SNO Electronic Calibration Constants
SNO-STR-2001-005

Steve Biller, Bill Frati, Phil Harvey, Vadim Rusu, Jeremy Sylvestre,
Richard Van de Water

August 18, 2001

Abstract

This document is a comprehensive and chilling description of how the SNO electronic calibration constants (ECA) are generated, validated, and then subsequently used for analysis. Also detailed are the behavior, various pathologies, and systematics of the constants themselves. There are many detailed tables, lists, and appendices, that are useful for technical reference.

The effects of the time and charge rate effects, known as TSLH and QSLH respectively, are not dealt with here. The ECA constants are taken at high rate, which makes the effects of these problems negligible. Because of this the ECA constants serve as a baseline from which the appropriate corrections can be made.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and General Description

The ECA (Electronic Calibration Analysis) constants are a first level, critical, and fundamental part of SNO data analysis. If these calibrations are not done right, or pathologies are not correctly accounted for, then serious errors can occur in up stream analysis, e.g. PCA, optics, energy, data cleaning, fitting, etc. Hence, a detail analysis and treatment is in order.

1.1 Electronics and DAQ

For a overview and description of the SNO electronics and trigger system, see [1], [3], and Appendix G. Electronic calibrations are required to generate the zero point charge (pedestals) and time slope calibration constants necessary to convert charge ADC counts into pedestal corrected counts above zero, and time ADC counts into nano-seconds. Note, the charge pedestal correction does not convert to photoelectrons, this is done at a later stage by the PCA (PMT Calibration Analysis). The ECA calibrations are also a insightful way to verify that the SNO electronics are functioning properly, which is a crucial systematic that needs constant diligence. As will be seen in the document, the electronics themselves are fairly robust and stable, with the zero point and slope calibrations being reproducible, even against many different variable environmental conditions.

The general idea is that a channel can fire via two methods. First, in regular data mode, where a PMT pulse trips the discriminator, which is level crossing and set on average at about 1/4 PE [2]. The second is a forced firing via a logic enable on the front end, and then a subsequent trigger Pedestal and Global Trigger issued by the central timing/trigger card. This second form of firing is completely controlled by the electronics, and is independent of the channel thresholds and the state of the high voltage. In this mode, the integrator is forced to integrate when no charge (except for some small random coincidences with PMT firing) is in the PMT/cable, and hence produces a zero point measurement. The high voltage board has a charge source that can be used to inject a controlled amount of charge of varying intensities, in conjunction with the pedestal pulse. This is used to run charge slopes, or back during the WEB wars to measure reflections from blown base terminator resistors. In pedestal mode, the time slopes can be generated by varying the time between the pedestal pulse, and the global trigger (PED-GT delay). The MTCD can be programmed to precisely generate these known delays. By sweeping through the entire channel time window (GTVALID) of the QUSN7 chip, the time slope for all channels can be generated. A distribution of GTVALID for the detector is shown in Figure 1.1. There are roughly twice the 10000 channels since there are two physical TAC's (TAC0 and TAC1) in the QUSN7 CMOS chip. Details on how the TAC timing was setup using the various Motherboard voltages can be found in [3].

The standard ECA calibration program is done twice a week, first a pedestal run (10 minutes) then a time slope run (80 minutes). A charge slope run is usually not performed as the information from this is not used except as a periodic electronics diagnostic check. The charge calibration (ADC counts to photoelectrons) is best done with PCA data. See on **surf** in /users/snodaq/doc/ECA_runs.ps for the operators procedure outlining how the runs are done. The runs, whether Pedestal, Charge or TAC slope, are done as a ECA calibration run, with the "ECA" bit in the Run header set. Appendix I describes more details about run methodology and frequency.

Once the ECA type (Ped, QSlope, TSlope) task is begun, the process is initiated by the DAQ issuing commands to the front end electronics setting the pedestal enable bit (PEDEN) for each discriminator channel that is to be pedestal pinged. A half crate at a time is done, as well, the channel triggers are disabled for this crate. Once enabled, the MTCD (trigger card) is programmed to fire the crate pedestal pulse of the selected crate at 10 Hz. When the front end discriminator receives the trigger pedestal pulse, it fires, starting the event chain as if there was a real PMT pulse. The integrator starts integrating, and the subsequent logic pulses that start the QUSN7 chip process are issued. Of course, for most pedestal firings, there is no charge, and hence, is a measure of the charge zero point. The random channel coincidence for a 400 nsec coincidence window between a 10 Hz pinging, and a 500Hz tube rate, is only 0.02%. A Global Trigger (GT) is then issued 150 nsec after the PED signal, at which point the CMOS chip begins the process of saving the data to its internal capacitors, and eventually read out and digitized by the Motherboard Sequencer (see [3] for a description of the sequencer). Each 32 channel Motherboard has four multiplexed ADC's (12 bit MAX 120, digitization range 0 to 4095), one for each charge, Q_{hs}, Q_{hl}, Q_{lx}, and the one TAC measurement. The half crate is pinged for 160 events (10 events per cell), before moving on to the next half crate.

The 150 nsec PED-GT delay during pedestal runs was chosen as the default, even though normal data is more like 250 nsec, and might make more sense. The reasoning is that if a channel has a bad GTVALID timing (too short), it might still be good in every other respect, but would be completely missed with the large delay. This way we can still make

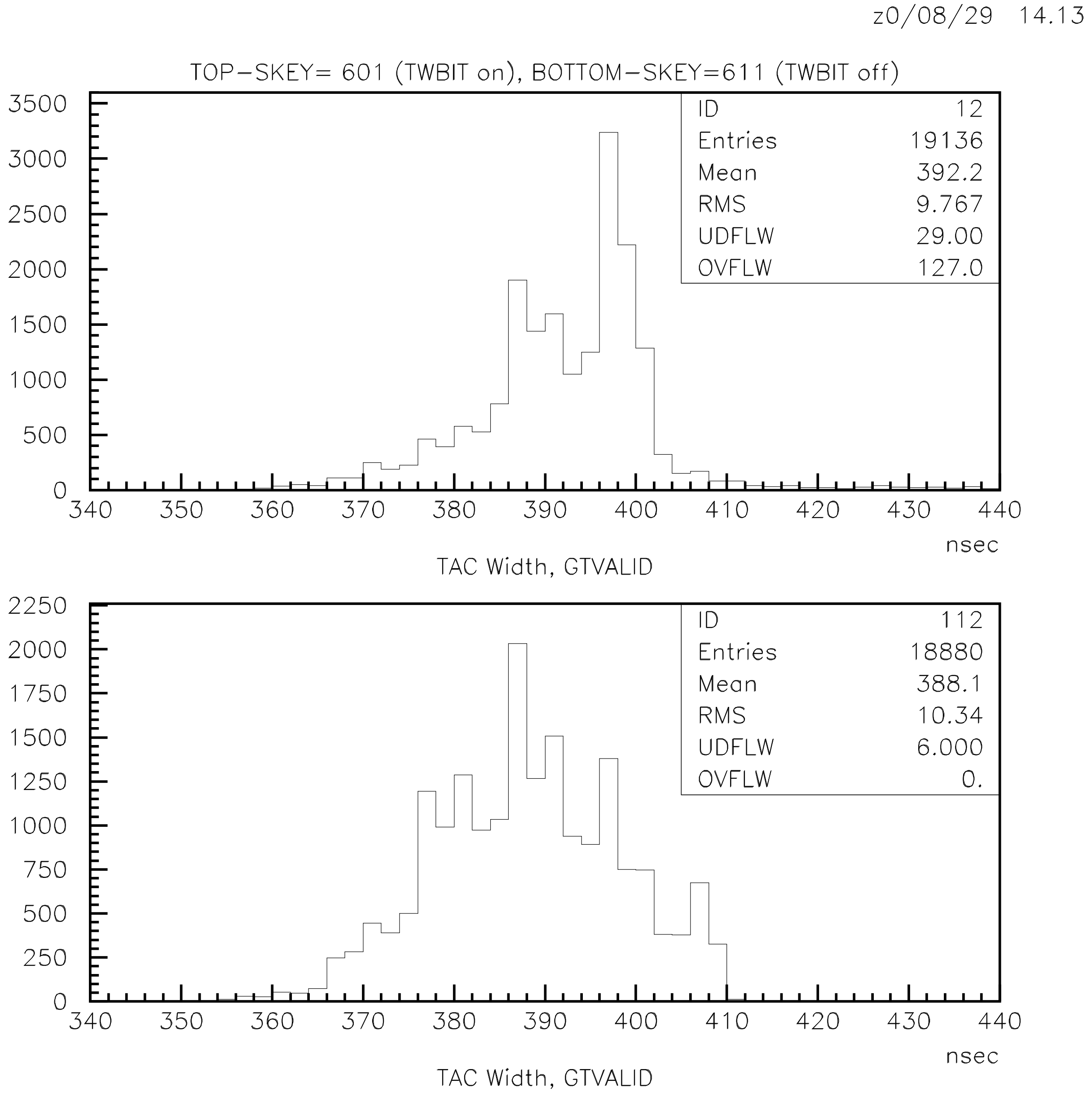


Figure 1.1: The measured TAC time window length (GTVALID) for all channels. The top figure is with TCMOS skey= 601 (twiddle bits on), and the bottom with TCMOS skey= 611 (twiddle bits off).

the channel fire and take data. Having it shorter gets too close to the short integrate time of 60 nsec, which could possibly induce noise in the Q_{hs}charge measurement.

Every channel has 16 banks of front end storage cells, where the four Qhl, Qhs, Qlx, and TAC measurements are stored. During normal data taking, the data (PMT hit plus a coincident Global Trigger) acquired by the front end is stored on one cell (bank of four capacitors inside the QUSN7 chip). The next cell is then in line for the next event. Eventually, every cell is cycled. Given that each cell has a different pedestal offset, and possibly TAC slope, ECA calibrations must be generated for each cell. It has been decided that 10 events per cell is a reasonable amount of statistics to calculate a robust median.

The ECA run is interleaved with high NHIT threshold (NHIT100MED= 50 hits) and ESUM triggered events so that we remain supernova live. The pedestal events are identified by the PEDESTAL trigger bit set in the trigger word. There is some possibility of collisions between pedestals and physics events on the order of 0.004%.

The TAC slope is done in exactly the same manner as the pedestals, but with the TAC slope sampled by changing the PED-GT delay at many different well defined intervals. Originally, the injected time slope was 19 points with 20 nsec intervals, starting with a delay of 20 nsec. After initial review, it was decided in the interest of increasing resolution in critical areas, to increase the number of points to 31 (maximum allowed by titles files structure), with a variable spacing of 20 and 10 nsec. The time ordering is shown in Table 1.1. The usual 10 events per cell is generated. The TAC slope for each cell is then constructed from the large number of PED-GT delay points (19 or 31). The data is taken at a much higher rate of 1kHz, with the estimated channel and event coincidence rates of 2.0% and 0.4%, respectively.

The actual TAC time were mapped out using an oscilloscope measuring the time difference between the start of GTVALID and the arrival of Global Trigger (GT). The difference between PED and the start of GTVALID is about 18 nsec, because of internal delay in the SNODISC/INT chips, and varies by up to a few nsec between channels. The 31 point variable PED-GT delay settings were each checked, and the corresponding true TAC time measured. The results are shown in Table 1.1. The true TAC time is consistently 10 nsec larger than the programmed PED-GT delay, even after expect delays were accounted for (e.g. 18 nsec). It is not known where or how this extra delay was put in, but since it is consistently applied, does no harm. To add to the confusion, SNOMAN adds 18.35 nsec to the programmed PED-GT delay, which then over estimates the true time by 8 nsec. Again, since it does no real harm, and that the analysis code is expecting this time delay, it is left unchanged.

The steps the DAQ control computer and eCPU take during a pedestal run are as follows:

```
1) Suspend the eCPU
2) Disable 20ns and 100ns triggers on all channels in crate
```

Point #	DAQ programmed PED-GT delay (nsec)	SNOMAN PED-GT delay (nsec)	Scope measured PED-GT delay (nsec)
1	10	28.35	20
2	20	38.35	30
3	30	48.35	39
4	40	58.35	49
5	50	68.35	59
6	60	78.35	69
7	70	88.35	80
8	80	98.35	89
9	90	108.35	100
10	100	118.35	109
11	120	138.35	128
12	140	158.35	149
13	160	178.35	170
14	180	198.35	188
15	200	218.35	209
16	220	238.35	229
17	240	258.35	248
18	260	278.35	268
19	280	298.35	288
20	300	318.35	308
21	310	328.35	319
22	320	338.35	329
23	330	348.35	339
24	340	358.35	350
25	350	368.35	360
26	360	378.35	370
27	370	388.35	379
28	380	398.35	390
29	390	408.35	400
30	400	418.35	410
31	410	428.35	420

Table 1.1: Programmed PED-GT delays for the variable TAC slope runs. The times stored in SNOMAN are the DAQ times plus 18.35 nsec (set in CECA bank). This was done to account for an internal circuit delay in the SNOI/SNOD chips. Also shown are the measured delays using an oscilloscope.

```
 3) Send Soft GT
 4) Write MTC registers if not done before (next version of SHaRC)
 5) Ship EPED START or CHANGE record
 6) Enable channel pedestals on the half crate
 7) Delay for 0.5 seconds to allow channels pedestals to settle
 8) Start the pulser firing at a fixed rate
 9) Resume the eCPU if pulsing for more than 1 second
10) Delay until specified number of pedestals are fired
11) Turn off the pulser
12) Reset the channel pedestals for the half crate
13) Resume the eCPU if not resumed in step  9)
14) Delay until eCPU has time to read out all data (assume 800 kB/s)
15) Suspend the eCPU
16) Ship EPED CHANGE record for 2nd half of crate
17) Repeat steps 6) to 15) for 2nd half of crate
18) Ship EPED STOP record
19) Send Soft GT
20) Enable 20ns and 100ns triggers on previous crate
21) Repeat steps 2) to 20) for all crates
22) Ship EPED END record
23) Send two Soft GT's
24) Resume the eCPU
```

The steps are similar for the QSlope and TSlope runs except there are repeated loops between steps 4 and 12 for each charge or PED-GT delay increment.

During pedestal pinging, the pulser is started with a fixed period and then turned off after an estimated number of events have been generated. Because of imperfections in the MAC multitasking, etc, the timing is not always exact, and causes a difference in the returned number of events relative to what is expected. Nominally, 10 events per cell is requested, with 11 usually returned. The ECA generation code only processes the first 10 events, and will throw out any remainders. A few less events can be tolerated before the validation code flags the channel.

A picture of the ZDAB data stream structure for a typical pedestal run is shown in Figure 1.2. The EPED records contains the information that feeds into SNOMAN the state of the electronics setup such as crate number, PED-GT delay, etc (see Table 2.1 for details). The PMT records contains the built raw PMT data ([4]), which is processed by the UNPACKER in SNOMAN.

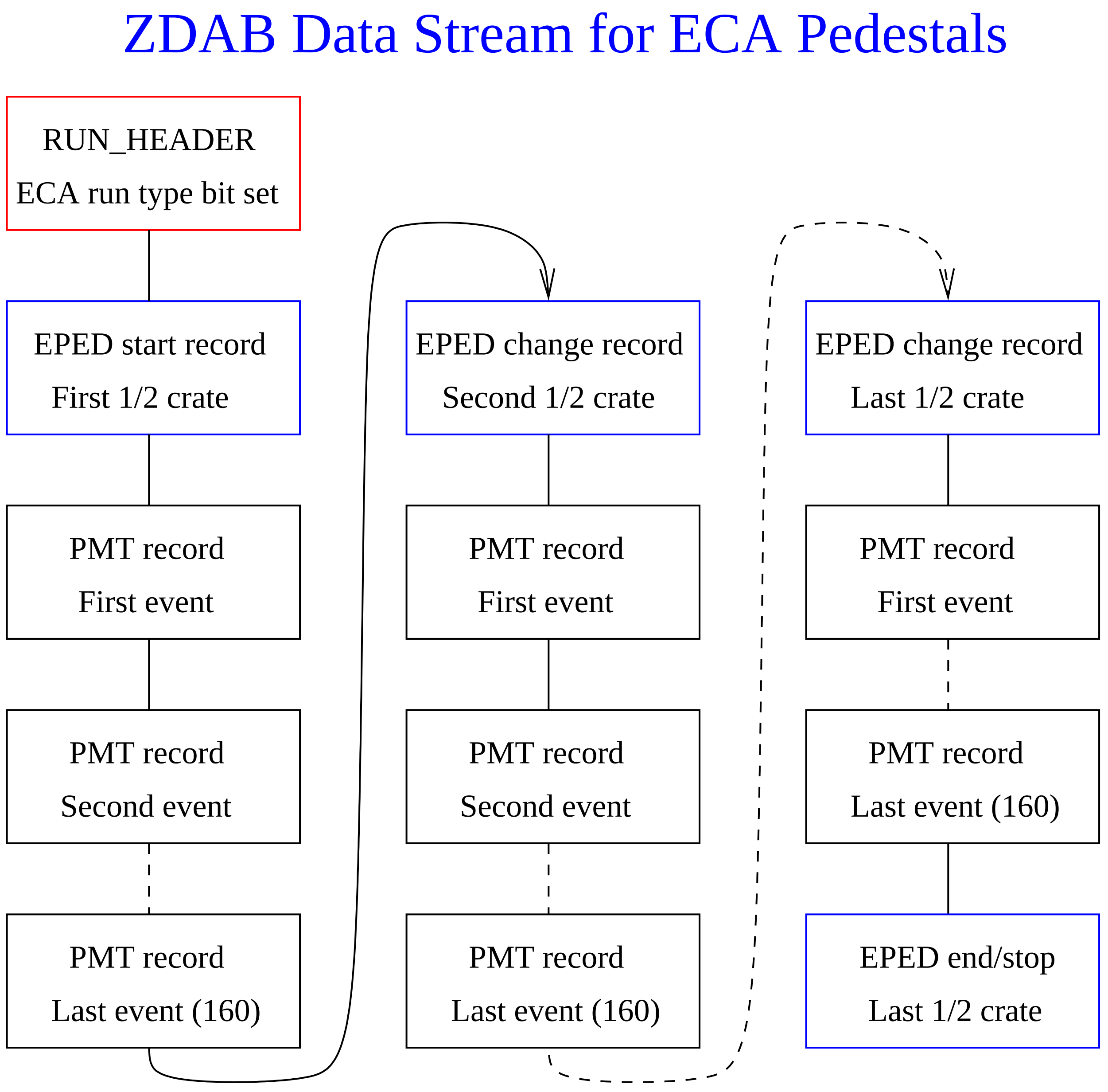


Figure 1.2: ZDAB data record stream for a typical pedestal run. Time slope record streams are identical except for the inclusion of an extra EPED record every time the PED-GT delay is changed by the DAQ control computer.

1.2 SAS and SNOMAN

At this point, the ZDAB's (raw data structure) are shipped to surface and SAS begins processing the data using SNOMAN and the current ECA generation software unit. At present, the constants are then reviewed by hand (onsite analysis/electronics expert), and if okayed, made available to be put into the SNO database. As well, Bill Frati runs his suite of hardware, TAC jump, and comparisons checks on the ECA data (detailed reports get generated, checked, and saved on **surf** in /bigdisk/penn/dawg/eca_frati).

With the new ECA validation code, the process will be the same, however, there are many more extensive checks and diagnostics done to the data automatically, with bad cells/channels flagged. The outputs are easy to view (ASCII), and are an excellent diagnostic tool for monitoring and identifying electronics problems (see Appendix H). The generated titles, log, and hbk outputs can be viewed on **surf** in /bigdisk/sas/eca. Also generated by the validation code are bit status words that are feed back into SNOMAN though the ECA title files. These words are made available to the analysis software so that bad channels can be flagged and removed from analysis, or used as a means of studying the systematics associated with bad electronics channels.

1.3 ECA Data Flow

Figure 1.3 shows an overall ECA data flow diagram which helps the reader understand the various processes and steps in the history of an ECA event.

The current chapter dealt mostly with the ECA data path up to SAS/SNOMAN. The chapters that follow discuss in detail the generation and validation of ECA constants as well as the application of the constants, all which are done within SNOMAN. A final chapter details analysis and systematic studies of the ECA constants.

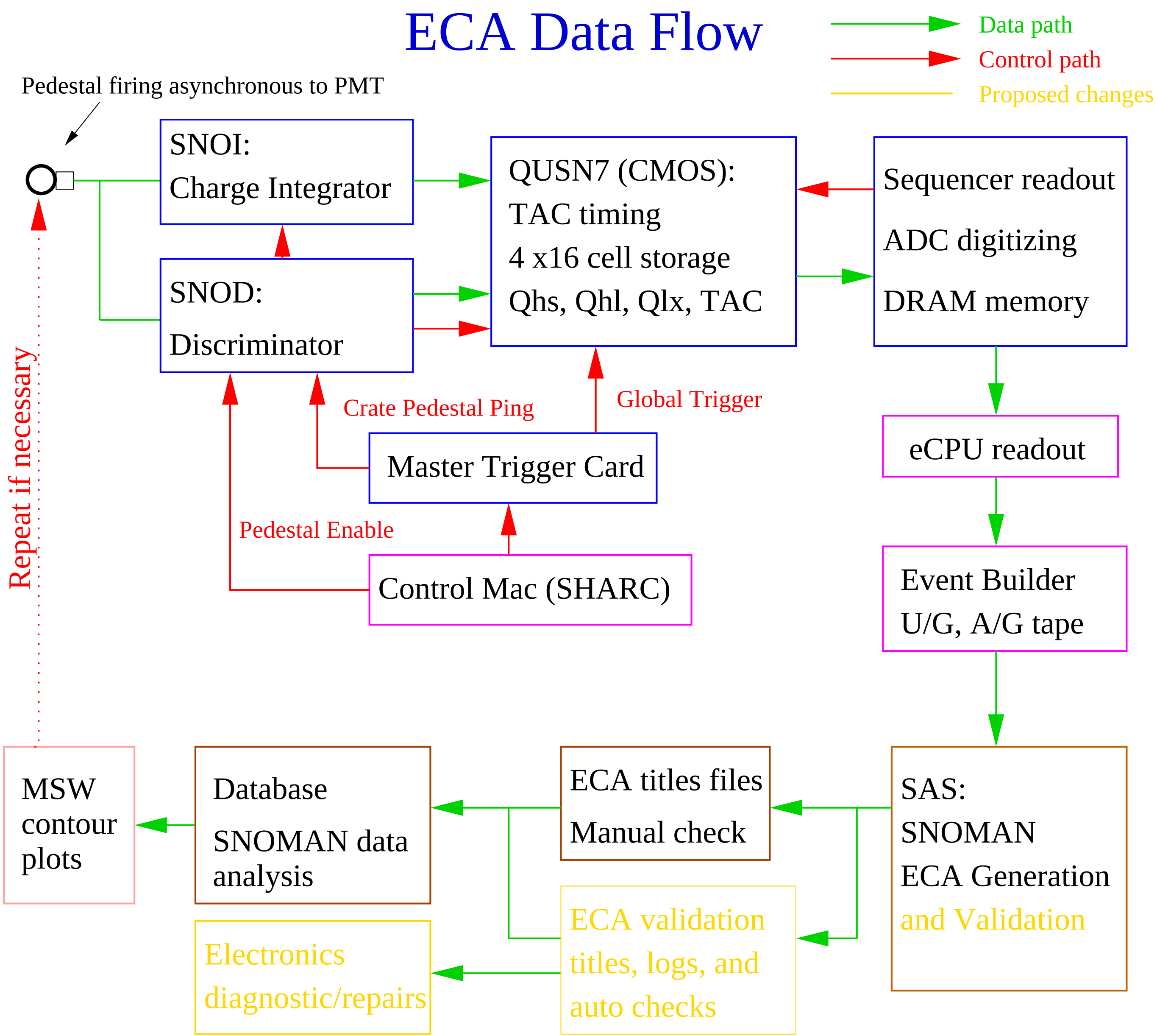


Figure 1.3: ECA data flow diagram.

Chapter 2

ECA Generation

The SNOMAN ECA processor analyses the ZDAB data containing electronics calibration, determines charge and time pedestals and slopes and creates titles files containing these data. The input control for determining what type of calibrations, and various limits or ranges for validation, etc, is dictated by the input control titles file. The details for the ensuing discussions are referred to Appendix A.

2.1 Performing the Calibration

The program **eca_electronics_calib.for** is the main routine performing the actual calibration constants generation. In a nutshell, it accumulates digitized charge and time data in 1/2 crate chunks, passes them to routines to find median values and perform fits, and then output all this information into banks containing calculated/fit parameters and/or packed histograms as a function of the known input values.

2.1.1 Initializations

The first stage of the ECA software unit (SU) checks the configuration choices given by the CECA bank (see Appendix A) and perform various initializations. For a pedestal (PED) or time slope (TSlope) run, the DQxx banks [1] for the current ECA run are loaded. As well, the current valid Ped or TSlope ECA constants are accessed from the SNO database directly, or read in via titles, so a difference check can be performed with the newly calculated constants. In the case of a charge slope run (QSlope) an optionally supplied pedestal file (described

[1]DQxx banks are generated by the DAQ and contain the detector state information, e.g. crates/channels online, thresholds, etc.

later) is loaded if the fit is to be constrained by the pedestals. It is therefore important to run a pedestal calibration first in such cases.

The event header (PMT record, see Figure 1.3) information is used to determine the type of event, where it came from and what to do with it. Events that are generated by a MTCD pedestal trigger are flagged by the Pedestal bit set in the trigger word. The only way for confusion to occur is if a real event, e.g. Radon, neutrino, pickup, etc, occurs in coincidence with the MTCD pedestal being issued, i.e. other trigger types being set in conjunction with Pedestal. In this case, the individual channels might have PMT's registering light. This overall coincidence is very low at the NHIT thresholds that are set during a ECA run. If an event does not have the Pedestal trigger bit set, regardless of other triggers set, the event is ignored.

As previously mentioned, the calibration is performed in 1/2 crate segments and the program assumes that calibration data has been taken by cycling through these 1/2 crates sequentially from the given starting to finishing crate numbers. On the start of a new 1/2 crate calibration, as indicated by the EPED record information (see Table 2.1), initialization of various counters is performed and, if it is the first 1/2 crate, then various output files and histograms are also initialized. These output files consist of generated calibration titles banks for each 1/2 crate, a detailed log, and a hbook histogram summary files. This allows details of the calibration values to be examined, and will be described in more detail later.

The information in the EPED record is important for directing the program flow, and providing critical information such as ECA run type, PED-GT delay, value of charge injection, etc. This is used by the ECA SU to properly process and bin the subsequent PMT data.

2.1.2 Accumulation of Data and Calculation of Median

For each valid event corresponding to the 1/2 crate being calibrated, the PMT bundle is first unpacked and the crate, card, channel and cell (CCCC) are determined. The three ADC values (Q_{hs}, Q_{hl}, and Q_{lx}) and the TAC are then stored for each CCCC (note that only 8 card values are needed for storage since we are dealing with 1/2 crates) in the fixed-length array *qt_events*, contained in **eca_qt_ev_temp.inc**. Up to *max_n_events* (currently set at 10) are stored here for each CCCC. Representative numbers are then taken to be the median of these values as computed by **eca_qt_med.for**. This routine also computes an effective width of the distribution for TSlope data to be use in fitting.

The median of a sample of values $x_1, ..., x_N$ is estimated by finding that value of x_i which has equal number of values above and below it (see Numerical Recipes). The values $x_j, j = 1, ...N$ are sorted into ascending order, then the formula for the median is:

```
Data Words
----------

   Data types: B(bit), I, F, D(double), H(hollerith 4 char)

 +1  I  KEPED_PEDESTAL_WIDTH              Width of Q pedestal
 +2  I  KEPED_PEDESTAL_GTDELAY_COARSE     Obvious from the name.
 +3  I  KEPED_PEDESTAL_GTDELAY_FINE       Ditto.
 +4  I  KEPED_DAC_SETTING                 Amplitude of Q pedestal
 +5  I  KEPED_CRATE_ID                    Crate ID. See note 1
 +6  I  KEPED_CALIBRATION_TYPE            Calibration type (1=q slope,
                                                 2=t slope, 3=ped)
 +7  I  KEPED_GT_ID                       Global trigger id.
 +8  I  KEPED_BEG_END                     Tells whether bank is start of
                                          calibration (1), change in
                                          injection value (2), or end of
                                          calibration (3), end of run (4)
 +9  I  RUN_NUMBER                        Run number
+10  I                                    Spare
 ..  I                                    ..
+14  I                                    ..
```

Table 2.1: The data words defined the SNOMAN EPED bank.

$$
\begin{aligned}
x_{med} &= x_{(N+1)/2} \qquad for \;\; N \;\; odd \\
&= \frac{1}{2}(x_{N/2} + x_{(N/2)+1}) \qquad for \;\; N \;\; even
\end{aligned}
$$

If a distribution has a strong central tendency, so that most of its area is under a single peak, then the median is an estimator of the central value. It is a more robust estimator than the mean is, which fails only if the area in the tails is large, while the mean fails if the first moment of the tails is large.

Because the median is insensitive to a few outliers, it is ideal for our application. Detailed studies of the electronics reveals that the typical RMS for a single cell, from internal stability and noise, is typically less than two ADC counts. However, other processes such as PMT light, external noise, etc, can cause deviations. These processes are typically random, and should only cause deviations from the mean for a few events. Other, known systematic effects, such as rate/time dependent TAC drift (Time Since Last Hit - TSLH), nonlinear ADC's (Wittich Jumps), Q_{hs} droop, long integrate noise, temperature drifts, etc, could potentially cause large enough deviations to affect the median. To first order, most of these phenomena have negligible effects under normal ECA run conditions, and can be ignored for the moment. A more detailed treatment and discussion of each of these effects will be done later in Chapter 6.

Figure 2.1 shows an example of the three charge and one TAC raw pedestal data for a typical cell. As well, Figure 2.2 shows a typical TAC slope for an individual cell using the 31 point variable PED-GT delay slope. Each point in the slope is the calculated TAC median for the injected PED-GT time delay. Notice the TAC curl at large times (early light), and the slight non-linearity at short times (late light).

As will be shown later in Chapter 4, the width and difference statistic demonstrates that the median is very stable and robust. Because no additional pruning of outliers based on RMS deviations is necessary, then for a pedestal calibration run, the calculated medians will simply become the pedestal values. For charge or time slope runs, the medians become ADC count for the know injected value of the charge or time.

2.1.3 Issued Error Statements

During the calibration, the following inconsistencies result in a call to **eca_stop** and the issue of an error statement:

1. The crate seen at the start of calibration is not the one designated as the starting crate

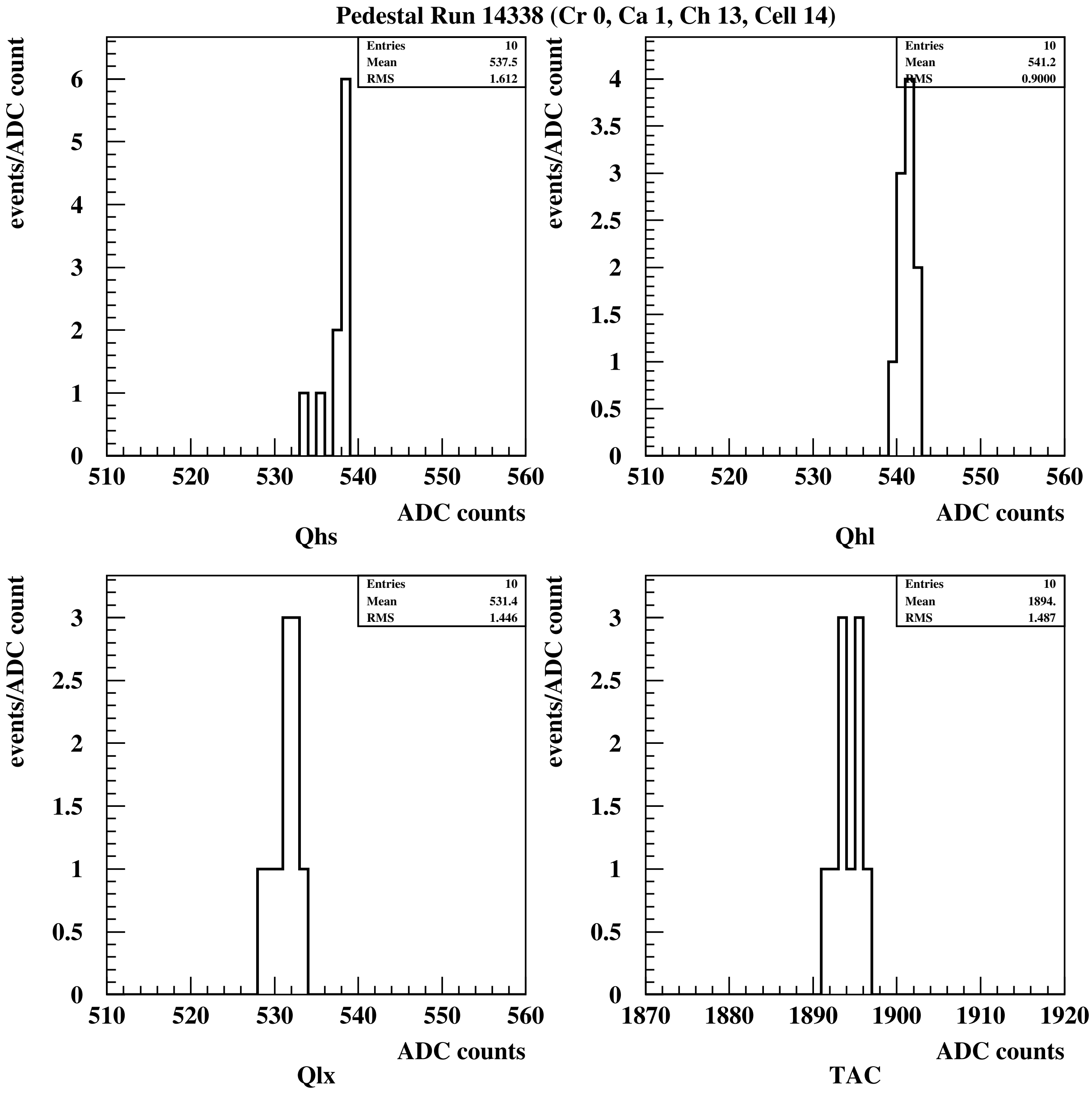


Figure 2.1: A single channel cell raw data for the three charge and one TAC measurements.

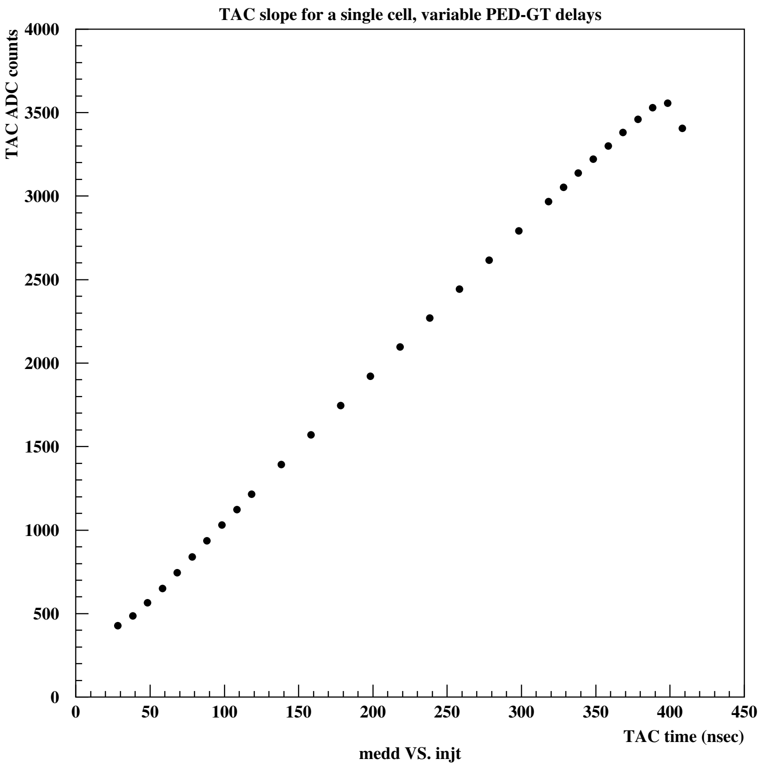


Figure 2.2: Time slope for a single cell using the 31 point variable PED-GT delays as shown in Table 1.1. Notice the increased time sampling at the two non-linear regions at either end of the TAC range. A software cut is used to remove the TAC curl region at large times (see Section 5.3).

in the CECA bank

2. There's no change in the crate number or injected charge value despite a header indicating a change has occurred

3. The crate number hasn't advanced despite a header indicating the calibration of the crate is finished

4. A header indicating the end of calibration is given before a header indicating the previous crate is done

5. A header indicating the end of calibration is given without the last crate listed in the CECA bank being done

6. The header type is not recognized by ECA

7. The 1/2 crate number is invalid

8. There is a mismatch between the given 1/2 crate number and the calculated value based on channels

2.1.4 Calibration Processing, Fitting, and Output

After each 1/2 crate is finished, data files corresponding to the type of calibration are either opened or re-opened, a bank header for the corresponding 1/2 crate is written (via **write_titles_header**) and the calibration data is processed and output to the file by **eca_qt_out.for**. This routine first calls **eca_qt_out_init.for** to initialize counters and write configuration information specific to the type of calibration being performed. After loading various values into local variables, **eca_qt_filter.for** is then called to perform the tagging of suspicious channels and also to identify "bad" charge/time samplings in the case of QSlope or TSlope calibrations. The algorithm for identifying bad samplings (data points), TAC curl, etc, is described in detail in Chapters 3 and 4.

One assumption in this algorithm is that data is taken so as to produce sequential, digitized values from lowest to highest. The TSlope data sampling should also be chosen so as to sufficiently reveal the TAC curl, in which case the filtering should effectively mark the calibration region to be ignored.

Next, **eca_qt_fit.for** is called to perform linear fits for the QSlope calibration, and linear and cubic fits for the TSlope calibration (the TSlope linear fit is used for diagnostics and as an input starting point for the cubic fit). For the QSlope calibration, if the resulting linear slopes are outside of the tolerances designated in the CECA bank for each calibration type, all data points are tagged as being bad. For the TSlope calibrations, a more sophisticated set of checks are done, which is outline in the next two chapters.

Both the initial linear and subsequent cubic fits are simple non-weighed least-squared algorithms. The purpose of these fits is not so much for actual calibration (though there is an option to use them as such), but more as a simple means of checking the sensibility and robustness of the data. The actual TAC calibration is done as an interpolation between adjacent injected time points, and will be described in more detail in Chapter 5.

Calibration flags and, in the case of TSlopes, data points are now packed in preparation for output. Calibration information is then written out to titles files and **eca_summary.for** updates summary files containing histograms and ntuples. The various file formats for these are described in the next chapter. Note that an important feature of this calibration code is that the basic TSlope calibration data are preserved in compressed form within the titles file. The current approach to applying the calibration involves the use of interpolation between these data, but other schemes (such as the use of the cubic fit parameters, also included) may thus also be used without regenerating the basic titles files.

Chapter 3

ECA Validation

In an effort to provide more detailed information concerning the validity of Electronic Calibration (ECA) constants to data analysis, the SNOMAN code which is responsible for reading in ECA runs and writing out ECA Titles Files has undergone a major revision which takes effect in SNOMAN version 4_0181. More specifically, new routines have been written to test and report on the validity of the Pedestal and Time Slope calibration constants.

3.1 New Code

Various routines were changed, and new ones created to make the changes necessary to perform the ECA validations. See Appendix C for a list of the new/changed routines.

As well, many additions have been made to `eca_control.dat`, refer to Appendix A for details.

3.2 New Naming Convention for Output Files

To make it easier to match up ECA runs with their titles file, the naming convention on Pedestal and Time Slope titles files has been altered to include the run number. The new naming convention is

x`eca_`*rrrrrrrrrr*_*yyyymmdd*_*hh*`.dat`

where *x* will be '`p`' or '`t`' for pedestals or time slopes, respectively, and *rrrrrrrrrr* will be the run number.

Similarly, the naming convention for the ntuple file has changed:

*x*eca_*rrrrrrrrrr*_*yyyymmdd*_*hh*.hbk

The extension has been changed to 'hbk' since the files no longer contain an ntuple of all the pedestal/time slope data (see the subsection on "New Diagnostic Histograms" in each of the "Pedestals" and "Time Slope" sections).

Finally, since much new diagnostic information will be available in the SNOMAN log files (see the "New Diagnostic Output to SNOMAN Log File" subsection in each of the "Pedestals" and "Time Slope" sections), these log files will be saved and renamed as

*x*eca_*rrrrrrrrrr*_*yyyymmdd*_*hh*.log

3.3 Pedestals

3.3.1 New Cell Status Word

The existing Pedestal Title Banks contained one word for every cell, in addition to the four pedestal value words, to be used to report whether those pedestal values were "suspicious" or "OK". The word has been redefined by the new ECA Validation code to be a bit-mask corresponding to the passing or failure of various tests on the pedestal data, as described in the following section, "Summary of Tests". Use the *ibits* function to test the status of each bit or a combination of bits in this bit mask. Alternately, one may unpack the word with the *blow* routine, but since the word was not *bunch*'ed to begin with, the 32nd unpacked flag will correspond to the 0-th bit, the 31st unpacked flag will correspond to the 1st bit, and so on.

See Table 3.1 for a bit-by-bit description of the Pedestal Cell Bit Mask. See the "New Title Bank Format" section for details about where in the PDST Title Bank this word is located.

Note: Bits 21 through 27 are ORs across cells for the channel. For example, bit 21 will be set if at least one cell in the channel that the current cell resides in fails at least one of the three QHS pedestal tests (median, width, difference).

3.3.2 Summary of Tests

A list of the tests and checks performed on ECA Pedestal data follows. Stated test limits/ranges can be changed in `eca_control.dat` (see Appendix A). A discussion on test limits/ranges selection is done in Chapter 4.

0	Overall cell mask (OR of bits 1 through 14)
1	QHS median
2	width
3	difference from previous ECA run
4	QHL median
5	width
6	difference from previous ECA run
7	QLX median
8	width
9	difference from previous ECA run
10	TAC median
11	width
12	difference from previous ECA run
13	Does cell have 10 events? (yes=0, no=1)
14	Does cell have >0 events? (yes=0, no=1)
15	Overall channel mask (OR of bits 16 through 27)
16	MB ID check
17	DB ID check
18	Overall QHS for channel
19	QHL
20	QLX
21	TAC
22	Overall median test for channel
23	width
24	difference
25	Channel sequencer status check (from DQCH banks)
26	Does channel have 10*(16 cells)=160 events? (yes=0, no=1)
27	Does channel have >0 events? (yes=0, no=1)
28–31	spare

Table 3.1: Cell Bit Mask (Pedestals)

Median Tests

Median pedestal values are tested against an acceptable range (upper and lower limit), which define the normal operating region of the electronics. Since each cell/channel offset (pedestal) is deterimined by the intrinsic chip characteristics and setup voltages, the acceptable range can be quite large. Note that the acceptable ranges can be independently altered, through `eca_control.dat`, for each individual charge type.

Width Tests

Typically, ten pedestal events are (should be) read back from each cell during an ECA Pedestal run. The width of this ten-event distribution is calculated for each pedestal type by sorting the events by pedestal value in ascending order and taking the difference between the sixth and fourth values of the ordered set. A check is then done on the width to ensure it is below a nominal level. Again, individual width checks can be programmed for each of the three charge and TAC channels common to a single Motherboard. This test is an excellent check on the stability of the ECA constants during the short period that they were taken.

Difference Tests

By reading in the PDST Banks from the most recent ECA Pedestals run, the validation code is able to compare each cell's current and previous pedestal values. The difference calculated is current minus previous. Note that if the previous pedestal value is -9999 (see discussion below test descriptions for information on when this occurs), the Difference Test will not be performed. If the difference is found to be greater than a preselected value, then the cell is flagged for the charge or TAC measurement in question. Like the width test above, the difference test is an excellent measure of stability, but over the period since the last time the ECA was done.

Number of Events Returned Tests

The validation routines test the number of events that each cell returned, flagging a cell if it returned less than 10 events or if it returned no events, with separate flags for each of the two conditions. If a cell returns no events then none of the other tests will be performed. A channel is flagged if any of its 16 cells returned less than 10 events (ie. the channel returned less than 160 events) or if no events were returned from any of the cells in that channel, again with separate flags for each of the two conditions.

Board ID Tests

The new Pedestals Title Bank format which will be introduced along with this code (see the following section for a description of the new titles format) includes board ID numbers at the end of each PDST Bank. The validation code reads in the board ID numbers from the PDST Bank from the most recent ECA Pedestals run and compares them with the board ID numbers from the current run's DQID bank. Both motherboard (MB) and daughterboard (DB) ID numbers are checked and a flag is set if any of the ID's differ. As well, the "Difference Tests" are not performed for the affected cells if any of the board ID's differ.

Channel Sequencer Test

For each channel, the DQCH bank is checked to see if the channel's sequencer was enabled. If it was not, a flag is set and the validation code does not perform any of the other tests mentioned above since no events would have been returned from any cell in that channel.

IMPORTANT NOTE

If a cell returns no events, either through malfunction or because the sequencer was disabled for the channel in which that cell resides, the pedestal values for that cell will be set to -9999. As stated above, if a cell returns no events or its channel sequencer is disabled then none of the other cell tests will be performed. In particular, the flags for the Median Tests will NOT be set, and thus it is very important to test the flags corresponding to the "Cell returned no events" and the "Channel sequencer disabled" conditions before deciding that a cell's pedestals are acceptable based on the Median Test flags. Even better would be to additionally check that the pedestal values for each cell are greater than zero, regardless of the bit mask for the cell.

3.3.3 New Pedestal Title Bank Format with Validation

Changes:

- Reordering of write-out sequence of data.
- Redefinition of cell status word, as described in the "New Cell Status Word" section.
- Addition of board ID numbers.
- Addition of Validation failure totals.
- Addition of 16 spare words.

To reflect the major format changes to the PDST Banks, Bank Numbers will run from 101 to 138, whereas the previous PDST Banks were numbered 1 to 38. Note that each of the Bank Numbers still corresponds to a particular half-crate — eg. bank numbers 107 and 108 correspond to the first and second halves of crate 3 respectively.

Data Words
Data Types: b (bit), i, f, d (double), h (hollerith 4 char)

Word	Type	Description

Pedestals:

2560*jcard + 80*jchannel + 5*(jcell-1)		
+1	i	Bit-masked status word for cell (see description below)
+2	i	QHS pedestal (in ADC counts) for cell
+3	i	QHL pedestal (in ADC counts) for cell
+4	i	QLX pedestal (in ADC counts) for cell
+5	i	T pedestal (in TDC counts) for cell

Card ID's:

5*jcard + 20480		
+1	h	Mother Board ID for card
+2	h	Daughter Board ID for DB 0 of card
+3	h	Daughter Board ID for DB 1 of card
+4	h	Daughter Board ID for DB 2 of card
+5	h	Daughter Board ID for DB 3 of card

HALF-CRATE Stats:

20520		
+1	i	# of channels w/ sequencer disabled
+2	i	# of channels which failed MB ID check
+3	i	# of channels which failed DB ID check
+4	i	# of channels which had <160 events
+5	i	# of channels which had 0 events
+6	i	# of bad QHS channels
+7	i	# of bad QHS median channels
+8	i	# of bad QHS width channels
+9	i	# of bad QHS difference channels
+10	i	# of bad QHL channels
+11	i	# of bad QHL median channels
+12	i	# of bad QHL width channels
+13	i	# of bad QHL difference channels
+14	i	# of bad QLX channels
+15	i	# of bad QLX median channels
+16	i	# of bad QLX width channels
+17	i	# of bad QLX difference channels
+18	i	# of bad TAC channels
+19	i	# of bad TAC median channels
+20	i	# of bad TAC width channels
+21	i	# of bad TAC difference channels
+22	i	# of bad QHS or TAC channels
+23	i	# of bad QHS or TAC median channels
+24	i	# of bad QHS or TAC width channels
+25	i	# of bad QHS or TAC difference channels
+26	i	total # of bad channels

There then follow 16 spare integer words.

Above, jcard=0,..,7; jchannel=0,..,31; jcell=1,..,16
(if in first half of crate, jcard = card#; if in second half, jcard = card#-8)

3.3.4 Pedestal Diagnostic Output to SNOMAN Log File

A large amount of new diagnostic information is dumped to the SNOMAN log file by the validation code. When generated by SAS, these output files will be saved on **surf** in /bigdisk/sas/eca/log with the name format described in Section 3.2.

0	Too many channels with 0 events
1	Too many channels with lt160 events
2	Too many channels failed board ID test
3	Too many channels failed QHS tests
4	QHL
5	QLX
6	TAC
7	Bad 1/2-crate QHS average pedestal
8	QHL
9	QLX
10	TAC
11	Bad 1/2-crate QHS average width
12	QHL
13	QLX
14	TAC
15	Bad 1/2-crate QHS abs. average difference
16	QHL
17	QLX
18	TAC
19	Too many flagged channels
20–25	spare
26	UC (or old DCR_ACTIVITY) bit was set
27	Missing DQID Bank - could not check board IDs
28	Missing DQCH Bank - could not check chan seqs
29	Missing PDST Bank - could not check diffs or IDs
30	Did not reach end of last crate
31	spare

Table 3.2: Overall Pedestals Run Bit Mask

First, the integrity of the entire ECA run is summarized in a 32-bit mask and is written to the `.log` file as an integer. Use your favorite integer-to-binary conversion routine to expand this code number. Table 3.2 contains a description of this bit mask; for a description of the tests which determine whether or not these bits are set, see Chapter 4.

If any of the bits are set then the integer version of this overall bit mask will be non-zero; in this case the phrase "ECA_PROBLEM" will be written to the `.log` file followed by the actual integer value of the mask. Thus, when SAS finishes processing an ECA run, it should *grep* the `.log` file for the phrase "ECA_PROBLEM". If this phrase is found, SAS should email the detector manager or an ECA expert so that the run can be examined more closely before the titles file is moved into the ECA titles directory and/or database.

If any of bits 27 through 29 are set, some of the ECA Validation summaries, test results, and histograms should not be trusted. The Pedestal values in the titles file are not affected by these missing banks, but cell status bit-masks will be affected and thus it is advised that the ECA run be analyzed again offline by an ECA or SAS expert, ensuring that the required banks are properly loaded. If bit 30 is set, then there was a serious error either during the actual ECA run or during the analysis by SAS/SNOMAN. In this case action must be taken as the Pedestal titles file will not be complete.

Following this, tables containing total numbers of failures for the various tests are written, with a separate table for the channel and cell summaries, respectively.

Next, average pedestal, width and difference values for each of the four quantities (QHS, QHL, QLX, and TAC) are written out by the half-crate. For the difference values, both $(\Sigma x)/n$ and $(\Sigma|x|)/n$ are provided.

After this, total numbers of failures for several of the tests, accumulated for each board, are written. Interspersed with this will be single-line messages for any channel with its sequencer enabled which returned zero events. These messages will appear directly after the summary for the board in which such a channel resides.

Next, a detailed description of all test failures is written out for each channel which contains at least one cell which failed at least one test.

Finally, if the corresponding flag is set in `eca_control.dat` then all 10 pedestal events for all cells will be dumped to the `.log`. The inclusion of this data greatly increases the size of the `.log` file and so this option should normally be left off. Another flag in `.log` allows one to specify whether or not the 10 pedestal events should be sorted by value, lowest to highest, for each cell; if this flag is set to "unsorted" then the events for each cell are dumped in the order they were read back and an asterisk (*) is placed next to the cell which was read from first in each channel.

All of the above mentioned tables and lists in the SNOMAN log file have been intentionally designed to facilitate easy *grep*'ing for desired diagnostic messages. For more information on the format of this diagnostic information it is best to refer directly to one of these log files, whenever they become available.

3.3.5 Diagnostic Histograms

See Table 3.3 for a list of the histograms which can be found in the pedestals `hbk` file. Note that histograms 8101 through 9916 are only available if the appropriate flag is set in `eca_control.dat`.

3.4 Time Slopes

3.4.1 Cell Status Words

Two bit-masked words per cell have been added to the Time Slope Title Banks, complementing the existing bit-masked word. Below is a description of all three cell status words.

1st 32-bit Word – Bad points flag

- Word consists of 32 packed flags — use the *blow* routine to unpack.
- Each flag corresponds to one of the (max.=31; typical=19) injected-time steps, with at least one spare flag.
- Flag is set if corresponding point was excluded from cubic fit calculation (TAC curl region and no/low event points).
- Unused 32nd flag is always set to avoid a Zebra peculiarity when reading in data from the titles file.
- This word already exists in current TSLP Bank format; no change was made.

2nd 32-bit Word – Suspicious points flag

- Word consists of 32 packed flags — use the *blow* routine to unpack.
- Each flag corresponds to one of the (max.=31; typical=19) injected-time steps, with at least one spare flag.
- Bit is set if corresponding point was "suspicious", where "suspicious"-ness is defined as in the description of the "Number of Suspicious Points Test" (see the "Summary of Tests" section).
- Unused 32nd flag is always set to avoid a Zebra peculiarity when reading in data from the titles file.
- This word is a new addition to the TSLP Banks.

3rd 32-bit Word – Validation status flag

- Word is a 32-bit mask — use the *ibits* function to test each bit or a combination of bits. Alternately, the word may be unpacked with the *blow* routine, but then the 32nd unpacked flag would correspond to the 0-th bit, the 31st unpacked flag would correspond to the 1st bit, and so on.
- Function of each bit is defined in Table 3.4. Each bit corresponds to the passing or failure of various tests on the Time Slope data, as described in the following section, "Summary of Tests".
- This word is a new addition to the TSLP Banks.

See the "New Title Bank Format" section for details about where in the TSLP Banks these words are located.

3.4.2 Summary of Time Slope Tests

A list of the tests and checks performed on ECA Time Slope data follows. Stated test limits/ranges can be changed in `eca_control.dat` (see Appendix A). A discussion on test limits/ranges selection is done in Chapter 4.

Fit Parameter Tests

A cubic fit is performed to each cell TAC slope, excluding the TAC curl and low event points (see below "Number of Points Used in Cubic Fit Test"). Three of the four cubic fit parameters (cubic, quadratic, and linear term coefficients) are tested against an acceptable range. The intercept is not tested since this is only an offset, as well, it is already checked by the TAC pedestal tests.

Fit Parameter Difference Tests

By reading in the TSLP Banks from the most recent ECA Time Slopes run, the validation code is able to compare each cell's current and previous cubic fit parameters, i.e. cubic, quadratic, and linear terms. The difference calculated is current minus previous. The linear term should be the most robust to variations, and should be tested the most stringently.

The difference test is not performed if the board ID has changed since the last ECA run nor if the slope parameter from the previous ECA run is exactly zero.

Note on Fit Parameter Tests: In the event that NO injected-time points are deemed suitable for inclusion in the cubic fit (see "Number of Points Used in Cubic Fit Test" below), the cubic fit parameter flags will not be set.

Point-by-Point Width Test

Typically, ten pedestal events are read back at each injected-time point from each cell during an ECA Time Slopes run. The width of this ten-event distribution is calculated for each injected-time point by sorting the events by median digitized TAC value in ascending order and taking the difference between the sixth and fourth values of the ordered set. A test is done on the width, and if a preselected number of points for the same cell time slope fail the width test, then the entire cell is flagged. By requiring more than one failed point per time slope, this increases the robustness of the test, making it less sensitive to random fluctuations.

Point-by-Point Number of Events Returned Test

The validation routines test the number of events that each cell returned at each injected-time point, if it returned less than 6 of the requested 10 events, then the points fails. As in the width test above, a certain preselected number of points have to fail before the entire time slope for the cell in question is flagged.

A separate flag is also set if the channel returned no events for all of the injected-time points (potential blind flasher). A channel is flagged if any of its 16 cells failed the Point-by-Point Number of Events Test, or if no events were returned from any of the cells in that channel, again with separate flags for each of the two conditions.

Number of Points Used in Cubic Fit Test (TAC Curl Identification)

If the median digitized TAC value at a given injected-time point does not show an increase from the previous injected-time point, that point and the two (or four if the number of injected time points is greater than 25) before it are excluded from the cubic fit parameter calculations. The purpose of this is to exclude the TAC curl from the fit. As well, if a cell returned less than 6 events at a given injected-time point, that point is not used in the fit. If the number of points excluded from the cubic fit by these two checks is more than a preselected value, then the cell is flagged.

Points that are identified above are flagged as “bad points”, and are due predominately to the TAC curl. This information is packed into the titles output file (section 3.4.1) and is used by the ECA calibration application routine to identify hits that occur in the TAC curl region. Points that fail internally in the TAC slope are removed, and interpolation is done without that point. See Chapter 5 for a discussion of how the above information is applied to calibrate the data.

Number of Suspicious Points Test

The median digitized TAC value at a given injected-time point is labeled as "suspicious" if that injected-time point failed either the "Point-by-Point Width Test" or the "Point-by-Point Number of Events Returned Test" (or both). If more than a set number of suspicious points are found, then the cell is flagged. Each point that has been identified as suspicious has a flag set which is packed into the titles output file (section 3.4.1) and is used by the ECA calibration application routine (see Chapter 5) to determine if the point is used for time interpolation.

Frati TAC Jump (nonlinear ADC) Test

During detector commissioning, nonlinear behavor of the MAX 120 ADC's was observed (called "Wittich Jumps", in honor of the discoverer, not to be confused with "Wittich Pulses", another famous SNO detector glitch), which seem to progressively get worse, i.e. more ADC's affected with time. A test was developed using the Motherboard calibration DAC (CALDAC) that clearly demonstrates the effect, as seen in Figure 3.1. More details of the nonlinear ADC systematic can be found in Chapters 5 and 6.

The problem with the CALDAC test is that it requires the detector HV to be turned off, and was done at most once a month. This is a low sampling rate given that about 20 or more ADC's can go bad in this period of time. A test using the time slope ECA data has been developed by Bill Frati that can identify bad TAC ADC's. Note that this test does not work for charge, since we are only sampling one point, i.e. pedestal. The Bill Frati's TAC Jump Test algorithm has been emulated as closely as possible in the ECA validation code.

The TAC digitizations are parametrized as a third order polynomial of the input time and the four parameters of the polynomial are obtained using a simple Chisquare analysis. TAC jumps are detected by looking at the residuals of the fit at all ECA calibration points between digitizations of 60 to 350. Should the residual at any point be greater than 1ns or the Chisquare of the fit be greagter than 100, then that chip is flagged a potential TAC jumper.

A somewhat simplified pseudo-code summary of this implementation follows:

Assume that $X(1:N)$ contains the injected-time values and $Y(1:N)$ contains the corresponding median digitized TAC values at each injected-time point, where N is typically 19 or 31 (as set in the Scheduler in SHaRC). Further assume that for the Alternate Cubic Fit Calculation that any "bad" injected-time points, as discussed in the description of the "Number of Events Used in Cubic Fit Test", have already been thrown out and all the "good" points have been collected, in order of injected-time, in arrays $X_1(1:N_1)$ and $Y_1(1:N_1)$, where the X_1-array holds the injected-time values, the Y_1-array holds the corresponding median

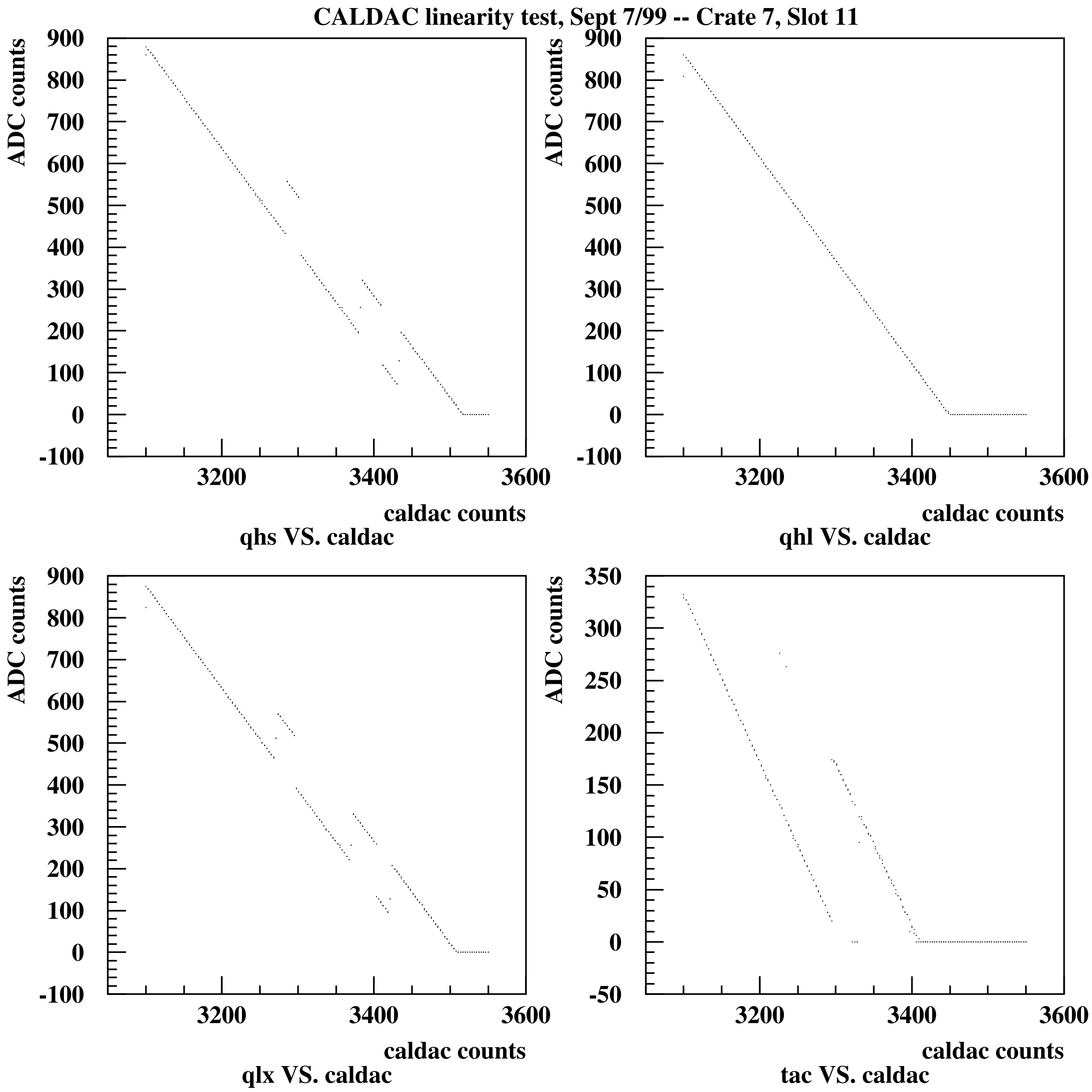


Figure 3.1: Example of nonlinear ADC's from a CALDAC test. Only Q_{hl} is okay, while the other channels exhibit nonlinearities (jumps).

digitized TAC values at those injected-time points, and N_1 is the number of "good" points.

Alternate Cubic Fit Calculation:

```
Do an initial linear fit on X_1 and Y_1 ⟶ Ỹ_1(x)
Do I = 1, N_1
  Let D_1(I) = |Ỹ_1(X_1(I)) − Y_1(I)|
Enddo
Let N_2 be the number of points (out of N_1) with D_1(·) < 50.0
If N_2 > 10 then let N_2 = 10
Place the N_2 points with the smallest D_1(·) in arrays X_2(1 : N_2) and Y_2(1 : N_2),
    in order of injected-time
Do a linear fit on X_2 and Y_2 ⟶ Ỹ_2(x)
Do I = 1, N_1
  Let D_2(I) = |Ỹ_2(X_1(I)) − Y_1(I)|
Enddo
Let N_3 be the number of points (out of N_1) with D_2(·) < 50.0
Place these N_3 points in arrays X_3(1 : N_3) and Y_3(1 : N_3), in order of injected-time
Do a linear fit on X_3 and Y_3, yielding parameters L_1, L_2
Set initial guess for cubic fit parameters to:
    C_1 = L_1
    C_2 = L_2
    C_3 = C_4 = 0.0
Do cubic fit on X_3 and Y_3 ⟶ Ỹ_C(x), using parameters C_1, ..., C_4 as an initial guess
If χ^2 for cubic fit satisfies χ^2 > 200.0 then
  Do I = 1, N_3
    Let D_C(I) = |Ỹ_C(X_3(I)) − Y_3(I)|
  Enddo
  Remove point with largest D_C(·) from X_3- and Y_3-arrays, and let N_3 = N_3 − 1
  Repeat linear fit on X_3 and Y_3, yielding parameters L_1, L_2
  Reset initial guess cubic fit parameters to:
      C_1 = L_1
      C_2 = L_2
      C_3 = C_4 = 0.0
  Repeat cubic fit on X_3 and Y_3 ⟶ Ỹ_C(x), using parameters C_1, ..., C_4 as an initial
      guess
Endif
```

TAC Jump Test Using Alternate Cubic Fit $\widetilde{Y}_C(x)$:

```
If χ^2 for Alternate Cubic Fit satisfies χ^2 > 10000.0 then
  Flag cell
Else
  Do I = 1, N
    If X(I) > 59.0 and X(I) < 350.0 then
```

```
      Let T be the root of Ỹ_C(x) − Y(I) = 0 nearest to X(I)
      Let D_T(I) = |T − X(I)|
    Else
      Let D_T(I) = 0.0
    Endif
  Enddo
  Let K be the number of points (out of N) with D_T(·) > 1.5
  If K ≥ 5 then flag cell
Endif
```

NOTES:

- The parameters for the cubic fit calculated in the above scheme are not the same as the ones written to the TSLP Banks — those are calculated by a single linear then cubic fit on the "good" points.
- Any injected-time points which fall outside the 59 to 350 ns range will appear in the Diagnostic Output in the `.log` file as having 0 residual.

Board ID Tests

The new Time Slope Title Bank format which will be introduced along with this code (see the following section for a description of the new titles format) includes board ID numbers at the end of each Bank. The validation code reads in the board ID numbers from the TSLP Bank from the most recent ECA Time Slope run and compares them with the board ID numbers from the current run's DQID bank. Both motherboard (MB) and daughterboard (DB) ID numbers are checked and a flag is set if any of the ID's differ. As well, the "Fit Parameter Difference Tests" are not performed for the affected cells if any of the board ID's differ.

Channel Sequencer Test

For each channel, the DQCH bank is checked to see if the channel's sequencer was enabled. If it was not, a flag is set and the validation code does not perform any of the other tests mentioned above since no events would have been returned from any cell in that channel.

IMPORTANT NOTE

If a cell returns no events, either through malfunction or because the sequencer was disabled for the channel in which that cell resides, the median digitized TAC values for that cell will be set to -9999. As stated above, if a cell returns no events or its channel sequencer is disabled then none of the other cell tests will be performed. In particular, the 2nd status word containing the point-by-point “suspicious” flags will not have any flags set, and thus it is very important to test the flags corresponding to the “Cell returned no events” and the “Channel sequencer disabled” conditions before deciding that a cell’s median digitized TAC values are acceptable based on this status word. Even better would be to additionally check that each median digitized TAC value for each injected-time point is greater than zero, regardless of the bit mask for the cell.

3.4.3 New Time Slope Title Bank Format with Validation

Changes:

- Reordering of write-out sequence of data.
- Addition of two new cell status words per cell.
- Addition of board ID numbers.
- Addition of Validation failure totals.
- Addition of 16 spare words.

To reflect the major format changes to the TSLP Banks, Bank Numbers will run from 101 to 138, whereas the previous TSLP Banks were numbered 1 to 38. Note that each of the Bank Numbers still corresponds to a particular half-crate — eg. bank numbers 107 and 108 correspond to the first and second halves of crate 3 respectively.

Data Words
Data Types: b (bit), i, f, d (double), h (hollerith 4 char)

Word	Type	Description
Preamble:		
0		
+1	i	Number of injected-time points (NT)
+2	i	Number of words per cell (NW)
+3	i	1st injected time (ns*100)
.		
.		
+2+NT	i	NT-th injected time (ns*100)

Time Slope Data:

2 + NT + 512*NW*jcard + 16*NW*jchannel + NW*(jcell-1)

+1	i	(Packed) Bit mask for Points Used in Fit Calculation
+2	i	(Packed) Bit mask for “Suspicious” Points
+3	i	(Not Packed) Bit mask - Validation Test Status Word
+4	i	1st Packed TSlope data word for cell
+5	i	2nd Packed TSlope data word for cell
.		
.		
+NW	i	(NW-3)-th Packed TSlope data word for cell

Card ID’s:

21 + 4096*NW + 5*jcard

+1	h	Mother Board ID for card
+2	h	Daughter Board ID for DB 0 of card
+3	h	Daughter Board ID for DB 1 of card
+4	h	Daughter Board ID for DB 2 of card
+5	h	Daughter Board ID for DB 3 of card

HALF-CRATE Stats:

21 + 4096*NW +40

+1	i	# of channels w/ sequencer disabled
+2	i	# of channels which failed MB ID check
+3	i	# of channels which failed DB ID check
+4	i	# of channels which had 0 events
+5	i	# of channels which failed width test
+6	i	# of channels which failed #pts in fit test
+7	i	# of channels which failed #events test
+8	i	# of channels which failed #suspicious test
+9	i	# of channels which failed TAC Jump test
+10	i	# of channels with bad fit parameter (cubic term)
+11	i	# of channels with bad fit parameter (cubic term,range)
+12	i	# of channels with bad fit parameter (cubic term,diff)
+13	i	# of channels with bad fit parameter (quad. term)
+14	i	# of channels with bad fit parameter (quad. term,range)
+15	i	# of channels with bad fit parameter (quad. term,diff)
+16	i	# of channels with bad fit parameter (linear term)
+17	i	# of channels with bad fit parameter (linear term,range)
+18	i	# of channels with bad fit parameter (linear term,diff)
+19	i	total # of bad channels

There then follow 16 spare integer words.

Above, jcard=0,..,7; jchannel=0,..,31; jcell=1,..,16
(if in first half of crate, jcard = card#; if in second half, jcard = card#-8)

PACKED DATA WORDS: The packed TSlope data words for each cell consist of NT+4 12-bit integer words packed into NW-3 32-bit words. Use the *blow* routine to unpack them. Once unpacked, the first NT integers are the digitized TAC values corresponding to the NT injected times. The last 4 integers are the cubic fit parameters, scaled to fit in 12-bit words. To unscale them, do:

$C(1) = \widetilde{C}(1)/2.0$
$C(2) = \widetilde{C}(2)/100.0$
$C(3) = \widetilde{C}(3)/10^5$
$C(4) = \widetilde{C}(4)/10^7$

where $\widetilde{C}(1), ..., \widetilde{C}(4)$ are the values read in from the TSLP Bank, $\widetilde{C}(1)$ being the intercept.

3.4.4 Time Slopes Diagnostic Output to SNOMAN Log File

As with the pedestals, a large amount of new diagnostic information is dumped to the SNOMAN log file by the validation code. When generated by SAS, these output files will be saved on **surf** in /bigdisk/sas/eca/log with the name format described in Section 3.2.

First, the integrity of the entire ECA run is summarized in a 32-bit mask and is written to the `.log` file as an integer. Use your favorite integer-to-binary conversion routine to expand this code number. Table 3.5 contains a description of this bit mask; for a description of the tests which determine whether or not these bits are set, refer to Chapter 4.

If any of the bits are set then the integer version of this overall bit mask will be non-zero; in this case the phrase "ECA_PROBLEM" will be written to the `.log` file followed by the actual integer value of the mask. Thus, when SAS finishes processing an ECA run, it should *grep* the `.log` file for the phrase "ECA_PROBLEM". If this phrase is found, SAS should email the detector manager or an ECA expert so that the run can be examined more closely before the titles file is moved into the ECA titles directory and/or database.

If any of bits 27 through 29 are set, some of the ECA Validation summaries, test results, and histograms should not be trusted. The Pedestal values in the titles file are not affected by these missing banks, but cell status bit-masks will be affected and thus it is advised that the ECA run be analyzed again offline by an ECA or SAS expert, ensuring that the required banks are properly loaded. If bit 30 is set, then there was a serious error either during the actual ECA run or during the analysis by SAS/SNOMAN. In this case action must be taken as the Pedestal titles file will not be complete.

Following this, tables containing total numbers of failures for the various tests are written, with a separate table for the channel and cell summaries, respectively.

Next, average cubic fit parameters and the average differences from the previous TSlopes run for three of the four terms (cubic, quadratic and linear) are written out by the half-crate.

For the difference values, both $(\Sigma x)/n$ and $(\Sigma|x|)/n$ are provided.

After this, total numbers of failures for several of the tests, accumulated for each board, are written. Interspersed with this will be single-line messages for any card that had at least one TAC Jump Test failure. These messages will appear directly after the summary for the board in which such a channel resides.

Next, a detailed description of all test failures is written out for each channel which contains at least one cell which failed at least one test.

Finally, if the corresponding flag is set in `eca_control.dat` then all 10 time slope events at each injected-time step for all cells will be dumped to the `.log`. The inclusion of this data greatly increases the size of the `.log` file and so this option should normally be left off. Another flag in `.log` allows one to specify whether or not the 10 time slope events should be sorted by value, lowest to highest, for each injected-time step; if this flag is set to "unsorted" then the events for each injected-time step are dumped in the order they were read back and an asterisk (*) is placed next to the cell which was read from first at each injected-time step in each channel.

All of the above mentioned tables and lists in the SNOMAN log file have been intentionally designed to facilitate easy *grep*'ing for desired diagnostic messages. For more information on the format of this diagnostic information it is best to refer directly to one of these log files, whenever they become available.

3.4.5 Time Slope Diagnostic Histograms

See Table 3.6 for a list of the histograms which can be found in the time slope `hbk` file.

3.5 Miscellaneous Final Notes

3.5.1 Titles File Output

The write-out sequence for ECA data has been altered for both Pedestal and Time Slope data. Previously the SNOMAN code for writing out the data for each half-crate was (in simplified psuedo-code):

```
do icell=1,ncells
  do ichannel=1,nchannels
    do icard=1,ncards/2
      Write out ECA data for current cell to titles file.
```

```
    enddo
  enddo
enddo
```

The new code reverses the order of the do-loops:

```
do icard=1,ncards/2
  do ichannel=1,nchannels
    do icell=1,ncells
      Write out ECA data for current cell to titles file.
    enddo
  enddo
enddo
```

However, by making the sacrifice of duplicating some code, the format for the QSLP titles files will remain unchanged; that is, the write-out sequence for the ECA QSLP data will not change along with the PDST and TSLP data.

3.5.2 Notes for SAS

- As it is unlikely that that DQxx banks will have made it into the database by the time SAS analyses an ECA run, it is recommended that SAS add an explicit "titles" command to the ECA `.cmd` file to load the DQxx banks. As well, the likelihood of the most recent ECA data making it to the database by the time the next ECA run is analyzed might be in question as well; it may be worth considering adding an explicit "titles" command to load previous ECA data as well.
- As mentioned in the "New Diagnostic Output to SNOMAN Log File" sections, SAS should *grep* the `.log` file looking for "ECA_PROBLEM". If found, SAS should email the detector manager or a designated ECA expert before the titles files are moved to `.../snoman/titles/eca`.
- SNOMAN `.log` files from ECA analysis contain important diagnostic information and these files should be saved along with the `.dat` and `.hbk` files. Refer to the "New Naming Convention for Output Files" section.

1101	QHS Pedestals	5303	# bad QLX-width channels (per 1/2 Crate)
1102	QHS Pedestals (avg. over 1/2 crates)	5304	# bad QLX-diff channels (per 1/2 Crate)
1201	QHS Widths	5401	# bad TAC channels (per 1/2 Crate)
1202	QHS Widths (avg. over 1/2 crates)	5402	# bad TAC-median channels (per 1/2 Crate)
1301	QHS Diffs	5403	# bad TAC-width channels (per 1/2 Crate)
1302	QHS Abs Diffs (avg. over 1/2 crates)	5404	# bad TAC-diff channels (per 1/2 Crate)
1303	QHS Diffs (avg. over 1/2 crates)	5501	# bad QHS-or-TAC channels (per 1/2 Crate)
2101	QHL Pedestals	5502	# bad QHS-or-TAC-median channels (per 1/2 Crate)
2102	QHL Pedestals (avg. over 1/2 crates)	5503	# bad QHS-or-TAC-width channels (per 1/2 Crate)
2201	QHL Widths	5504	# bad QHS-or-TAC-diff channels (per 1/2 Crate)
2202	QHL Widths (avg. over 1/2 crates)	6000	TOTAL # bad channels (per 1/2 Crate)
2301	QHL Diffs	8101	CMOS Droop (QHS,med-first)
2302	QHL Abs Diffs (avg. over 1/2 crates)	8102	CMOS Droop (QHS,med-second)
2303	QHL Diffs (avg. over 1/2 crates)	8201	CMOS Droop (QHL,med-first)
3101	QLX Pedestals	8202	CMOS Droop (QHL,med-second)
3102	QLX Pedestals (avg. over 1/2 crates)	8301	CMOS Droop (QLX,med-first)
3201	QLX Widths	8302	CMOS Droop (QLX,med-second)
3202	QLX Widths (avg. over 1/2 crates)	8401	CMOS Droop (TAC,med-first)
3301	QLX Diffs	8402	CMOS Droop (TAC,med-second)
3302	QLX Abs Diffs (avg. over 1/2 crates)	9101	CMOS Droop (QHS,med-first,wo dsbld ch)
3303	QLX Diffs (avg. over 1/2 crates)	9102	CMOS Droop (QHS,med-second,wo dsbld ch)
4101	TAC Pedestals	9201	CMOS Droop (QHL,med-first,wo dsbld ch)
4102	TAC Pedestals (avg. over 1/2 crates)	9202	CMOS Droop (QHL,med-second,wo dsbld ch)
4201	TAC Widths	9301	CMOS Droop (QLX,med-first,wo dsbld ch)
4202	TAC Widths (avg. over 1/2 crates)	9302	CMOS Droop (QLX,med-second,wo dsbld ch)
4301	TAC Diffs	9401	CMOS Droop (TAC,med-first,wo dsbld ch)
4302	TAC Abs Diffs (avg. over 1/2 crates)	9402	CMOS Droop (TAC,med-second,wo dsbld ch)
4303	TAC Diffs (avg. over 1/2 crates)	9901	CMOS Droop (TAC,med-firsts)
5001	#channels w/ Seq disabled (per 1/2 Crate)	9902	CMOS Droop (TAC,med-seconds)
5002	#channels MBoard ID failed (per 1/2 Crate)	9903	CMOS Droop (TAC,med-thirds)
5003	#channels DBoard ID failed (per 1/2 Crate)	9904	CMOS Droop (TAC,med-fourths)
5004	#channels w/ lt160 events (per 1/2 Crate)	9905	CMOS Droop (TAC,med-fifths)
5005	#channels w/ 0 events (per 1/2 Crate)	9906	CMOS Droop (TAC,med-sixths)
5101	# bad QHS channels (per 1/2 Crate)	9907	CMOS Droop (TAC,med-sevenths)
5102	# bad QHS-median channels (per 1/2 Crate)	9908	CMOS Droop (TAC,med-eighths)
5103	# bad QHS-width channels (per 1/2 Crate)	9909	CMOS Droop (TAC,med-ninths)
5104	# bad QHS-diff channels (per 1/2 Crate)	9910	CMOS Droop (TAC,med-tenths)
5201	# bad QHL channels (per 1/2 Crate)	9911	CMOS Droop (TAC,med-elevenths)
5202	# bad QHL-median channels (per 1/2 Crate)	9912	CMOS Droop (TAC,med-twelfths)
5203	# bad QHL-width channels (per 1/2 Crate)	9913	CMOS Droop (TAC,med-thirteens)
5204	# bad QHL-diff channels (per 1/2 Crate)	9914	CMOS Droop (TAC,med-fourteenths)
5301	# bad QLX channels (per 1/2 Crate)	9915	CMOS Droop (TAC,med-fifteenths)
5302	# bad QLX-median channels (per 1/2 Crate)	9916	CMOS Droop (TAC,med-sixteenths)

Table 3.3: Histogram List (Pedestals)

0	overall cell flag (OR of bits 1 through 12)
1	cell returned no events at any injected time step
2	too few points used in fit
3	too many suspicious points
4	x-cubed coeff. failed range test
5	x-cubed coeff. failed diff test
6	x-squared coeff. failed range test
7	x-squared coeff. failed diff test
8	slope coeff. failed range test
9	slope coeff failed diff test
10	too many points failed width test
11	too many points failed #events test
12	Frati TAC jump flag
13	overall channel flag (OR of bits 14 through 27)
14	channel returned no events
15	some cell had too few points used in fit
16	some cell had too many suspicious points
17	overall x-cubed coeff. checks
18	overall x-squared coeff. checks
19	overall slope coeff. checks
20	overall fit coeff. range checks
21	overall fit coeff. diff checks
22	overall width check
23	overall #events check
24	Motherboard ID check
25	Daughterboard ID check
26	channel sequencer check
27	some cell failed the Frati TAC jump test
28–31	spare

Table 3.4: Cell Bit Mask (Time Slopes)

0	Too many channels failed board ID test
1	Too many channels with 0 events
2	Too many channels failed #events test
3	Too many channels failed width test
4	Too many channels failed #points used in fit test
5	Too many channels failed #suspicious points test
6	Too many channels failed Frati TAC Jump test
7	Too many channels failed cubic coeff test
8	quadratic
9	linear
10	Bad 1/2-crate average coeff - cubic term
11	quadratic
12	linear
13	Bad 1/2-crate average coeff diff - cubic term
14	quadratic
15	linear
16	Too many flagged channels
17–25	spare
26	UC (or old DCR_ACTIVITY) bit was set
27	Missing DQID Bank - could not check board IDs
28	Missing DQCH Bank - could not check chan seqs
29	Missing TSLP Bank - could not check diffs or IDs
30	Did not reach end of last crate
31	spare

Table 3.5: Overall Time Slopes Run Bit Mask

1001	C-Fit, cubic term	1301	C-Fit, intercept term
1002	C-Fit, cubic term (1/2 Crate Avg)	2001	# inj.time steps used in fits
1003	C-Fit, cubic term diffs	2002	# ch failed #Tsteps in fit tests (per 1/2 Crate)
1004	C-Fit, cubic term abs. diffs (1/2 Crate Avg)	2101	Widths
1005	C-Fit, cubic term diffs (1/2 Crate Avg)	2102	#steps per cell with bad width
1006	# ch bad cubic term range (per 1/2 Crate)	2103	# ch failed Width tests (per 1/2 Crate)
1007	# ch bad cubic term diffs (per 1/2 Crate)	2201	TStep: #Events returned
1101	C-Fit, quadratic term	2202	#steps per cell with bad #events
1102	C-Fit, quadratic term (1/2 Crate Avg)	2203	# ch failed #Ev tests (per 1/2 Crate)
1103	C-Fit, quadratic term diffs	2204	# ch which returned 0 ev (per 1/2 Crate)
1104	C-Fit, quadr term abs. diffs (1/2 Crate Avg)	2301	# ch failed sequencer check (per 1/2 Crate)
1105	C-Fit, quadr term diffs (1/2 Crate Avg)	2401	# ch failed MB ID check (per 1/2 Crate)
1106	# ch bad quadratic term ranges (per 1/2 Crate)	2501	# ch failed DB ID check (per 1/2 Crate)
1107	# ch bad quadratic term diffs (per 1/2 Crate)	2601	Frati Time Residuals
1201	Cubic Fit, linear term	2602	Frati ChiSq
1202	C-Fit, linear term (1/2 Crate Avg)	2603	#steps per cell with bad residual
1203	Cubic Fit, linear term diffs	2604	# ch failed Frati TAC jump test (per 1/2 Crate)
1204	C-Fit, linear term abs. diffs (1/2 Crate Avg)	2701	# suspicious (bad width OR bad #ev) steps per cell
1205	C-Fit, linear term diffs (1/2 Crate Avg)	2702	#ch failed #Suspicious test
1206	# ch bad linear term range (per 1/2 Crate)	9000	Total # flagged ch (per 1/2 Crate)
1207	# ch bad linear term diffs (per 1/2 Crate)		

Table 3.6: Histogram List (Time Slopes)

Chapter 4

ECA Validation Cut Selection

The choice of cut values for the various validation tests are described here. The overall reasoning in defining sensible cut values is two fold. First, we want to base these cuts on reasonable physics grounds, e.g. what is an acceptable range for pedestals, TAC slopes, etc, given our overall physics analysis requirements and tolerable systematics. The second, what performance can we expect from the electronics, and what are the normal limits we observe from the distributions themselves for properly working channels. We want to ensure that we are flagging only problematic channels, and not some normal variations. The two, sometimes conflicting, points have to be balanced.

4.1 ECA Pedestals

The pedestal validation tests discussed in Section 3.3 have various ranges, limits, etc. These are discussed below, as well as overall run checks.

4.1.1 Median Test

The choice of the charge and time median test range is dictated essentially by the variation that we expect for normal behaving electronic channels. For a normally operating channel, we expect the charge pedestal to be around 600 ADC counts, but can be as low as 300 and high as 900 counts. Figure 4.1 shows the pedestals for a typical ECA run, where clearly the charge pedestals are strongly peaked and fall mostly in the range mentioned above. Some of the channels above/below this range were investigated and were indeed found to have problems and require attention. When possible, they will be repaired. The acceptable range is large, but that is more a function of channel to channel response/setup variations and does not necessarily indicate a problem with the electronics.

The large tail at lower ADC pedestals exhibited by $Q_{hl}$is due to bad VBAL setup voltages. As long as they remain constant, it is not a problem as it is corrected by the calibration. However, very low values outside of the acceptable range should eventually be repaired.

The TAC median pedestal is dependent on the value of the Ped-GT delay that is programmed by the DAQ. Normal ECA pedestal runs are done with a 150 nsec delay. For normal TAC channels, this represents 1850 ADC counts. The range that covers the majority of the channels, see Figure 4.1, is from 1350 to 2400 ADC counts. This is more a function of the TAC slope, and better flagged by TAC slope validation checks. However, if a TAC value is widely out of range, possibly indicating a severe problem with a channel, it is worth flagging and providing the analysis with the added information.

4.1.2 Width Test

The choice of width cuts should be guided by physics requirements. How much fluctuation in a channel charge measurement are we willing to tolerate? For the high gain short integrate charge (Q_{hs}) channels, a 1/4 PE is appropriate (about 8 counts), since this feeds directly into the time correction for discriminator walk. For the long integrate charge, both high and low gain, the requirements in not so stringent. For Q_{hl}, about 1/2 PE (16 counts) is sensible, while for Q_{lx}, 16 counts corresponds to 6 PE.

From an electronics functionality perspective, the above limits are well within the expected capability of the circuitry, where the nominal under ideal conditions is less than a couple of counts.

For the TAC, a sensible limit is about 1.5 nsec width fluctuation that can be tolerated. For a normal TAC calibration, this is about 13 ADC counts. Figure 4.2 shows a plot of the pedestal widths for a typical ECA run.

Pedestal fluctuations within a cell can be caused by many sources. One possibility is random coincidences with real light, which would only happen for only a single cell. However, the use of the mean to calculate the pedestal, as well as the low rate of coincidences (see Chapter 1), helps to minimize this effect. Another possibility, is a bad integrator balance or electronic noise (e.g. pedestal pulse pickup in channel 17) which would increase pedestal fluctuations. This is why the long integrate channels have larger widths than the short integrate charge. There are also a significant fraction of flagged cells that appear to be just random variations, since they don't repeatedly show up, and there is nothing else flagged for the channel.

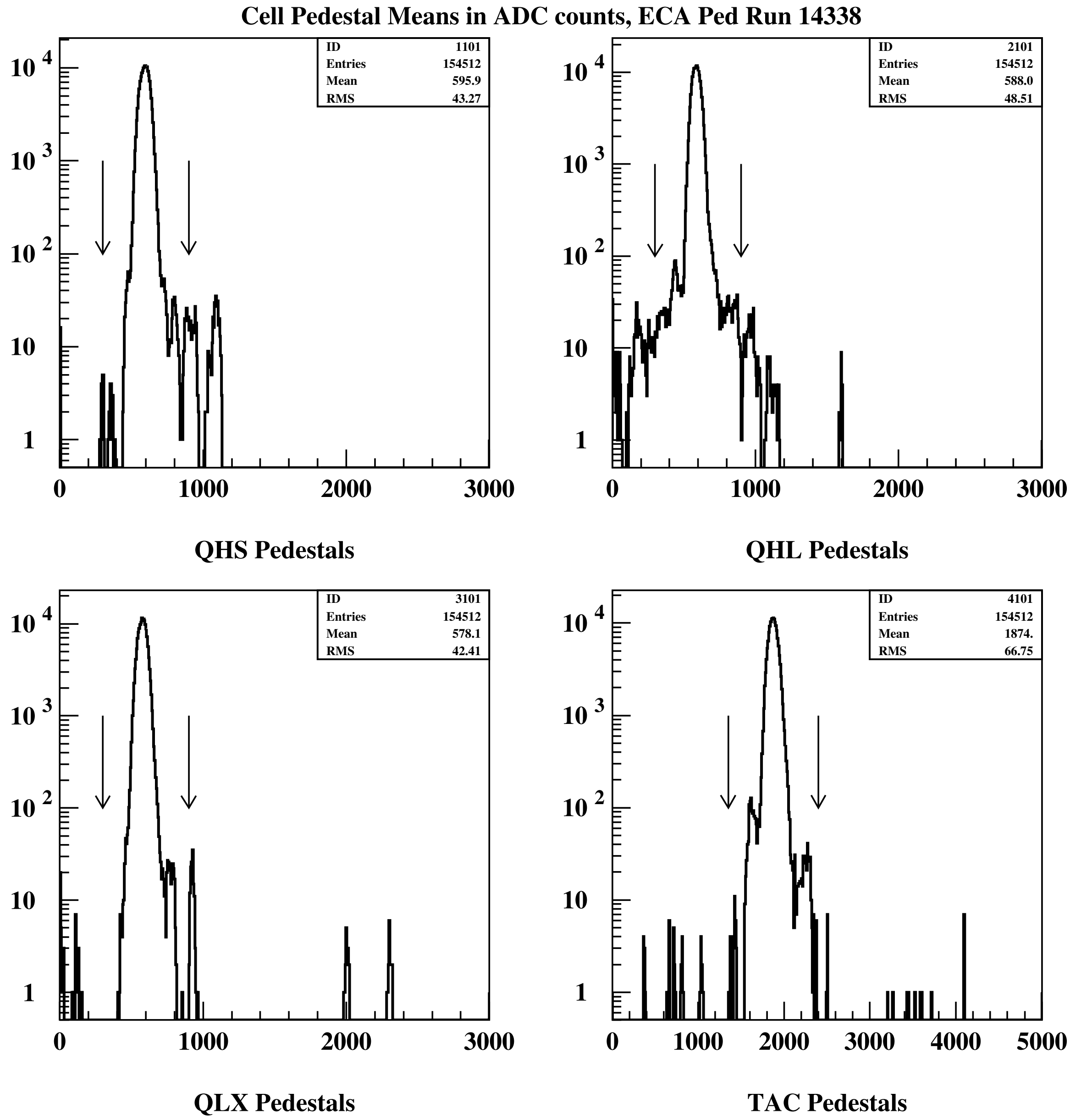


Figure 4.1: Cell pedestal means for the three charge and one TAC channel.

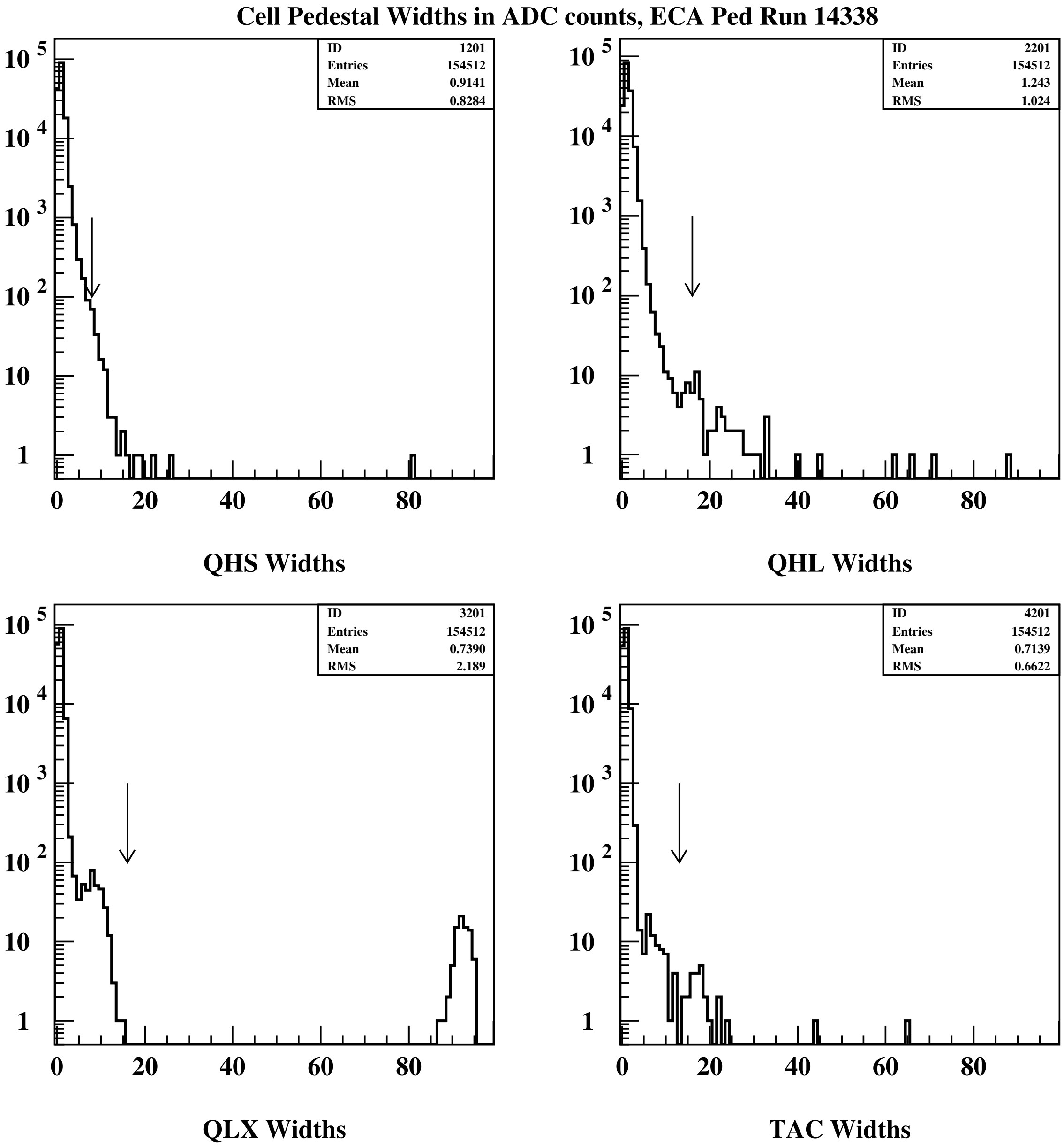


Figure 4.2: Cell pedestal widths for the three charge and one TAC channel.

4.1.3 Difference Test

An important monitor of the stability of the ECA constants is to compare the new constants with the ones taken last. This allows a check of the ECA stability from run to run, and gives the analyzer some confidence that the electronics are functioning well, and are not drifting from their nominal response.

The limits used for the difference tests are the same as those used in the width test above, i.e. 8 ADC counts (1/4 PE) for Q_{hs} 16 ADC counts (1/2 and 6 PE) for Q_{hl} and Q_{lx} and 13 ADC counts (1.5 nsec) for the TAC . These are sensible limits, as can be observed in Figure 4.3.

The low gain charge spread is due to channel 17 pedestal pulse pickup and bad integrator balance, which tends to increase the amount and level of fluctuations. For Q_{hs}, a large fraction of the flagged cells appear just to be random variations since it disappears on the next ECA, and the rest of the channel has no other flagged problems. As will be discussed later, the TAC variations are usually due to Wittich jump ADC's, or a malfunctioning TAC, which makes this test important in identifying bad channels/slots so they can be repaired promptly.

4.1.4 Number of Events Test

Channels that have no ECA events are potentially dangerous, as they can be a source of blind flashers since the electronic circuitry is not working properly, but the PMT still has HV applied. It might be as simple as the pedestal enable or pedestal pulse not getting to the channel, in which this channel will not be a blind flasher since during normal neutrino runs it will still have valid raw PMT data. This sort of question can be resolved by looking at the most recent PCA/occupancy data, which is done on a monthly basis. When channels with no ECA data are flagged, they are investigated as soon as possible, and if found to be dead, the high voltage resistor removed. To date, six channels have been found with this condition.

Channels that have less than the expected number of events, but more than zero, are not as problematic. They are still flagged, and investigated, but do not need to be turned off immediately. If the channel is found to be broken, and a source of potential blind flashers, then the channel is disabled.

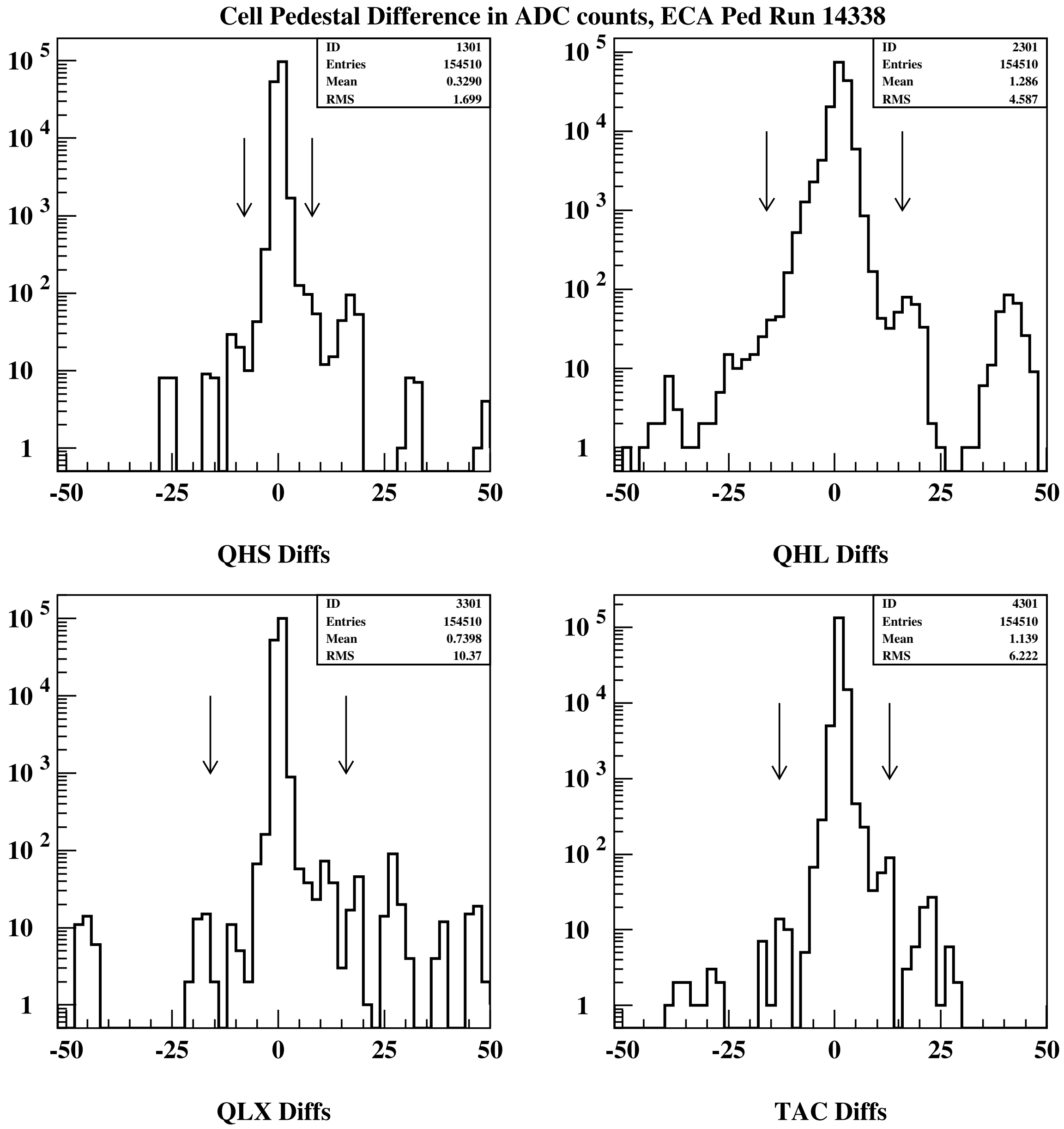


Figure 4.3: Cell pedestal diffs for the three charge and one TAC channel.

4.1.5 Board ID Test

Since the Mother and Daughter board ID's of the current valid ECA calibration are stored, when a channel is calibrated, a check can be done to determine if the ID's match. If not, the channel is not calibrated and a status flag bit is set. It is possible that after a ECA calibration, a board has to be swapped out because of problems. There might be a period of time (hours to days), before the new board can have ECA calibrations done.

The procedures these days is to do a ECA within a few hours soon after a new board is activated. However, in the past, especially the early months of 1999 production running, we weren't as diligent, and periods of days might elapse before a calibration is done.

This new scheme will correct for these circumstances and ensure we are not applying the wrong calibrations.

4.2 Sample Pedestal Diagnostic Summary Output

Below is a summary of failures for a typical ECA pedestal run 14338. The detailed crate/channel/cell diagnostic breakdown, with stats, is much longer, and is not shown here (see on **surf** in /bigdisk/sas/eca/log). For a historical breakdown for all ECA runs to present, see Chapter 6.

```
--------------------------------------
CHANNEL SUMMARY ECA PEDESTAL RUN 14338
--------------------------------------
# of channels with
  sequencer disabled  =  71
# of channels which
  failed MB ID check  =  0
# of channels which
  failed DB ID check  =  0
# of channels which
  had <160 events     =  49
# of channels which
  had 0 events        =  0
---------------------------
# of bad QHS channels =  84
----> bad medians     =  34
----> bad widths      =  43
----> bad diffs       =  33
```

```
# of bad QHL channels =  109
----> bad medians     =  53
----> bad widths      =  12
----> bad diffs       =  54
# of bad QLX channels =  44
----> bad medians     =  14
----> bad widths      =  8
----> bad diffs       =  40
# of bad TAC channels =  27
----> bad medians     =  7
----> bad widths      =  12
----> bad diffs       =  21
# of bad QHS OR TAC
  channels            =  109
----> bad medians     =  41
----> bad widths      =  55
----> bad diffs       =  54
---------------------------
Total # bad channels  =  323
---------------------------
---------------------------
CELL SUMMARY
---------------------------
# with no events =  0
# with lt 10 events =   66
---------------------------
# of bad QHS cells    =  750
----> bad medians     =  466
----> bad widths      =  76
----> bad diffs       =  381
# of bad QHL cells    =  1361
----> bad medians     =  774
----> bad widths      =  49
----> bad diffs       =  579
# of bad QLX cells    =  683
----> bad medians     =  208
----> bad widths      =  80
----> bad diffs       =  606
# of bad TAC cells    =  277
----> bad medians     =  64
----> bad widths      =  25
----> bad diffs       =  240
# of bad QHS OR TAC
  cells               =  1015
----> bad medians     =  530
```

```
----> bad widths      =  99
----> bad diffs       =  619
---------------------------
Total # bad cells     =  2432
---------------------------
```

4.3 Pedestal Cut Summary

```
#.     Pedestal Validation Parameters (ADC counts)
 300.    #. QHS Tolerances: lower median
 900.    #.                 upper median
 8       #.                 width
 8       #.                 abs. diff
 300.    #. QHL Tolerances: lower median
 900.    #.                 upper median
 16      #.                 width
 16      #.                 abs. diff
 300.    #. QLX Tolerances: lower median
 900.    #.                 upper median
 16      #.                 width
 16      #.                 abs. diff
 1350.   #. TAC Tolerances: lower median
 2400.   #.                 upper median
 13      #.                 width
 13      #.                 abs. diff
```

4.4 ECA Pedestal Run Checks

After a ECA pedestal run is complete, SAS will then execute the SNOMAN job that will generate the constants, validation flags, and various statistics. A set of checks are then done to these results. If there are enough failures, the run is flagged, the SIC alerted (via email), and the constants will not proceed to the database till they have been checked manually, and cleared for use by an expert.

The set of ECA pedestal run tests are displayed in Table 4.1. The ECA SU keeps track of the 1/2 crate median, width, and difference for each of the three charge and single TAC.

Test Number	Test Type	Min Test Value	Max Test Value
1	no. channels with zero events	NA	1
2	no. channels with 160 events	NA	32
3	no. failed board ID check	NA	1
4a	no. flagged Q_{hs} channels	NA	256
4b	no. flagged Q_{hl} channels	NA	256
4c	no. flagged Q_{lx} channels	NA	256
4d	no. flagged TAC channels	NA	256
5a	1/2 crate Q_{hs} median	530	630
5b	1/2 crate Q_{hl} median	530	630
5c	1/2 crate Q_{lx} median	530	630
5d	1/2 crate TAC median	1825	1925
6a	1/2 crate Q_{hs} width	NA	3
6b	1/2 crate Q_{hl} width	NA	3
6c	1/2 crate Q_{lx} width	NA	3
6d	1/2 crate TAC width	NA	3
7a	1/2 crate Q_{hs} abs. difference	NA	3
7b	1/2 crate Q_{hl} abs. difference	NA	3
7c	1/2 crate Q_{lx} abs. difference	NA	3
7d	1/2 crate TAC abs. difference	NA	3
8	Total flagged channels	NA	500

Table 4.1: List of ECA pedestal run tests. The min and max test values are inclusive. Test values 4 thru 7 are in units of ADC counts.

The max value for test 4 is the limit of the number of channels typically flagged by the cell/channel validation tests, any more than this would indicate a substantial change in hardware performance, and required immediate attention. The range for test 5 is the typical spread in 1/2 crate means for all the crates. The maximum test value for tests 6 and 7 will catch situations where many channels start to exhibit problems, e.g. a rack power supply problem, etc. When a single channel has zero events, this is flagged and immediate inspection required since this can indicate a blind flasher. The failure of a few channels not having all 160 events can be tolerated, but if a slots worth of channels fail, it is worth investigation. Failed board ID checks are not considered serious, as we expect boards to be changed out. However, if the flag is set and a board not swapped, it could indicate a serious problem, and requires investigation.

4.5 ECA Time Slopes

Similarly for the pedestals, the TAC slope tests identified in Section 3.4, have various ranges, limits, etc. These are discussed below, as well as overall time slopes run checks. As in the pedestal case, we want to make cut selections based on sensible physics goals and electronics performance limits.

4.5.1 TAC Fit Parameter Tests

A cubic fit is done to the TAC data for each cell to test if the curve is sensible. No cut is done on intercept, as we are not interested in absolute offsets, as this is essentially tested by ECA TAC pedestal. Current limits are, cubic term $\pm 4.0 \mathrm{x} 10^{-5}$, quadratic term ± 0.02, and the linear term from 5.0 to 13.0 (units of ADC counts/nsec for the linear term). Since there is no absolute right or wrong slope value, the choice of limits is best done by inspection of the TAC slope range exhibited by the electronics under normal conditions. Such an example can be seen in Figure 6.3.

4.5.2 TAC Fit Parameter Difference Tests

A difference is done between the cubic fit parameters for the current and previous ECA time slope runs. Current absolute limits were arrived at in a similar fashion as those for the fit parameter tests above, e.g. cubic $5 \mathrm{x} 10^{-5}$, quadratic 0.03, and linear 2.0. See Figure 4.5 for the distributions and limits. The cubic and quadratic terms allowed range are rather loose, since these terms are not as robust as the linear, which is given tighter limits.

4.5.3 TAC Width Test

A check on width is made for each point in a cell. Current limits on the width is 5 counts (about 0.5 nsec), and the cell is flagged if more than 3 (5 when the number of injected points is greater than 25) points fail, as seen in Figure 4.6. The width cut is tightened up since it takes more than one point to fail before the cell is flagged. An effect that works against this is the higher detector rate (1kHz), which results in an expected increase in coincidences with random PMT firing, resulting in a wider TAC width. This can be seen by comparing the pedestal TAC width distribution in Figure 4.2 with that in Figure 4.6. However, by requiring multiple TAC points to fail, we are surpressing failures due to random fluctuations and flagging only systematic width failures.

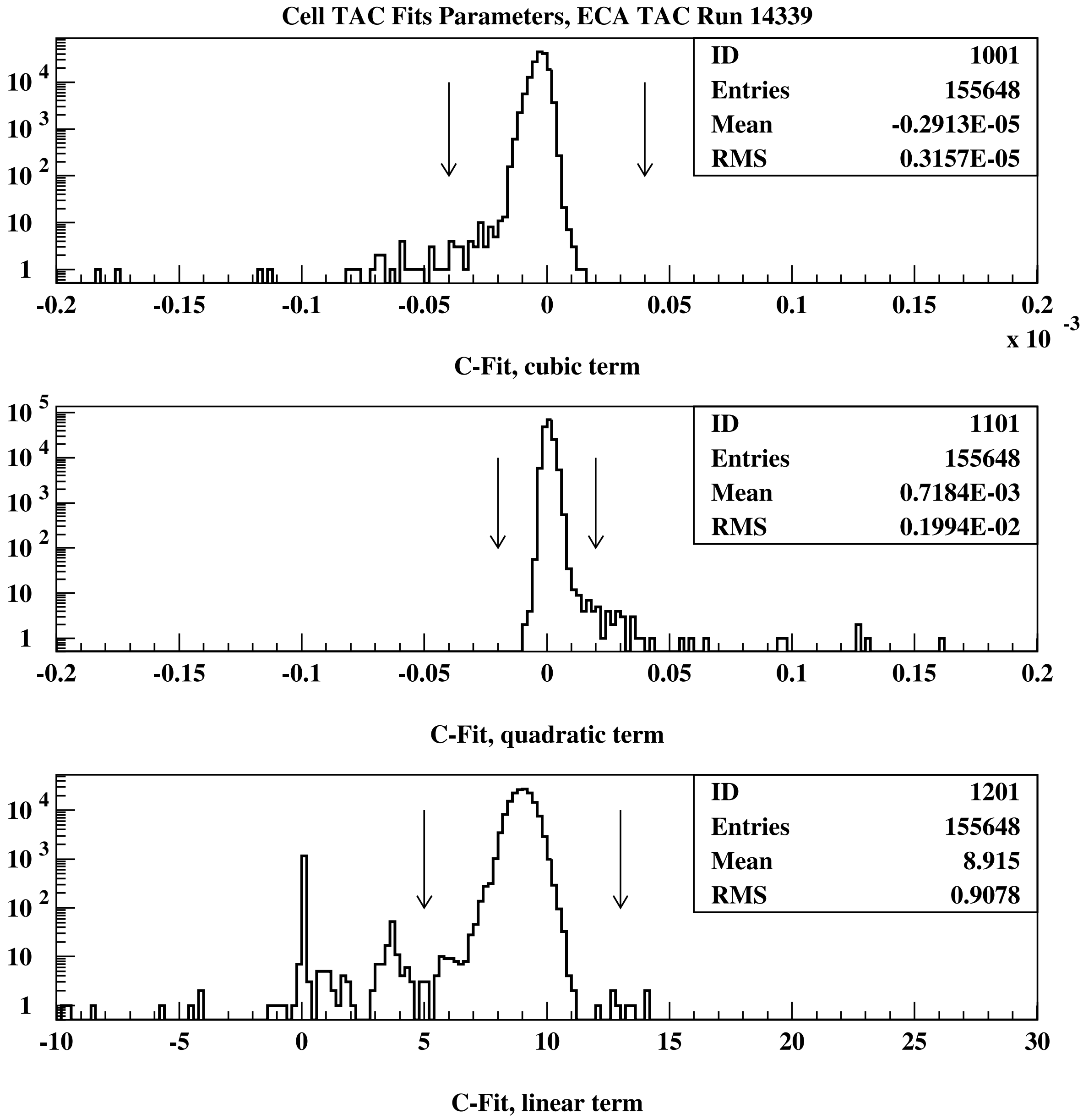


Figure 4.4: Cell TAC fit parameters.

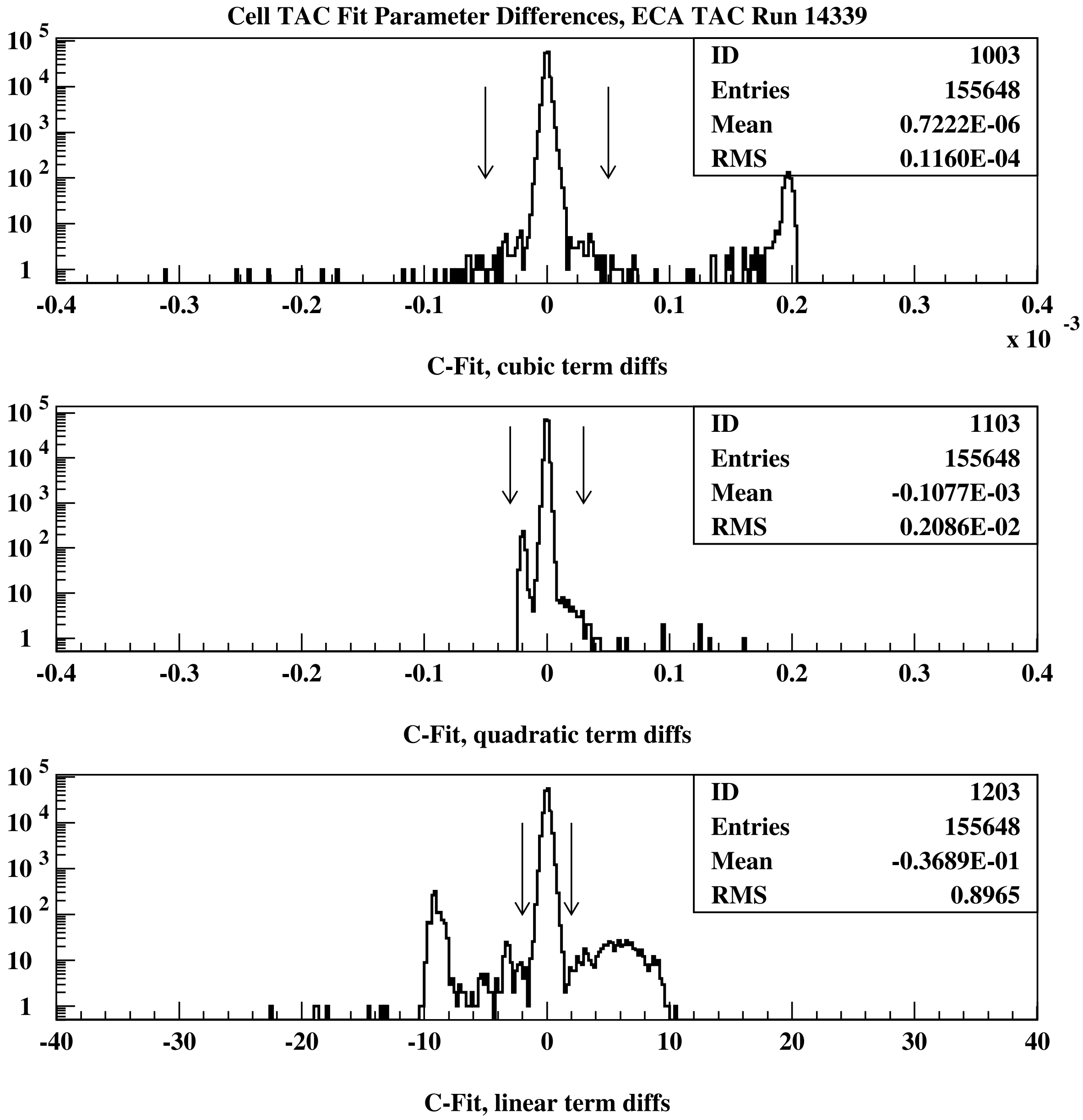


Figure 4.5: Cell TAC fit parameters difference.

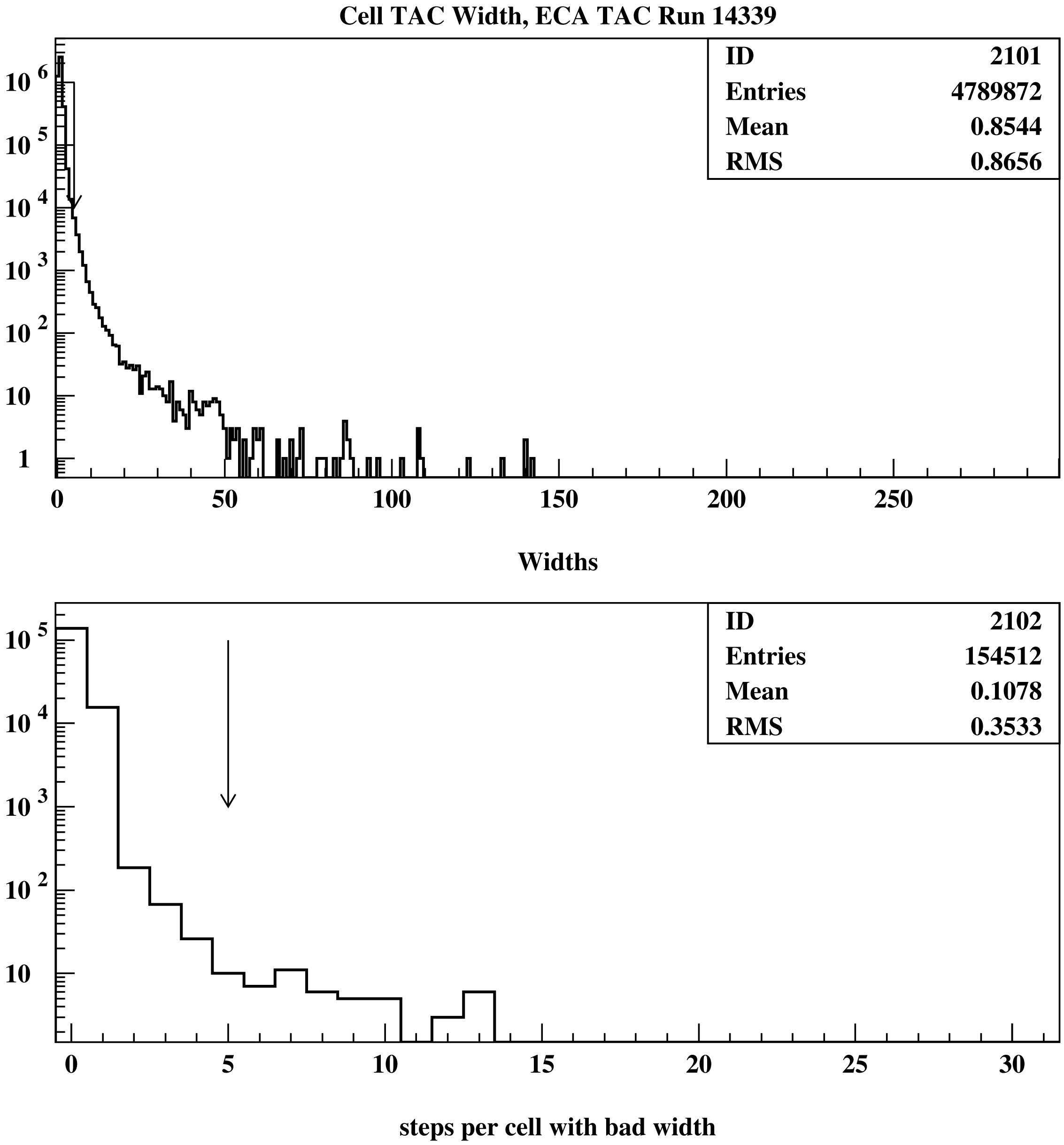


Figure 4.6: Top figure is the TAC width for all injected time points (individual cells). The bottom figure is the number of injected time points (steps) per cell with bad widths.

4.5.4 Point-by-Point Number of Events Returned Test

For each injected time point, the number of events are checked. The current limits are a combination of a minimum of 6 events per step per cell, and 3 (8 when the number of injected points is greater than 25) or more injected time points rejected because of insufficient number of events. The distributions can be viewed in Figure 4.7.

The mean of about three steps per cell with bad number of events is due to the sampling range (for the 31 point case only) being longer than most TAC time windows (GTVALID). This oversampling allows particularly wide channels to still be properly calibrated.

The astute reader will notice that the peak of the distribution in Figure 4.7 is not at 10, as expected, but more like 11. This is due to the the MAC timing imperfections described in Section 1.1.

4.5.5 Number of events used in the Cubic Fit Test

After all bad points have been removed (see Section 3.4.2 "Number of Points Used in Cubic Fit Test"), a check is made flagging all cells that have less than 12 (18 when the number of injected points is greater than 25) remaining points for a cubic fit, see Figure 4.8. The mean of this distribution is 25, which is much less than 31 injected points. These are due mostly to the TAC curl and out of GTVALID range points.

4.5.6 Number of Suspicious Points Test

The cell is flagged if the number of suspicious points (see Section 3.4.2 "Number of Suspicious Points Test") is equal to or greater than 5 (9 when the number of injected points is greater than 25), as shown in Figure 4.8.

4.5.7 Frati TAC Jump Test

Besides the monthly CALDAC checks, which directly tests the ADC's using the onboard calibration DAC, a algorithm has been produced by Bill Frati to flag bad ADC's (see Section 3.4.2). It only works for the TAC channel because of the multi-point sampling. It is estimated, based on comparisons to the CALDAC data, that the technique has a $\sim$75% positive identification of bad ADC, and a $\sim$25% misidentification. Figure 4.9 shows the various distributions and the cuts made by Frati's algorithm, e.g. a time residual cut of $\pm$1.5 nsec, with more than five failed residuals flagging the channel.

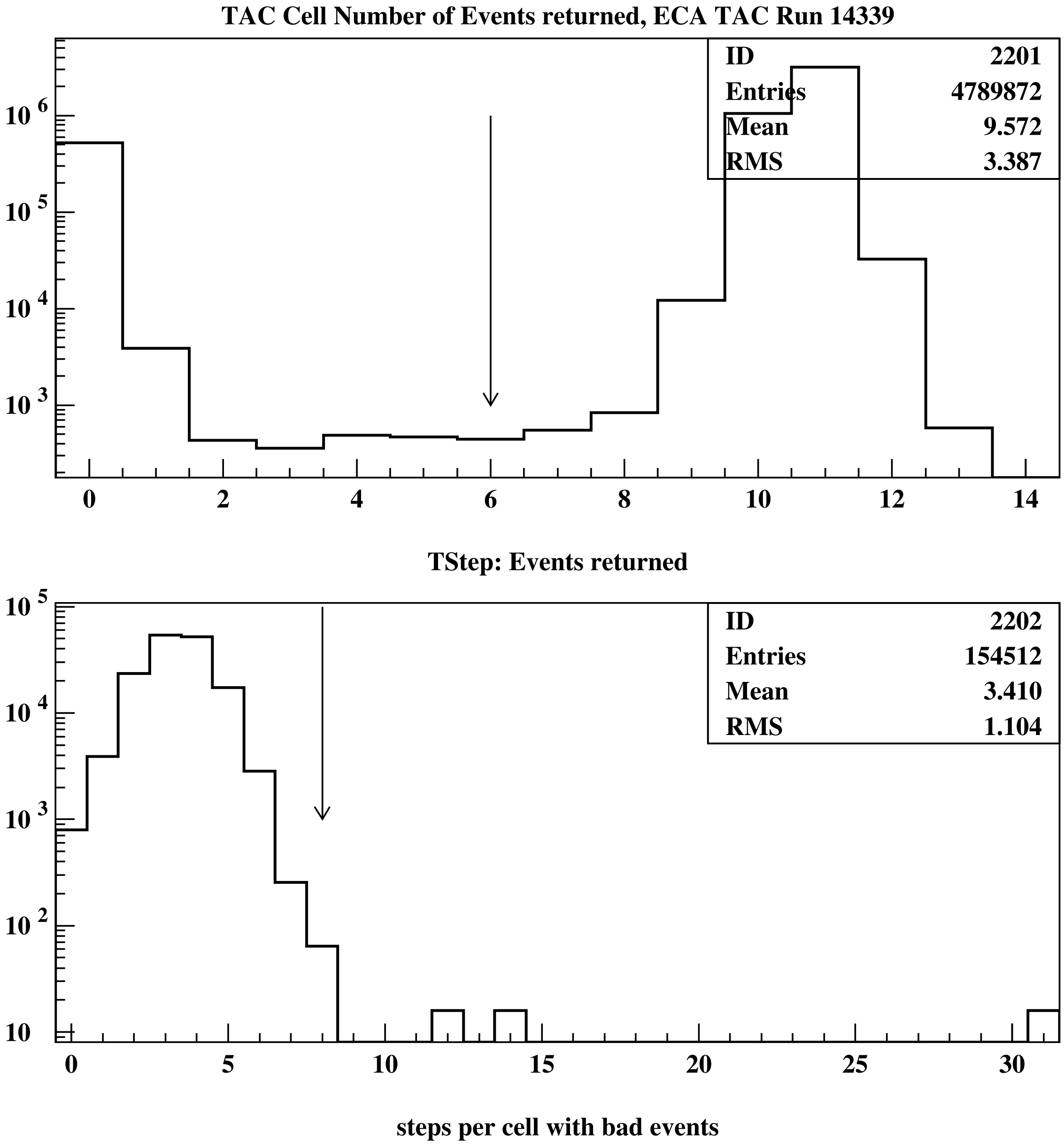


Figure 4.7: Top figure is number of events per step per cell, the bottom is number of injected time points (steps) per cell with bad number number of events.

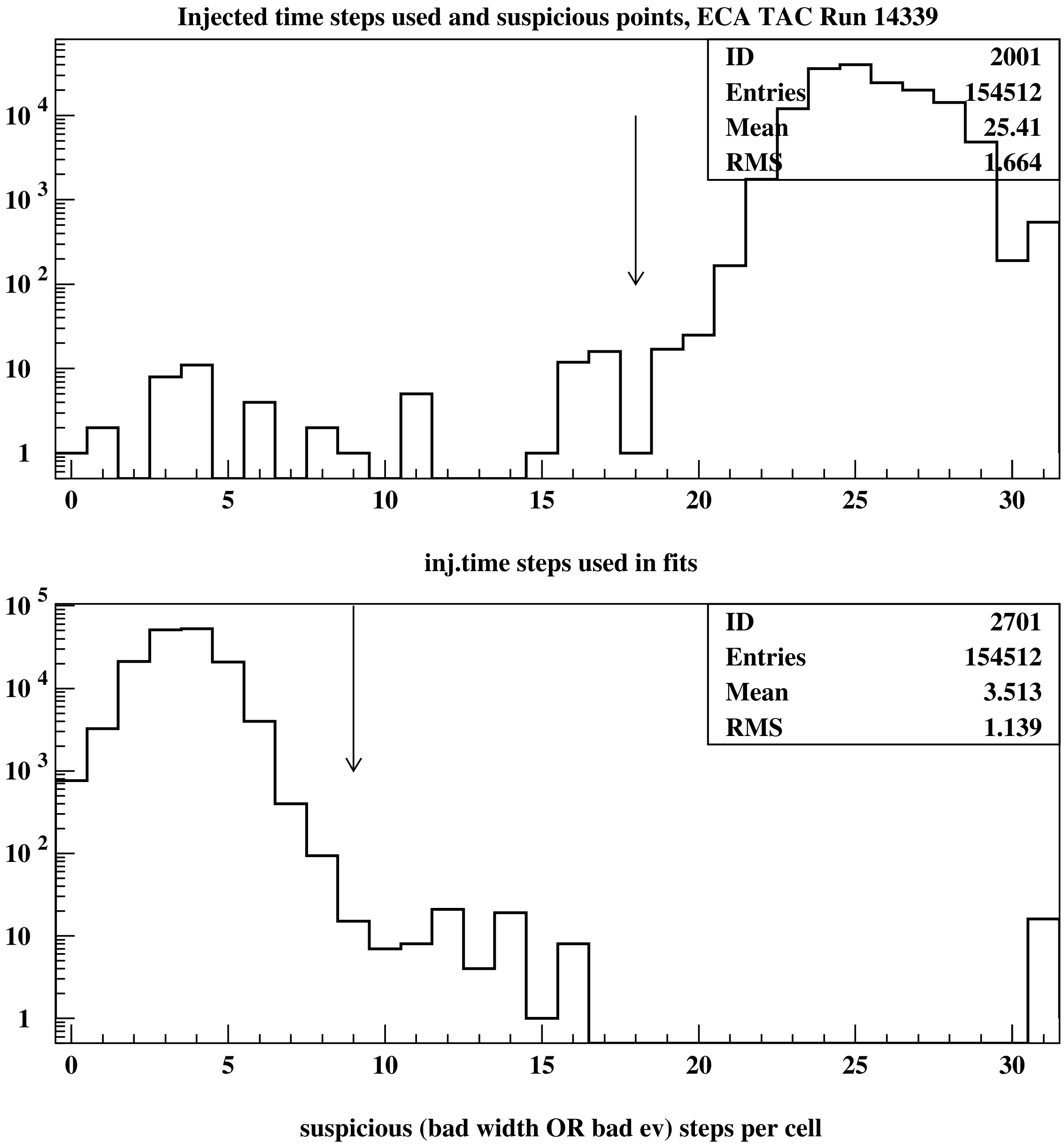


Figure 4.8: Top figure is number of events per cell used in the cubic fit, the bottom is number of suspicious points per cell.

Every month before the ADC's were finally fixed with the VSS change, a CALDAC run was made, then bad ADC's heated treat to temporarily repair them. The list is available to SNOMAN through title files, and is detailed in Chapter 5.

4.6 Sample Time Slope Diagnostic Summary Output

Below is a summary of failures for a typical ECA time slope run 14339. The detailed breakdown, with stats, is much longer, and is not shown here (see on surf in /bigdisk/sas/eca/log). For a historical breakdown for all ECA runs to present, see Chapter 6.

```
---------------------------------
CHANNEL SUMMARY ECA TAC RUN 14339
---------------------------------
# with Seq. off =  71
# changed MB ID =  0
# changed DB ID =  0
# with no events =  0
# failed #events tests =  7
# failed width tests =  8
# failed #points in fit tests =  6
# failed #suspicious pts. tests =  11
# failed Frati TAC jump tests =  11
---------------------------
# Flagged cubic     coeffs =  72
 Out of range =  31
 Bad diff =  64
# Flagged quadratic coeffs =  35
 Out of range =  35
 Bad diff =  16
# Flagged linear    coeffs =  128
 Out of range =  34
 Bad diff =  121
---------------------------
Total flagged channels =  207
---------------------------
---------------------------
CELL SUMMARY
---------------------------
# with no events =  0
# with too many points that had
too few events =  112
```

y

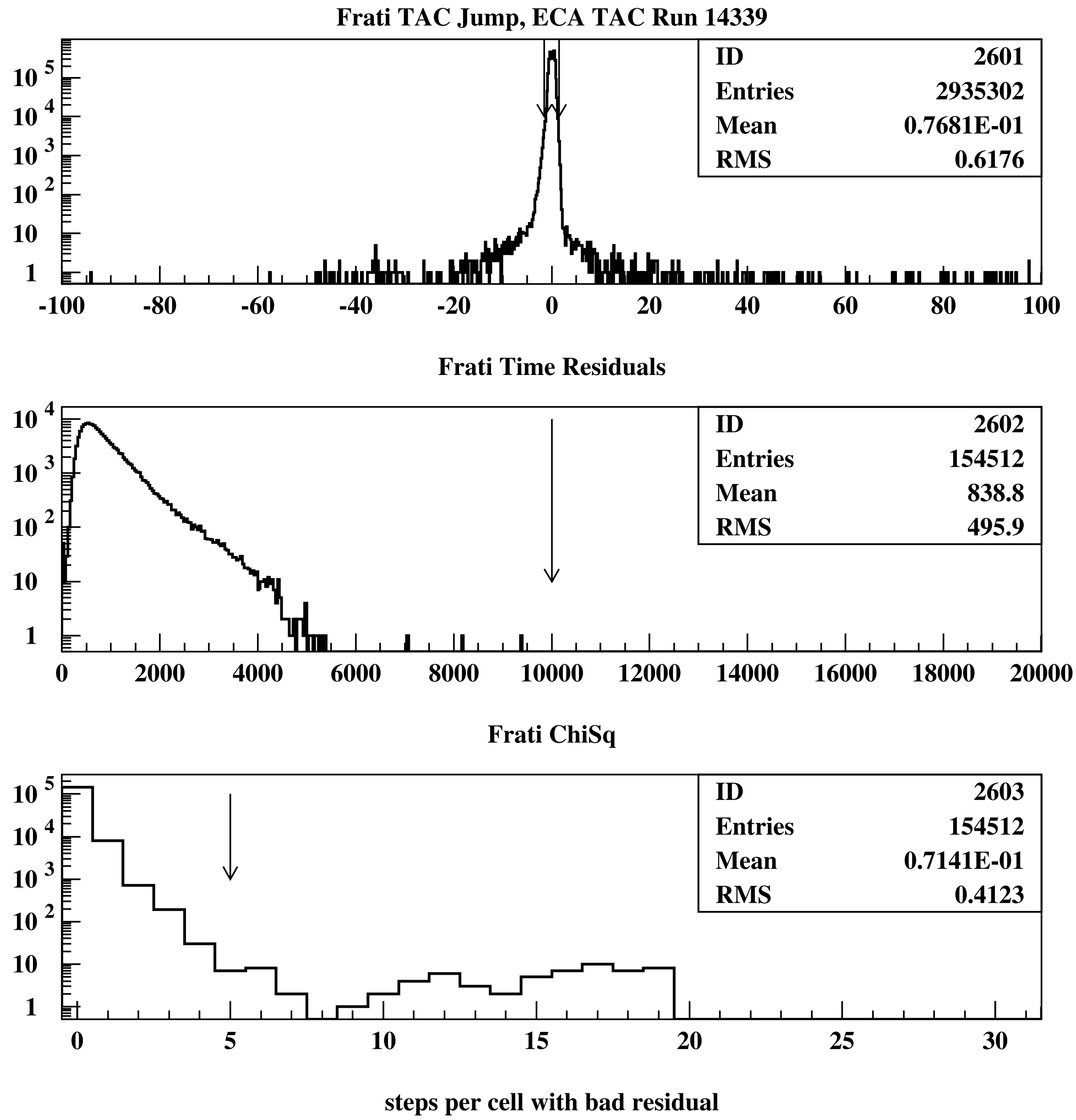


Figure 4.9: Frati's TAC jump test.

```
# with too many points that had
bad widths =  41
# with too few points in fit =  63
# with too many suspicious points =  87
# failed Frati TAC jump test =  72
---------------------------
# Flagged cubic     coeffs =  537
 Out of range =  34
 Bad diff =  526
# Flagged quadratic coeffs =  37
 Out of range =  37
 Bad diff =  16
# Flagged linear    coeffs =  770
 Out of range =  179
 Bad diff =  678
---------------------------
Total flagged cells =  891
---------------------------
```

4.7 TAC Slope Cut Summary

```
#.    TSlope Validation Parameters
-0.00004 #. Tolerance: Min.,
 0.00004 #.            Max.,
 0.00005 #.          & Abs. Max. Diff  for x-cubed coeff (cubic fit)
-0.02    #. Tolerance: Min.,
 0.02    #.            Max.,
 0.03    #.          & Abs. Max. Diff for x-squared coeff (cubic fit)
 5.      #. Tolerance: Min.,
 13.     #.            Max.,
 2.      #.          & Max_Diff for slope coeff (cubic fit)
 18      #. Min. acceptable no. of TSteps used in cubic fit
 5       #. Tolerance on step-by-step width check
 5       #. Min. no. of bad widths in a given cell for cell to be flagged
 6       #. Min. acceptable no. of events returned per TStep
 8       #. Min. no of TSteps which failed nevents check for cell to be flagged
 9       #. Min. no. of suspicious TSteps for cell to be flagged
 10000.  #. Tolerance: ChiSq tolerance
 1.5     #.            and abs. tolerance on point-by-point residual
```

```
5       #. Min. number of bad residuals for cell to be flagged (TAC jump test)
6       #. Min acceptable no. of events returned per TStep for that TStep
        #.   to be used in fit
```

4.8 ECA Time Slope Run Checks

After a ECA time slope run is complete, SAS will then execute the SNOMAN job that will generate the constants, validation flags, and various statistics. A set of checks are then done to these results. If there are enough failures, the run is flagged, the SIC alerted (via email), and the constants will not proceed to the database till they have been checked manually, and cleared for use by an expert.

The set of ECA time slope run tests are displayed in Table 4.2. The ECA SU keeps track of the 1/2 crate medians and differences for each of the cubic fit parameters, as well as statistics on each of the tests performed. As in the case for pedestals, the various run test limits represent reasonable expectation of failures and variations in the electronics. Any large variations in 1/2 crate fit slopes, etc, can signal a problem at the crate level, such as failing power supplies, etc.

A successful ECA run with no gross problems can give the SIC confidence that the electronics overall are performing well. Smaller problems flagged at the slot/channel/cell level are details that are best left to an electronics expert for investigation and repair.

Test Number	Test Type	Min Test Value	Max Test Value
1	Number of channels with zero events	NA	1
2	Number of failed board ID checks	NA	1
3	No. channels failed no. events per point test	NA	50
4	No. channels failed width test	NA	50
5	No. channels failed no. points in fit test	NA	50
6	No. channels failed suspicious points test	NA	50
7	No. channels failed Frati TAC jump test	NA	25
8a	Number of failed cubic fits	NA	50
8b	Number of failed quadratic fits	NA	50
8c	Number of failed linear fit	NA	50
9a	1/2 crate average cubic coeff	-0.00005	0.00005
9b	1/2 crate average quadratic coeff	-0.02	0.02
9c	1/2 crate average linear coeff	5	13
10a	1/2 crate cubic difference	NA	0.00005
10b	1/2 crate quadratic difference	NA	0.03
10c	1/2 crate linear difference	NA	2
11	Total flagged channels	NA	250

Table 4.2: List of ECA time slope run tests. The min and max test values are inclusive.

Chapter 5

Application of ECA Constants

Once the ECA constants have been generated, processed by SAS, verified to be okay, and put into the SNODB, they are now available for analysis. The subjects of reading in ZDAB data files, titles files, validity ranges and the correct time period application of constants can be reviewed in the SNOMAN companion.

This chapter deals with the details of how the ECA constants are applied to the raw ZDAB data and corrected charge and time measurements are produced. Also discussed is the software rejection of the TAC curl region, as well as the application of validation flags to the data. This last part is particularly important in having confidence that the constants applied are robust and correct.

5.1 TCAL Control File

The basic specification for how the derived ECA constants are used in conjunction with data is given by the TCAL titles bank in the file **calibration.dat** (this file also defines how the PCA is implemented as well). Four of the most relevant entries for ECA calibrations in this bank are words 4, 5, 6, 9, 15, and 16. Their meanings and default settings are as shown in Table 5.1.

These choices highlight four basic aspects of the calibration:

- The ability to use "old-style" ECA titles files as initially generated by Bill Frati has been preserved, though the files themselves cannot be generated in the current SNOMAN code. The actual code segment which applies these old-style constants is **alt_eca_calib_channel.for**. As the use of these files has effectively been discontinued as an "official standard," further details of the old-style ECA are not given here. Those

```
14  #. Flag to control calibration mode.  The choices are:
    #.   Charge: 10 -counts above pedestel (see charge zero flag)
    #.           20 -ECA linear correction
    #.           30 -ECA quadratic correction - old ECA, (same as 2 new ECA)
    #.           40 -PCA charge ( uses ECA counts above pedestel
    #.                            see charge zero flag)
    #.   Time:   1  -counts above pedastel (see time zero flag)
    #.           2  -ECA time linear correction - old ECA (same as 3 new ECA)
    #.           3  -ECA time ( new ECA -linear interp or cubic ,
    #.                          old ECA -cubic )
    #.           4  -PCA walk corrected time
    #.
0   #.Charge zero flag (0 = pedestals, 1 = intercepts)
0   #.Time zero flag   (0 = pedestals, 1 = intercepts)
    .
    .
    .
4   #. Type of ECA calibration constants to use (eca_type):
    #. 1=old ECA
    #. 2=new ECA (linear interp TSlope)
    #. 3=new ECA (cubic TSlope)
    #. 4=newer ECA (linear interp TSlope, and validation)
    #. 5=newer ECA (cubic TSlope, and validation)
    .
    .
    .
0   #. ECA Validation flag -- applicable if ECA type.ge.4
    #. 0= generate ECA validation status word in PMT bank
    #. 1= same as 0, plus, set cell Q/T to -9999 for pathologically
    #.    bad ECA channels (see ECA document for details).
    #. 2= same as 0, plus, set cell Q/T to -9999 if failed cell validation
    #. (E.G. If only Qlx and TAC failed cell validation, they are set to
    #.  -9999, while Qhs and Qhl are left unaffected).
    #. 3= same as 2, plus, if cell (Qhs.or.TAC) failed validation, set
    #.    cell TAC= -9999
    #. 4= same as 2, plus, if failed channel Q/T validation, set
    #.    all cell Q and T= -9999
0   #. Bad ADC (Wittich jumps) flag -- applicable if ECA type.ge.4
    #. 0= generate bad ADC status word in PMT bank
    #. 1= if ADC failed CALDAC, then set corresponding Q/T to -9999
    #. 2= if ADC failed Frati jump test, set corresponding TAC to -9999
```

Table 5.1: Relevant words from TCAL bank for ECA calibrations. The values displayed are the defaults for proper application of ECA calibrations to ZDAB data. When PCA gain data becomes available in the future, the user might want to use calibration mode = 44.

who are interested in it's use should contact Bill Frati.

- Charges and times may be calibrated either in terms of standard units ("picofratis" and nanoseconds) or simply channels above pedestal. The value for pedestal used for this may be specified as either coming explicitly from pedestal calibration data or from a linearly extrapolated intercept based on charge/time samplings. The default is to use the pedestals from ECA calibrations.

- By default the time conversion ("TSLOPE") is derived by linear interpolation between valid time samplings (which have been stored is compressed format – see Section 3.4.1). Very early times are always excluded due to the double-valued problem of the TAC curl, as well, very late times given the uncertainly in time extrapolation. The interpolation scheme means that some channels may have calibration ranges which are only *partially* valid if one or more time samplings have been labeled as bad. While not ideal, a general solution for what to do in such situations does not presently exists and this is thus adopted as the safest option. All this can be overridden by choosing to use a cubic fit parameterization of the TSLOPE. However, the resulting time calibrations are then by no means guaranteed!

- To generate ECA validation flags and pack them into the PMT bank status word. As well, if selected, to have the ECA calibration code set flagged channels to -9999.

5.2 Calibration Code

The main segment of code which actually applies the calibration is **cal_eca_calib_channel.for**. This subroutine takes the following inputs:

```
ccc      in     Card/Crate/Channel
cell     in     Cell number
q_lx     in     raw ADC counts
q_hl     in     raw ADC counts
q_hs     in     raw ADC counts
tac      in     raw ADC counts
```

and produces the following output for the TCAL settings in Table 5.1:

```
q_lx     out    Pedestal corrected ADC counts (low gain charge)
q_hl     out    Pedestal corrected ADC counts (high gain long charge)
q_hs     out    Pedestal corrected ADC counts (high gain short charge)
tac      out    TAC time in nanoseconds
```

A charge/time value of -9998 is returned if no calibration data exists, e.g. no ECA constants or board IDs changed, for the requested channel/cell. A value of -9999 is returned if the data exists but the calibration is deemed to be bad, e.g. TAC curl region, not enough points for interpolation, failed validation, etc. This last point will de discussed in more detail below.

The algorithm for ECA calibrating charge is a simple pedestal subtraction,

$$Q\_adc\_corr(i, cccc) = Q\_adc\_raw(i, cccc) - Q\_pedestal(i, cccc)$$

where $i = 1, 2, 3$ is Q_{lx}, Q_{hl}, and Q_{hs} respectively, $Q\_pedestal$ is the ADC pedestal value from the valid ECA titles file, and *cccc* is the appropriate crate, card, channel, cell number.

For the time interpolation, the algorithm is to find the two nearest good TAC slope points that bracket the TAC value to be calibrated,

$$XTslope(j, cccc) < XT < XTslope(j + a, cccc), \qquad a \geq 1$$

where XT is the raw ADC TAC to be calibrated, $XTslope$ is the stored ADC slope value for the *cccc* in question, and j is the slope point index. The interpolation slope points $XTslope$ cannot be flagged as bad (TAC curl) or suspicious, see Section 3.4.1 and 3.4.2 for details. A simple linear interpolation is done between the bracket calibration points to determine the calibrated TAC time in nano-seconds,

$$TAC = ((XT - XTslope(j + a, cccc))TIM(j + a) - (XT - XTslope(j, cccc))TIM(j))/$$
$$(XTslope(j, cccc) - XTslope(j + a, cccc))$$

where TIM is the known PED-GT delay time (in nsec) of the calibration points. When the TAC is outside the valid calibration range, e.g. TAC curl region or late times (low TAC), then the TAC value returned is set to -9999, and a bit in the validation flag is set (see Table 5.2).

Figures 5.1 and 5.2 show the improvements of using the time interpolation scheme for ECA calibration of the TAC.

Both calibration and uncalibration are handled in this same code. Uncalibration is signaled by calling the routine with CCC set to -(CCC+1) . One word of warning is that, in order to provide a more general use, "uncalibration" simply performs the calibration in reverse. So, to mimic the electronics you must nint() the resulting charges and times and clip them to between 0-4095 counts checking for failures.

5.3 TAC Curl Identification

Because the TAC curl is a very nonlinear region, as well as having the potential for double valued times, a software cut is necessary to flag this region and cut it out for subsequent

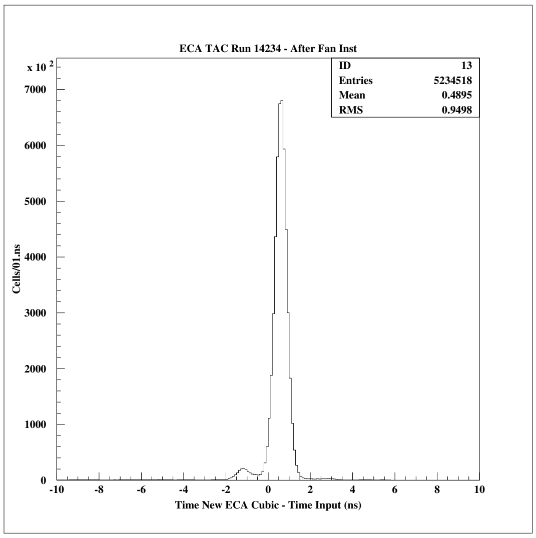


Figure 5.1: Generated TAC time residules for new ECA cubic fit algorithm. The small bump at -2nsec is a hardware artifact (channel 17 pickup) that does not present in the full ECA generation.

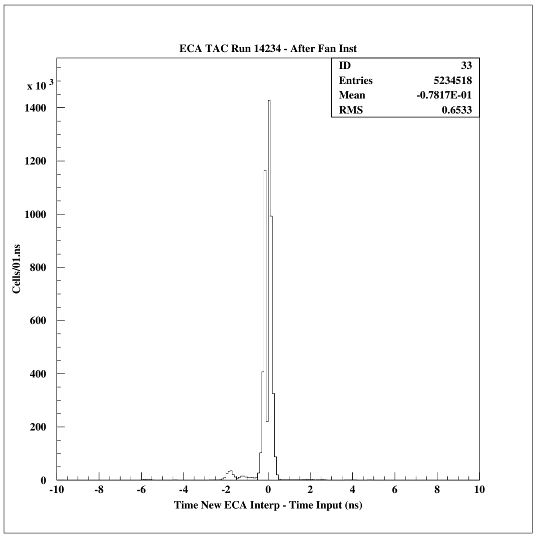


Figure 5.2: Generated TAC time residules for the new ECA interpolation scheme.

analysis. It is known from hardware studies that this region is about 40-60nsec at the end of the TAC window (large ADC values). The GTVALID window was intentionally set long, about 400 nsec, to allow a software cut in the window that would not adversely affect the acquisition of early light.

The process of identifying the TAC curl region is done during the generation and validation of the ECA data, see Section 3.4.2 for details. For a normal TAC curve, the successive points should be monotonically increasing. When a point in the TAC curve is found where the next injected point is less than the current one, then this point is identified as the cusp of the curl and the preceding 2 (or 4 points if using 31 injected points) points are flagged a "bad". The TAC curve is effectively truncated at this point when calibrating using the interpolation scheme.

Figure 5.3 shows the results of the TAC curl cut, as well as the removal of hits in the low region that are not extrapolated. The number of hits that are removed is about 2%, with less than half in the TAC curl region. The important point to note is that the leading edge of the in-time peak is not affect by this cut.

5.4 Application of Validation Flags to Analysis

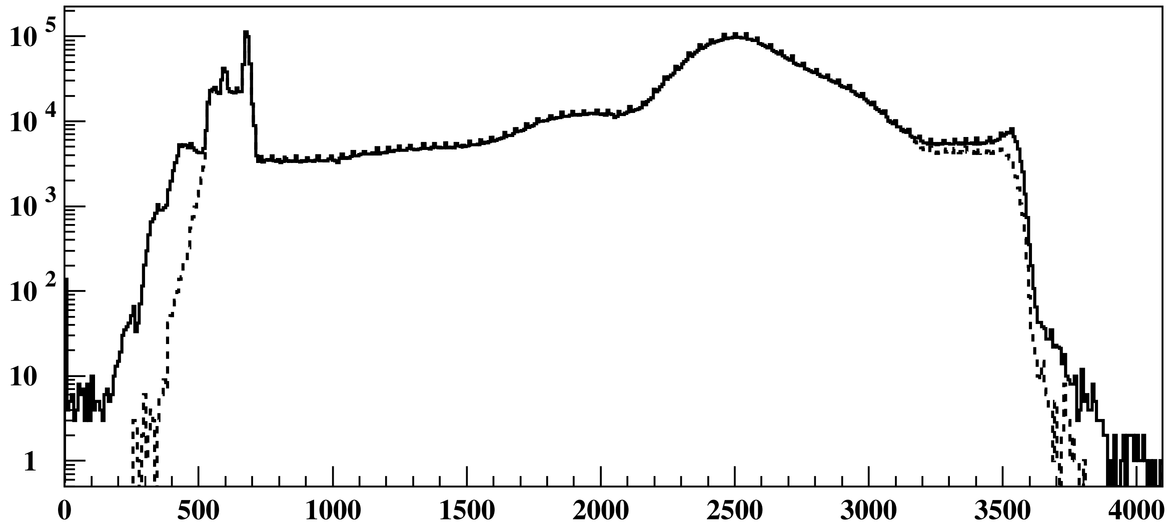
After the ECA validation data has been extracted and saved to the titles files, they then become available to the analysis within SNOMAN. There are two ways the the analyzer can choose how the the validation information can be used, as described below.

5.4.1 ECA Status Word

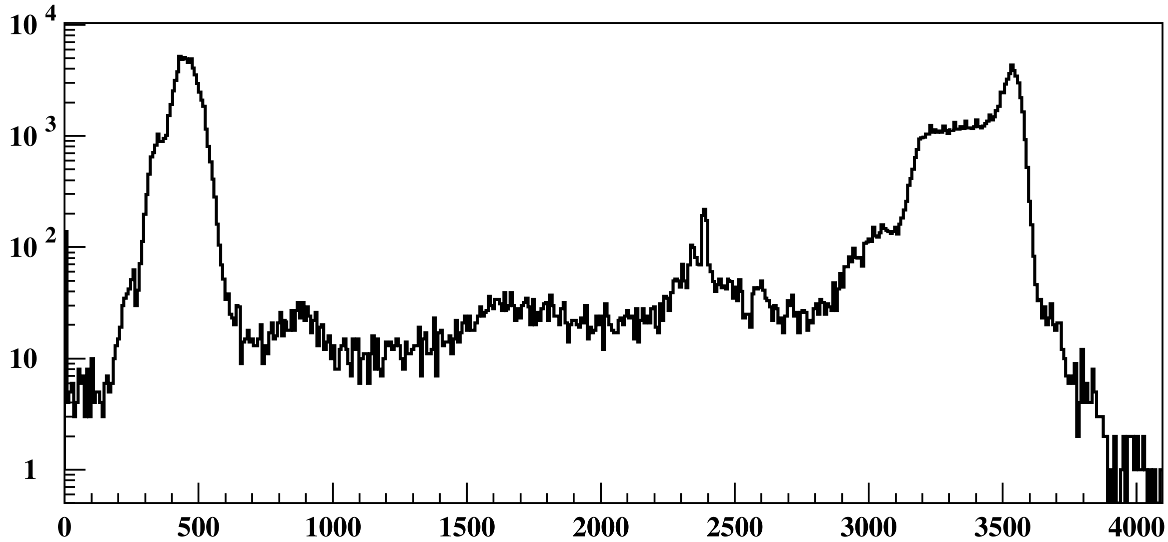
The first method, (set bank TCAL 1 word 15 to 0) is to pack the ECA status word in the PMT bank (KPMT_ECA_VAL) as shown in Table 5.2. The data is ECA calibrated normally, and the status word is made available so subsequent SNOMAN code can check the status to identify potential problems with the ECA calibration. For example, a fitter can do a *btest(eca_status,6).and.btest(eca_status,7)* test to determine if the TAC calibration is okay and whether the tube should be included in the fit.

When the raw data is ECA calibrated, the validation word packed in the titles bank associated with the specific *cccc* is checked, and a number of tests performed and the appropriate bits set in the status word. A summary of these tests, and the bit definitions in Table 5.2, are given below:

- Bit 0: An "or" of bits 1 thru 7.



Raw TAC - all hits



Raw TAC - curl/low region

Figure 5.3: The solid line in the top figure shows the raw TAC spectrum for all hits in a neutrino run 12233, sub run 0. The dotted lines shows the remaining hits after the TAC curl and low end has been removed (bit 15 set in ECA status word, Table 5.2). The second plot show the TAC distribution of rejected curl/low hits. The origin of the structure at late time (600) in the first plot is not known.

- Bit 1: If the current Mother/Daughter-board ID from the DQXX banks are different than in the ECA titles being applied, the flag is set and the returned Q/TAC is set to -9998.
- Bit 2: No ECA constants available, the returned Q/TAC is set to -9998.
- Bit 3-5: If the cell failed either the *mean*, *width* or *diff* test (see Table 3.1), then the corresponding Q_{xx} pedestal bit is set.
- Bit 6: If the cell failed the TAC pedestal *diff* test (see Table 3.1), then the bit is set.
- Bit 7: If any of the the cell TAC slope tests fail (see Table 3.4, "or" of bits 1 to 10), then the bit is set.
- Bit 8: If bit 15 of the pedestal mask (Table 3.1), or bit 14 to 26 of the TAC slope mask (Table 3.4) is high, then the bit is set.
- Bit 9: If the Frati cell jump test (bit 12 of Table 3.4) fails, then set the bit.
- Bit 10: If the Frati channel jump test (bit 27 of Table 3.4) fails, then the bit is set.
- Bit 11-14: If the appropriate Q/TAC ADC is bad based on the CALDAC test (see Section 5.5), then the bit is set.
- Bit 15: Flags the case when the TAC is set to -9999 because the hit is in the TAC curl or low region.
- Bit 16-19: If the cell failed two of the pedestal *mean*, *width* or *diff* tests (see Table 3.1), then the corresponding Q_{xx}/TAC bit is set. This flags particularly bad cells.
- Bit 20: If one of the time slope cell test bits 2,3, or 8 is set (see Table 3.4), then the bit is set. This flags particularly bad cells.

The status checks are specific to either cell or channel depending on what level the failure applies. For instance, bits 3 in Table 5.2 is cell specific. If for a certain channel the Q_{hs} pedestal calibration is bad for cell 5, it might then be okay for cell 6. However, bit 11 applies to all channels on a slot, since a bad ADC is common to all Q_{hs} channels on a Motherboard.

Note, other than bits 1 and 2 of the ECA status flag, no action is taken to calibrated Q/TAC data. The appropriate ECA calibration is performed, and the data sent on its way, with only error flags set.

These bits are available to SNOMAN and can be used within PCA, Optics, Energy and fitting routines, or where ever the PMT bank is accessible, to flag or turn off potentially bad ECA channels.

5.4.2 Automatic Failure Flagging

The second method (set bank TCAL 1 word 15 to 1,2,3, or 4) is a quick and dirty means of applying the ECA validation information to turn off channels (set to -9999). The ECA status flag is generated as above, and when a ECA status bit 3 thru 8 is set, then the corresponding Q/TAC cell is set to -9999. This way, no modifications need to be made to SNOMAN since if it finds a Q/TAC value of -9999, then at least for the fitters, it will throw out the bad hit.

In more detail, the three user selected possibilities are as follows:

- Set bank TCAL 1 word 15 to 1: As well as generating the ECA status word, if bit 16 thru 20 is set, then return -9999 for the appropriate Q/TAC cell. This is to turn off particularly bad, or pathological channels. This would also include TAC slopes that are out of specification.
- Set bank TCAL 1 word 15 to 2: As well as generating the ECA status word, if bit 3 thru 7 is set, then return -9999 for the appropriate Q/TAC cell, e.g. if only bit 4 set, return Q_{hl} of -9999, while other Q/TAC calibrated normally.
- Set bank TCAL 1 word 15 to 3: Same as 1 above, plus, if bits 3, 6, or 7 are set, then return -9999 for the TAC. The idea here is if Q_{hs} is bad, then since it feeds into the TAC via the PCA walk correction, then the TAC time will be wrong.
- Set bank TCAL 1 word 15 to 4: Same as 1 above, plus, if bit 8 is set, then return -9999 for all three charges and TAC.

5.5 Bad ADC Flag

As described in Section 3.4.2 and 4.5.7, bad ADC's plagued the detector for the first year of running (till August of 2000) before they were permanently fixed with the LED modification. This systematic is handled in the analysis by having the capability to turn off (or flag) bad slots that had misbehaving ADC's identified via either the CALDAC test, or Frati's jump method.

The data from the CALDAC tests are hand scanned, and the bad ADC's (see Figure 3.1) are flagged by setting a bit in the BADC titles. The validity of the bank is backwards, since when the test is done the ADC's were subsequently fixed by heating. There is one bit for each of Q_{hs}, Q_{hl}, Q_{lx}, and TAC. If the ADC is flagged, then all 32 channels associated with the Q/TAC for the Motherboard in question are flagged as bad.

Bit no.	Test Type
0	failed overall ped/TAC cell check (OR of bits 1-7)
1	failed MB/DB ID test (channel/slot)
2	no ECA events/sequencer off (cell)
3	failed Qhs ped test (cell)
4	failed Qhl ped test (cell)
5	failed Qlx ped test (cell)
6	failed TAC ped test (cell)
7	failed TAC slope test (cell)
8	failed overall pedestal/TAC check (channel)
9	Frati TAC jump flag set (cell)
10	Frati TAC jump flag set (channel)
11	Q_{hs} failed CALDAC test (slot)
12	Q_{hl} failed CALDAC test (slot)
13	Q_{lx} failed CALDAC test (slot)
14	TAC failed CALDAC test (slot)
15	TAC curl or low region (cell)
16	pathological Qhs pedestal (cell)
17	pathological Qhl pedestal (cell)
18	pathological Qlx pedestal (cell)
19	pathological TAC pedestal (cell)
20	pathological TAC slope (cell)

Table 5.2: ECA status word available in PMT bank. If the bit is set to 1, then the corresponding condition is true. Bit 15 flags the case where the raw TAC value lies in the curl or low region, and is not calibrated.

During application of the ECA constants, the user can choose how to handle bad ADC's with word 16 in the TCAL bank. A description of the options follows,

- Set bank TCAL 1 word 15 to 0: Set the bits 9-14 in the ECA status word. See Table 5.2.
- Set bank TCAL 1 word 15 to 1: Same as 0, plus, if Q/TAC ADC is flagged as bad from CALDAC test, set associated Q/TAC -9999. This affects all 32 common channels on a Motherboard.
- Set bank TCAL 1 word 15 to 2: Same as 0, plus, if Frati jump test failed for channel/slot, set associated Q/TAC to -9999.

Chapter 6

Analysis and Systematic Studies of ECA Constants

The purpose of this chapter is to study the various systematics of the ECA constants as well as pointers on how to perform more detailed systematics studies of the application of the ECA constants to the data (PCA, data cleaning, optics, energy, reconstruction, etc). This final topic is a major component of the neutrino analysis and is beyond the scope of this report. The ECA validation machinery is now available to perform such studies.

6.1 ECA Regeneration and Trends

With the new validation code and titles format, a regeneration of the old ECA data had to be done. As well, it provided an opportunity to review and investigate ECA trends, weed out bad runs, and flag detector configuration changes which might affect the constants. The working area on **surf** with all the log and hbk and control files is in /bigdisk/penn/eca/generation.

Appendix D shows a list of ECA runs from the start of production running up to Mar 26, 2001. The list shows the runs that were deemed okay to be used in the regeneration, as well as a separate list of the runs that were identified as problematic. Also documented are special conditions or unusual circumstances during the ECA, as well as when major detector configuration changes were performed. This provides a basis for understanding the behavior of the constants as well as when to flag detector changes that requires turning off the difference test.

Appendix E and F shows the summary data for the ECA regeneration up to runs 15402/15401. Figures 6.1 to 6.4 shows time plots for some of the summary data quantities.

To first order, the detector wide average pedestals are constant within a few ADC counts. However, there are a few notable exceptions, as well as bumps and wiggles in some of the other trend quantities. A discussion of these is in order,

- Between runs ∼11200 to ∼12300 there is much instability in the pedestal averages (Figure 6.1), especially the large droop in Q_{lx} . During this period the effects of the nonlinear ADC's was increasing. Figure 6.5 shows the overall increase in the number of bad ADC's, which track this period. As well, gain op-amps and ADC's were dying because of the reheating treatment required to temporarily fix the ADC's. Since Q_{lx} is not critical to neutrino analysis, they were not fixed immediately, which accounts for the significant drop in the Q_{lx} average. They were not fixed till the ADC LED mod was done in August of 2000.
- A Sharp jump in TAC pedestal average at run 12468 (Figure 6.1). This is due to the change in GTVALID timing when turning off the QUSN7-CMOS twiddle bits, i.e. going from TCMOS skey=601 to TCMOS skey=611. This was done to lessen the effects of the Time Since Last Hit (TSLH) problem in the QUSN7 chip.
- The fit coefficients change from run 12425 to 13390 (Figure 6.3). This is a result of going from 19 TAC fit points to 31. After run 13390, the 31 point variable TAC slope was used for TSlope generation. Because of the sampling differences at low TAC values and a slight non-linearity in the QUSN7-CMOS response in this region, the slope coefficients changed. Overall, there was no real change in slope characteristics, which only serves to highlight the sensitivity of the fit method, and why we do not use it for calibrations.
- Up run 12340, the error rates for various quantities progressively increased (Figures 6.2 and 6.4). This is a result of the non-linear ADC's, which was eventually repaired by changing VSS to -12V (know as the LED mod). Figure 6.5 shows the general increase in the number of bad ADC's based on monthly CALDAC tests, which clearly tracks the error rates. Also, in Figure 6.4, the number of identified Frati jump bad ADC channels also shows the same trend. This further supports the claim that the general degradation of the ECA constants was due to the non-linear ADC problem.
- Starting at run 15306, the charge pedestals drooped significantly (Figure 6.1). This is due to the new VBAL (skey 621) constants that was required to compensate for the crate fan upgrade. The 5C drop in crate temperatures necessitated a recalibration (ECAL) of the integrator balance.

To summarize, the biggest effect on the stability of the ECA constants was the non-linear ADC's. Once these were permanently fixed with the LED modification, the constants have shown remarkable stability and consistency. If the detector configuration or environment does not significantly change, then the charge and time pedestals are good to within a couple ADC counts over a period of a year. This is remarkable stability!

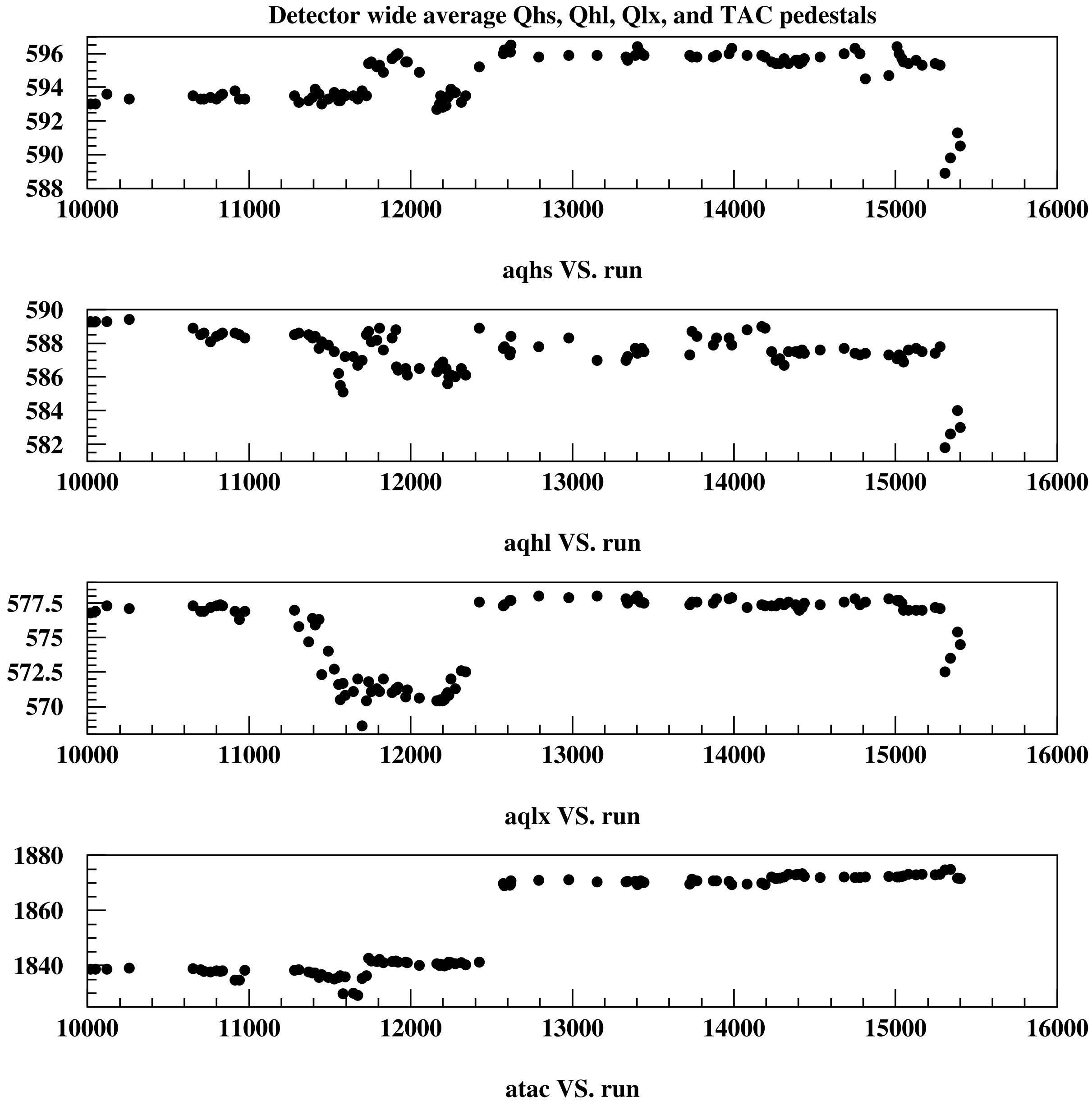


Figure 6.1: Detector wide pedestal averages since the start of production running (see Appendix E for details and definition of labels).

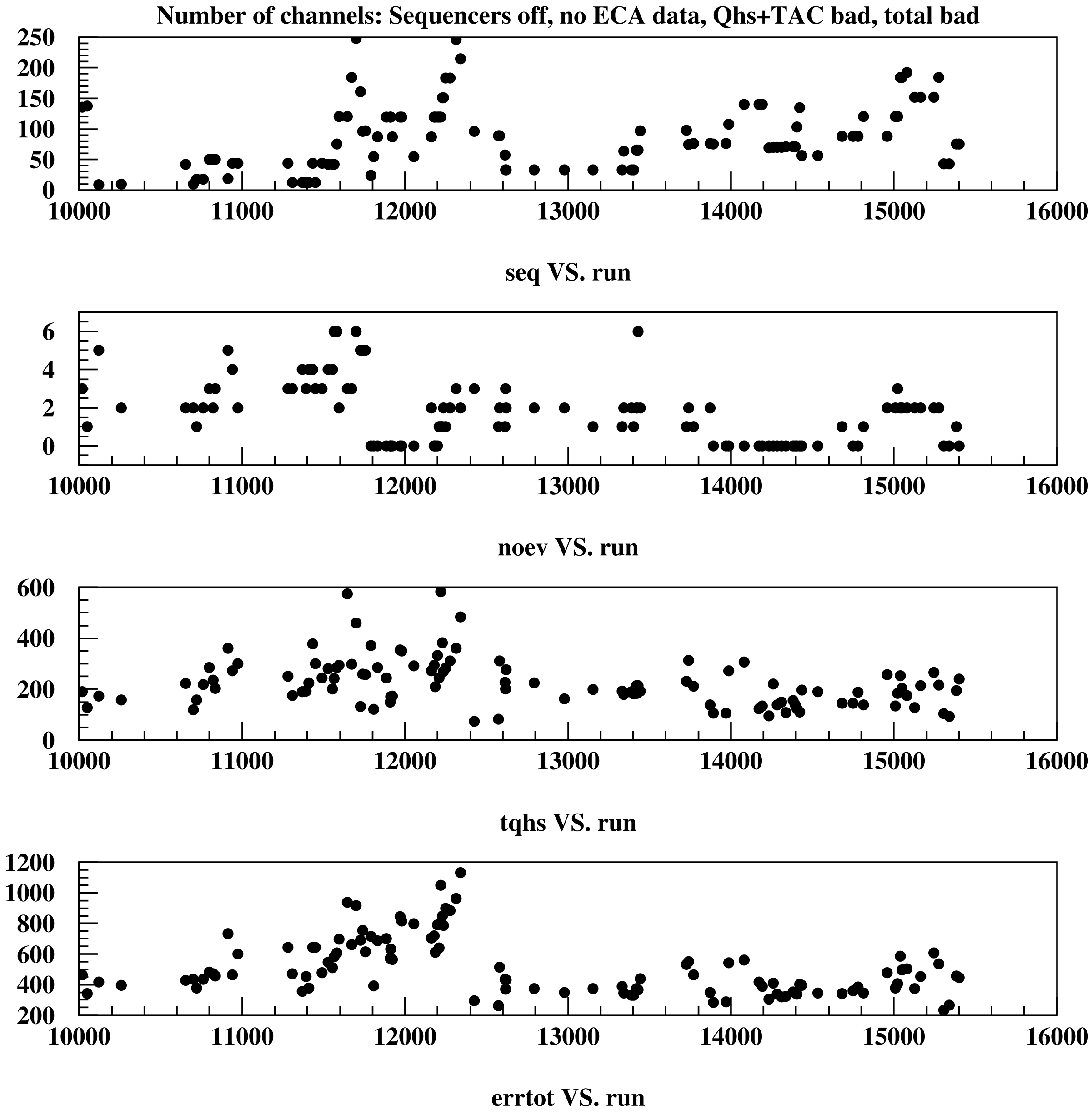


Figure 6.2: Various ECA pedestal errors since the start of production running (see Appendix E for details and definition of labels).

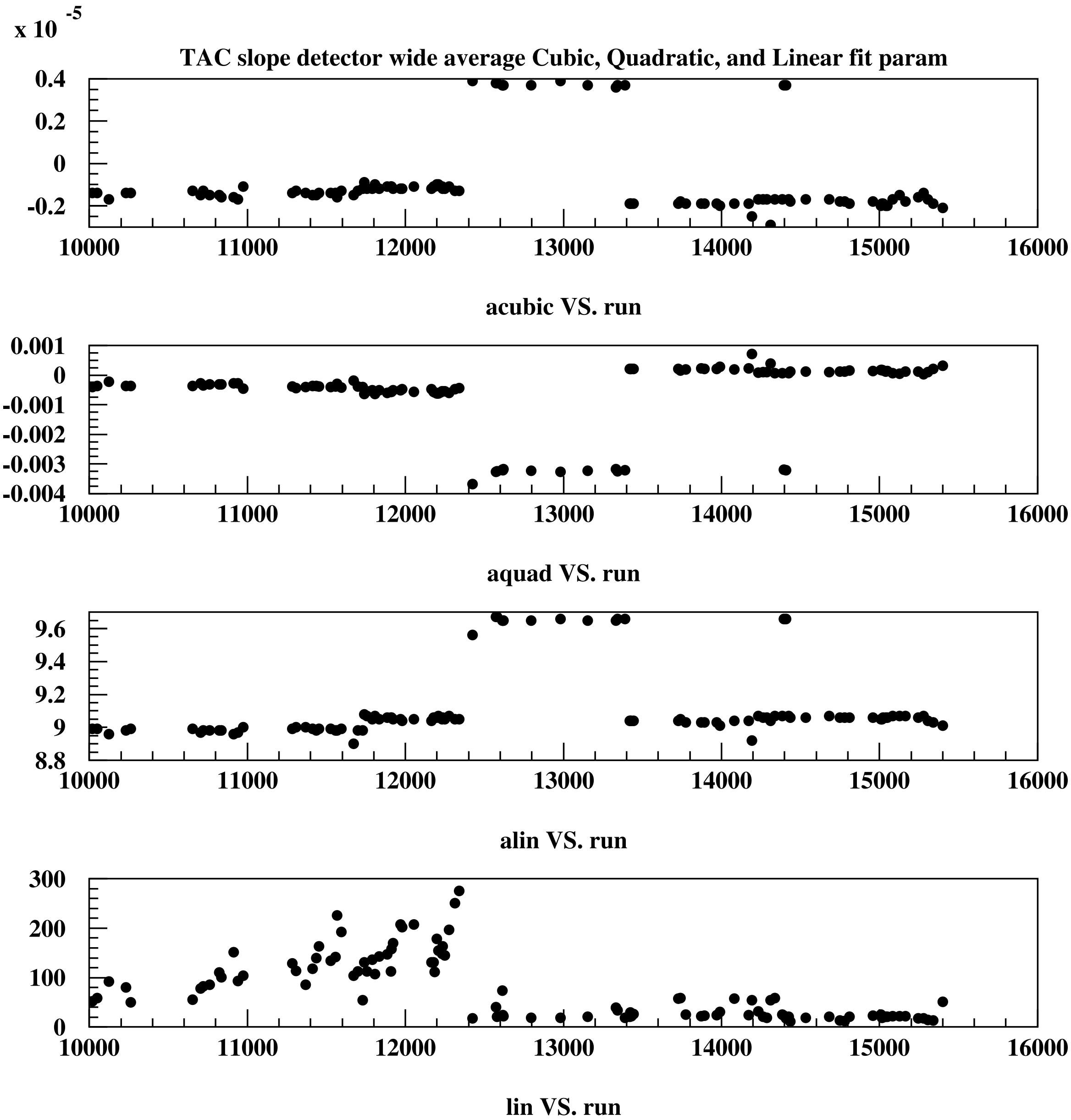


Figure 6.3: Detector wide time slope fit averages since the start of production running (see Appendix F for details and definition of labels). The last plot (lin vs. run) is the total number of channels with bad linear coefficients.

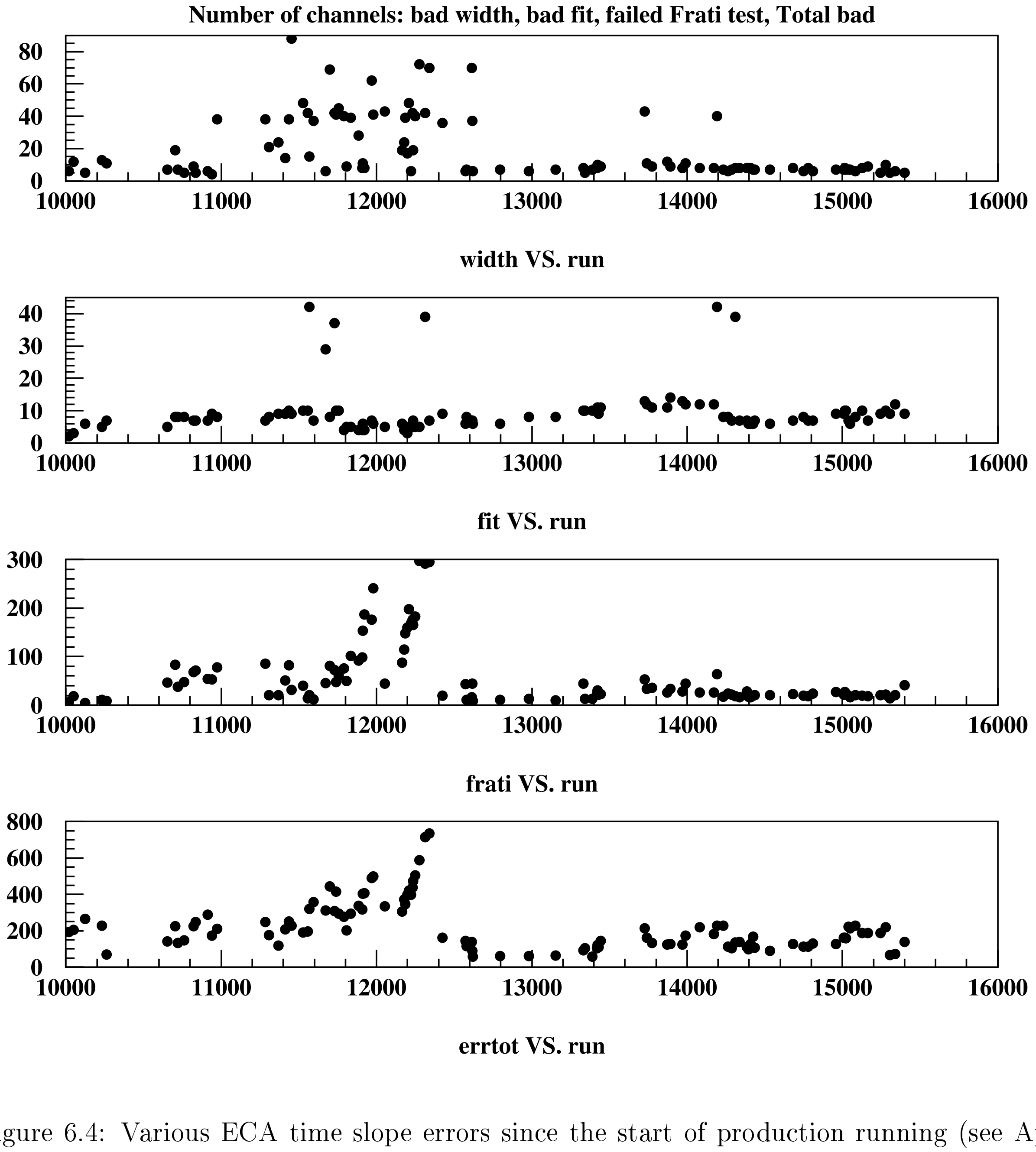


Figure 6.4: Various ECA time slope errors since the start of production running (see Appendix F for details and definition of labels).

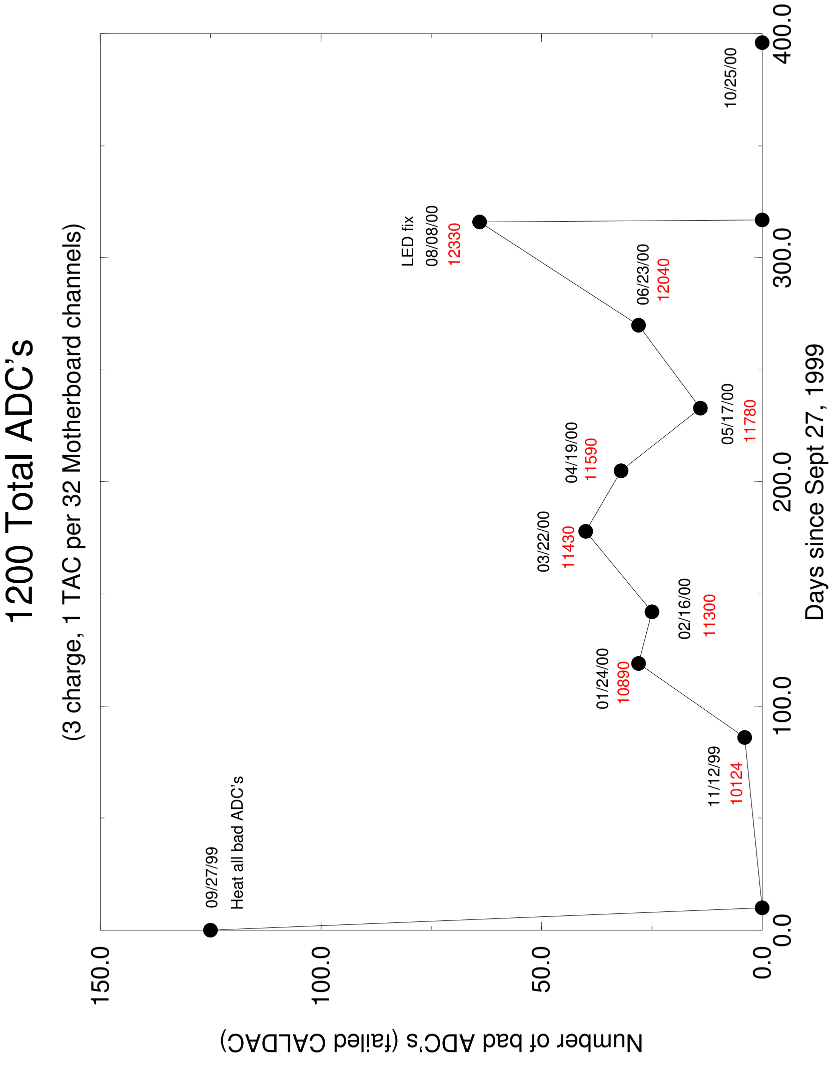


Figure 6.5: Bad ADC history based on CALDAC tests. Shown at every point is the date, and the preceding run number (red) when the tests occurred. No new bad ADC's have been seen up to the latest CALDAC run of March 8, 2001 (day 530).

6.2 Behavior, Pathologies, and Systematics of ECA's

Some noteworthy ECA systematic effects are discussed below.

6.2.1 Rate Dependencies

Normally the ECA data is taken at 10Hz for pedestals, and 1 kHz for TAC slopes. There is a known rate effect in the QUSN7-CMOS chip called the Time Since Last Hit (TSLH) effect. At time constants of 10's of seconds between tube hits with associated Global Trigger, the TAC response can drift by many nanoseconds from true. This effect is well studied and documented (see SNO analysis homepage). The solution was turning off the QUSN7-CPOS twiddle bits and installing upgraded cooling fans. This reduced the effect significantly. As well, the old data, as well as any new data should it be required, can be corrected with SNOMAN (known as the HCA correction, see TCAL and SNOMAN companion).

From the perspective of the ECA constants, they are generated at a high enough rate that they are not affected by the TSLH problem. Thus, the ECA constant can be regarded as the normal baseline for correcting the effect. For these reasons, the TSLH effect is mostly ignored in this study.

A similar effect has been reported for the Q_{hs} charge, and studied at some length by the electronics group. Here, the dependency is not trigger rate, but is discriminator rate, whereby the Q_{hs} charge will show droop at low discriminator rates (less than a Hz). Thus, at normal tube rates at nominal HV, this effect is negligible. However, recently there has been some indications that there might also be a trigger rate dependency in the Q_{hs} response (and the other two charges) somewhat similar to the TSLH effect. This QSLH effect is currently being studied and a correction will hopefully be developed soon.

6.2.2 Temperature Dependencies

ECA pedestal run 11940 was taken during a period of relatively high deck temperature and humidity (Chiller problems). The deck temperature was 21.7C (normally 19.6C), and humidity 62% (normally 55%).

This pedestal run was analyzed and found not to be too different from a normal run (see on **surf** in /bigdisk/penn/vdwater/snoman/eca for log and hbk files). There was no significant change in the crate average charge and TAC pedestals. However, about 735 channels were flagged by the validation, when normally there are only 400 during this period. The increase is mostly from the difference test. This might be expected if the change is due to the increased temperature. Therefore, even a 2C change in deck temperature can have a non-negligible effect on the ECA constants.

The current procedure is to set the UC bit if the deck temperature gets around 23C. It might be worth dropping this threshold to 22C. As well, if the temperature excursion is for many days, and relatively constant, then a special ECA runs should be done to calibrate the data during this high temperature period.

6.2.3 ECAL and Low Voltage Dependencies

The ECA constants are extremely dependent on the values of the electronics calibrations constants (ECAL) loaded into the hardware by the DAQ (initialized either by PENNDAQ or SHARC). An example is back before production running, SHARC got the VBAL constants mixed up and was initializing the hardware with the wrong values. This was only discovered a few months later (which is one of the motivations for the ECA validation code) by Bill Frati when studying the ECA constants, as they were wildly fluctuating.

Fortunately, the ECAL constants (VBAL, TDISC, TCMOS, and VTHR, see Appendix G) are maintained and controlled tightly by the electronics group. Boards are recalibrated only when new ones are put into the detector, or when significant detector changes warrant a recalibration, e.g. TCMOS twiddle bits off, crate fans, etc.

ALL of the voltages for the front end electronics and digitization chain are derived from stable Motherboard power regulators, and are mostly independent of the five crate supplies (+5, -5, +8, +24, -24 Volts). This implies that the ECA constants themselves are fairly robust as all critical voltage references and power are regulated and stable. However, the main supply for the QUSN7-CMOS chip is VCC (+5V), which is drawn from the main rack supply. In principle, the charge and time pedestals and TAC slopes should be independent of VCC, however, this has been found not to be the case (yet another QUSN7-CPOS feature). A change in the crate VCC does affect the ECA constants.

An example of this effect can be found in ECA runs 11728 and 11729, where VCC for crates 2 and 3 was adjusted (running too high) from about 5.5V to 5.0V. It was observed that the three charge pedestals increased by about 14 ADC counts, and the TAC pedestal increased by 70 counts. As well, the TAC slope went from 8.5 counts/nsec, to 9.4 counts/nsec. These are significant changes, and is why changes in VCC need to be monitored carefully. Under normal operation, the PowerOne low voltage supplies deliver reliable and stable voltage (well, except the 24V, but that has been solved now). Coupled with gobs of filtering capacitance close to sensitive inputs, the ECA constants are fairly immune to this design feature of the QUSN7-CPOS chip. However, small effects of load, i.e. number of firing channels, are evident on channel pedestals, as described below.

6.2.4 Other Detector Condition Dependencies

Other dependencies studied were changes in conditions such as crate high voltage on/off, channel thresholds nominal/max, and different crate loads (16 boards versus one board).

Figures 6.6 and 6.7 shows the effects of crate loading on pedestals. For Q_{hs} there is a five count shift going from pedestal pinging one slot to the entire crate, while for the TAC the shift is three counts. The other two charge channels also show similar behavior. The physical cause of this is the fact that the QUSN7-CMOS charge and time measurements are somehow sensitive to VCC (chip power +5V), as describe in the section above. This voltage comes from an unregulated rack supply (direct from PowerOne supply), which probably has voltage sags under heavy loads, i.e. large number of channels firing.

Since ECA's are always taken with HV on, and channel thresholds at nominal, the effects of changes in the state of HV or thresholds is not relevant. However, for completeness, a number of runs have been taken with HV off, or channel thresholds set to max. The results show that as expected, the changes in these conditions have only small effects on the ECA constants. These differences are consistent with the expected contribution from random hits, since PMTs with HV on, or nominal thresholds, run at higher discriminator rates.

6.2.5 Channel 17 Q_{hl} and Q_{lx} Jitter

It has been observed that there is noise being injected into channel 17 Q_{hl} and Q_{lx} pedestals. This can be seen in Figure 4.2 and 4.3, where the widths and differences are larger for these two charge channels than for Q_{hs}. The offending contribution can be traced to channel 17. This is due to the pedestal ping lines from the Motherboard that go to the PMTIC via the EUROCONN connectors on pins adjacent to channel 17. Even though these lines are low noise ECL differential, there is still some pickup injected. The long integrate channels are most affected, and is mostly like due to resonance matching of the frequency of the noise and the time scale of the long integrate.

Once the problem was identified, the pedestal ping width was increased up to 1000 nsec (via SHARC), which is much longer than the 400 nsec channel width. This has decreased the effect since the back part of the pedestal pulse ping pulse is now beyond the long integrate time. However, there is still some effect remaining, probably from the front end of the pedestal ping pulse.

As well, this noise also accounts for channel 17's firing in the half of the crate that is not being directly pinged. The pickup from the pedestal ping causes the other channel 17's in the crate to fire, even though they are not programmed to fire. These channels are easily removed during analysis of the ECA constants, since we know which channels should be firing.

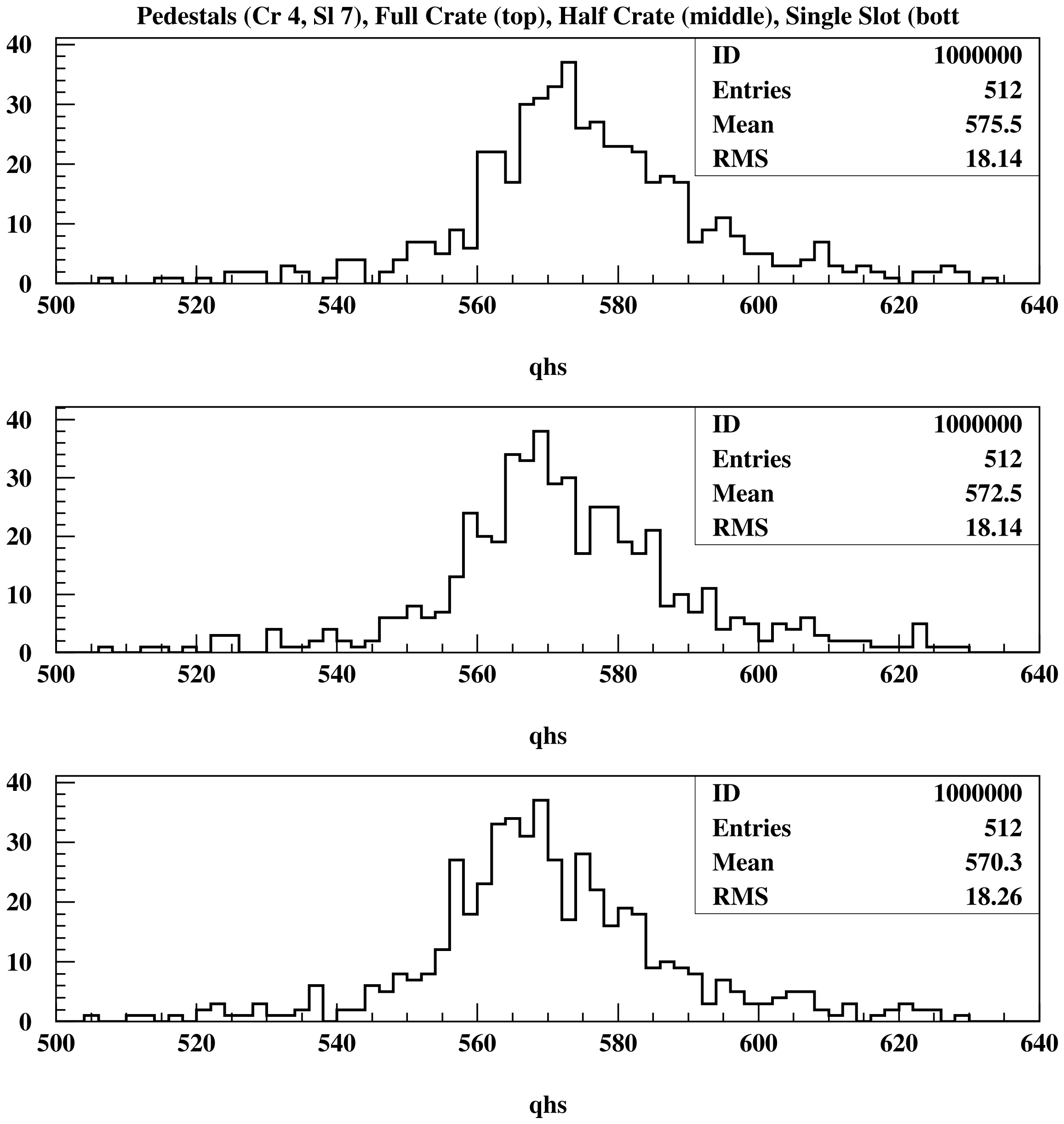


Figure 6.6: Effects of crate load on Q_{hs} pedestals.

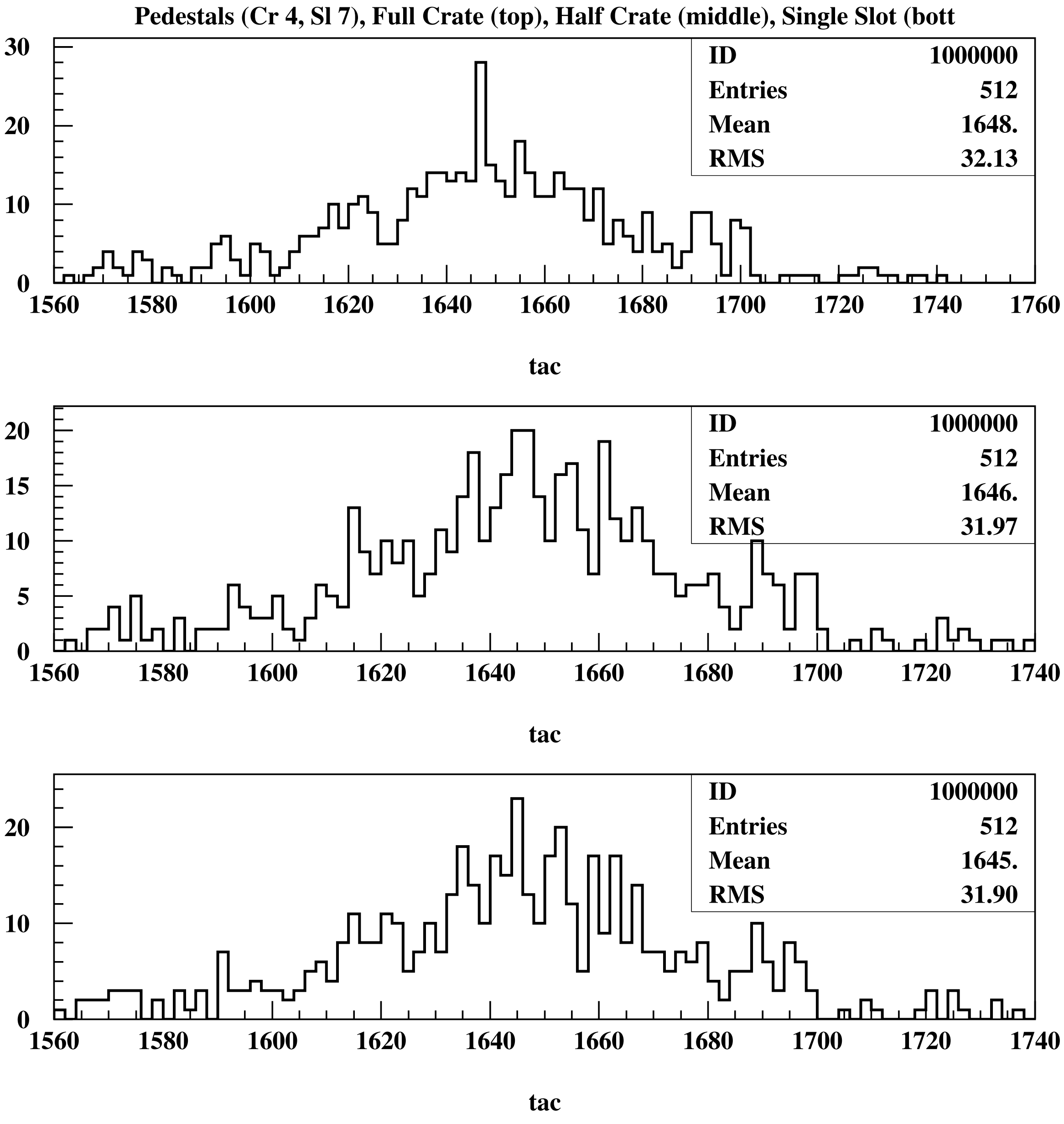


Figure 6.7: Effects of crate load on TAC pedestals.

6.3 Strategy for Systematics Studies of ECA on data analysis

```
Handling of ECA Validation in Analysis:
--------------------------------------

The biggest affects of bad ECA constants are on PCA, data cleaning,
reconstruction, and in-time energy measurements (optics should also be
on this list, but its a done deal at the moment).  Below is a
discussion of how to apply the new ECA validation code and regenerated
data to the analsysis and the various systematic studies required to
gain confidence in our results.

As part of the ECA review, an analysis of all ECA runs using the new
validation code has been done.  Overall stability of the constants has
been proven (see ECA document) and when deviations have occured, been
explained by a either a change in detector configuration, or
unexpected effects.  If the change is big enough, then the ECA
difference test is turned off for the ECA runs in question.  Also, a
handful of bad ECA runs (missing crates, etc) has been identified and
removed the regeneration list.  A new ECA run list has been made (see
Appendix D of the ECA document) and a rengeneration of the constants
with the new validation code has been done.  As well, a few minor bugs
in the old generation and application of the ECA constants were fixed
(see ECA doc).  A summary of all the runs can be found in Appendix E
and F.  In general, the value of the ECA constants have not changed
significantly, but now include validation information indicating the
robustness of the constants.  The regenerated ECA constants are in
SNODB (as of mid March 2001), and supersede the old ones.

As well, a list of bad (nonlinear) ADC's has been generated based on
monthly CALDAC runs.  This list is now available as title files in
snoman with appropriate flags/code to remove bad ADC's during ECA
calibration of the data.

I am assuming below that one is using SNOMAN 4_0183 or later, which has ECA
updates necessary for the analysis below.


1) PCA:
```

The PCA constants have been regenerated using the new ECA list. The default generation is with validation turned on, but bad channels not set to -9999:

```
  set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
  set bank tcal 1 word 15 to 0  (validation type: generate status words only)
  set bank tcal 1 word 16 to 0  (CALDAC type: generate status only)
```

The PCA has its own set of tests to determine if a channel is suspect. This is also packed into the PMT bank and made available to analysis. One noteworth and substantial PCA cut is that for cells with low Qhs near pedestal, the TAC is set to -9999, indicating failed PCA walk calibration. As much as 15% of hits in a neutrino run fail this cut, and are due mostly to pickup hits. However, the ECA calibrated data is still available in the following new PMT bank words, this allows the analysis to proceed without walk corrections if desired:

```
        RQ(LPMT+KPMT_ELX)=      q_lx
        RQ(LPMT+KPMT_EHL)=      q_hl
        RQ(LPMT+KPMT_EHS)=      q_hs
        RQ(LPMT+KPMT_EPT)=      tim
```

2) Data cleaning:

The situation here is the same as for PCA, e.g.

```
  set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
  set bank tcal 1 word 15 to 0  (validation type: generate status words only)
  set bank tcal 1 word 16 to 0  (CALDAC type: generate status only)
```

This is probably the safest thing to do, as we do not want to increase contamination by turning off channels. It is not clear exactly what the direction of the effects are, e.g. improved/worse rejection. However, There is some indication that ECA validation does have an effect on some data cleaning cuts, and should be investigated in the future.

3) Reconstruction:

For reconstruction, a systematic study of bad ECA channels, and nonlinear ADC's should be performed.

a) For starters, all final reconstruction (published) should be performed with the following minimum ECA validation checks:

```
  set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
  set bank tcal 1 word 15 to 1  (val type: set pathological cells to -9999)
  set bank tcal 1 word 16 to 0  (CALDAC type: generate status only)
```

This does a minimum of checking and sets calibrated cells to -9999 that fail a number of serious ECA validation tests. The ECA constants for these cells are considered very bad and should not be used for reconstruction (see ECA doc for explaination of pathological channels). These include cells with flat TAC slopes, fail multiple validation tests, etc.

b) The first systematic check is to repeat reconstruction studies of neutrino and calibrations data with the following validation flags set:

```
  set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
  set bank tcal 1 word 15 to 2  (val type: set bad cells to -9999)
  set bank tcal 1 word 16 to 0  (CALDAC type: generate status only)
```

This sets calibrated cells to -9999 that fail any ECA validation test (see ECA doc for details of cuts, to summarize diffs of 1/4pe for Qhs and 1.5 nsec for TAC). In this mode anywhere from 1% to 5% of cells will fail and not be included in the fit (assuming that the path fitter does not include time channels with -9999). A check should then be made to see if there are any significant systematic shifts in reconstructed positions, widths, etc.

c) If time permits repeat 3 b) but with:

```
  set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
  set bank tcal 1 word 15 to 3  (val type: set bad cells to -9999)
  set bank tcal 1 word 16 to 0  (CALDAC type: generate status only)
```

This is the same a 3 b), plus, if Qhs fails the ECA validation, then the associated TAC values is also set to -9999. If we have a bad Qhs measurement, then the TAC will be suspect since the walk correction will be wrong. This is equivalent to applying ECA validation checks to the PCA directly, which is not done (see above).

d) check for effects of bad ADC's on reconstruction:

```
set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
set bank tcal 1 word 15 to 0  (val type: generate status words only)
set bank tcal 1 word 16 to 1  (CALDAC type: set bad ADC cells to -9999)
```

For any slot with a flagged ADC, all channels/cells for the appropriate ADC is set to -9999. This affects the data up to August 8/01, after which a hardware fix has eliminated this problem.

This is an important/large systematic that needs to checked, and can affect as much as 10% of the channels, and possibly in concentrated parts of the detector.

4) Intime Energy (RSP):
   ----

a) As in 3 a) above, all energy reconstruction should be done with a minimum of:

```
set bank tcal 1 word  9 to 4  (ECA type: validation and interp TSlope)
set bank tcal 1 word 15 to 1  (val type: set pathological cells to -9999)
set bank tcal 1 word 16 to 0  (CALDAC type: generate status only)
```

to eliminate pathologically bad channels.

b) Since the RSP relies on a tight intime window, having a bad TAC measurement can push an inappropriate hit into or out of the window. The effects of this is not clear, and requires some investigation of the sensitivity of the intime NHIT to bad channels. The ECA validation tests above can be applied, e.g. 3 b,c, and d. A better technique would be to apply a check mask directly to each cell ECA validation word to decide if the tube hit should be included in the sum, as well as included in the total live cell sum. A suggested mask of 1638606 (digital) applied to the ECA status word in Table 5.2 should be used. This mask excludes the CALDAC and Frati tests. As of SNOMAN 4_0185,

```
the check mask has been included (see RSP documentation, and
energy_control.dat).

Another point is that the TAC difference/width cut is only 1.5nsec,
which is much smaller than the RSP window, e.g. a tube that wobbles by
only 1.5nsec probably will not jump out of the time box.  An option
for future RSP analysis might be to rerun the ECA validation but with
a bigger TAC cut on the order of the time window itself, say 5 to 7
nsec.  This would then flag channels that are more likely to miss the
intime window and would also cut down on the number of bad ECA
channels, flagging only the real bad ones.  A similar increase would
be done for Qhs, say about 1pe.  A set of regenerated ECA constants
with looser validation as described above has been done, and are
available in /bigdisk/lanl/vdwater/eca/gen\_2.
```

6.4 Effects of ECA Validation on Reconstruction and Energy

A full report on this topic can be found in [5]. The conclusion reached is that ECA validation (including CALDAC) does have a significant effect on RSP energy scale, up to about 1%, and must be included in the neutrino analysis errors. The effects on reconstruction are much less significant.

Appendix A

ECA Control File

The basic specification for the calibration to be performed is given by the CECA bank, in the titles file **eca_control.dat** . This bank dictates the type of calibration (charge slopes, time slopes or pedestals), starting and ending crate numbers to use, tolerance limits denoting "bad" and "questionable" calibration results, the nature of the output summary file and whether to make use of pedestal information when fitting the charge slopes. Symbolic names have been set up for most of these choices as follows:

$qslope	do charge slope calibration
$tslope	do time slope calibration
$pedestals	do pedestal calibration
$firstcrate xx	start with crate xx
$lastcrate yy	finish with crate yy
$eca_short_summary	produce shortened summary
$qslope_free	don't constrain qslopes by pedestals

```
*LOG
*---  Standard Titles File: eca_control.dat              31 Aug 2000
*---                  Banks: CECA
*LOG OFF
*-----------------------------------------------------------------------------
*---
*---    Calibration Control of Electronics Slopes and Pedestal
*---
*---    Contact:  R. Van de Water   UPenn
*---
*-----------------------------------------------------------------------------
*---
*---
*DO CECA 1 -i(30I 3I 16F 2I 2F 2I 2F 2I 2F 2I 2F 5I 9F 6I 2F 9I 16F 11I 9F 2I)
                                                     #. Electronics Calibration
```

```
#.
#.      Standard Database Header
#.
19750101        0 20380516   223319 #.  1..4   Intrinsic validity
      0        0        0           #.  5..7   Data type, Task type, Format no.
      0        0        0           #.  8..10  Creation Date, Time, Source Id.
19750101        0 20380516   223319 #. 11..14  Effective validity
      0        0                    #. 15..16  Entry Date Time
4*0                                 #. 17..20  Spare
10*0                                #. 21..30  Temporary data (not in database)
#.
#.   End of Standard Database Header
#.
#.   General Set-up parameters
 2       #. 1 = Q Slope, 2 = T Slope, 3 = Pedestal
 0       #. First and
 18      #.   Last Crate Number (0 to 18)
#.
#.   Tolerances on Q/T Slope medians (step by step)
#.   and linear charge slope fits
 6.0     #. Tolerance: Num sigmas of ABSOLUTE Q/T values from overall avg
         #.            to determine whether to tag channel as "questionable"
 3.0     #. Tolerance: Num sigmas of RELATIVE Q/T values from overall avg
         #.            to determine whether to tag channel as "questionable"
 0.      #. Spare.
 0.      #. Spare.
 1.      #. Tolerance: Min &
 100.    #.                  Max slopes determining "bad" Q_HS Slope
 1.      #. Tolerance: Min &
 100.    #.                  Max slopes determining "bad" Q_HL Slope
 1.      #. Tolerance: Min &
 100.    #.                  Max slopes determining "bad" Q_LX Slope
#.
#.   More general parameters
 5.      #. DAQ to pe conversion constant
 2.5     #. DAQ to pe conversion constant
 18.35   #. DAQ to ns Conversion constant
 255.    #. DAQ to ns Conversion constant
 10.     #. DAQ to ns Conversion constant
 0.1     #. DAQ to ns Conversion constant
 2       #. QSLP - (1) Short or (2) Long summary output distributions
 2       #. QSLP - (1) Normal linear fit or (2) Force fit through pedestal
#.
#.   Pedestal Validation Parameters
 300.    #. QHS Tolerances: lower median
 900.    #.                 upper median
 8       #.                 width
 8       #.                 abs. diff
 300.    #. QHL Tolerances: lower median
 900.    #.                 upper median
 16       #.                 width
```

```
 16       #.                 abs. diff
 300.    #. QLX Tolerances: lower median
 900.    #.                upper median
 16       #.                 width
 16       #.                 abs. diff
 1350.   #. TAC Tolerances: lower median
 2400.   #.                upper median
 13       #.                 width
 13       #.                 abs. diff
 2       #. (1) Do or (2) Don't produce CMOS droop histograms
#.
#.    Pedestal/TSlope Common Parameters
 2       #. (1) Do or (2) Don't dump all pedestal/tslope events to log file,
 2       #.   and (1) Sorted or (2) Unsorted
#.
#.    TSlope Validation Parameters, for 31 injection points
-0.00004 #. Tolerance: Min.,
 0.00004 #.            Max.,
 0.00005 #.          & Abs. Max. Diff  for x-cubed coeff (cubic fit)
-0.02    #. Tolerance: Min.,
 0.02    #.            Max.,
 0.03    #.          & Abs. Max. Diff for x-squared coeff (cubic fit)
 5.      #. Tolerance: Min.,
 13.     #.            Max.,
 2.      #.          & Max_Diff for slope coeff (cubic fit)
 18      #. Min. acceptable no. of TSteps used in cubic fit
 5       #. Tolerance on step-by-step width check
 5       #. Min. no. of bad widths in a given cell for cell to be flagged
 6       #. Min. acceptable no. of events returned per TStep
 8       #. Min. no of TSteps which failed nevents check for cell to be flagged
 9       #. Min. no. of suspicious TSteps for cell to be flagged
 10000.  #. Tolerance: ChiSq tolerance
 1.5     #.            and abs. tolerance on point-by-point residual
 5       #. Min. number of bad residuals for cell to be flagged (TAC jump test)
 6       #. Min acceptable no. of events returned per TStep for that TStep
         #.   to be used in fit
#.
#.    Pedestal Overall Tolerances
 1       #. Max. acceptable no. chs that failed 0ev test
 32      #. Max. acceptable no. chs that failed lt160 ev test
 1       #. Max. acceptable no. chs that failed board ID check
 256     #. Max. acceptable no. flagged chs.: QHS
 256     #.                                  QHL
 256     #.                                  QLX
 256     #.                                  TAC
 530.    #. Min.
 630.    #.      and Max. acceptable 1/2-crate average on QHS pedestals
 530.    #. Min.
 630.    #.      and Max. acceptable 1/2-crate average on QHL pedestals
 530.    #. Min.
 630.    #.      and Max. acceptable 1/2-crate average on QLX pedestals
```

```
 1825.   #. Min.
 1925.   #.      and Max. acceptable 1/2-crate average on TAC pedestals
 3.      #. Max. acceptable 1/2-crate average on widths: QHS
 3.      #.                                              QHL
 3.      #.                                              QLX
 3.      #.                                              TAC
 3.      #. Max. acceptable 1/2-crate average on absolute diffs: QHS
 3.      #.                                                      QHL
 3.      #.                                                      QLX
 3.      #.                                                      TAC
 500     #. Max. acceptable total flagged channels
#.
#.    TSlope Overall Tolerances
 1       #. Max. acceptable no. chs that failed board ID checks
 1       #. Max. acceptable no. chs that had 0ev
 50      #. Max. acceptable no. chs that failed no. events per point test
 50      #. Max. acceptable no. chs that failed width
 50      #. Max. acceptable no. chs that failed no. points in fit test
 50      #. Max. acceptable no. chs that failed no. suspicious points test
 25      #. Max. acceptable no. chs that failed TAC jump test
 50      #. Max. acceptable no. flagged cubic
 50      #.                             quadratic
 50      #.                             linear fit parameters
-0.00005 #. Min. and
 0.00005 #.          Max. acceptable 1/2-crate average on cubic coeff
-0.02    #. Min.
 0.02    #.      and Max. acceptable 1/2-crate average on quadratic coeff
 5.      #. Min.
 13.     #.      and Max. acceptable 1/2-crate average on linear coeff
 0.00005 #. Max. acceptable 1/2-crate avg on abs. diffs for cubic
 0.03    #.                                                 quadratic
 2.      #.                                                 linear coeff
 250     #. Max. acceptable total flagged channels
#.
#.
 0       #. Perform ECA difference test (0= yes, 1= no)
#.          Can also disable difference test by setting run UC bit
```

Appendix B

Old style Output File Formats

B.1 Pedestal Titles Files

These files have the naming convention *peca_yyyymmdd_vv.dat* , where "yyyymmdd" is the date and "vv" is the effective version number.

The bank number specifies the 1/2 crate to which the calibrations refer. These numbers run from 1-18, so bank numbers 1 and 2 refer to the first and second half of crate 0.

The offset to the main link for a particular calibration value is given by,

$$OFFSET = 5*[(ice-1)*(nchannels*ncards/2)+(ich-1)*(ncards/2)+(jca-1)]+1+N \tag{B.1}$$

where "ice" is the cell index (1 to ncells), "ich" is the channel index (1 to nchannels), "jca" is the card index for this 1/2 crate (1 to ncards/2), and "N" refers to either QHS,QHL,QLX or T (thus running from 1 to 3). The word at location N=0 is a bit-mask flag for the following 4 numbers to indicate any values deemed "bad".

```
Data Words
----------

    Data types: B(bit), I, F, D(double), H(hollerith 4 char)

   +1   I   Bit-masked status word for values of 1st cccc (defined by OFFSET)
   +2   I   QHS pedestal (in ADC counts) for 1st cccc (defined by OFFSET)
```

```
   +3   I   QHL pedestal (in ADC counts) for 1st cccc (defined by OFFSET)
   +4   I   QLX pedestal (in ADC counts) for 1st cccc (defined by OFFSET)
   +5   I   T pedestal (in TDC counts) for 1st cccc (defined by OFFSET)

   +6   I   Bit-masked status word for values of 2nd cccc (defined by OFFSET)
   +7   I   QHS pedestal (in ADC counts) for 2nd cccc (defined by OFFSET)
   +8   I   QHL pedestal (in ADC counts) for 2nd cccc (defined by OFFSET)
   +9   I   QLX pedestal (in ADC counts) for 2nd cccc (defined by OFFSET)
   +10  I   T pedestal (in TDC counts) for 2nd cccc (defined by OFFSET)
     .
     .
     .
     .
```

B.2 QSlope Titles Files

These files have the naming convention *qeca_yyyymmdd_vv.dat* , where "yyyymmdd" is the date and "vv" is the effective version number.

The bank number specifies the 1/2 crate to which the calibrations refer. These numbers run from 1-18, so bank numbers 1 and 2 refer to the first and second half of crate 0.

The offset to the main link for a particular calibration value is given by

$$OFFSET = 6*[(ice-1)*(nchannels*ncards/2)+(ich-1)*(ncards/2)+(jca-1)]+2*(N-1) \tag{B.2}$$

where "ice" is the cell index (1 to ncells), "ich" is the channel index (1 to nchannels), "jca" is the card index for this 1/2 crate (1 to ncards/2), and "N" refers to either QHS,QHL,QLX or T (thus running from 1 to 3). The word at location N=0 is a bit-mask flag for the following 4 numbers to indicate any values deemed "bad".

```
Data Words
----------

    Data types: B(bit), I, F, D(double), H(hollerith 4 char)

   +1   I   linear fit QHS slope times 100 for 1st cccc (defined by OFFSET)
   +2   I   linear fit QHS intercept times 10 for 1st cccc (defined by OFFSET)
   +3   I   linear fit QHL slope times 100 for 1st cccc (defined by OFFSET)
```

```
+4   I   linear fit QHL intercept times 10 for 1st cccc (defined by OFFSET)
+5   I   linear fit QLX slope times 100 for 1st cccc (defined by OFFSET)
+6   I   linear fit QLX intercept times 10 for 1st cccc (defined by OFFSET)

+7   I   linear fit QHS slope times 100 for 2nd cccc (defined by OFFSET)
+8   I   linear fit QHS intercept times 10 for 2nd cccc (defined by OFFSET)
+9   I   linear fit QHL slope times 100 for 2nd cccc (defined by OFFSET)
+10  I   linear fit QHL intercept times 10 for 2nd cccc (defined by OFFSET)
+11  I   linear fit QLX slope times 100 for 2nd cccc (defined by OFFSET)
+12  I   linear fit QLX intercept times 10 for 2nd cccc (defined by OFFSET)
  .
  .
  .
  .
```

B.3 TSlope Titles Files

These files have the naming convention *teca_yyyymmdd_vv.dat* , where "yyyymmdd" is the date and "vv" is the effective version number.

The bank number specifies the 1/2 crate to which the calibrations refer. These numbers run from 1-18, so bank numbers 1 and 2 refer to the first and second half of crate 0.

The 1st data word in the bank specifies the number of time samples (call this "NT") used in the TSlope calibration (a maximum of 31 are allowed). The calibration parameters themselves are stored in a packed format. The 2nd data word therefore specifies how many of these packed words (call this "NP") are associated with each channel. There then follows the list of the NT time samples, in units of ns*100. After this, come the packed calibration values: the 1st packed word in a given set (i.e. for a given cccc) consists of bit-flags (31 + 1 extra which is needed to get around a zebra problem), one for each time sample, which are set if the digitization value corresponding to a given time sample is deemed to be bad. The remainder of the words in the set are 12-bit parameters packed into 32-bit words. The first of these are the digitization values for each time sample. These are used as a basis for interpolation when applying the calibration to raw data. Following this are 4 cubic-fit parameters, which may also be used as an alternative, or in conjunction with, the interpolation method. All the packed words were formed using the BUNCH routine in CERNLIB and are expanded again in the calibration routine with BLOW.

The offset to the main link for a particular set of calibration values is

$OFFSET = 2 + N\_Steps + N_Words * [(ice - 1) * (nchannels * ncards/2) + (ich - 1) * (ncards/2) + (jca - 1)]$

where "ice" is the cell index (1 to nells), "ich" is the channel index (1 to nchannels), "jca" is the card index for this 1/2 crate (1 to ncards/2), and "N" refers to either QHS,QHL or QLX charges (thus running from 1 to 3).

```
Data Words
----------

    Data types: B(bit), I, F, D(double), H(hollerith 4 char)

 +1           I  Number of time samplings (call this "NT")
 +2           I  Number of  packed words/channel (call this "NP")
 +3           I  1st injected time (ns*100) for 1/2 Crate
 +4           I  2nd injected time (ns*100) for 1/2 Crate
  .
  .
 +2+NT        I  NTth injected time (ns*100) for 1/2 Crate

 +2+NT+1      I  (Packed) Bit-mask flag for any values deemed "bad" in 2nd cccc
 +2+NT+2      I  2nd packed tslope word (see notes above) of 1st cccc
 +2+NT+3      I  3rd packed tslope word (see notes above) of 1st cccc
  .
  .
 +2+NT+NP+1   I  NPth packed tslope word (see notes above) of 1st cccc

 +2+NT+NP+1+1 I  (Packed) Bit-mask flag for any values deemed "bad" in 2nd cccc
 +2+NT+NP+1+2 I  2nd packed tslope word (see notes above) of 2nd cccc
 +2+NT+NP+1+3 I  3rd packed tslope word (see notes above) of 2nd cccc
  .
  .
 +2+NT+2NP+1  I  NPth packed tslope word (see notes above) of 2nd cccc
  .
  .
  .
  .
```

B.4 Summary Histograms/NTuples

These files are of the form *pdst_yyyymmdd_vv.ntp, qslp_yyyymmdd_vv.ntp* and *tslp_yyyymmdd_vv.ntp* for the 3 different calibrations, respectively. Each contains a set of summary histogram which show the total number of "good", "suspicious" and "bad" calibrations in each 1/2 crate. "Bad" values have also been flagged in the titles files and by default will not be used in application of the calibration constants. "Suspicious" values are for diagnostic purposes only and are defined by comparing a given parameter value with a running average of typical values accumulated during the ECA run. as previously described. For QSLP, histograms of average QSlope values are also given for all 3 charges and, for TSLP, a histogram of average TSlope values (linear fit) is also shown.

If a "long summary" has been requested in the CECA bank, then a much more detailed ntuple is also included, containing information on individual channels. Information has been packing into variables as much as possible to reduce the size, but these files can still be quite large and may take a while to interact with via PAW. Nevertheless, it affords one the opportunity to examine exactly why a given channel may have been failed by the software. To make it somewhat easier to use the ntuple, first execute **eca_qtslope.kumac** which will define the variables card, crate, chan, cell, qtyp (for choosing QHS, QHL or QLX) and qslope.

B.4.1 PDST

The ntuple simply contains the charge and time pedestals for each CCCC. Negative values indicate pedestals which are considered "suspicious" (the true value = the magnitude). There is currently no criterion for establishing a "bad" pedestal value, aside from one which is listed as zero.

B.4.2 TSLP

The ntuple actually contains the injected (TINP)and digitized (TDIG) values used to obtain the calibration, along with an average linear-fit slope (SLP), for each CCCC. A negative TINP indicates a "suspicious" values and a negative TDIG indicates a value deemed to be "bad" (remember, only "bad" values are tagged in the actual titles file, "suspicious" values are only for diagnostics). As before, the true values of TINP and TDIG are given by their absolute values.

B.4.3 QSLP

The ntuple contains similar information the the TSLP ntuple, but the injected and digitized values (QINP and QDIG) are given for each of the 3 different charges. This can be selected by using qtyp=1 for QHS, qtyp=2 for QHL, or qtyp=3 for QLX. The slope must be examined here through the variable qslope defined by **eca_qtslope.kumac**, since the SLPQ parameter in the ntuple has been used to pack additional information as well.

Appendix C

Validation Code Changes

The following subroutines were added to SNOMAN:

digtoch_10	Modification of digtoch_8.for to allow up to 10 characters
eca_pdiagnostic	Main validation routine for Pedestals
eca_peds_out	Writes Pedestal values out to titles file
eca_pt_fulldump	Dumps raw ECA pedestal/tslope data to log file
eca_pt_halfcratesum_out	Writes validation test failure totals to log file
eca_pt_histo_out	Outputs various values to HBOOK histograms
eca_pt_ids_out	Writes board identification (ID) numbers out to titles file
eca_pt_out	Main output routine for Pedestals and Time Slopes (altered version of old eca_qt_out)
eca_pt_summary_out	Writes summary of validation test results out to log file
eca_qslp_out	Main output routine for Charge Slopes (altered version of old eca_qt_out)
eca_tdiagnostic	Main validation routine for Time Slopes
linear_fit	Performs a linear fit on provided points
poly_solve	Finds a root for a given polynomial, given an initial guess

The following subroutines already existed in SNOMAN but were modified in some way:

eca_ini	eca_electronics_calib
eca_qt_filter	eca_qt_fit
eca_qt_med	eca_qt_out_init
eca_stop	eca_trm
median	

The following `.inc` files were added:

eca_check_vars.inc	Contains variables for keeping track of total test failures
eca_dqxx_access	Contains link variables for access DQxx banks

The following `.inc` files already existed but were modified in some way:

eca_com.inc	eca_qt_output.inc
eca_var.inc	

Appendix D

ECA Run Numbers and Dates

The tex file for the data below can be found on **surf** in the directory /bigdisk/penn/vdwater/snoman/eca/00run_dates.tex.

```
ECA RUN NUMBERS AND DATES:
*************************

Last Updated: Mar 25/01

Good ECA runs used for analysis:
********************************

PEDS             TSLOPES
Run#    Date     Run#    Date
----    ----     ----    ----

15402   20010326   15401   20010326
15382   20010324   -----   -------- ->TAC slope missing from crates 8-18
15338   20010316   15339   20010316 ->no diff, crate 12/13 low
                                      volts supply swapped.
15306   20010309   15307   20010309 ->no diff, change in charge pedestals

**Gain op-amp resistor change

15277   20010305   15278   20010305
15244   20010228   15245   20010228
15163   20010222   15164   20011022
15127   20010219   15128   20010219
15081   20010215   15082   20010215
15049   20010212   15050   20010212
15040   20010210   15041   20010210
15023   20010205   15024   20010205
-----   --------   15017   20010202 ->WEB druing PED run, orphan spew during TAC
15009   20010131   15010   20010131
14959   20010126   14960   20010126
14813   20010122   14810   20010122 ->first PED run botched (TAC after PED run).
14779   20010119   14780   20010119
14748   20010116   14749   20010116
14682   20010113   14683   20010114
14534   20010109   14535   20010109
14436   20010103   14337   20010103
```

```
14423   20010101   14424   20010101
14406   20001228   14407   20001228 ->TAC 31 points non variable
14395   20001225   14396   20001225 ->TAC 31 points non variable
14383   20001221   14384   20001221
14338   20001219   14339   20001219
14309   20001215   14310   20001215 ->All okay now
14302   20001214   14303   20001214 ->TAC run prematurely ends at crate 8
14285   20001211   14286   20001211
14261   20001207   14262   20001207
14233   20001204   14234   20001204 ->no diff for pedestals only

**Crate fan upgrade, board temps lower by ~5C
**new VBAL skey= 621 due to lower temps.

14191   20001127   14192   20001127
14172   20001123   14173   20001123
14081   20001120   14082   20001120
14052   20001117   14053   20001117 ->network problems, TCMOS twiddle bits ON
13988   20001112   13989   20001112 ->GPS not working
13970   20001110   13971   20001110
13893   20001107   13894   20001107
13872   20001103   13873   20001103
13772   20001030   13773   20001030
13739   20001027   13740   20001027
13727   20001024   13728   20001024
13564   20001020   13565   20001020 ->TCMOS twiddle bits ON, no diff
13445   20001016   13443   20001016
13429   20001012   13430   20001012
13424   20001009   13425   20001019
13419   20001005   13420   20001005  ->new variable TAC time ramp

**All runs from this date to present use new variable TAC time ramp, unless
explicitly stated.
**All runs from this date to present should have pedestal width= 1000nsec

13403   20001002   -----   --------  ->TAC test run
13390   20000928   13391   20000928  ->TAC 31 points, ped width= 750nsec
13342   20000925   13343   20000925  ->TAC 31 points, ped width= 750nsec
13332   20000921   13333   20000921  ->TAC 31 points, ped width= 750nsec
13152   20000915   13153   20000915  ->TAC 31 points, ped width= 750nsec
12978   20000912   12980   20000912  ->TAC 31 points, ped width= 750nsec
12794   20000908   12795   20000908  ->TAC 31 points, ped width= 750nsec
12620   20000905   12621   20000905  ->TAC 31 points, ped width= 750nsec
12616   20000902   12617   20000902  ->TAC 31 points, ped width= 750nsec
12612   20000831   12613   20000831  ->TAC 31 points, ped width= 750nsec
12579   20000828   12580   20000828  ->TAC 31 points, ped width= 750nsec
12573   20000825   12574   20000825  ->TAC 31 points, ped width= 750nsec, no diff

**Runs 12558 to 12469 not used due to problems.  Validity date on 12573/12574
changed to 20000817 to cover range.

**All runs from this date to present have TCMOS skey= 611 (twiddle bits OFF),
unless explicitly stated.  All runs before this date to the start of
production (Nov 2/99) have TCMOS skey= 601 (twiddle bits ON).

12425   20000814   12426   20000814  ->TAC 31 points, ped width= 500nsec, no diff

**LED mod to ADCs from Aug 8 to 14/2000.  Should not use neutrino data in this period.
**Note, all runs before this date have number of TAC points = 19 (dt= 20 nsec)

12340   20000807   12339   20000807  -> TAC 19 points, ped width= 500nsec
12314   20000803   12315   20000803
12276   20000731   12277   20000731
12248   20000727   12249   20000727
```

```
12235   20000724   12236   20000724
12230   20000720   12231   20000720
12220   20000717   12221   20000717
12208   20000713   12209   20000713
12198   20000710   12199   20000710
12185   20000706   12186   20000706
12179   20000704   12180   20000704
12163   20000629   12164   20000629
12052   20000624   12053   20000624
11979   20000619   11980   20000619
11968   20000615   11969   20000615
11921   20000608   11923   20000608
11912   20000605   11913   20000605
11909   20000601   11910   20000601
11885   20000530   11886   20000530
11832   20000525   11833   20000525
11806   20000520   11807   20000520
11791   20000518   11792   20000518
11757   20000515   11758   20000515
11740   20000512   11741   20000512
11728   20000511   11729   20000511 ->Crate 2 and 3 VCC change, no diff
11699   20000504   11700   20000504
11672   20000427   11673   20000427
11645   20000424   -----   -------- ->tac run missing crates 17-18
11594   20000420   11595   20000420
11581   20000418   no TAC  -------- ->ped width = 500ns
11566   20000413   11567   20000413 ->ped width = 500ns up to run 12468
11555   20000411   11556   20000411 ->all runs before this have ped width= 100nsec
11526   20000404   11527   20000404
11491   20000330   xxxxx   -------- ->tac run missing crates 4-12
11450   20000323   11452   20000323
11434   20000316   11435   20000316
11408   20000309   11412   20000309
11394   20000302   -----   -------- ->tac run missing crates 11-18
11369   20000225   11370   20000225
11307   20000218   11308   20000218
11282   20000214   11283   20000214
10973   20000204   10974   20000204
10940   20000128   10941   20000128
10912   20000125   10914   20000125
10834   20000115   10835   20000115
10822   20000113   10823   20000113
10798   20000107   -----   -------- ->TAC run ended prematurely, breakdown
10778   20000103   -----   -------- ->Charge slope run
10760   19991230   10761   19991230
10721   19991222   10722   19991222
10702   19991217   10703   19991217
10653   19991210   10654   19991210
10259   19991129   10261   19991129
-----   --------   10233   19991126 ->wrong PED-GT delay for pedestal run
10122   19991113   10123   19991113
10050   19991108   10051   19991108
10018   19991104   10019   19991104
05724   19991024   05725   19991024
05561   19991016   05562   19991016

Other problem or test ECA runs not used in regeneration:
*******************************************************

10207
10210
10229
10230
10231
```

```
10232 ->wrong PED-GT delay for pedestal run
10260
10262
10652
10720
10810
10799 ->flat tac, crate 1 HV tripped.
11395 ->tac run missing crates 11-18
11409
11410
11492 ->tac run missing crates 4-12
11564
11648 ->tac run missing crates 17-18
11671
11738
11823   20000523 ->ped timepulser=3Hz
11822   20000523 ->ped timepulser=1kHz
11821   20000523 ->ped timepulser=100Hz
11858   20000526
11856   20000526
11855   20000526
11854   20000526 ->peds analysis error, there is no year zero?!?
11853   20000526
11888   20000530 ->experimental peds with thresholds upped on all channels
11944   20000612 -> *EXPERIMAL* (pulser=1Hz)
11943   20000612 -> *EXPERIMENTAL* (#ev/cell=5, pulser=45 Hz)
11942   20000612 -> *EXPERIMENTAL* (#events/cell=5, pulser=100Hz)
11940   20000612 -> *EXPERIMENTAL* (high deck temp/humidity)
11941   20000612 -> *EXPERIMENTAL* (high deck temp/humidity)
11922
11939
11967
12001
12339
12341 ->TAC 31 points, ped width= 500nsec, test run
12537
12540
12543
12544
12546
12548
12558   20000823 ->TSlopes wouldn't run -> ped width= 750nsec
12532   20000822 ->Detector went down during TSlopes run ->ped width= 750nsec
12468   20000817 ->crates 14&15 offline, TAC 31 points, ped width= 500n
12469   20000817 ->crates 14&15 offline, TAC 31 points, ped width= 500n
12559
12560
12563
13397
13398
13399
13402
13404 ->TAC 19 points, ped width= 1000nsec, test run
13442
13520
13521
13870
14809 -> mistakenly started TAC run at end of this pedestal run
15016 -> WEB druing PED run
15037 -> fec fifo overflow
15038 -> fec fifo overflow
15080 -> 10 Mhz clock not working
15400 -> builder dies during ped run.
15383   20010324 ->TAC slope missing from crates 8-18
```

Appendix E

ECA Pedestal Summary Data

The following pedestal validation summary data is a result of the ECA regeneration of March, 2001. At the end of the appendix is a description of the table labels. The detailed validation summary logs for these runs can be found on **surf** in /bigdisk/penn/eca/generation/log. The perl script to extract the summary numbers in this appendix can be also found on **surf** in /bigdisk/penn/vdwater/snoman/eca_develop/eca_trends. This script can and should be used for future ECA trend analysis.

Runno	Status	Seq	MBid	DBid	160ev	0ev	QHS	QHL	QLX	TAC	TQHS	ErrTot	Cells	aQHS	aQHL	aQLX	aTAC
5561	536871936	9	0	0	8	0	55	12	5	12	65	91	229	593.1	589.3	576.7	1837.9
5724	492548	9	32	0	30	1	108	78	68	77	184	306	2818	593.1	589.5	576.8	1838.3
10018	1027	136	0	0	71	3	69	24	60	122	190	465	2728	593.0	589.3	576.8	1838.7
10050	263168	137	0	0	11	1	62	39	43	66	128	340	1556	593.0	589.3	576.9	1838.6
10122	394245	9	160	16	22	5	53	43	38	124	172	415	2225	593.6	589.3	577.3	1838.7
10259	67470341	10	128	128	32	2	103	35	62	59	158	394	1991	593.3	589.4	577.1	1839.1
10653	427009	42	0	0	4	2	121	60	140	103	223	429	3111	593.5	588.9	577.3	1838.9
10702	361477	10	64	0	13	2	74	145	118	48	118	435	3340	593.3	588.5	576.9	1838.5
10721	459780	18	0	8	7	1	79	77	163	84	158	378	2912	593.3	588.6	576.9	1837.9
10760	361473	18	0	0	6	2	120	84	185	108	218	436	4011	593.4	588.1	577.2	1837.7
10798	394243	50	0	0	33	3	134	71	120	157	284	482	3903	593.3	588.4	577.3	1838.2
10822	328705	50	0	0	15	2	133	75	164	127	236	469	3379	593.5	588.5	577.4	1838.0
10834	132101	50	32	0	11	3	131	55	158	74	203	458	2536	593.6	588.6	577.3	1838.2
10912	1016837	19	96	104	31	5	202	100	233	188	360	733	7204	593.8	588.6	576.9	1834.8
10940	492545	44	0	0	16	4	112	49	125	194	272	462	4474	593.3	588.5	576.3	1834.8
10973	951299	44	0	0	36	2	70	63	216	237	301	602	5926	593.3	588.3	576.9	1838.3
11282	1016835	44	0	0	92	3	110	97	225	164	251	643	5160	593.5	588.5	577.0	1838.3
11307	67601413	12	32	32	16	3	83	78	210	95	176	470	4417	593.1	588.6	575.8	1838.5
11369	493059	12	0	0	47	4	85	47	104	109	189	354	3691	593.2	588.5	574.7	1837.7
11394	500739	12	0	0	121	3	93	65	121	103	193	453	3618	593.4	588.3	576.4	1837.3
11408	500737	12	0	0	6	4	115	51	122	114	224	378	3575	593.9	588.4	575.9	1837.4
11434	1025025	44	0	0	7	4	206	93	185	178	377	645	6007	593.6	587.7	576.3	1835.7
11450	1017379	12	0	0	43	3	106	55	280	195	300	645	7828	593.0	588.1	572.3	1836.8
11491	493059	44	0	0	48	3	67	59	122	182	245	479	5073	593.3	587.9	574.0	1835.8
11526	919043	42	0	0	69	4	76	68	131	210	280	547	5580	593.7	587.5	572.7	1835.2
11555	1017603	42	0	0	34	4	126	98	179	79	200	512	5169	593.2	586.2	571.6	1835.7
11566	1017601	42	0	0	28	6	82	172	164	164	241	583	6849	593.2	585.5	570.5	1836.3
11581	1017601	75	0	0	29	6	81	112	177	209	286	608	6124	593.6	585.1	571.7	1829.7
11594	68027909	120	32	24	31	2	79	88	189	215	293	696	6646	593.5	587.2	570.8	1836.0

11645	1017415	120	32	0	47	3	124	132	139	466	575	937	10936	593.5	587.2	571.1	1830.0
11672	1017345	184	0	0	32	3	87	68	120	214	298	662	5496	593.3	586.7	572.0	1829.2
11699	1017409	248	0	0	29	6	111	79	145	354	461	917	8950	593.8	587.0	568.6	1835.4
11728	537396867	161	0	0	220	5	74	48	151	58	131	689	4295	593.5	588.5	570.4	1836.3
11740	951815	96	128	8	57	5	139	79	160	128	260	756	6288	595.4	588.7	571.8	1842.6
11757	1017347	97	0	0	60	5	82	69	162	178	258	613	5781	595.5	588.1	571.1	1841.6
11791	984646	24	64	40	58	0	72	144	148	301	371	714	7696	595.2	588.2	571.3	1841.5
11806	493056	55	0	0	15	0	76	78	147	46	121	390	3960	595.3	588.9	571.1	1842.3
11832	1017346	87	0	0	82	0	87	111	156	202	285	688	6212	594.9	587.6	572.0	1841.1
11885	1017350	119	0	8	87	0	81	106	176	165	243	700	6260	595.7	588.3	571.0	1841.4
11909	1021442	119	0	0	99	0	83	71	157	67	150	570	4339	595.9	588.8	571.3	1841.7
11912	886530	119	0	0	77	0	89	116	173	82	171	631	5207	595.9	586.6	571.2	1841.4
11921	1017600	87	0	0	21	0	90	113	199	83	173	565	5409	596.0	586.4	571.4	1841.3
11968	1017602	119	0	0	34	0	133	157	213	226	355	844	8502	595.5	586.5	570.7	1841.3
11979	984832	119	0	0	22	0	128	149	213	223	349	815	8400	595.5	586.1	571.2	1841.0
12052	1017604	55	32	24	32	0	109	180	248	187	291	797	8116	594.9	586.5	570.6	1840.1
12163	1017605	87	0	8	29	2	119	138	215	159	273	703	7536	592.7	586.3	570.4	1840.6
12179	1025792	119	0	0	17	0	156	151	186	142	294	718	7399	593.0	586.7	570.4	1840.1
12185	886528	119	0	0	22	0	133	122	167	77	209	612	6012	593.5	586.7	570.5	1840.4
12198	1017600	119	0	0	24	0	138	171	206	195	332	792	8019	592.8	586.9	570.4	1840.3
12208	886528	119	0	0	11	1	122	144	162	123	244	639	6059	592.9	586.5	570.5	1839.9
12220	960322	119	0	0	42	1	187	197	196	403	584	1050	12114	592.9	586.5	570.8	1840.2
12230	960256	151	0	0	15	1	161	152	177	226	382	847	8356	593.4	585.6	571.0	1840.2
12235	960257	151	0	0	15	2	140	161	225	130	269	789	7435	593.5	586.0	570.8	1841.3
12248	1025794	183	0	0	109	1	169	145	242	115	283	898	7605	593.9	586.1	572.0	1841.1
12276	1017633	183	0	0	19	2	180	144	285	135	310	885	7785	593.7	586.0	571.3	1840.6
12314	1031973	246	32	0	20	3	167	162	260	200	361	962	7676	593.1	586.5	572.6	1841.1
12340	952163	215	0	0	40	2	138	175	299	348	484	1134	9987	593.5	586.1	572.5	1840.3
12425	603980803	96	0	0	58	3	52	58	21	22	73	294	1381	595.2	588.9	577.6	1841.3
12573	536870914	89	0	0	37	1	66	49	22	17	82	263	1475	596.0	587.7	577.3	1869.8
12579	819201	89	0	0	18	2	118	88	34	196	311	515	5388	596.2	587.8	577.4	1868.9
12612	327682	57	0	0	42	1	78	120	27	152	227	433	4615	596.4	587.3	577.7	1869.2
12616	294919	33	0	8	56	3	115	85	38	90	201	369	3842	596.1	587.5	577.7	1869.4
12620	67403777	33	0	0	24	2	106	90	32	178	276	431	4414	596.5	588.4	577.7	1870.8
12794	67403777	33	0	0	22	2	94	91	26	133	225	373	4163	595.8	587.8	578.0	1871.0
12978	3	33	0	0	56	2	107	78	43	55	161	349	3130	595.9	588.3	577.9	1871.2
13152	278530	33	0	0	66	1	85	94	37	118	199	375	3426	595.9	587.0	578.0	1870.3
13332	344066	33	0	0	47	1	83	120	46	114	192	387	3694	595.8	587.0	577.8	1870.3
13342	262145	64	0	0	24	2	78	69	44	105	180	345	3123	595.6	587.2	577.5	1870.6
13390	491521	33	0	0	22	2	75	92	30	116	189	329	3563	595.9	587.7	577.8	1870.5
13403	491520	33	0	0	13	1	92	94	33	96	181	330	3297	596.4	587.4	578.0	1869.4
13419	344065	65	0	0	12	2	85	79	27	133	213	373	3938	596.1	587.6	577.6	1870.3
13424	262147	65	0	0	34	2	81	73	35	106	184	361	3338	596.0	587.6	577.6	1870.7
13429	278529	65	0	0	25	6	93	72	28	124	214	366	3365	596.0	587.7	577.6	1870.3
13445	491523	97	0	0	51	2	94	103	30	103	193	439	3500	595.9	587.5	577.5	1870.1
13727	1015810	98	0	0	80	1	84	127	34	151	232	533	4931	595.9	587.3	577.4	1869.6
13739	1024005	74	0	24	26	2	110	137	44	211	314	549	5666	595.8	588.7	577.6	1871.3
13772	483330	76	0	0	87	1	92	92	51	127	212	464	3551	595.8	588.4	577.6	1870.7
13872	73731	76	0	0	48	2	81	82	39	62	138	349	2633	595.8	587.9	577.5	1870.7
13893	208896	75	0	0	11	0	73	100	40	36	105	284	2400	595.9	588.3	577.8	1870.7
13970	67575808	76	0	0	21	0	78	88	37	34	105	286	2318	596.0	588.3	577.8	1870.5
13988	68149250	108	0	0	54	0	127	134	53	158	273	541	4982	596.3	587.9	577.9	1869.3
14081	1024000	140	0	0	8	0	121	127	49	189	306	561	5300	595.9	588.8	577.2	1869.5
14172	466946	140	0	0	70	0	77	82	43	50	123	418	2394	595.9	589.0	577.4	1869.9
14191	466944	140	0	0	24	0	70	76	41	67	134	389	2892	595.8	588.9	577.3	1869.4
14233	536883202	69	0	0	75	0	79	67	23	17	95	304	1750	595.5	587.5	577.3	1872.1
14261	401410	70	0	0	35	0	106	88	44	117	220	411	3742	595.4	587.0	577.3	1871.6
14285	65536	70	0	0	29	0	92	94	36	48	138	336	2819	595.4	587.1	577.5	1871.7
14309	499712	70	0	0	16	0	82	91	54	67	148	321	3029	595.7	586.7	577.4	1872.2
14338	458754	71	0	0	49	0	84	109	44	27	109	323	2432	595.4	587.5	577.6	1873.2
14383	434176	71	0	0	29	0	97	98	50	62	156	351	2836	595.6	587.5	577.4	1872.9
14395	401410	71	0	0	46	0	105	79	52	36	139	343	2592	595.6	587.5	577.2	1873.2
14406	303104	103	0	0	13	0	97	97	51	30	124	339	2794	595.4	587.4	577.0	1873.1
14423	434178	135	0	0	64	0	91	102	34	23	111	401	2566	595.5	587.6	577.2	1873.4

14436	67600388	56	32	16	25	0	118	126	75	115	197	394	3459	595.7	587.4	577.5	1872.3
14534	67608576	56	0	0	16	0	90	90	48	101	190	346	3081	595.8	587.6	577.4	1872.0
14682	434176	88	0	0	27	1	82	96	35	65	144	341	2760	596.0	587.7	577.6	1872.2
14748	67510274	88	0	0	33	0	93	91	39	55	144	358	2657	596.3	587.4	577.8	1871.9
14779	67608576	88	0	0	13	0	84	100	53	108	188	383	3559	596.0	587.3	577.4	1871.9
14813	67608576	120	0	0	18	1	110	82	30	33	139	345	2752	594.5	587.4	577.6	1872.2
14959	270339	88	0	0	56	2	123	99	47	136	256	478	4200	594.7	587.3	577.8	1872.4
15009	172035	120	0	0	43	2	105	81	34	30	134	378	2801	596.4	587.1	577.7	1872.1
15023	237569	120	0	0	20	3	84	82	44	102	183	405	2948	596.0	587.3	577.7	1872.2
15040	68132865	184	0	0	22	2	123	129	55	133	253	587	4460	595.7	587.1	577.5	1872.4
15049	204801	184	0	0	13	2	84	102	49	122	204	495	3511	595.5	586.9	577.0	1872.5
15081	761859	192	0	0	56	2	107	91	39	71	176	502	2750	595.4	587.6	577.0	1873.1
15127	67610625	152	0	0	17	2	90	92	35	40	128	375	2472	595.6	587.7	577.0	1873.0
15163	491521	152	0	0	4	2	116	101	43	134	214	454	3924	595.3	587.5	577.0	1873.2
15244	1024003	152	0	0	105	2	133	113	36	135	265	608	4192	595.4	587.4	577.2	1873.0
15277	1024003	184	0	0	50	2	119	96	44	100	216	534	3615	595.3	587.8	577.1	1873.2
15306	536879104	43	0	0	18	0	88	82	25	16	103	233	1813	588.9	581.8	572.5	1874.8
15338	603987970	43	0	0	57	0	74	83	26	19	92	265	1857	589.8	582.6	573.5	1875.0
15382	501122	75	0	0	76	1	94	142	43	107	195	455	3641	591.3	584.0	575.4	1871.7
15402	501122	75	0	0	35	0	152	144	61	91	239	446	4177	590.5	583.0	574.5	1871.6

- **Runno**: run number.
- **Status**: ECA run status return code (base 10), see Table 3.2.
- **Seq**: number of channel sequencers disabled.
- **MBid**: number of Motherboards with differing ID's.
- **DBid**: number of Daughterboards with differing ID's.
- **160ev**: number of channels with less than 160 events.
- **0ev**: number of channels with zero events.
- **QHS**: number of channels failed Q_{hs} tests.
- **QHL**: number of channels failed Q_{hl} tests.
- **QLX**: number of channels failed Q_{lx} tests.
- **TAC**: number of channels failed TAC tests.
- **TQHS**: number of channels failed Q_{hs} or TAC tests.
- **ErrTot**: total number of failed channels (including Seq).
- **Cells**: total number of failed cells (including Seq).
- **aQHS**: detector averaged Q_{hs} pedestal.
- **aQHL**: detector averaged Q_{hl} pedestal.
- **aQLX**: detector averaged Q_{lx} pedestal.
- **aTAC**: detector averaged TAC pedestal.

Appendix F

ECA TAC Slope Summary Data

The following time slope validation summary data is a result of the ECA regeneration of March, 2001. At the end of the appendix is a description of the table labels. The detailed validation summary logs for these runs can be found on **surf** in /bigdisk/penn/eca/generation/log. The perl script to extract the summary numbers in this appendix can also be found on **surf** in /bigdisk/penn/vdwater/snoman/eca_develop/eca_trends. This script can and should be used for future ECA trend analysis.

Runno	Status	Seq	MBid	DBid	Oev	#ev	Width	Fit	Susp	Frati	Cub	Quad	Lin	ErrTot	Cells	aCub	aQuad	aLin
5562	536870914	9	0	0	2	2	12	1	13	12	2	2	8	31	295	-0.0000012	-0.00043	8.99
5725	1	9	32	0	1	1	5	1	5	9	1	1	43	87	232	-0.0000012	-0.00045	9.00
10019	514	136	0	0	2	6	6	2	8	6	2	2	52	193	288	-0.0000014	-0.00040	8.99
10051	512	137	0	0	0	8	12	3	15	18	3	2	59	206	319	-0.0000014	-0.00037	8.99
10123	66051	9	160	16	3	7	5	6	10	4	4	4	92	265	430	-0.0000017	-0.00023	8.96
10233	67109377	10	128	128	1	8	13	5	17	11	7	4	80	227	387	-0.0000014	-0.00036	8.98
10261	67108864	10	0	0	0	8	11	7	19	9	7	3	50	71	337	-0.0000014	-0.00036	8.99
10654	576	42	0	0	0	10	7	5	15	46	6	5	55	143	491	-0.0000013	-0.00036	8.99
10703	577	10	64	0	0	12	19	8	29	83	14	12	78	224	804	-0.0000015	-0.00028	8.97
10722	577	18	0	8	0	13	7	8	14	38	10	7	82	134	492	-0.0000013	-0.00035	8.98
10761	576	18	0	0	0	13	5	8	18	48	9	5	86	147	726	-0.0000015	-0.00032	8.98
10823	576	50	0	0	0	12	9	7	18	68	11	11	110	225	760	-0.0000015	-0.00032	8.98
10835	577	50	32	0	0	14	5	7	15	71	14	18	101	247	632	-0.0000016	-0.00031	8.98
10914	66113	20	96	104	0	13	6	7	14	54	31	26	151	288	1042	-0.0000016	-0.00027	8.96
10941	576	44	0	0	0	13	4	9	15	53	27	29	93	173	740	-0.0000017	-0.00027	8.97
10974	576	44	0	0	0	12	38	8	47	78	12	12	104	211	1249	-0.0000011	-0.00046	9.00
11283	576	44	0	0	0	11	38	7	46	85	13	11	129	247	1128	-0.0000014	-0.00038	8.99
11308	67109377	12	32	32	0	14	21	8	33	21	10	9	114	177	611	-0.0000013	-0.00044	9.00
11370	512	12	0	0	0	14	24	9	33	21	14	16	86	120	535	-0.0000014	-0.00040	9.00
11412	576	44	0	0	1	14	14	9	24	51	15	14	118	207	661	-0.0000015	-0.00037	8.99
11435	66112	44	0	0	0	13	38	10	47	82	13	15	140	251	1084	-0.0000015	-0.00037	8.98
11452	616	12	0	0	0	14	88	9	98	31	15	16	163	228	1330	-0.0000014	-0.00039	8.99
11527	608	42	0	0	0	17	48	10	59	40	19	33	134	192	964	-0.0000014	-0.00041	8.99
11556	544	42	0	0	0	18	42	10	53	14	16	18	142	196	1216	-0.0000014	-0.00038	8.98
11567	66080	42	0	0	0	49	15	42	61	21	20	18	226	319	1421	-0.0000016	-0.00030	8.98
11595	67174913	120	32	24	0	12	37	7	47	12	9	5	192	358	941	-0.0000013	-0.00042	8.99
11673	66113	184	32	0	0	14	6	29	16	45	42	32	104	311	974	-0.0000015	-0.00019	8.90
11700	66152	248	0	0	1	14	69	8	81	81	17	32	113	443	1583	-0.0000013	-0.00038	8.98
11729	536937056	161	0	0	0	18	42	37	54	72	40	39	54	310	1308	-0.0000012	-0.00040	8.98

11741	66145	96	128	8	1	20	41	10	52	48	13	15	131	414	851	-0.0000009	-0.00063	9.08
11758	66144	97	0	0	1	20	45	10	57	61	16	13	112	294	1213	-0.0000012	-0.00055	9.07
11792	66113	24	64	40	0	15	40	4	47	76	11	15	136	277	1180	-0.0000012	-0.00051	9.05
11807	576	55	0	0	0	12	9	5	17	50	6	7	107	203	688	-0.0000010	-0.00063	9.07
11833	67174976	87	0	0	0	16	39	5	47	101	38	38	143	294	1194	-0.0000012	-0.00052	9.05
11886	66113	119	0	8	0	13	28	4	44	92	21	6	147	337	1277	-0.0000011	-0.00060	9.06
11910	66112	119	0	0	0	14	8	4	15	98	7	9	112	316	928	-0.0000011	-0.00057	9.06
11913	66112	119	0	0	0	16	11	6	33	153	16	15	158	405	1167	-0.0000011	-0.00056	9.06
11923	66112	87	0	0	0	17	8	4	27	187	21	24	170	407	1557	-0.0000012	-0.00051	9.05
11969	66152	119	0	0	0	14	62	7	71	176	20	21	207	489	1867	-0.0000012	-0.00052	9.05
11980	66144	119	0	0	0	14	41	6	61	241	23	24	202	500	1956	-0.0000012	-0.00048	9.04
12053	66113	55	32	24	0	12	43	5	48	44	16	9	207	334	1247	-0.0000011	-0.00056	9.05
12164	66113	87	0	8	1	13	19	6	48	88	15	12	131	307	893	-0.0000012	-0.00048	9.04
12180	66112	119	0	0	0	14	24	4	44	115	16	21	131	373	1266	-0.0000011	-0.00056	9.06
12186	66112	119	0	0	0	12	39	4	46	148	16	22	111	345	1759	-0.0000011	-0.00058	9.06
12199	66112	119	0	0	0	14	17	3	35	160	16	11	178	399	1927	-0.0000010	-0.00062	9.06
12209	66144	119	0	0	0	14	48	5	59	198	12	11	155	420	2111	-0.0000010	-0.00062	9.07
12221	66112	119	0	0	0	25	6	5	15	168	26	23	153	398	1667	-0.0000011	-0.00059	9.06
12231	66144	151	0	0	0	23	42	5	54	176	19	24	148	437	1853	-0.0000011	-0.00054	9.05
12236	66112	151	0	0	1	24	19	7	47	165	17	21	163	474	1713	-0.0000012	-0.00055	9.06
12249	66144	183	0	0	0	18	40	5	51	182	13	11	145	505	1836	-0.0000012	-0.00054	9.05
12277	98920	183	0	0	0	16	72	5	83	297	49	46	197	588	2508	-0.0000011	-0.00061	9.07
12315	66529	246	32	0	0	50	42	39	86	292	86	66	251	716	3100	-0.0000013	-0.00047	9.05
12339	66152	215	0	0	0	16	70	7	79	295	43	48	275	736	3295	-0.0000013	-0.00044	9.05
12426	603979776	96	0	0	0	8	36	9	42	20	6	6	18	162	761	0.0000039	-0.00367	9.56
12574	536870976	89	0	0	1	3	6	6	8	43	22	26	40	145	601	0.0000038	-0.00326	9.67
12580	0	89	0	0	1	4	7	8	9	11	4	3	21	116	257	0.0000038	-0.00324	9.67
12613	520	57	0	0	0	5	70	7	46	16	3	3	74	138	894	0.0000037	-0.00319	9.65
12617	65	33	0	8	1	4	37	7	40	44	4	5	24	90	757	0.0000037	-0.00322	9.65
12621	67108864	33	0	0	0	6	6	6	11	9	3	2	22	59	256	0.0000037	-0.00318	9.65
12795	67108864	33	0	0	1	5	7	6	10	11	2	3	19	61	262	0.0000037	-0.00323	9.65
12980	67108864	33	0	0	1	5	6	8	9	13	3	3	19	61	265	0.0000039	-0.00326	9.66
13153	0	33	0	0	0	6	7	8	9	10	4	2	21	63	288	0.0000037	-0.00323	9.65
13333	64	33	0	0	0	7	8	10	12	44	4	4	39	92	811	0.0000036	-0.00318	9.65
13343	0	64	0	0	1	6	5	10	10	13	3	3	34	104	338	0.0000037	-0.00324	9.66
13391	0	33	0	0	1	6	7	10	12	13	2	3	19	59	301	0.0000037	-0.00322	9.66
13420	0	65	0	0	0	11	8	10	14	24	4	4	21	103	302	-0.0000019	0.00020	9.04
13425	64	65	0	0	0	12	10	11	17	30	5	3	29	121	390	-0.0000019	0.00020	9.04
13430	2	65	0	0	2	11	9	9	15	22	3	2	22	109	398	-0.0000019	0.00020	9.04
13443	0	97	0	0	1	11	9	11	14	23	6	5	26	144	378	-0.0000019	0.00021	9.04
13728	578	98	0	0	2	12	43	13	47	53	5	6	57	215	1086	-0.0000019	0.00021	9.04
13740	577	74	0	24	1	12	11	12	19	34	6	3	59	161	901	-0.0000018	0.00015	9.05
13773	64	76	0	0	0	14	9	11	17	36	2	1	25	133	438	-0.0000019	0.00019	9.03
13873	64	76	0	0	0	12	12	11	16	26	3	3	22	124	387	-0.0000019	0.00022	9.03
13894	64	75	0	0	0	14	9	14	17	33	3	1	23	128	413	-0.0000019	0.00021	9.03
13971	67108928	76	0	0	0	14	8	13	17	28	4	2	24	124	407	-0.0000019	0.00021	9.03
13989	67108928	108	0	0	0	14	11	12	18	44	7	5	30	174	446	-0.0000020	0.00028	9.01
14082	576	140	0	0	0	12	8	12	15	26	7	3	57	220	676	-0.0000019	0.00019	9.04
14173	64	140	0	0	0	13	8	12	18	26	5	2	24	183	397	-0.0000019	0.00022	9.04
14192	53824	140	0	0	1	13	40	42	49	64	37	36	54	227	947	-0.0000025	0.00072	8.92
14234	1	69	128	56	0	7	7	8	12	17	14	1	32	227	365	-0.0000017	0.00008	9.07
14262	0	70	0	0	0	5	6	8	12	24	5	4	21	112	265	-0.0000017	0.00010	9.06
14286	0	70	0	0	0	6	7	7	10	22	5	3	19	104	272	-0.0000017	0.00010	9.06
14310	512	70	0	0	0	40	8	39	44	18	36	35	54	136	821	-0.0000029	0.00038	9.04
14339	512	71	0	0	0	7	8	7	11	16	37	5	59	139	823	-0.0000017	0.00007	9.07
14384	64	71	0	0	0	7	8	7	10	28	5	3	25	110	331	-0.0000017	0.00007	9.07
14396	0	71	0	0	0	3	8	6	9	17	11	9	23	98	250	0.0000037	-0.00320	9.66
14407	0	103	0	0	0	3	8	6	9	16	2	1	18	128	230	0.0000037	-0.00322	9.66
14424	0	135	0	0	0	7	7	6	10	20	3	2	21	167	285	-0.0000017	0.00006	9.07
14437	67108865	56	32	16	0	7	7	7	9	21	4	2	11	107	237	-0.0000018	0.00011	9.06
14535	67108864	56	0	0	0	6	7	6	8	21	3	2	19	90	275	-0.0000017	0.00011	9.06
14683	0	88	0	0	0	6	8	7	11	23	4	3	21	128	300	-0.0000017	0.00010	9.07
14749	67108864	88	0	0	0	6	6	8	10	19	3	2	13	114	221	-0.0000018	0.00012	9.06
14780	67108864	88	0	0	0	7	8	7	10	18	5	3	12	112	252	-0.0000018	0.00011	9.06

14810	67108864	88	0	0	0	9	6	7	10	24	4	3	21	129	325	-0.0000019	0.00015	9.06
14960	64	88	0	0	1	9	7	9	12	27	10	8	23	128	390	-0.0000018	0.00013	9.06
15010	0	120	0	0	1	8	7	9	12	22	6	3	25	163	331	-0.0000020	0.00017	9.05
15017	64	120	0	0	1	9	8	10	12	27	4	3	19	162	345	-0.0000019	0.00017	9.05
15024	0	120	0	0	1	9	7	10	12	24	4	5	20	159	344	-0.0000019	0.00015	9.06
15041	67108864	184	0	0	1	7	7	7	12	19	6	3	21	221	326	-0.0000020	0.00012	9.06
15050	0	184	0	0	1	6	7	6	12	16	3	3	21	215	298	-0.0000020	0.00014	9.06
15082	0	192	0	0	1	7	6	8	11	21	5	4	22	228	314	-0.0000017	0.00006	9.07
15128	67108864	152	0	0	1	7	8	10	12	20	4	3	22	187	320	-0.0000015	0.00005	9.07
15164	0	152	0	0	1	8	9	7	13	18	3	2	22	189	331	-0.0000018	0.00011	9.07
15245	0	152	0	0	0	11	5	9	14	21	7	4	17	188	327	-0.0000016	0.00011	9.06
15278	0	184	0	0	1	9	10	10	14	22	7	6	18	220	311	-0.0000014	0.00003	9.07
15307	536870912	43	0	0	0	6	5	9	11	14	4	4	14	66	230	-0.0000017	0.00010	9.04
15339	603979776	43	0	0	0	10	6	12	14	21	8	6	13	74	294	-0.0000019	0.00020	9.03
15401	576	75	0	0	0	9	5	9	14	41	27	33	51	139	739	-0.0000021	0.00032	9.01

- **Runno**: run number.
- **Status**: ECA run status return code (base 10), see Table 3.2.
- **Seq**: number of channel sequencers disabled.
- **MBid**: number of Motherboards with differing ID's.
- **DBid**: number of Daughterboards with differing ID's.
- **0ev**: number of channels with no data
- **#ev**: number of channels failed number of events test.
- **Width**: number of channel failed width test.
- **Fit**: number of channels failed number of points in fit test.
- **Susp**: number of channels failed number of suspicious points test.
- **Frati**: number of channels failed Frati jump test.
- **Cub**: number of channels with flagged cubic fit coefficients.
- **Quad**: number of channels with flagged quadratic fit coefficients.
- **Lin**: number of channels with flagged linear fit coefficients.
- **ErrTot**: total number of failed channels (including Seq)..
- **Cells**: total number of failed cells (including Seq).
- **aCub**: detector averaged cubic fit coefficients.
- **aQuad**: detector averaged quadratic fit coefficients.
- **aLin**: detector averaged linear fit coefficients.

Appendix G

Electronics Calibration Setup Constants (ECAL) and Detector Configuration Changes

Before the Motherboard+Daughterboard electronics can operate properly, the setup voltages that determine such things as integrator balance, channel thresholds, front end channel timing, etc, have to be determined and loaded into the board. These constants are generated by the PENNDAQ code and saved in a database that is keyed/stored based on time (Pkey), and a number index from 1 to 1000 (Skey). Eventually these constants are ported over to SHARC which then can initialize the electronics with the proper constants. Changes in the ECAL constants will be reflected in the ECA calibrations, as they are sensitive to changes in the electronics. Whenever a new board is put into the system, a motherboard voltage has changed, or a major detector configuration change is made, then ECAL needs to be done for the affected boards, and a ECA run to reflect the new changes.

Table G.1 lists the type of calibrations and which Motherboard voltages can affect it.

The nominal values of the ECA constants for the three charge pedestals depend mostly on VBAL, and slightly on TDISC and TCMOS. The TAC pedestals/slopes depend only on TCMOS. The channel thresholds VTHR have no effect.

A list of detector configurations and associated ECAL constants up to July 25, 2001, is shown in Table G.2.

An Skey number for a specific set of ECAL constants are incremented only for major detector configuration changes that affect the constants, or when there is a change/improvement in the algorithm that generates the constants. When new boards are swapped in and new ECAL constants generated, it does not change the overall Skey as these new constants are

ECAL Name	Description	Default Skey	MB Voltages
VBAL	Integrator balance	621	+6.5V, +4Ref -2Ref
TDISC	Discrim. timing (min= 420 nsec)	601	+5Ref, +/-1Ref, -3.3V, -2Ref
TCMOS	GTVALID timing (max= 410 nsec)	611	+6.5V, +5Ref
VTHR	Discriminator thresholds	610	+1Ref, -1Ref

Table G.1: Table of front-end ECAL constants and their current default database Skey (as of July 25, 2001). The list of Motherboard (MB) voltages are those that directly affect the calibration constants. As well, the five main voltages, VCC, VEE, +8V, +24V, and -24V, can have dramatic effects on ECAL, and the ECA constants, if they change/fluctuate from nominal.

Date	Detector Config Change	VBAL	TDISC	TCMOS	VTHR
2/11/99	Begin production running	601	601	601	610
5/11/00	Crate 2 and 3 VCC changed by -0.5V	601	601	601	610
8/18/00	ADC LED mod	601	601	601	610
8/25/00	CMOS twiddle bits off	601	601	611	610
11/29/00	Crate fan upgrade	621	601	611	610
12/21/00	Lower trigger thresholds	621	601	611	610
3/7/01	Gain op-amp resistor change	621	601	611	610
5/28/01	Salt added	621	601	611	610

Table G.2: Shown are all the major detector configuration changes and ECAL constant Skeys for VBAL, TDISC, TCMOS, and VTHR. This table is valid up to July 25, 2001. The TCMOS Skey= 601 constants are with the CMOS twiddle bits enabled.

considered as part of the current detector configuration. The constants from the new boards are tracked by the Pkey, and the board ID's (which are stored in the database along with the values). Since the start of production running, the algorithms used to generate the constants have remained the same, except when the CMOS twiddle bits were turned off, requiring a regeneration of the TCMOS constants for the entire detector.

Appendix H

Electronics Problems Diagnosis

The ECA constants are an excellent way to check the health and stability of the electronics. Below in Table H.1 is a list of flagged ECA pedestal conditions and a diagnosis of the problem. It is not exhaustive, but represents some of the experience I gained (RVdW) while onsite monitoring and debugging the electronics. Similarly, Table H.2 is for the ECA TAC slopes.

A number of problems flagged by the ECA validation (at the cell or channel level) are single occurrences or intermittent, indicating that the problem is probably not real, and is only a random fluctuation about the cut selections. It is only when the problem is persistently flagged over time, or occurs for an entire discrimintator (4 channels), Daughterboard, Motherboard, or crate, that it is real and needs some attention.

As more problems are discovered or understood, the tables in this Appendix should be expanded.

ECA Flag	Cause of Problem	Action
Channel(s) with 0 events	Broken pedestal enable or front end digital chain	Disable channel, possible blind flasher!
Channel(s) with <160 events	Flaky pedestal enable or front end digital chain	Check ECAL, occupancy and cmos rates
Failed MB or DB ID check	New boards or ID readback failure or missing DQxx/titles	Check that new boards were swapped in
Channels with sequencers disabled	Channel sequencer turned off, not being read out	Check that the right number of seq. disabled
Large number of bad channels	More than normal, or 500, channels flagged might indicate a problem	Investigate detailed log for source of problems
Most channels on a slot failed difference test	Bad Op-amp, gain resistor, ADC, DAC, or ECAL changed	Disable or repair the slot
Most channels on a slot failed median/width test	Bad Op-amp, gain resistor, ADC, DAC, or ECAL changed	Disable or repair the slot
Most channels on a DB failed diff/med/width test	Bad DB voltage, DAC, or ECAL changed	Disable or repair the DB
Crates with bad pedestals, widths, or large diffs	Bad LV crate supply voltage	Check crate LV supplies

Table H.1: Table of pedestal ECA validation problems, possible causes, and required action.

ECA Flag	Cause of Problem	Action
Channel(s) with 0 events	Broken pedestal enable or front end digital chain	Disable channel, possible blind flasher!
Channel(s) with <160 events	Flaky pedestal enable or front end digital chain	Check ECAL, occupancy and cmos rates
Failed MB or DB ID check	New boards or ID readback failure or missing DQxx/titles	Check that new boards were swapped in
Channels with sequencers disabled	Channel sequencer turned off, not being read out	Check that the right number of seq. diabled
Large number of bad channels	More than normal, or 500, channels flagged might indicate a problem	Investigate detailed log for source of problems
Too many failed "widths", "nfit", or "suspicious"	Problems with electronics or ECAL, monitor changes	Disable channel if persistent
Too manu failed "cubic", "quadratic", or "linear"	Problems with electronics or ECAL, monitor changes	Disable channel if persistent
Too many Frati TAC jumps on a slot	Failing/broken ADC mod	Repair/repalce ADC

Table H.2: Table of time slope ECA validation problems, possible causes, and required action.

Appendix I

ECA Run Methodology and Frequency

During routine detector neutrino operation, ECA runs (PED then TAC slope) are done twice a week, Monday and Thursday. Runs are also done immediately after a scheduled/unscheduled detector power down, and after substantial hardware changes or maintenance. It was found that this calibration schedule is the best compromise between too much calibration, and enough to flag potentially bad channels and/or detector problems on a reasonable time scale.

Furthermore, after a long detector shutdown, or any detector work that might affect ECA calibrations, two sets of ECA runs within an hour of each other should be performed to resync the *diff* checks. The other possibility is to turn off the *diff* checks by performing the ECA runs with the UC bit set, or to explicitly turn it off by hand in the CECA (see Appendix A) bank.

After a detector/crate powerdown, a period of at least one hour should be allowed before performing an ECA to let the electronics warm up. During this wait period the neutrino run should have the UC bit set to indicate that trusted ECA constants are not available.

Finally, ECA runs should be done within a day before PCA and N16 stability calibrations, and any other source runs. This ensures that proper ECA calibrations and validation flags are available for these critical runs.

Bibliography

[1] Various electronics papers/reports on **surf** in the directory /users/snodaq/doc, with the following filenames: elec_fec32.ps, elec_int_disc.ps, elec_paper.ps, elec_tests.ps, elec_ieee-98.ps, elec_overview.ps, elec_system.ps, elec_trig.ps.

[2] SNO-STR-99-021, SNO-STR-99-038, and SNO-STR-99-039.

[3] Peter Wittich thesis, "First Measurement of the Flux of Solar Neutrinos from the Sun at the Sudbury Neutrino Observatory", 2000.

[4] See on **surf** in the directory /users/snodaq/doc, the following, elec_fec32.ps.

[5] R.G. Van de Water, *Effects of ECA Validation on Reconstruction and Energy*, SNO Analysis Notebook, May 2001.