Analysis of beam test data from the CMS TOB Cosmic Rack

Henning Gast

20 August 2004

Abstract

This article documents my work as a summer student in the CMS TOB Cosmic Rack group at CERN in July and August of 2004. During that time, I have set up a running analysis environment based on the ORCA framework, created a simple event display, examined the signal over noise-distribution of the silicon modules, and conducted a pulse shape scan, among other things.

Contents

1 The Cosmic Rack and the CMS detector

This report deals with an analysis of data recorded with the CMS TOB Cosmic Rack during the May 2004 test beam. Therefore, a brief overview of the CMS detector and the purpose of the Cosmic Rack will be given in this first chapter. The second chapter will deal with the actual work conducted with the data.

1.1 A short overview of the CMS detector

The Compact Muon Solenoid (CMS) is a multi-purpose high energy physics experiment currently under construction at CERN. It will be one of four detectors observing proton-proton and heavy ion collisions in the Large Hadron Collider (LHC) which will become operative in 2007. Its unprecedentedly high centre-of-mass energy of $14\,TeV$ will make searches for new particles possible, for instance the Higgs boson or particles predicted by supersymmetric extensions to the standard model, while at the same time being able to look for unexpected new phenomena.

A schematic view of the CMS detector can be seen in figure 1. The protons will collide in the centre of the detector and the properties of the particles produced will be measured by several subdetectors to be described in the following.
Next to the beam, the **tracker**[3] is located. It uses silicon modules, in which

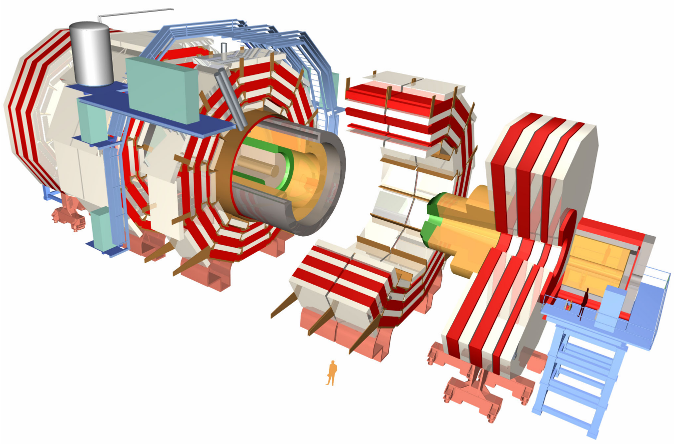

Figure 1: Schematic view of the CMS detector

charged particles deposit energy due to ionisation, to accurately measure points in the particles' trajectories. The innermost part is made up by a silicon pixel detector, whereas farther out, silicon strip detectors are employed. This part of the tracker is divided into the inner and outer barrel (TIB and TOB), in the area perpendicular to the beam axis, and the endcap (TEC) in the forward region.
A solenoid **magnet** produces a magnetic field of strength $4\,T$ thus bending the tracks of charged particles due to the Lorentz force. This allows a measurement of the particles' momenta from their radii of curvature.
Next to the tracker is the **calorimeter**, divided into two parts. The first is the electromagnetic calorimeter, consisting of $PbWO_4$ crystals. Here, electrons and photons, but not hadrons and muons, will produce an electromagnetic shower, whose size is a function of the incoming particle's energy. The second is the much larger hadronic calorimeter. It is made of brass plates, where hadrons will form a hadronic shower, and interleaved with plastic scintillators. The calorimeter as a whole thus provides an energy measurement and particle identification. In addition, it is also used for triggering.
The outermost part of CMS is formed by the **muon system**. It measures the trajectories of muons, that are not absorbed in the inner detector, using drift tubes in the barrel and cathode strip chambers in the forward region.
The physically interesting data will be extracted by a sophisticated trigger and data acquisition system.

1.2 The Cosmic Rack

The Cosmic Rack (figure 2) serves as an intermediate step between a single silicon module and the final TOB, enabling one to study the properties of the detector, test the infrastructure, e.g. cooling and readout systems, and find possible errors.
As will be the case in the TOB, the Cosmic Rack consists of rectangular rods, like the one depicted in figure 3, each of which carries six or twelve silicon modules, depending on whether the rod is single-sided (SS) or double-sided (DS). In the latter case, pairs of active layers of silicon are mounted back-to-back, one of them being tilted by an angle of $\alpha = 100\,mrad$ with respect to the other. This allows a measurement of the coordinate parallel to the strips.
The rods are then arranged in layers. As can be seen from figure 2, the Cosmic Rack has six layers at the moment. The two outermost layers are equipped with

Figure 2: Photograph of the Cosmic Rack

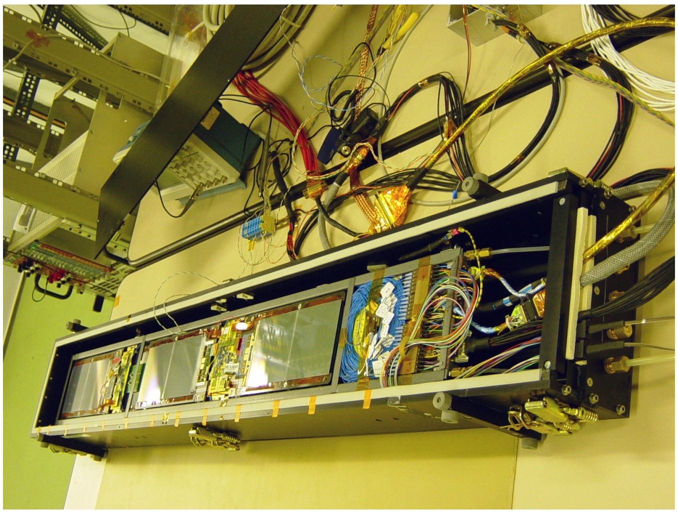

Figure 3: Photograph of a TOB rod

DS modules.
In the coordinate system adapted for this analysis, the z-axis points in the direction of the beam, while the x-axis is perpendicular to the strips.

2 Analysis of Cosmic Rack beam test data

The CMS TOB Cosmic Rack, presented in the previous chapter, was exposed to the X5 test beam at CERN in May 2004. The beam could be chosen to consist of either pions or muons, and during that time, numerous runs, each containing several thousand events, were recorded.
These data are available in the form of ROOT[4] trees from which the physical information desired can then be extracted using some suitable analysis programme. For this purpose, the ORCA[5] framework was chosen. ORCA is basically a set of

C++ classes designed to conduct the CMS physics analyses. However, since it is far from finished at the moment, installing and running an instance of it is not a trivial task. In order to do analyses of the test beam data, an ORCA-based programme called *General.cpp* exists in the CVS repository[1]. Among its important features are cluster finding and track fitting. The first task therefore was to get this analysis programme to run on a local workstation.

2.1 Setting up the analysis environment

To install the analysis programme on a local workstation, the code that is either specific to the test beam analysis or is to be modified locally has to be checked out from a central CVS repository and then be compiled using ORCA's `scram` utility. However, care has to be taken to take the right version of each module since taking the wrong one can result in compability crashes or missing features. In general, the version of ORCA used was 8.1.3. An installation script for setting up the analysis environment, resulting from these efforts, can be found on the world wide web[2].

2.2 A simple event display

In order to visualize the test beam data, a simple event display was created using the ROOT package. This is important for getting a first idea of the physical processes occuring and to quickly check if certain parts of the analysis are working properly, e.g. track finding. An example of a test beam event can be seen in figure 4 which depicts the passage of a charged pion.

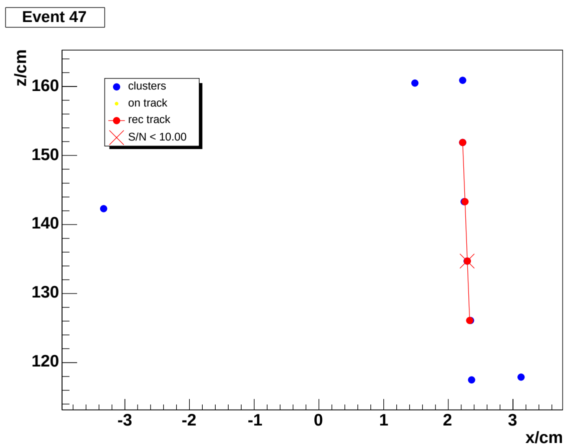

Figure 4: Event 47 of run 80735

When a charged particle traverses the detector, it will usually deposit energy in more than one strip. Such a group of strips with a signal that exceeds certain thresholds is called a cluster.
Marked with blue circles, the clusters found by the clustering algorithm are drawn in the x-z-plane. A track reconstructed by the track finder is also visible. The cluster with the red cross on it has low signal over noise, to be defined later. Four issues that can be seen from this randomly chosen event should be noted:
First, the effect of the stereo detectors on the first ($z \approx 160\,cm$) and sixth ($z \approx 118\,cm$) plane can be seen: On them, two clusters are shown. Since the x-coordinate

of the tilted detector is measured in the local reference frame, the distance between the two x-values allows reconstruction of the y-position of the cluster in the detector frame.
Second, the reconstructed track does not extend to all the layers, although there are clearly also hits on the outer layers belonging to it. A visual inspection of a large sample of events using the event display suggests that, for a vast majority of events, clusters on one or both outermost layers are not included in a track.
Third, the single layers are not perfectly aligned with respect to each other, the misalignment being of the order of a few hundred μm. This may be the reason for the reconstruction problem just mentioned.
Fourth, there are clusters likely generated by noise, like the one on the third layer, outside the track.

2.3 The signal-over-noise distribution

The signal read out from a certain strip when no signal is present will fluctuate around a given value called the **pedestal**. Given n such measurements,$\{d_i\}$, the pedestal p is calculated as the arithmetic mean:

$$p = \frac{1}{n}\sum_{i=1}^{n} d_i$$

The standard deviation of these values is called the **noise** σ:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - p)^2}$$

The **signal-over-noise** S/N of a cluster can then be defined as the cluster charge divided by the total cluster noise.
In figure 5, the distribution of S/N values, obtained from all modules, in a sample of events from run 80297 can be seen. The solid line is a Landau distribution fit

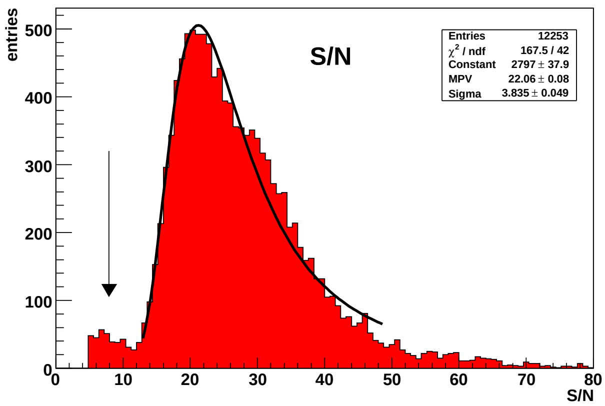

Figure 5: A S/N distribution

to the data. It describes the measured distribution nicely in the interval $S/N \in [12, 40]$. However, there is also a significant number of clusters with very low S/N, as indicated by the arrow. The question thus arose what the reason for this behaviour was.

Low S/N can be due to either low signal or high noise. Therefore, a separate look on both the signal and noise distributions needs to be taken. On the left side of

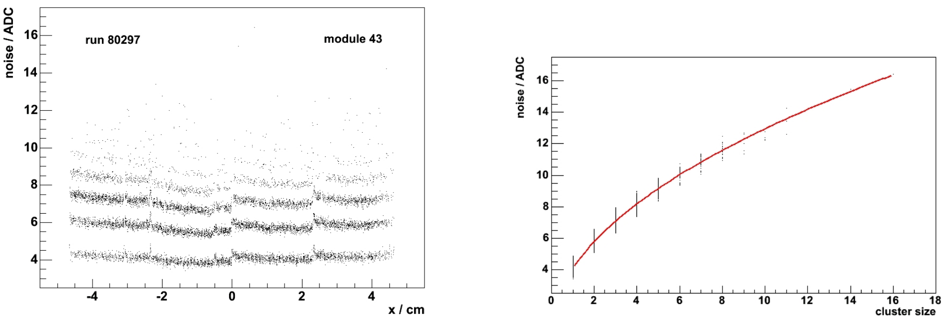

Figure 6: The distribution of noise vs strip position (left) and the dependence of noise on cluster size (right), for module 43 and events from run 80297

figure 6, a scatter plot of noise vs x-position, i.e. strip number, can be seen. The values are taken from clusters on module 43, during run 80297. The fact that the values obtained for the noise form bands instead of being distributed continuosly is curious at first. However, a glance at the right side of figure 6 reveals the reason for this. Here, the noise values are plotted against the number of strips in the corresponding cluster. The red line is a best fit of the function $n(s) = a_0\sqrt{s}$, where s denotes cluster size, n denotes noise, and a_0 is a free parameter. Such a behaviour is expected if the total cluster noise is calculated by adding the noise values of the constituent strips in quadrature. Indeed, such a calculation was found in the source code in two different places.
Coming back to the problem of low S/N, figure 6 then suggests that the reason has to be looked for in the charge distribution, since no unexpected pattern is apparent in the noise distribution, apart from the bands. Figure 7 sheds light on what is happening. The histogram in the background, plotted in green, shows the position of clusters with normal S/N on module 43. The data was taken during run 80297, with muons as incident particles. The strip number n_S of a given cluster was calculated using the formula

$$n_S = 512 - \frac{1}{p}(x + 256p)$$

where $p = 183\,\mu m$ is the pitch of the strips and x is the local coordinate. 512 is the number of strips on the module, and the inversion is needed because of an as yet unresolved problem with the geometry files.
The profile of the location of incident particles is nicely flat and falls off only towards the edges of the detector. This effect is due to the width of the scintillator used for triggering during the test beam. Another histogram, plotted in red colour, marks the positions of clusters with low S/N. It can clearly be seen that these clusters are only found on certain strips. In addition, clusters with low energy deposition are histogramed using black triangles. Comparing this with the position of the low S/N clusters shows that the low S/N is indeed due to low signal.
Furthermore, the blue lines indicate clusters that have been marked either dead or noisy by the strip masking algorithm. The explanation for the appearance of low S/N-clusters can therefore be given as follows: When a particle traverses the area of a bad strip, it will deposit a large amount of energy there, but also a smaller amount in the neighboring strips. If this amount still exceeds the thresholds used

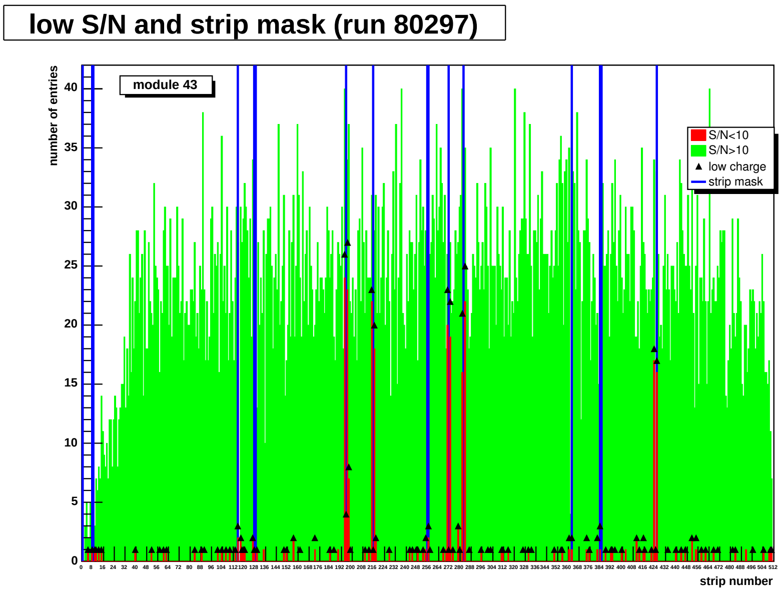

Figure 7: The explanation for the low S/N clusters

in the cluster finding algorithm and the algorithm simply ignores the bad strip, a cluster will be found with a mean position on the bad strip, but with comparatively small charge.

As a technical aside, it should be mentioned that the extraction of the strip mask from the ORCA result files was not possible using the default `.orcarc` parameters. This would lead to the strip mask being updated every 50 events and its consequent changing on that timescale. In order to obtain a reliable strip mask, the `.orcarc` parameters `PedestalCalculator:eventsRequiredToCalibrate` and `NoiseCalculator:eventsRequiredToCalibrate` should be set to N, N being of the order of a few thousand events, and `General` should be evoked with the arguments `General A B`, where $A > 2N$ and $B = 10A$. Since only events with event numbers greater than $A - 2N$ will be used for cluster and track finding, it is recommended to generate files with the calibration data, using the `GeneralApv-AnalysisFactory:saveCalibrationData` option in `.orcarc`, followed by another run of the analysis, this time reading the calibration data from the files.

2.4 A pulse shape scan

The signals generated in a silicon strip by an ionizing particle are first integrated and amplified in a preamplifier. Then, a shaper generates pulses which are sampled every $25\,ns$, corresponding to the frequency of bunch crossings in the LHC, and then stored in an analogue pipeline on the readout chip. In order to obtain the best S/N-ratio, the sampling has to occur at the moment that the pulse height reaches its peak. The sampling time can be adjusted in multiples of $25\,ns$ using the readout chip's latency register and fine tuning can be done in steps of $1.04\,ns$ using the PLL chip, whose job is to decode trigger signals.
Because of the very short time between bunch crossings and subsequent "pileup" of particles from different events in the detector, the modules will be read out in *deconvolution mode* featuring a shorter shaping time at the expense of higher noise,

as opposed to the normal operation (*peak mode*).
In order to obtain the time dependence of the pulse height, a pulse shape scan was performed. The principle idea can be seen in figure 8. Several pion runs

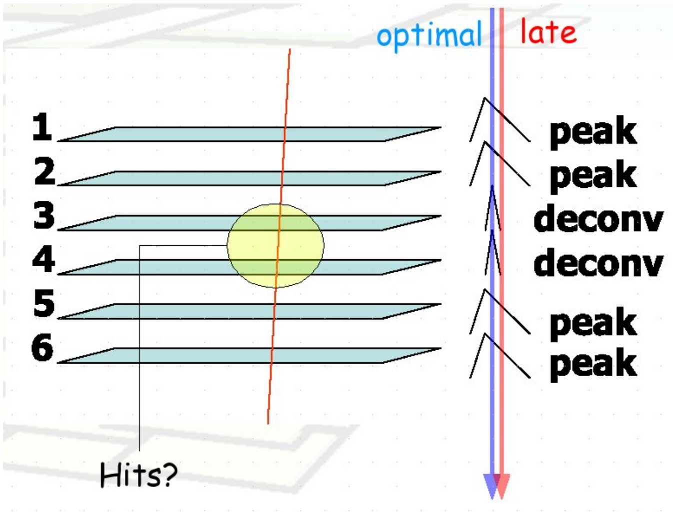

Figure 8: How to do a pulse shape scan. *(courtesy Martin Weber and Alexander Dierlamm)*

were conducted, during which the pll delay and/or latency values of the modules on layers 3 and 4 were set to different Δt with respect to the supposedly optimal sampling time. These layers used the deconvolution mode, while the settings on the remaining layers were left unchanged and peak mode was used for them. In this way, reliable track finding can still be done.
The tracks found, required to have at least five clusters, are then extrapolated to the third and fourth layers, and a cluster is looked for in the vicinity ($\Delta x_i < 1\,mm$) of the extrapolated intersection point. If one is found, its charge is histogramed.
The histograms resulting from this pulse shape scan are displayed in figure 9. Each histogram belongs to a different run, and on the horizontal axis, the charge in ADC counts is plotted, while the vertical axis contains the number of entries. A Landau distribution is fitted to each histogram, and the resulting most probable values (MPV) can be seen in figure 10. As is obvious from the number of entries in the charge histograms, almost no clusters are found for delays $\Delta t \geq 48\,ns$ or $\Delta t \leq -24\,ns$. The corresponding points in figure 10 are therefore not reliable. However, the secondary peak around $\Delta t = 35\,ns$ is prominent and needs to be explained. In addition, the charge level after $25\,ns$ seems to be at a quarter of the peak level, still.
It should be noted that this method of conducting a pulse shape scan is fast, but not optimal. Instead, it should be done after alignment and not using layers 3 and 4 for track fitting, and the signal height should be calculated from the raw data, not using cluster finding.

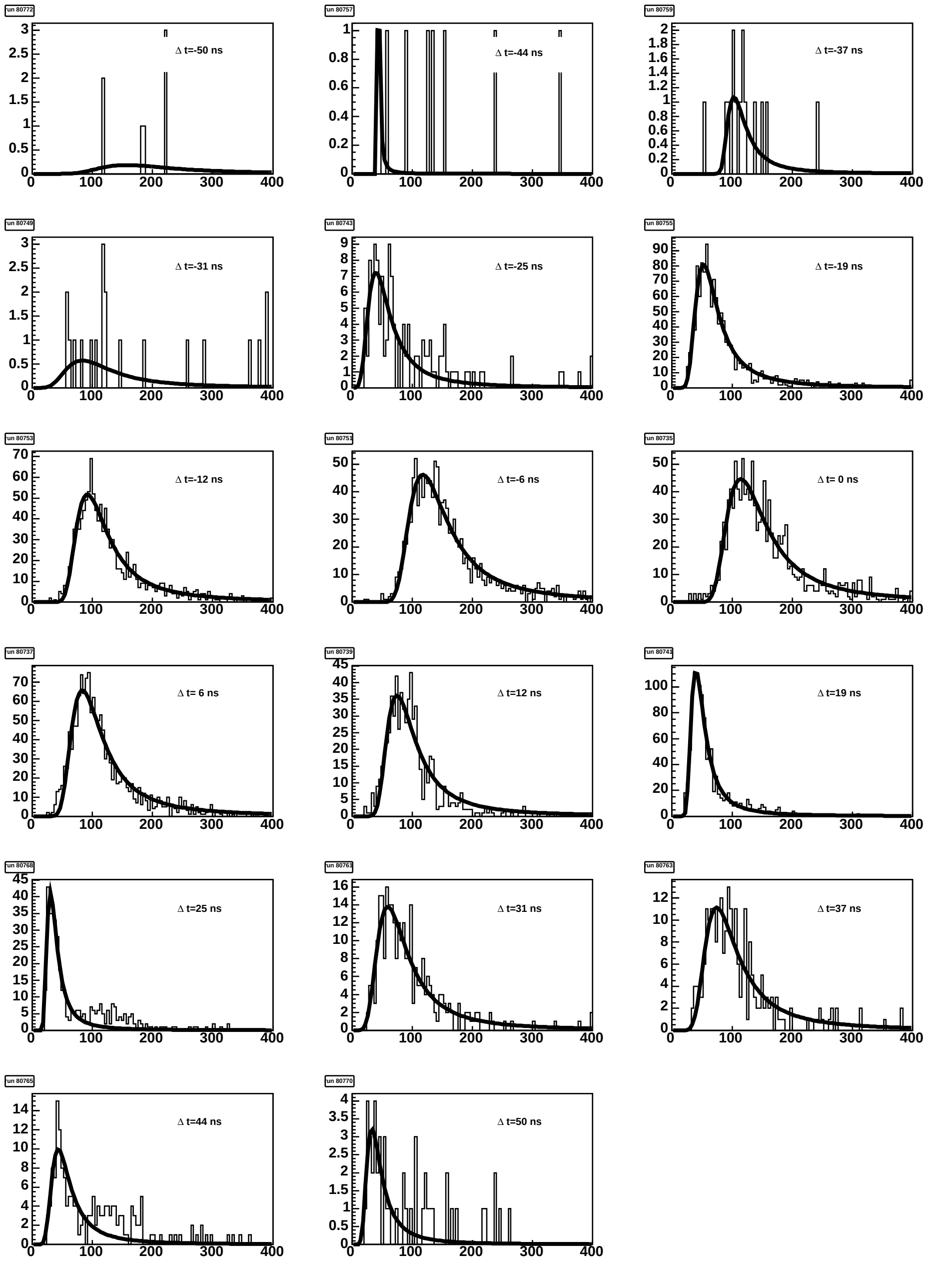

Figure 9: Charge histograms from the pulse shape scan.

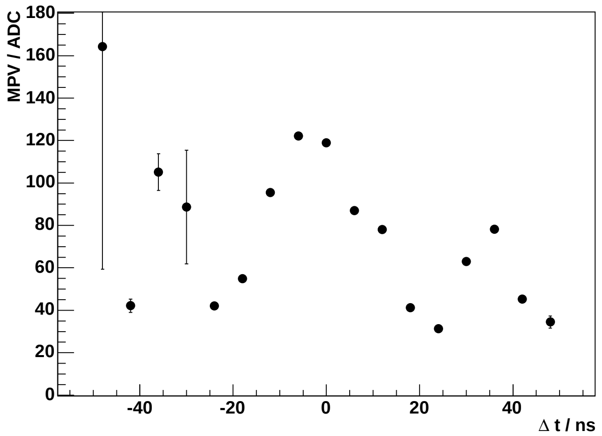

Figure 10: Pulse shape (MPV distribution in time)

Acknowledgements

I would like to thank Duccio Abbaneo for accepting me as a member of his group during this summer and for useful discussions about the issues presented above.
My supervisor, Martin Weber, has patiently instructed me on all aspects of my work here. In addition, I highly appreciated his many explanations on a large variety of topics related to particle physics.
Thanks also to Roberto Chierici for insights on how to do a pulse shape scan and to Alexander Dierlamm for looking after me in the first week of my stay.

References

[1] `http://cms-btau-datahandling.web.cern.ch /cms-btau-datahandling/TestBeam/tbeam_software.html`

[2] `http://cms-btau-datahandling.web.cern.ch /cms-btau-datahandling/TestBeam/X5may2004/Default.asp`

[3] *CMS collaboration*, The Tracker Project, Technical Design Report, CERN/LHCC 98-6

[4] *R.Brun et al.*, ROOT – an Object-Oriented Data Analysis Framework, `http://root.cern.ch`

[5] *CMS Software and Computing Group*, Object Oriented Reconstruction for CMS Analysis, CMS-IN 1999/001