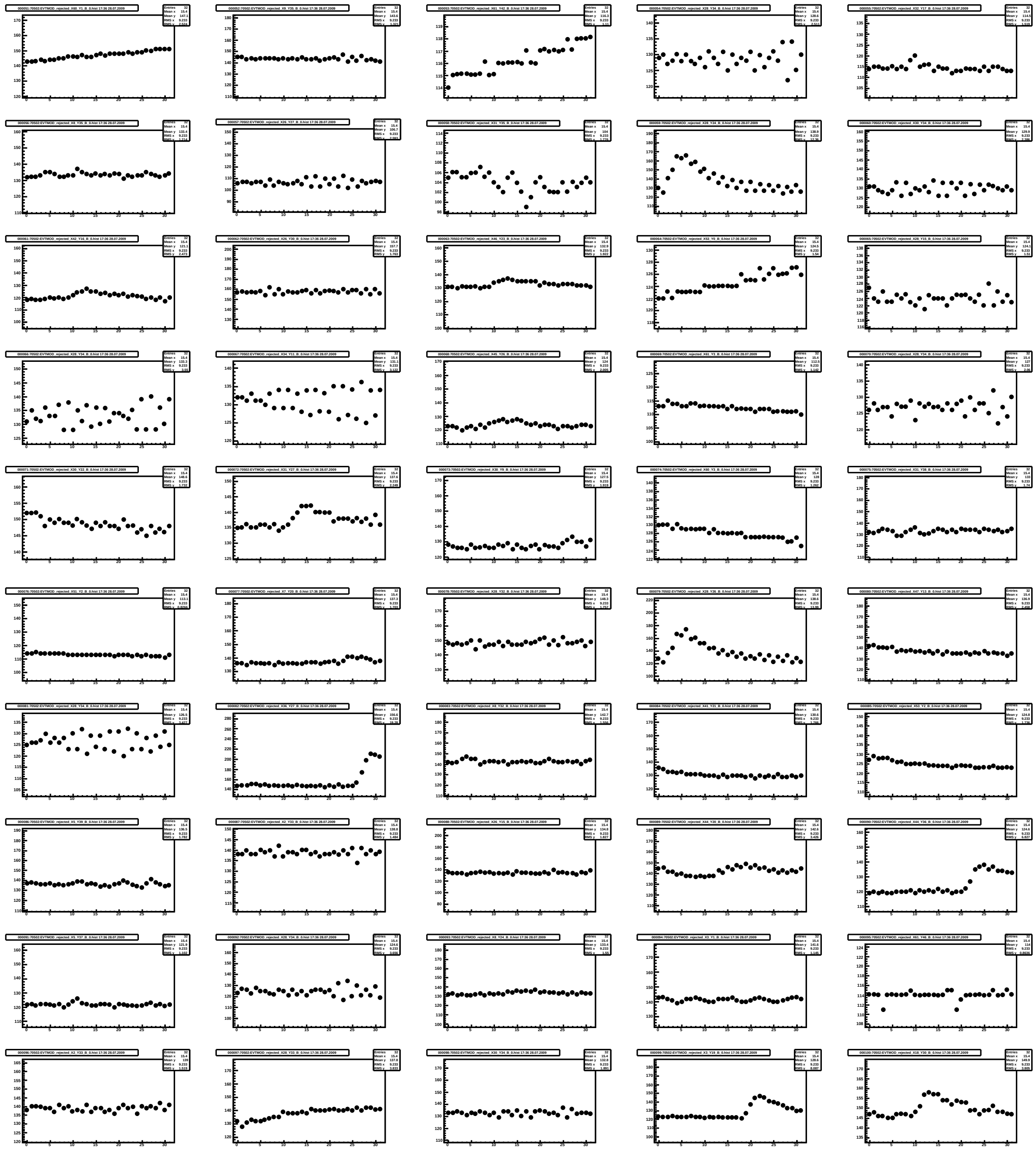




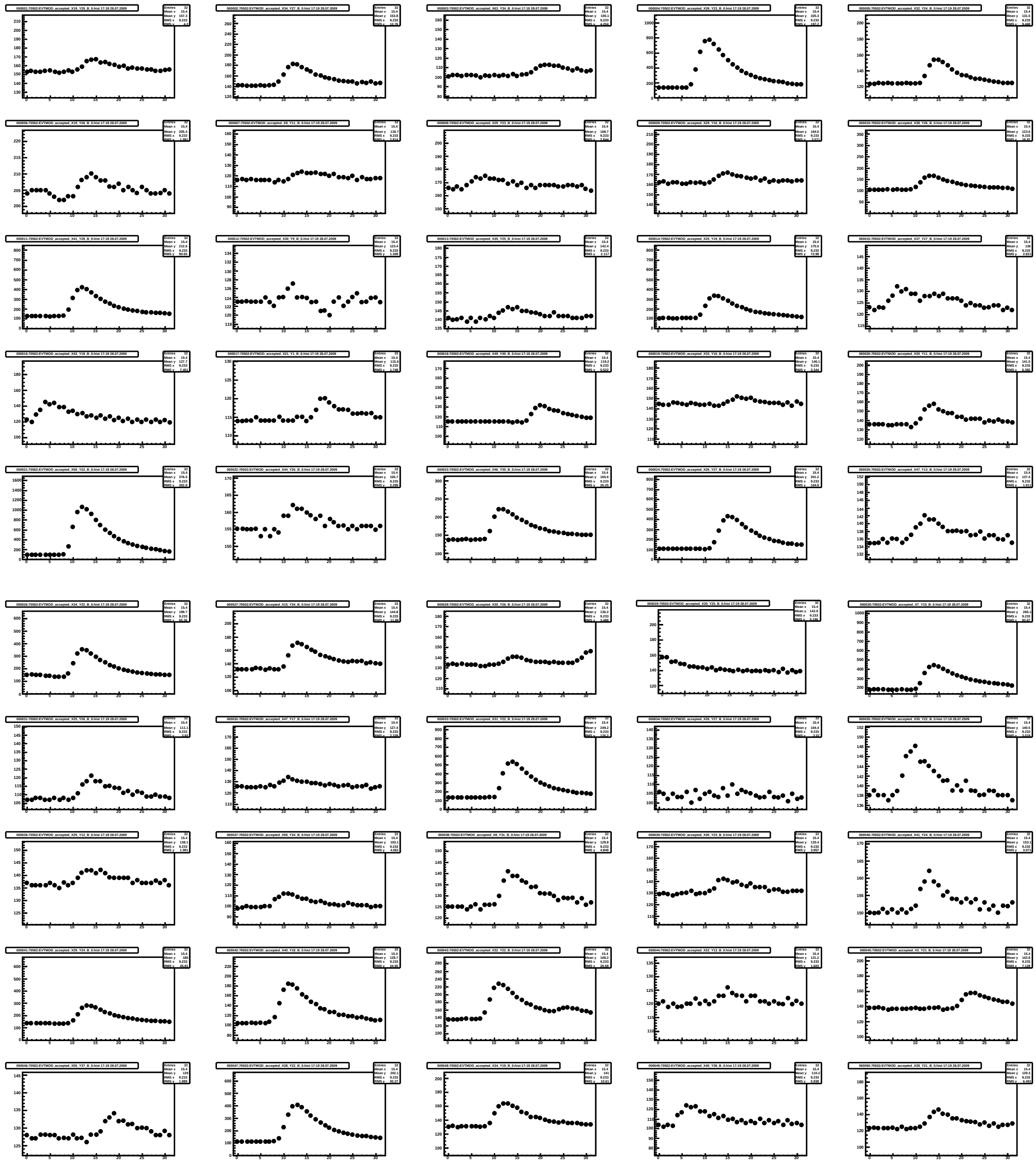
ECAL2 rejected signals

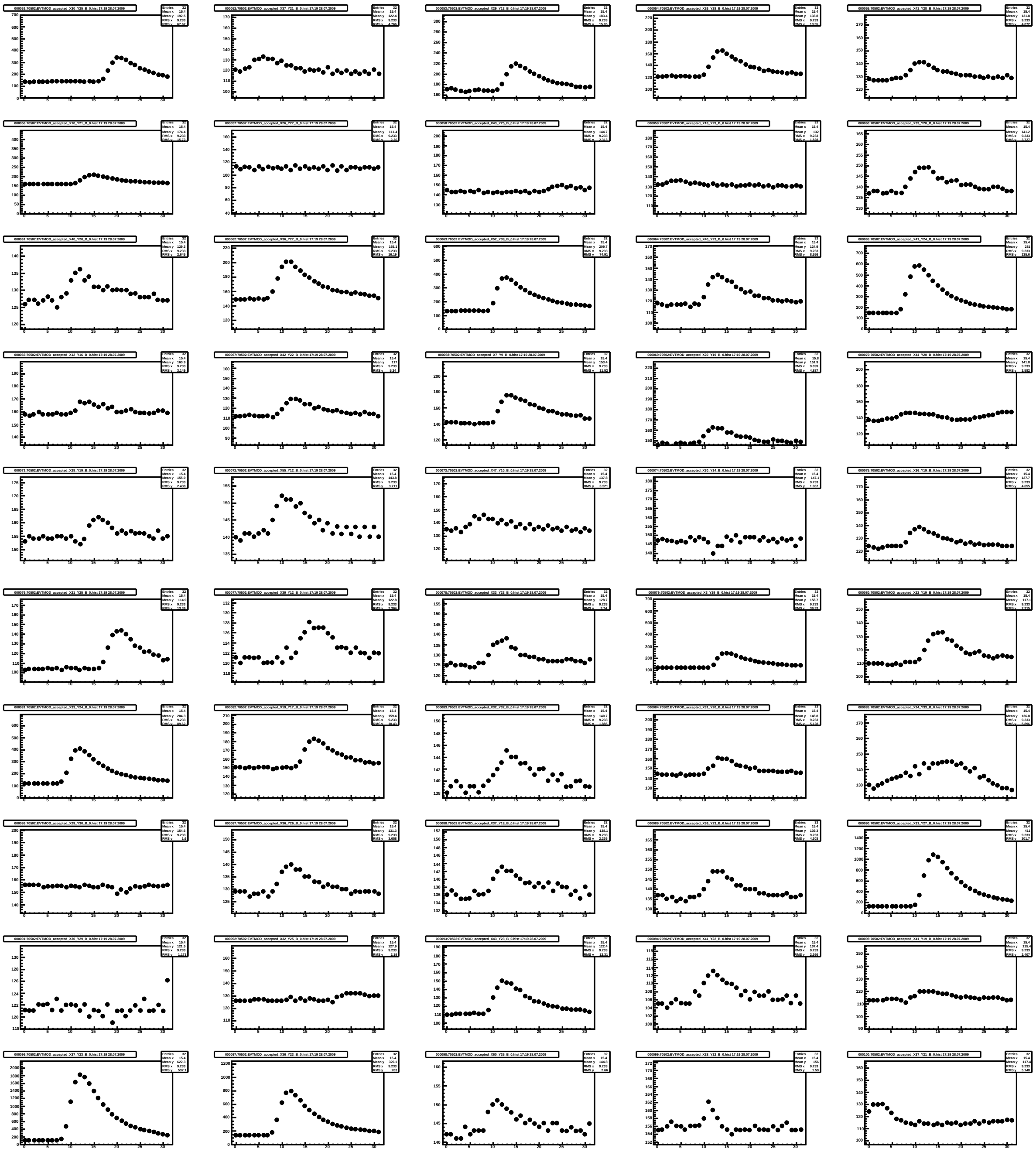






ECAL2 accepted signals





Appendix B

Configuration of ECAL related modules in Cinderella

This section s written for DAQ experts to configure Cinderella for ECAL data processing. All configuration parameters from relevant module for ECAL data processing will be explained. Standard parameters which appear in the configuration of every module, like the <Active></Active> parameter for switching the corresponding module on or off, will not be explained.

The SADC module

Mappings: All file names listed in this array will be parsed. It is expected the files are in the standard format for ECAL1 or ECAL2 mapping files.

TBnames: All tbnames (or detector plane names) listed in this array will be processed by the module. Every tbname which does not appear here will be ignored. The order of the tbnames in this array is important for the detectors size information.

Xsize, Ysize: These two represent the X and Y size of one cell of the ECAL. Attention is need when setting these values. They are also arrays and the values which correspond to a certain tbname, have to be at the same array position like the corresponding name in the **TBname**s array.

Xtep, Ystep: Arrays of the distance from one cell to the next cell in X or Y direction. The corresponding values have to be at the same position like the tbname they belong to in the **TBnames** array. The distance to the next cell is not always the cell size!

XRefsys, YRefsys: The COMPASS coordinates of the (0,0) cell in detector coordinates. Can be retrieved from the detectors.dat file with the help of an ECAL expert. Again the order has to be the same like in the **TBnames** array.

XOffsetPos, YOffsetPos: If there is an offset inside one plane, like at the MAINZ and OLGA part of ECAL1, the corresponding X and Y detector coordinate has to be filled into this array. If there is no offset it should be set to '-1'. The order of the values has to correspond to the order of tbnames in the **TBnames** array.

XOffset, YOffset: The size of the offset in cm. Should be at the same array position like the offset detector coordinates in **XOffsetPos** and **YOffsetPos**.

ZPos: The Z-position of the detector in COMPASS coordinates.

intwordnum: Indicates the number of integral words for every channel. Only needed when data from before 2006 is processed.

FEMReadout: When set to '1' the SADC module is running in FEM mode. It is writing all results into a special FEM structure which is needed by the "ecal02_an" to normalise calibration event with the corresponding FEM values.

The ecal02_an module

TBnames: Array of tbnames which should be processed. Tbnames not listed here will be skipped.

Slopeint: The interval used for slope calculation. A value of '1' means the slope is calculated from one to the next sample. A value of '2' from one to the one after the next,.... .

Barrier: Threshold in ADC channels for the bld algorithm. Will be ignored if a valid calibration is available and the **EnergyThres** parameter if bigger than '0'.

BarrierInterval: The amount of slopes to be summed up and compared with the **Barrier**.

EnergyThres: A threshold in MeV from which an individual barrier will be calculated for each channel. If there is no valid calibration available it will fall back to the **Barrier** setting.

IntDiffThres: The Interval Difference Threshold indicates to minimum difference between the two intervals in the smd algorithm, above which the signal is considered as good. If **smdEnThres** is set to a value bigger than '0' and ECAL calibrations are available this parameter will be ignored.

smdEnThres: The smd Energy Threshold in MeV will be translated to an interval difference for every channel based on the calibration coefficients (similar to **EnergyThres**). Thus an individual threshold can be applied for every channel.

DetectorName: This parameter has to be set to the detector name needed to obtain the calibration file from the calibration database. At ECAL2 the detector name is equal to the tbname, at ECAL1 it is different.

ChiSquareFit: Boolean parameter to turn on/off experimental chi square fit to determine the signal offset against a reference.

EcalMonDBTable: Table in the COMPASS database where ECAL calibration event amplitudes should be written to. It is needed to specify it int he format: "Database.TableName"

DBTimeout: Maximum time one query is allowed to take in seconds.

WriteSpills: Specifies the amount of spills over which the calibration event amplitudes should be averaged.

UseFEM: Boolean parameter. If set to '1' all calibration amplitudes will be normalised to the corresponding FEM value. Requires a SADC module configured with turned on **FEMReadout** parameter.

The cluster module

TBnames: Array of tbnames which should be processed. Tbnames not listed here will be skipped.

RatioBG: The ratio threshold of bad/good cells inside a cluster. All clusters with a ratio above this threshold will be considered as good and all channels will be kept regardless of their noise tag.

The pi_ECAL module

TBnames: Array of tbnames which should be processed. Tbnames not listed here will be skipped.

TargetX, TargetY, TargetZ: COMPASS coordinates of the target in cm.

EnergyCut: clusters with an energy below this threshold will not be used for π^0 reconstruction.

The evtmod module

TBnames: Array of tbnames which should be processed. Tbnames not listed here will be skipped.

Compression: Boolean parameter. If set to '1' all (M)SADC data from **TBnames** will be Huffman encoded with the tree specified in **HufftreeFile**.

DecompCheck: Boolean parameter. If set to '1' a decompression check will be done for every channel after Huffman encoding. Should be only used for debugging.

HufftreeFile: Name of the mapping file containing the Huffman tree inside the mapping directory.

hufftreedumpfile: During every run the evtmod module is collecting statistics for Huffman tree generation. If a path to a folder is specified a Huffman tree in ASCII format is written to a file called "hufftree". If this file is already existing it will be overwritten. Leaving this parameter empty will disable Huffman tree generation.

FilterNoise: Boolean parameter. If set to '1' all channels tagged for rejection will be finally rejected.

Bibliography

[Col96] The COMPASS Collaboration. Common muon and proton apparatus for structure and spectroscopy, 1996.

[Gro08] Particle Data Group. *REVIEW OF PARTICLE PHYSICS*. 2008.

[HHC02] V. Lindenstruth D. Röhrich B. Skaali T. Steinbeck K. Ullaland A. Vestbø A. Wiebalck H. Helstrup, J. Lien and ALICE Collaboration. *High Level Trigger System for the LHC ALICE Experiment*. 2002.

[Huf52] David A. Huffman. A method for the construction of minimum-redundancy codes. 1952.

[Ket09] Bernhard Ketzer. Physics with hadronic probes at compass. 2009.

[Kuh07] Roland Kuhn. *Gluon Polarization in the Nucleon*. 2007.

[Nag05] Thiemo Nagel. Cinderella: an online filter for the compass experiment. Master's thesis, 2005.

[V.K08] A.Magnon F.Nerling V.Kolosov, O.Kouznetsov. Present performances of compass electromagnetic calorimetry from data analysis, 2008.