Measurement Noise and Time-Base Calibration in ACDC Data

Charles Whitmer

TerraPower LLC, Bellevue, Washington, USA

December 19, 2018

Abstract

ADC data collected on an ACDC card with a generated 100 MHz sine wave input are analyzed with the goal of understanding the measurement noise in the system. Best fits to models are performed, and the residuals of the fits are examined. We take the point of view that correlations of the residuals with any system parameters represent systematic errors that can be measured and used for calibration.

After all calibration is complete, we find that remaining noise levels are in the range of 0.6 mV to 0.9 mV regardless of whether the PSEC4 is powered by a linear voltage regulator or a switching voltage regulator. These noise levels are consistent with earlier measurements.

One result of this analysis is the development of a method to perform time-base calibration for a PSEC chip using only the data from 50 software triggers. This method is shown to have an accuracy comparable with the 'zero crossing method' with 1×10^6 zero crossings, which takes about 50 000 triggers of data.

1 Introduction

In August of 2018, Audrey Whitmer collected data sets of 50 software triggers each from modified and unmodified ACDC boards. The modification consisted of providing the 1.2 V supply voltage for the PSEC4 chips from an external switching power regulator instead of the usual linear regulator on the ACDC board. A Tektronix AWG (Arbitrary Wave Generator) supplied a 100 MHz sine wave input to a single channel on the ACDC board.

In her report on her work [1], Audrey compared only 4 triggers from a board (ACDC board 24) in each state, modified and unmodified, and analyzed them using a simple least squares fit and the Excel Solver. The standard deviation of the residuals of the fits for both board states amounted to approximately 12.5 ADC units, or about $12.5 \times (1.0\,\mathrm{V})/4096 = 3.1\,\mathrm{mV}$. This level of noise differs from the 3 ADC units, or about 0.7 mV, reported by Oberla, et al. [2] Audrey speculated that the discrepancy was due to the lack of time-base calibration.

In light of this, we wish to investigate whether it is possible to use her existing data sets to calibrate a PSEC4 chip well enough to match the previously observed level of noise.

2 The Method

After performing a best fit of a model to collected data, we can look for correlations of the fit residuals to various system parameters. True noise would not correlate with anything, so we assume that any statistically significant observed correlation points to an area of the system that needs to be calibrated. There are four correlations in particular that we will look at.

1. Correlation of residuals with the voltage being measured.
2. Correlation of residuals with the voltage being measured on a per-cell basis.
3. Correlation of residuals with the trigger number.
4. Correlation of residuals with the time derivative of the voltage.

Each of these will be handled in a section below.

3 The General Problem

We feed a generated sine wave into a channel on an ACDC board and record ADC measurements for m triggers. We call the measurements d_{kj} for trigger number k and cell number j. A 'software' trigger is used, so that the phases of the wave captured by each trigger are random.

In an ideal case, we would know that our input is an AC coupled sine wave, with pedestals subtracted, so we could try to fit to a simple sine wave with no baseline. In real cases we do observe that a baseline offset survives. So we would try to minimize $\sigma^2\chi^2$ in:

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}(Asin(\theta_j + \phi_k) + B - d_{kj})^2 \tag{1}$$

where the free parameters are an amplitude A, a baseline B, a set of phases ϕ_k for each of m triggers, and a set of time sample points θ_j with one for each of n cells.

Note that it is convenient for the math to measure 'times' θ_j in radians. Since we know the frequency of the input signal, we can convert to conventional time units later.

A typical best fit problem would have the σ^2 dividing each of the summands on the right hand side of equation 1, and it would represent the variance on each measurement. We do not know this value, so we are using a bootstrap method. After performing the best fit, we will assume that $\chi^2 \approx ndof$, where $ndof$ is the number of degrees of freedom, and that will let us estimate σ^2. The resulting σ is what we will report as 'noise'.

If the parameters θ_j and ϕ_k were known, then solving for the optimal A and B would be a trivial problem in linear algebra. But with θ_j and ϕ_k being inside the sine function, and there being hundreds of dimensions to search, a direct solution is not possible.

As a result we will approach the problem by finding approximate solutions for θ_j and ϕ_k and then incrementally improve them to reach the optimal solution. We will cycle through

a process of fitting the shape of the waveform, and then updating θ_j and ϕ_k, and then fitting the shape again with those new parameters. This process converges quickly.

The first approximation we make is to assume that all time steps are equal so that $\theta_j = j\Delta\theta$, for some $\Delta\theta$. The period of the sine wave is $2\pi/\Delta\theta$ when expressed as a number of cell samples. We know the frequency of our generated sine wave at the start and the nominal cell capture rate of the ACDC card. Using those we make a preliminary guess at the value of $\Delta\theta$.

If we take a value for $\Delta\theta$ as a given, then it is easy to find the optimum (minimized) solution for the following:

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}(C_k sin(j\Delta\theta) + D_k cos(j\Delta\theta) + B_k - d_{kj})^2 \tag{2}$$

Equation 2 has three free parameters for each trigger and we identify these as $C_k \approx Acos(\phi_k)$, $D_k \approx Asin(\phi_k)$, and $B_k \approx B$, where A, B, and ϕ_k are free parameters from equation 1. This is nearly a set of independent fits for each trigger, except that they are bound together by a common value for $\Delta\theta$.

After performing the above fit for the whole set of triggers, we have the approximate solutions for parameters in equation 1:

$$tan(\phi_k) = \frac{D_k}{C_k} \tag{3}$$

$$A^2 = \frac{1}{m}\sum_{k=0}^{m-1}(C_k^2 + D_k^2) \tag{4}$$

$$B = \frac{1}{m}\sum_{k=0}^{m-1} B_k \tag{5}$$

Note that equations 3 and 4 have two solutions, where ϕ_k differs by π and A can be positive or negative. We will choose the solution that makes A positive. That is, we choose the solution where the sign of $cos(\phi_k)$ is the same as the sign of C_k.

Since we have taken $\Delta\theta$ as given, we will need to vary it up and down to 'manually' search for the best wave period. Our initial guess is close to correct, so this is not difficult. (The optimization problem involves such simple linear algebra that 100 guesses for $\Delta\theta$, covering some range, can be evaluated in less than a second.)

After performing these fits over all triggers, we will have a set of phases ϕ_k that are quite close to the solution we are looking for.

For every real data set we have looked at, there is a significant value observed at this point for the baseline B. So we remove it here, and from this point on use a modified d'_{kj} as our dataset:

$$d'_{kj} = d_{kj} - B \tag{6}$$

4 Examples using real data

We will use samples collected by Audrey Whitmer in her test of switching power supplies in August 2018. For ACDC boards 20 and 24, connected to ACC board 4, she collected data with each card in a modified and unmodified state. (The modified state consisted of a lifted output pin from a 1.2 volt linear voltage regulator, and the voltage supply replaced by an external switching power supply board.) For each board in each state she recorded 50 software triggers on each of channels 3, 9, 15, 21, and 27. The channels are spaced 6 apart so that each test employs a different PSEC4 ASIC on the ACDC card. (This makes a total of 20 sets of 50 triggers, of which we will only examine 4 sets in this paper.) The supplied signal was a 100 MHz sine wave generated by the Tektronix AWG (Arbitrary Wave Generator) model 5012. The AWG was set to supply waves of 1 V peak-to-peak, but produced only about 0.2 V peak-to-peak, probably due to an incorrect mode setting. The AWG does not generate true sine waves, but instead an approximate function derived from a limited number of sample points. At 100 MHz its data rate (1.2 GS/s) restricts us to a maximum of 12 points; this turns out to be important and we will need to take it into account.

We will begin by looking at data from channel 3 on the modified board 24. The data from all 50 triggers was loaded from the file *aw.4.24.3.acdc.dat*.[1] The file begins with a set of measured pedestals[2] for each cell, and these were subtracted from the data before the fit. Figures 1 and 2 show the data for the first 25 triggers before and after the pedestal data was subtracted. Removing the pedestal measurements makes an obvious difference.

[1] Collected on 8/14/2018 with the logData shell script.

[2] Collected by the takePed command, which was run before each logData.

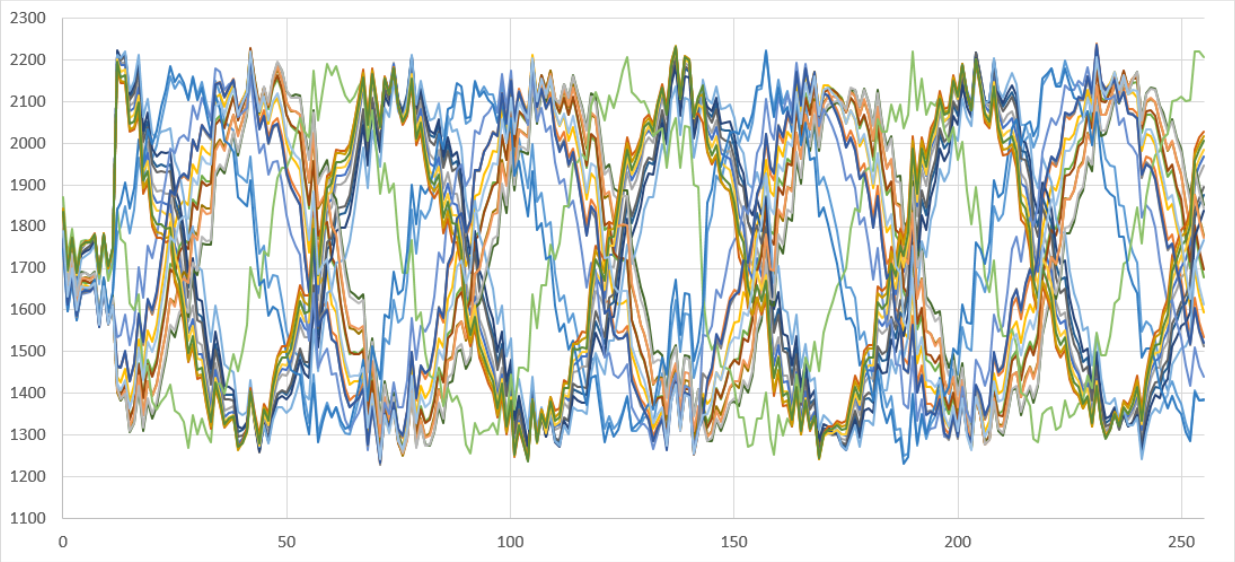


Figure 1: Raw data from the capture of 25 triggers. ADC values are plotted on the vertical axis, and cell numbers on the horizontal axis. There is a lot of noise to be seen as each cell measures voltage from a slightly different baseline. Note that the measured ADC values range only from about 1300 to 2200, instead of spanning the whole 0 to 4095 range.

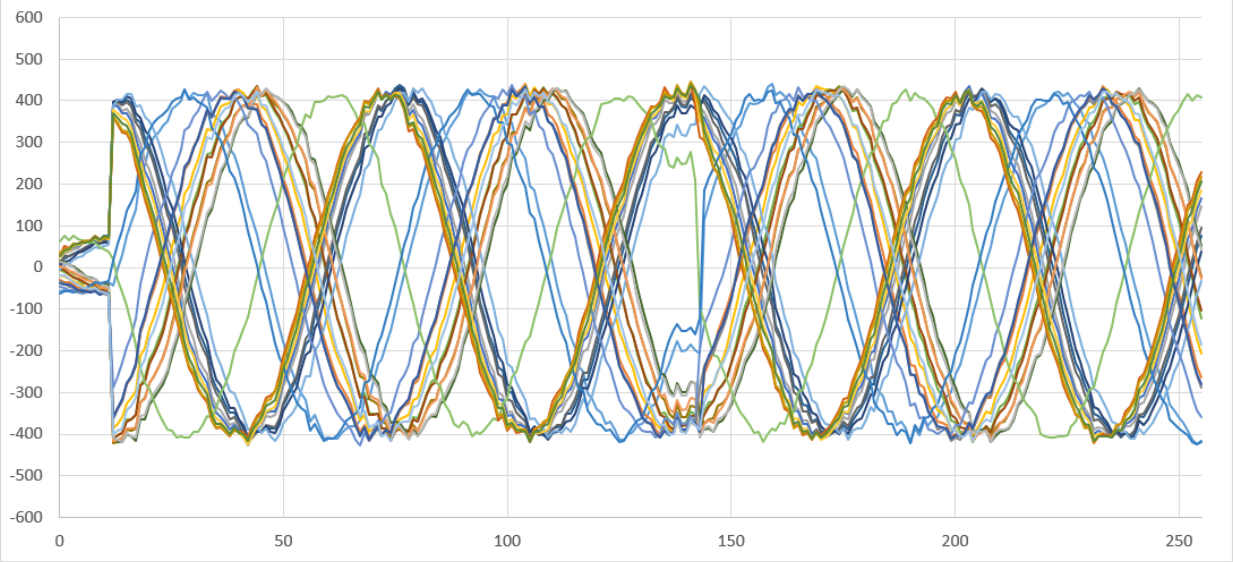


Figure 2: The same 25 triggers as in figure 1 except that the pedestal measurements from the header of the data file have been subtracted.

In figure 2 we can see 'artifacts' at the beginning and in the middle of each trigger data set. This is likely the result of a firmware bug, and we observed similar behavior in most channels on both ACDC cards that were tested. Something causes the recorded data to

flatten out over certain cell ranges. (These cell ranges vary by channel and by ACDC card.) As a result, we perform our fits by ignoring selected ranges of cells. We consider the cells to be numbered from 0 to 255, and by that method of counting the fits we will now perform will ignore cells 0 to 20 and cells 135 to 150 (a total of 37 cells).

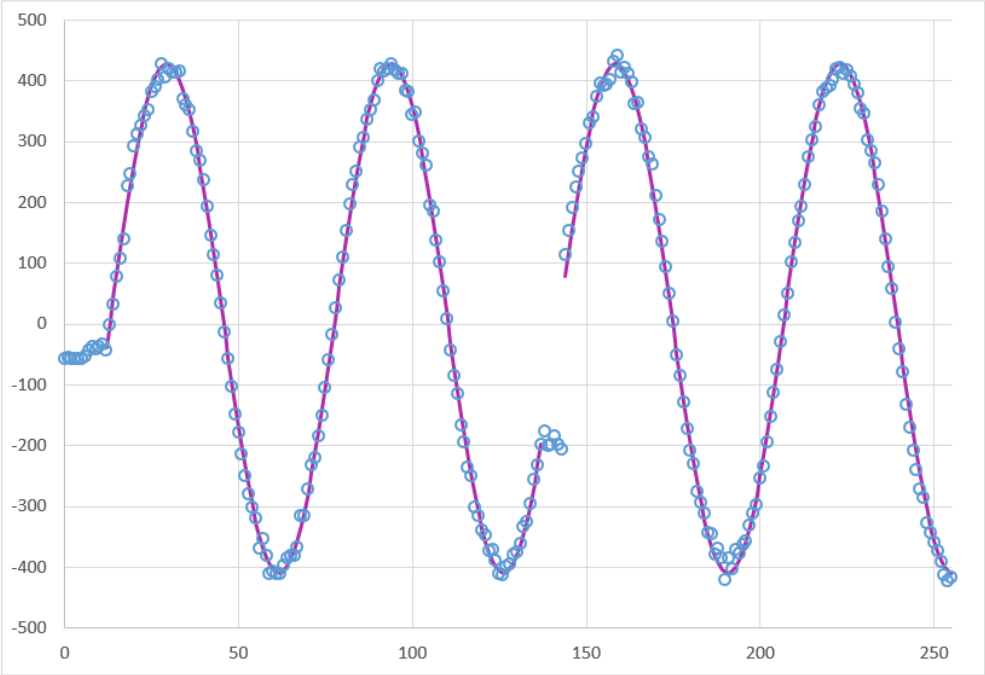


Figure 3: Modified ACDC board 24, trigger 0 capture on channel 3. The horizontal axis shows the cell number, the vertical axis is in ADC units with pedestals subtracted. Open circles show collected data; the solid line shows the best fit. Gaps in the solid line show where data was ignored due to artifacts. The residuals in the fit have a standard deviation of 12.7 ADC units.

After performing the fit for equation 2, the fit for the first trigger is plotted in figure 3. The standard deviation of the residuals is consistent with Audrey's result for a simple uncalibrated sine fit.

Combining the 50 triggers, we get the set of estimates $A = 416.3$, $B = 5.24$, a period of 64.56 cells, and a set of phases ϕ_k. B is subtracted from the dataset as described above. The period gives us the estimate $\Delta\theta = (2\pi\,\text{rad})/64.56 = 0.097\,33\,\text{rad}$, which gives us a starting point for the set of θ_j.

5 Correlating Residuals and Measured Voltage

We find the optimal fit for the parameter A in the following, assuming the θ_j and ϕ_k are fixed as described above:

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}(Asin(\theta_j + \phi_k) - d'_{kj})^2 \tag{7}$$

and then plot the residuals as a function of the best fit ADC value; see figure 4.

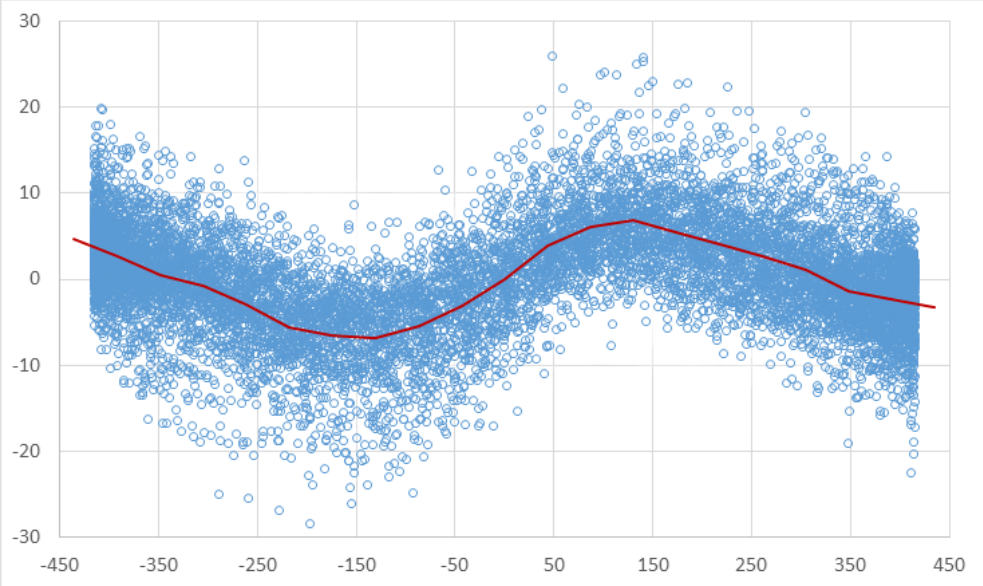


Figure 4: For a sine function fit to all triggers, the residuals in ADC units are individually plotted on the vertical axis and the fit ADC values on the horizontal axis. There is a strong correlation. A piecewise linear function consisting of 20 segments shows a smoothing of the data.

There is a strong correlation seen, and it means one of two things. It could be that there is a non-linear response to input voltages, or that we might be fitting to a curve with the wrong shape. An additional fact we have noted, but not illustrated here, is that this correlation curve changes depending on whether the sine wave is rising or falling. That would not be the case if this was completely the result of non-linear response. So we will proceed with the assumption that this correlation is due to an error in the shape of the wave form.

Given that the AWG is using a small number of sample points to generate the sine, it will certainly produce some number of harmonics. Investigation shows that including more than 7 harmonics does not significantly improve the fit, so we fit to 7 harmonics:

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}\left[\left(\sum_{h=1}^{7} A_h sin(h(\theta_j + \phi_k))\right) - d'_{kj}\right]^2 \tag{8}$$

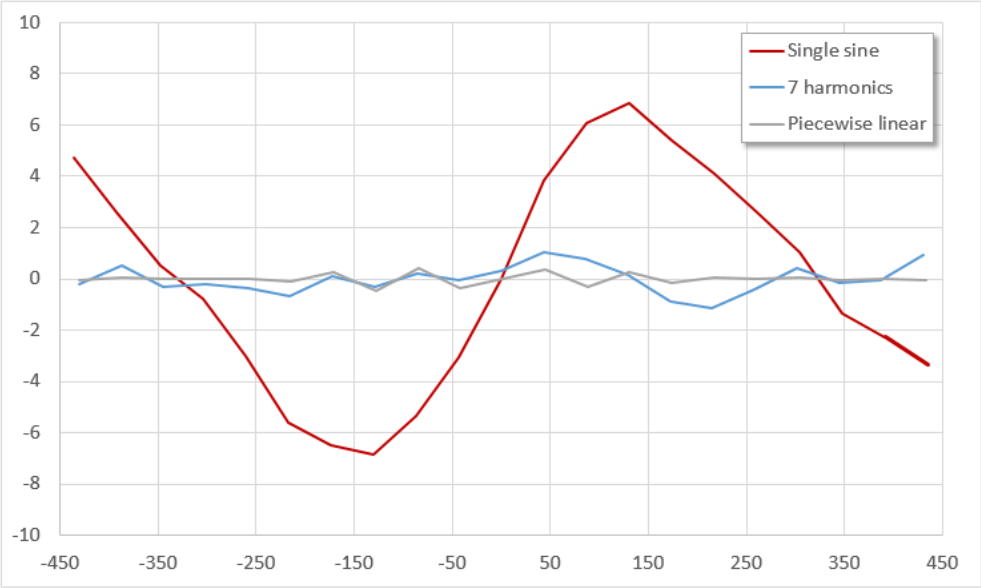


Figure 5: These are the piecewise linear smoothings of the residuals from a sine fit, a fit with 7 harmonics, and a periodic piecewise linear fit of the waveform. The red line is the same as in figure 4.

Including 7 harmonics reduces the correlation greatly, as seen by comparing the red and blue lines of figure 5. However, it is possible to do even better by fitting the waveform to a periodic piecewise-linear function (PPLF). In theory, enough harmonics can match any waveform, but it is not practical to spend the time to locate the specific harmonics that are needed, and including all harmonics up to some fixed number is wasteful of our degrees of freedom, as most of the harmonics have negligible amplitude. Using the PPLF is general and easy.

The PPLF is built on periodic triangular waves defined as follows:

$$P_n(x) = \begin{cases} 1 - \frac{n}{2\pi}x, & \text{if } 0 \le x < \frac{2\pi}{n} \\ 0, & \text{if } \frac{2\pi}{n} \le x < 2\pi(1 - \frac{1}{n}) \\ 1 - \frac{n}{2\pi}(2\pi - x), & \text{if } 2\pi(1 - \frac{1}{n}) \le x < 2\pi \\ P_n(x \pmod{2\pi}), & \text{otherwise} \end{cases} \tag{9}$$

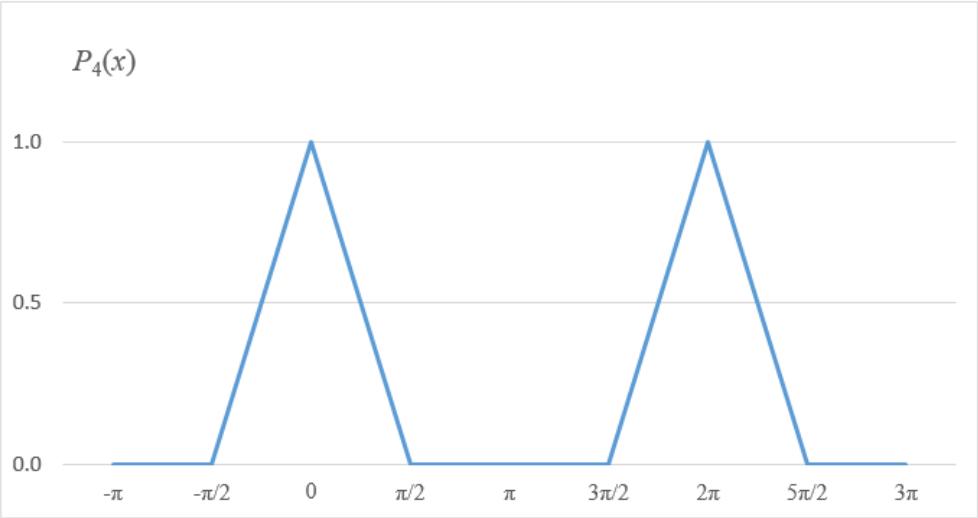


Figure 6: The periodic triangular waveform $P_4(x)$. Using sums of four displacements of these waves, we can construct any piecewise linear periodic function made of four segments on $[0, 2\pi]$.

Figure 6 illustrates an example of the function for the case $n = 4$. Linear combinations of four translations of P_4 can produce any PPLF having four equal-length line segments.

We find that 36 segments fit the waveform well, and the gray line in figure 5 shows improvement over using the harmonics. Using up 36 degrees of freedom to describe the waveform may sound wasteful, but we did start with $50 \times 256 = 12\,800$ dof, so we can afford it.

The model fit that is being performed now consists of minimizing equation 10. The calculation of the best fit is just as easy as the one for the harmonics since the fit parameters, Y_p, are simple coefficients of functions representing the triangular waveform with 36 different phases.

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}\left[\left(\sum_{p=0}^{35} Y_p P_{36}(\theta_j + \phi_k - \tfrac{2\pi}{36}p)\right) - d'_{kj}\right]^2 \tag{10}$$

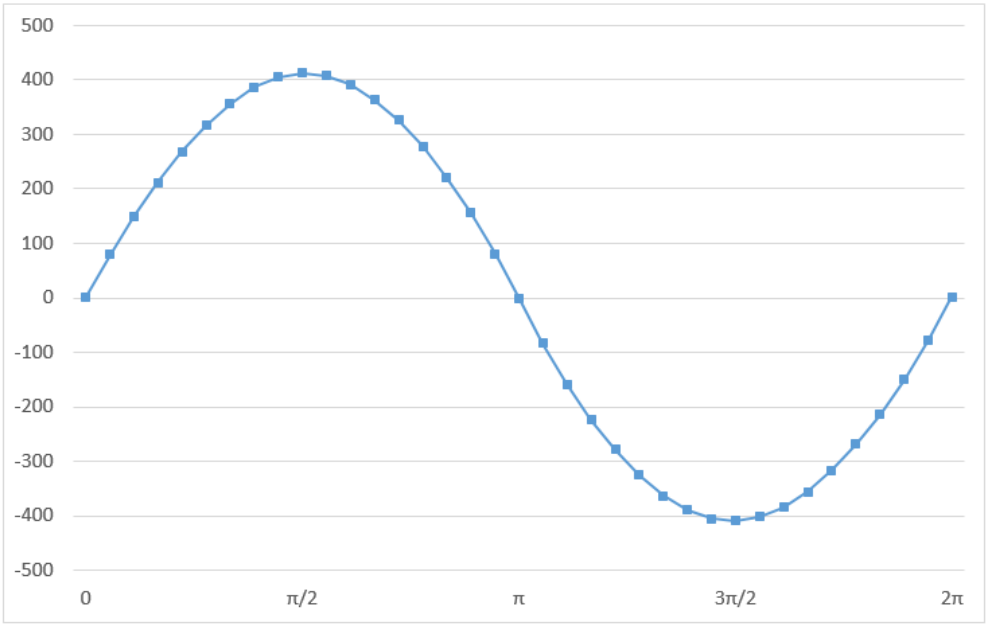


Figure 7: The waveform represented by a periodic piecewise linear function (PPLF) with 36 segments.

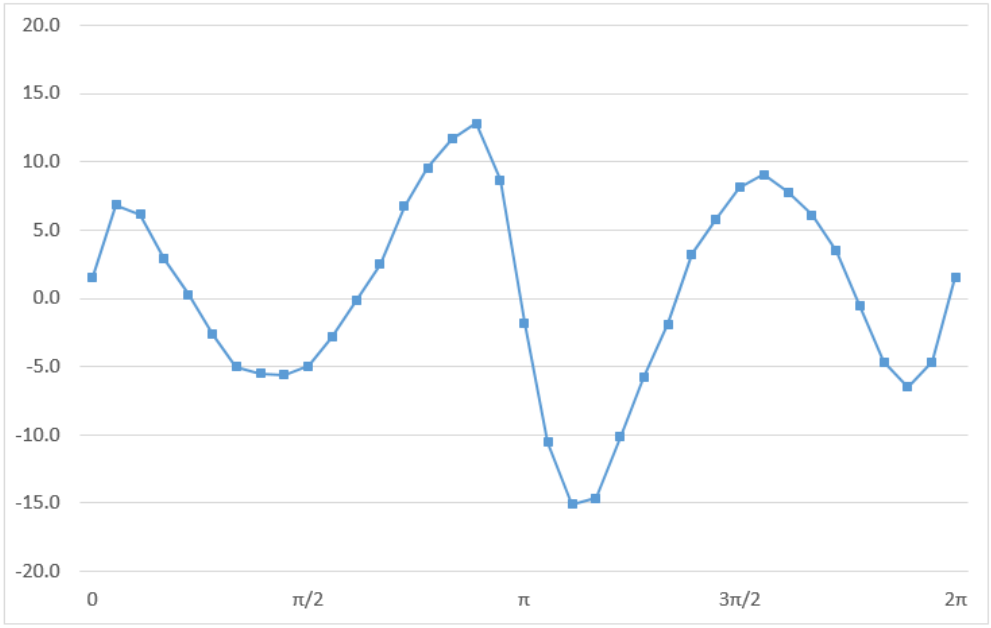


Figure 8: The difference between the piecewise linear function of figure 7 and a true sine.

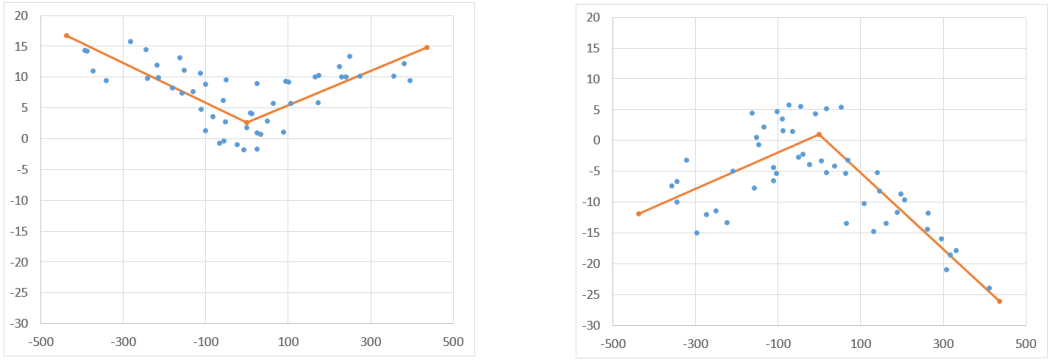
Figure 7 shows the waveform as fit by the PPLF. It looks like a sine wave at this scale, but figure 8 shows the result of subtracting a true sine, and there are large differences. First, the peaks and valleys of the actual wave are wider than a sine. Secondly, the shape of the

peak is different from the shape of the valley. The difference ranges from about 15 ADC units below the sine to 15 ADC units above. If our goal is to get the residual noise down to just a few ADC units, this systematic error must be removed.

We cannot be certain how much of this correction is adjusting for an inaccurate sine wave input, and how much is compensating for a global non-linear response to input voltage. Using a pure sine input, or the same input with a few different pedestals could resolve this issue, but we cannot investigate it with the data in hand. So what we have performed is a combined adjustment for the waveform shape and non-linear response.

6 Correlating Residuals and Measured Voltage on a per-Cell Basis

A non-linear response to input voltage exists and differs on a cell-by-cell basis.



(a) Cell 57

(b) Cell 119

Figure 9: Correlation of measured ADC value (horizontal axis) vs. residuals (vertical axis) for two cells. Each plot shows 50 points, one for each trigger.

Figure 9 shows two examples where there is a large curvature in the response to input voltage, and shows that the curvature can be positive or negative, depending on the cell. Each of these examples has a statistically significant departure from linearity; being nearly 8 standard deviations.

The curves in general, over all cells, show a sharp change in direction in the middle of the range. Normally, when looking for non-linearity we would start with the quadratic behavior, i.e. a constant curvature over the range. It turns out that a quadratic fit to the data under-represents the sharpness of the curve near zero. If we had input data over a larger dynamic range, say something approaching the 4096 ADC units that are possible, we could find hints to the true shape of the response curve. But with a range of only 800 ADC units, the two segment fit is the best we can find.

So we compute a simple two-segment model of the response for each cell. We call the resulting functions $\lambda_j(x)$, and then subtract this non-linear response from the observed data points. This costs us another 3 degrees of freedom per cell.

Now our model consists of:

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}\left[\left(\sum_{p=0}^{35} Y_p P_{36}(\theta_j + \phi_k - \tfrac{2\pi}{36}p)\right) - (d'_{kj} - \lambda_j(d'_{kj}))\right]^2 \tag{11}$$

7 Correlating Residuals and Trigger Number

All of the triggers for a chosen channel and board state were collected in a single run lasting no more than two minutes. In a stable environment, we would expect no change in behavior between triggers, except for the random phase at which the software trigger occurred. In particular, the distribution of residuals should not depend on the trigger number. To check this, we plot all residuals versus trigger number for the unmodified ACDC board 24. See figure 10.

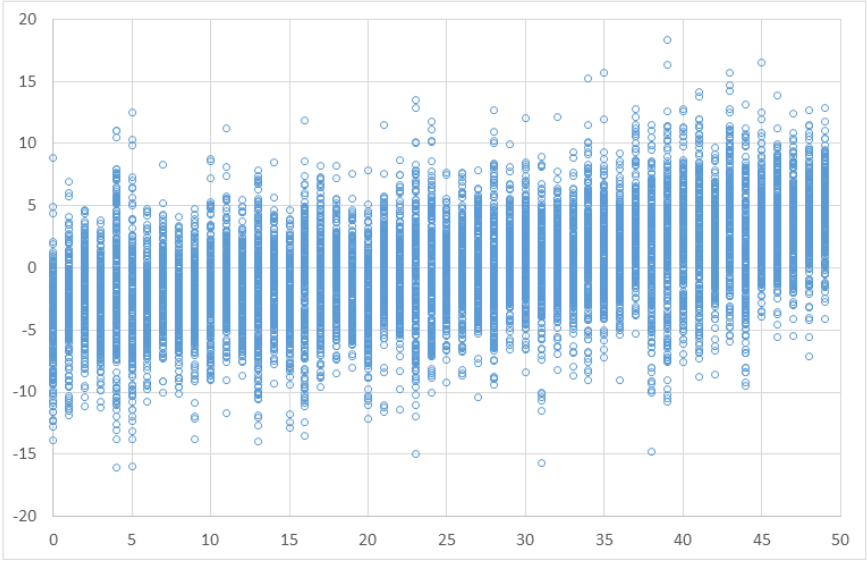


Figure 10: All residuals, on the vertical scale in ADC units, versus trigger number (0 to 49).

Surprisingly, there is a clear drift seen in the residuals even over this short period of time. There were issues maintaining connections between the ACC and ACDC boards for more than several minutes at a time, and Audrey speculated that it could be temperature related. We could be seeing a drift in behavior due to temperature change.

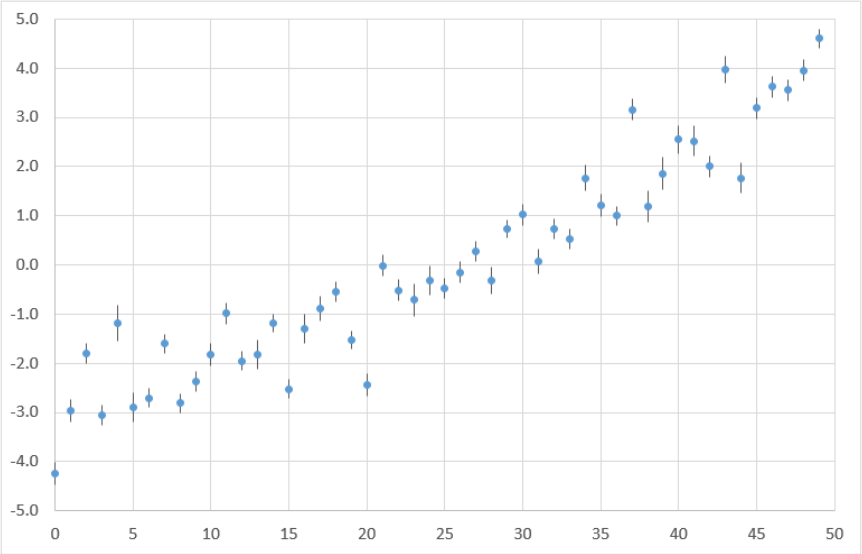


Figure 11: The means of the residuals plotted in figure 10, for the unmodified ACDC board 24 with the linear power regulator. The error bars represent the error on the mean.

Figure 11 shows that averages of the residuals that were scatter-plotted in figure 10. This shows a cleaner view of the drift.

Figure 12 shows the averaged residuals, but for the modified ACDC board 24. There is drift to be seen, but with different behavior than on the unmodified board.

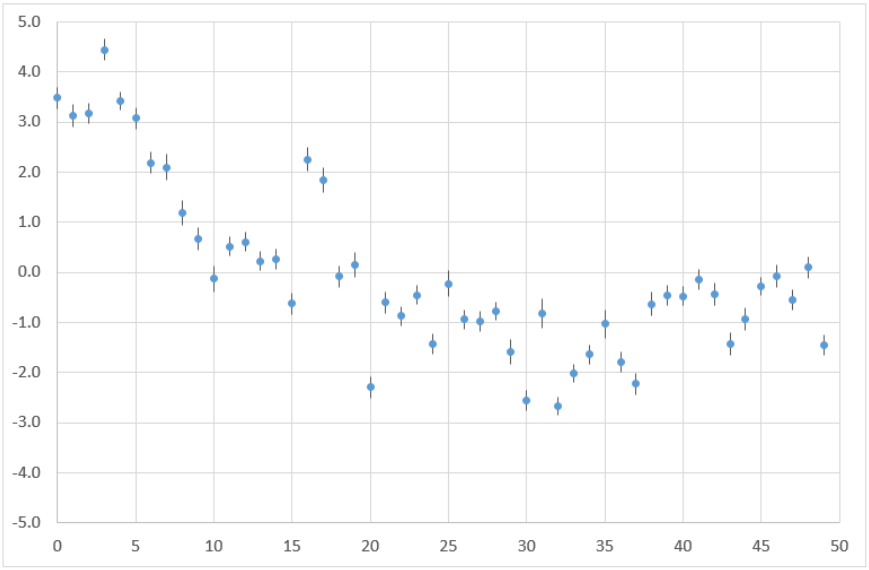


Figure 12: The same plot as in figure 11, but for the ACDC board with the switching power regulator. The error bars represent the error on the mean.

It would take more research under controlled conditions to determine the cause of the

drift per trigger. For our purposes, we have noted a statistically significant systematic error, and we will remove it. This can be done by subtracting a trigger-dependent baseline B_k from each data point.

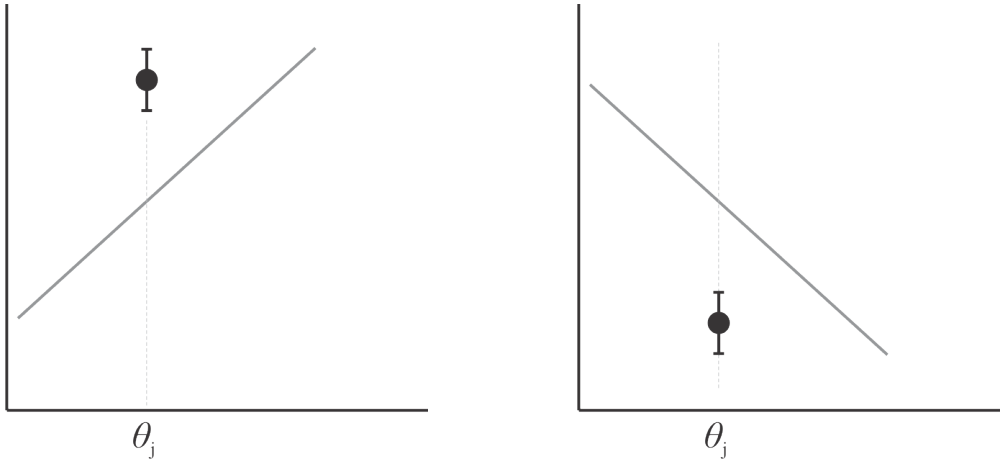
The final version of our model is:

$$\sigma^2\chi^2 = \sum_{k=0}^{m-1}\sum_{j=0}^{n-1}\left[\left(\sum_{p=0}^{35} Y_p P_{36}(\theta_j + \phi_k - \tfrac{2\pi}{36}p)\right) - \left(d'_{kj} - \lambda_j(d'_{kj}) - B_k\right)\right]^2 \tag{12}$$

8 Correlation of Residuals with the Slope of the Fit

Finally, we will show how to improve the estimates of θ_j and ϕ_k.

Suppose that, for a given cell j, we compute the average residual, but only when the slope of the fit curve is positive, and we find that the average data point lies above the fit curve and therefore has a positive residual. And likewise suppose that when the slope is negative we find that the average data point lies below the fit curve. This would be a case of the residuals being correlated with the slope. This is illustrated in figure 13.

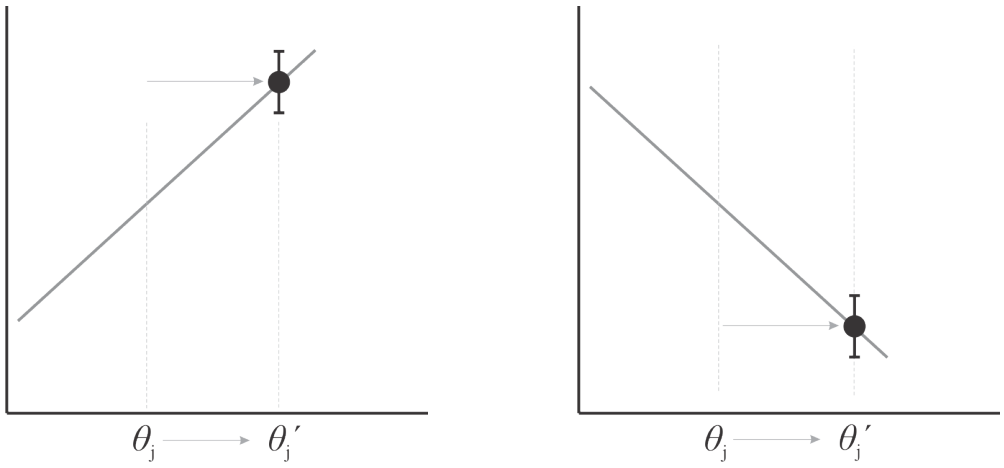


(a) Average data with a positive slope, showing a positive residual

(b) Average data with a negative slope, showing a negative residual

Figure 13: Schematic illustration showing a correlation between residuals and slope.

Having this sort of correlation suggests that our guess at the location of the sample point, θ_j, is not quite right. Consider figure 14 where, having observed the positive correlation, we shift our θ_j to the right. The data points move closer to the fitted curves, and the residuals are reduced.



(a) Adjustment with positive slope (b) The same adjustment with negative slope

Figure 14: These would be the same data points shown in figure 13, but we adjust our estimate of the time, θ_j, when they were measured.

Now, looking at real data, consider figure 15. It resembles the illustrated case in that there is an obvious positive correlation of the slope of the fit curve and the residuals.

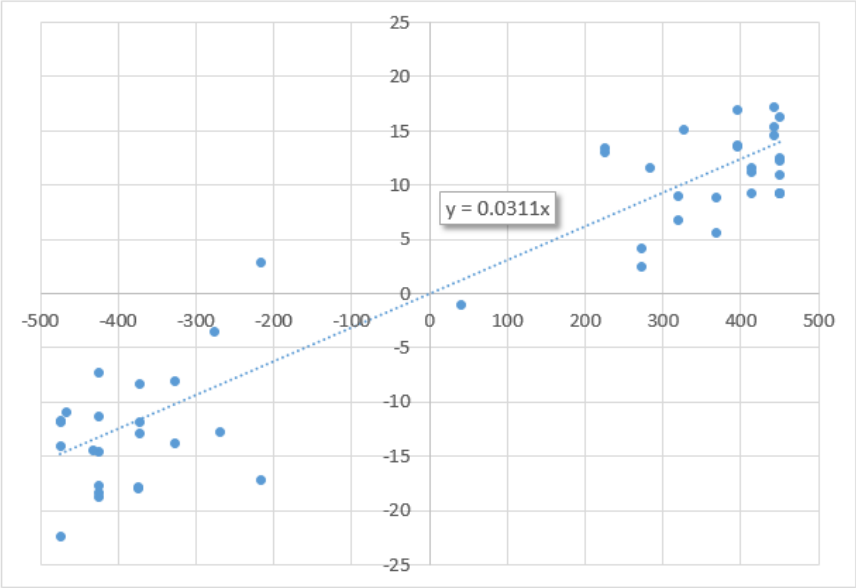


Figure 15: For cell 22 on the modified ACDC board 24, the residuals are plotted on the vertical axis in ADC units, versus the slope of the best fit waveform in ADC units per radian. Because the fitted wave shape is piecewise linear, there are only 36 different slopes observed. A'trendline' is shown with slope 0.0311.

Suppose that for cell j we have a set of slopes s_k and residuals r_k. If $\sum_k r_k s_k \neq 0$ then they have a correlation. If we change the time of sampling by an amount a, so that $\theta_j \to \theta_j + a$,

then we would expect, to first order, that the residuals would change $r_k \to r_k - as_k$, and the correlation would then be $\sum_k (r_k - as_k)s_k$. To make the correlation vanish, we would set:

$$a = \frac{\sum_k r_k s_k}{\sum_k s_k^2} \tag{13}$$

For the general case of fitting a line through the origin to a set of points (s_k, r_k), this value of a is the slope of that fit line. Figure 15 shows a 'trendline' through the data and displays the slope of the trendline. Note that the vertical units on the plot are ADC units and the horizontal units are (ADC units)/rad. This means that the slope of a line on the plot is measured in radians. The slope of the trendline is the first order adjustment we would like to make to θ_j, and in this case, the adjustment is $0.0311(16)$ rad. The initial guess for θ_{22} was $22 \times \Delta\theta = 22 \times 0.097\,33\,\text{rad} = 2.1413\,\text{rad}$. The adjustment is nearly a third of $\Delta\theta$.

The uncertainty on a in equation 13 is given by:

$$\delta a = \frac{\sigma}{\sqrt{\sum_k s_k^2}} \tag{14}$$

where σ is the uncertainty on each r_k. This is how the uncertainty on the angle adjustment is computed, but this is also the uncertainty on the adjusted angle. After this adjustment we consider $\theta_{22} = 2.1724(16)$ rad.

In all of our model equations, from equation 7 to equation 12, the treatment of θ_j and ϕ_k is symmetric. Because of that, the same technique used to update θ_j can also be used to update ϕ_k. The only difference is that we will have more sample points in our slope-vs-residual graph, since we get one sample per cell. Also, the initial guesses for phases were very good, and they are only ever adjusted slightly.

9 A Summary of the Fitting Process

We have gone through all the steps of the process of finding a best fit model. We list them here in summary as a sequence of steps to follow.

Note that we repeatedly compute a best fit to the shape of the waveform. Even though this looks complicated, it is equivalent to solving a system of 36 linear equations in 36 variables, and takes only a millisecond of computation. (If we are instead using a 7-harmonic sine, this is 7 linear equations in 7 variables, and so even faster.) Each time we change fit parameters other than the shape coefficients, we recompute the best shape fit because it may have changed a little.

1. Load the data for a particular channel from the .dat file. Subtract the pedestal data that is in the header of that file.

2. Fit to equation 2 in a loop that tries various guesses of $\Delta\theta$ over a range. The best $\Delta\theta$ gives us a first estimate of the θ_j. The other best-fit parameters, C_k, D_k, and B_k, give estimates for ϕ_k and a baseline B. Subtract the B from all data points.

3. Initialize the per-cell linearity adjustments $\lambda_j(x)$ to zero, and also the trigger-drift baseline adjustments B_k to zero.

4. Compute a best fit of the shape according to equation 12. Use the method of section 6 to adjust the $\lambda_j(x)$ functions to remove per-cell nonlinearity correlations.

5. Compute a best fit of the shape according to equation 12. Now use the method of section 7 and adjust the B_k values to remove trigger drift correlations.

6. Compute a best fit of the shape according to equation 12. Use the method of section 8 to adjust the θ_j estimates.

7. Compute a best fit of the shape according to equation 12. Use the method of section 8 to adjust the ϕ_k estimates.

8. Repeat steps 4 though 7 several times.

In practice, we go through the cycle of steps 10 times, and it is overkill. The system has converged by cycle 4, and no improvement at all is seen after cycle 8.

The steps that bring the largest improvement to the fit, i.e. the largest reduction in $\sigma^2\chi^2/ndof$, are:

- The first shape fit after the initial sine fits.
- The correction for non-linear cell response.
- The improvement of the sample times θ_j.

10 The Residual Noise in Linear versus Switching Regulator Tests

As a follow up to Audrey Whitmer's report on her tests performed in August of 2018, after careful calibration of ACDC boards we can calculate the following noise values. Modified boards supplied 1.2 V to the PSEC4 ASICs from a switching power regulator.

Each of these noise measurements is the result of calibration and analysis of 50 triggers. On ACDC board 24 the fits are performed with 9938 degrees of freedom and data from 219 cells[3][4] not affected by artifacts. On ACDC board 20 the fits are performed with 11 088 degrees of freedom and data from 244 cells[5][6].

ACDC Card	State	Channel	Noise (ADC units)	Noise (mV)
24	unmodified	3	3.59	0.88
24	modified	3	3.21	0.78
20	unmodified	3	2.57	0.63
20	modified	3	2.63	0.64

[3]-range 21,135,151,256 (This is an input parameter to the fitting program; cells 21 to 134 are included in the fit, as are cells 151 to 255.)

[4]datafiles: aw.4.24f.3.acdc.dat and aw.4.24.3.acdc.dat

[5]-range 12,256

[6]datafiles: aw.4.20.3.acdc.dat and aw.4.20m.3a.acdc.dat

These results are consistent with previous measurements of noise in the PSEC4, and show no significant difference between cards powered with a linear regulator and a switching regulator.

11 Cell-to-Cell Time Step Variations

The fitting process that we have performed gives us a set of 'time' points, θ_j, with errors determined by equation 14. Using the generated 100 MHz sine wave as our time standard we can convert these into t_j measured in conventional units. Note that the times we have calculated are absolute times, measured from an arbitrary zero point, and not differences between successive cell times. The typical error on the measured sampling time, for both the modified and unmodified ACDC board 24, is 2.6 ps. If we do look at the differences between successive sampling times, they average 154.8 ps with a standard deviation of 12.3 ps. The smallest time step seen is 123 ps and it comes between cells 114 and 115. The largest time step is 191 ps between cells 235 and 236.

In the paper of Oberla, et al., an average cell time step of 95.9 ps is reported with a standard deviation of 12.1 ps. The difference between their and our average cell time steps is explained by a 40 MHz system clock in their case, versus a 25 MHz clock in our case. The agreement of the standard deviations could be more than coincidence. The per-transistor variations in delay could be fixed times, rather than being frequency dependent.

The error on our measurement of a particular difference, say $t_{22} - t_{21}$ would be $\sqrt{2} \times 2.6\,\text{ps} = 3.6\,\text{ps}$. If this difference were measured with the zero-crossing method, with 1×10^6 zero crossings, the statistical error would be $(155\,\text{ps})\sqrt{256/10^6} = 2.5\,\text{ps}$, which is comparable. The method presented here required data collected from 50 triggers; data for 10^6 zero crossings requires around 50 000 triggers.

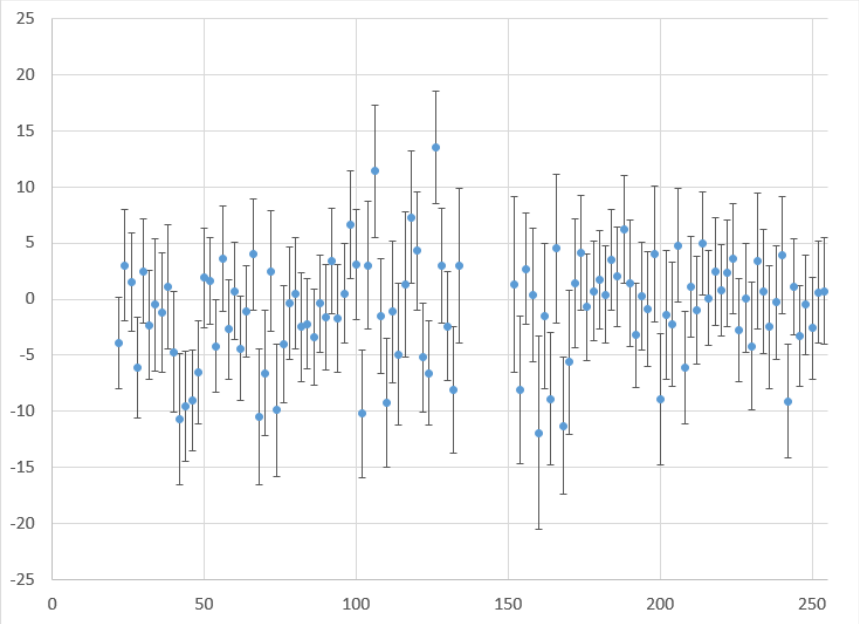


Figure 16: Values of $(t'_{j+1} - t'_j) - (t_{j+1} - t_j)$ are plotted for odd j values. The vertical axis is in picoseconds; the horizontal axis is the cell number j.

To ensure that we are measuring physical properties of a PSEC4 chip, we need to measure the t_j more than once. We do not have any repeated measurements made under the same conditions, but we do have measurements on the modified and unmodified ACDC board 24. We can make the assumption that the different power regulator will not change the cell timing, and compare.

We can compute successive cell time differences $t_{j+1} - t_j$ for the unmodified board, and do the same for the modified board, but call those $t'_{j+1} - t'_j$. These ought to be the same up to statistical errors. Figure 16 shows a plot of $(t'_{j+1} - t'_j) - (t_{j+1} - t_j)$ for odd values of j. (Note that the error on, say, $t_{31} - t_{30}$ is correlated with that of $t_{32} - t_{31}$, so we avoid plotting points with correlated errors.)

The mean of the differences plotted in figure 16 is $-1.0\,\text{ps}$ and the standard deviation is is $4.8\,\text{ps}$. Since we are taking a difference of differences we expected a standard deviation of $\sqrt{2} \times (3.6\,\text{ps}) = 5.2\,\text{ps}$. This shows that our error estimate is not far off.

This accuracy was achieved with only 50 triggers. Using 100 triggers would reduce the errors by a factor of $\sqrt{2}$. We can do even better than that by simply increasing the amplitude of the input sine wave. The s_k values in the denominator of equation 14 are slopes of the sines, and these would increase linearly with the amplitude of the wave. Increasing the wave (peak-to-peak) amplitude from our $0.2\,\text{V}$ to a reasonable $0.6\,\text{V}$ would cut the errors by a factor of 3, so that the successive cell time differences could be measured to an accuracy of $1.2\,\text{ps}$ with only 50 triggers.

12 Code to do the Fitting

Code which automates this whole fitting process has been written in C#, and can be made available on GitHub. It would not be difficult for someone to translate it into Python, if they so desire. An external package, called ALGLIB[7], was used for the linear algebra. Any code that can solve a set of linear equations can be substituted.

References

[1] A. Whitmer, "Proposed Changes to the ACDC Board (A. Whitmer summer report 2018)", LAPPD Document Library

[2] E. Oberla, J. Genat, H. Grabas, H. Frisch, K. Nishimura, G. Varner, "A 15 GSa/s, 1.5 GHz Bandwidth Waveform Digitizing ASIC", arXiv:1309.4397v1

[7] See: www.alglib.net