SuperCDMS Cross Channel Correlated Noise Analysis

Caleb Fink

UNIVERSITY of CALIFORNIA BERKELEY

May 10, 2018

Abstract

The goal of this analysis is to separate the uncorrelated noise from the cross channel correlated noise. The uncorrelated noise can then be used to better estimate the true energy resolution of the detector.

1 Method

Let us assume that the measured signal on any channel can be written as a standard linear model,

$$y_i(t) = \mathrm{A}_i n^c(t) + n_i^{\mathrm{r}}(t), \tag{1}$$

Where $y(t)$ is the measured current as a function of time, $n^{\mathrm{r}}(t)$ is the random intrinsic noise as a function of time, $n_c(t)$ is the correlated noise, A is the weighting coefficient, and the i subscript is the channel index. *Note that this model explicitly assumes that there is only one cross correlated noise source.*[1]

Let us work in frequency space by taking the Fourier transform of Eq. 1.

$$Y_{Ti}(\omega) = \mathrm{A}_i n_T^c(\omega) + n_{Ti}^{\mathrm{r}}(\omega). \tag{2}$$

Let us assume that A is complex, and calculate the cross correlation,

$$S_{ij}(\omega) = \lim_{T\to\infty} \frac{2}{T} \left\langle Y_{Ti}(\omega) Y_{Tj}^*(\omega) \right\rangle. \tag{3}$$

$$S_{ij}(\omega) = \lim_{T\to\infty} \frac{2}{T} \left\langle \left[\mathrm{A}_i n_T^c(\omega) + n_{Ti}^{\mathrm{r}}(\omega)\right] \left[\mathrm{A}_j n_T^c(\omega) + n_{Tj}^{\mathrm{r}}(\omega)\right]^* \right\rangle, \tag{4}$$

$$S_{ij}(\omega) = \lim_{T\to\infty} \frac{2}{T} \left\langle \mathrm{A}_i \mathrm{A}_j^* |n_T^c(\omega)|^2 + n_{Ti}^{\mathrm{r}}(\omega)\left(n_{Tj}^{\mathrm{r}}(\omega)\right)^* + n_{Ti}^{\mathrm{r}}\left(\mathrm{A}_j n_T^c(\omega)\right)^* + \mathrm{A}_i n_T^c(\omega)\left(n_{Tj}^{\mathrm{r}}(\omega)\right)^* \right\rangle, \tag{5}$$

$$\begin{aligned} S_{ij}(\omega) = \lim_{T\to\infty} \frac{2}{T} \Big[&\left\langle \mathrm{A}_i \mathrm{A}_j^* |n_T^c(\omega)|^2 \right\rangle + \left\langle n_{Ti}^{\mathrm{r}}(\omega)\left(n_{Tj}^{\mathrm{r}}(\omega)\right)^* \right\rangle \\ &+ \left\langle n_{Ti}^{\mathrm{r}}\left(\mathrm{A}_j n_T^c(\omega)\right)^* \right\rangle + \left\langle \mathrm{A}_i n_T^c(\omega)\left(n_{Tj}^{\mathrm{r}}(\omega)\right)^* \right\rangle \Big]. \end{aligned} \tag{6}$$

By definition there is no correlation between the cross channel correlated noise and the intrinsic noise on each channel, thus we claim that

$$\left\langle n_T^c(\omega)\left(n_{Tj}^{\mathrm{r}}(\omega)\right)^* \right\rangle = \left\langle n_{Ti}^{\mathrm{int}}\left(n_T^c(\omega)\right)^* \right\rangle = 0. \tag{7}$$

Also, the covariance between the the intrinsic noise for any two channels should be

$$\lim_{T\to\infty} \frac{1}{T} \left\langle n_{Ti}^{\mathrm{int}}(\omega)\left(n_{Tj}^{\mathrm{int}}(\omega)\right)^* \right\rangle = \left(\sigma_i^{\mathrm{int}}\right)^2 \delta_{ij}, \tag{8}$$

[1] If an additional correlated noise source and weighting coefficients were included in the model, we would always end up with more variables than data and system would be under constrained. This will be explored further in future analysis.

Where σ_i^2 is the variance on the i^{th} channel. Similarly, by definition,

$$\lim_{T\to\infty} \frac{1}{T} \left\langle |n_T^c(\omega)|^2 \right\rangle = \sigma_c^2. \tag{9}$$

Using these definitions the cross correlation becomes,

$$S_{ij}(\omega) = 2\mathrm{A}_i\mathrm{A}_j^*\sigma_c^2 + 2\left(\sigma_i^{\mathrm{int}}\right)^2 \delta_{ij}. \tag{10}$$

Since we are letting A be complex, we write $\mathrm{A} = \alpha + i\beta$.

$$S_{ij}(\omega) = 2(\alpha_i + i\beta_i)(\alpha_j - i\beta_j)\sigma_c^2 + 2\left(\sigma_i^{\mathrm{int}}\right)^2 \delta_{ij}, \tag{11}$$

$$S_{ij}(\omega) = 2(\alpha_i\alpha_j + \beta_i\beta_j + i(\beta_i\alpha_j - \alpha_i\beta_j))\sigma_c^2 + 2\left(\sigma_i^{\mathrm{int}}\right)^2 \delta_{ij}. \tag{12}$$

We can look at the Real and Imaginary parts of the cross correlation,

$$\boxed{\begin{array}{l} \mathrm{Re}\left(S_{ij}(\omega)\right) = 2(\alpha_i\alpha_j + \beta_i\beta_j)\sigma_c^2 + 2\left(\sigma_i^{\mathrm{int}}\right)^2 \delta_{ij} \\ \mathrm{Im}\left(S_{ij}(\omega)\right) = 2(\beta_i\alpha_j - \alpha_i\beta_j)\sigma_c^2 \end{array}}. \tag{13}$$

For a detector with n channels, we will have n^2 different equations. We have 3 parameters to fit per channel, plus one additional parameter for σ_c^2, so $3n + 1$ DOF. Therefor we must have 4 channels to perform this analysis.
Without loss of generality, we continue this analysis with $n = 4$
Since we have 4 channels, this means 16 equations and 12 unknowns.
We can construct a matrix of S_{ij}'s,

$$\bar{\mathbf{S}} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} & \mathbf{S}_{14} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} & \mathbf{S}_{24} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} & \mathbf{S}_{34} \\ \mathbf{S}_{41} & \mathbf{S}_{42} & \mathbf{S}_{43} & \mathbf{S}_{44} \end{pmatrix} \tag{14}$$

Since the elements of this matrix are symmetric under exchange of index, we can define 4^2 unique equations by making the lower triangle real and the upper triangle imaginary (and discarding the duplicate data).

$$\bar{\mathbf{S}} = \begin{pmatrix} \mathbf{S}_{11} & \mathrm{Im}\,\mathbf{S}_{12} & \mathrm{Im}\,\mathbf{S}_{13} & \mathrm{Im}\,\mathbf{S}_{14} \\ \mathrm{Re}\,\mathbf{S}_{21} & \mathbf{S}_{22} & \mathrm{Im}\,\mathbf{S}_{23} & \mathrm{Im}\,\mathbf{S}_{24} \\ \mathrm{Re}\,\mathbf{S}_{31} & \mathrm{Re}\,\mathbf{S}_{32} & \mathbf{S}_{33} & \mathrm{Im}\,\mathbf{S}_{34} \\ \mathrm{Re}\,\mathbf{S}_{41} & \mathrm{Re}\,\mathbf{S}_{42} & \mathrm{Re}\,\mathbf{S}_{43} & \mathbf{S}_{44} \end{pmatrix}, \tag{15}$$

where it is understood that the diagonal components will inherently be real.
We can estimate the cross spectral density using the `scipy.signal.csd()` function, which estimates the CSD using the Welch method.
We have N measurements of for each value, so each member of $\bar{\mathbf{S}}$ is a vector of length N. We estimate the CSD of each N elements, calculate the mean and the variance of the CSD's.
We then construct a χ^2 statistic of the form,

$$\chi^2 = [\mathbf{S}_{ij} - \langle \mathbf{S}_{ij} \rangle]^\top \mathbf{C}_{ijk\ell}^{-1} [\mathbf{S}_{k\ell} - \langle \mathbf{S}_{k\ell} \rangle], \tag{16}$$

If we assume there is no autocorrelation within a single channel as a function of frequency, this equation simplifies and we can write the covariance matrix for the $\bar{\mathbf{S}}$ matrix as

$$\mathbf{C}^{-1} = \frac{1}{\sigma_i^2\sigma_j^2}\mathbf{I}. \tag{17}$$

where the $\sigma_{i/j}^2$ are estimated by the variance of the CSD's for the N traces.
Since we assert there is no frequency dependence, we can construct a system of coupled equations at each frequency bin and minimize the χ^2 at each frequency using the `scipy.optimize.least_squares()` function, which is a non-linear least squares solver using the Levenberg-Marquardt method.

2 Fitting The Noise

The fitting of the noise will be done and each frequency bin independently, so each method described below is only in reference to a single frequency. The process will then be repeated for every frequency.

2.1 Residual Function

We will define a residual function

$$\text{res}(\omega)_{ij} = \left[\bar{\mathbf{S}}(\omega)^{\text{data}}_{ij} - \mathbf{S}(\omega)^{theory}_{ij}\right]\sqrt{\frac{N}{\sigma^2_{ij}}}, \tag{18}$$

where $\bar{\mathbf{S}}(\omega)^{\text{data}}_{ij}$ is the mean of all the ij CSDs calculated from `scipy.signal.csd()`, σ^2_{ij} is the calculated std from N estimates of $\mathbf{S}(\omega)^{\text{data}}_{ij}$. $\mathbf{S}(\omega)^{theory}_{ij}$ is the theoretical prediction from the corresponding equation in Eq 13. We will use the convention that if $i <= j$ we use the Real valued equation and if $i > j$ we use the Imaginary version of the equation.
We include a weighting factor of $\sqrt{\frac{N}{\sigma^2_{ij}}}$ in the residual because `scipy.optimize.least_squares()` does not allow the user to provide uncertainties. Since the residual passed to `scipy.optimize.least_squares()` will be squared, we take the square root of the uncertainty estimate to be the weight.

2.2 Initial Guessing of Parameters

We initially guess all the α_i an β_i's to be 0.5 and we bound these parameters between [0,1]. In the corresponding fitting code, these variables are referred to as `reA` and `imA` respectively.
The single correlated noise source ($\sigma^2_c \to$ `sigC`: in the code) is guessed to be

$$\sigma^2_{c\,guess} = \text{average}\left(\frac{1}{2}\frac{\left|\bar{\mathbf{S}}(\omega)^{\text{data}}_{ij}\right|}{\text{corrCoeff}_{\text{ij}}}\right), \quad \text{for } i \neq j \tag{19}$$

where corrCoeff_{ij} is the correlation coefficient of the data for the i^{th} and j^{th} channels.
Lastly, to guess the uncorrelated noise source ($\sigma^2_r \to$ `sigR`: in the code) we calculate

$$(\sigma^r)^2_{i\,guess} = \frac{\left|\bar{\mathbf{S}}(\omega)^{\text{data}}_{ii}\right|}{2} - \sigma^2_{c\,guess} \tag{20}$$

The guessing is done this way because there should only be a contribution from $(\sigma^r)^2$ along the diagonal off the CSD matrix, so for $i \neq j$ we should only have contributions σ^2_c.

3 Performance

3.1 Performance With Fake Data

Four channels of time series data were simulated per the noise model described in this document. A different white noise spectrum was generated for each channel (random noise). On top of this, one set of white noise data was generated (correlated noise) and then summed with the random noise on each channel with a different weighting coefficient A_i.
The combined data were processed through the de-correlation algorithm described above. The fitted results were then compared with the known values. Using 500 traces per channel and 500 samples per trace, the following values were calculated:

Parameter	True Value (arbitrary units)	Value from Fit (arbitrary units)
Intrinsic Noise ch A	$(3.2 \pm 0.1) \times 10^{-12}$	$(3.2 \pm 0.3) \times 10^{-12}$
Intrinsic Noise ch B	$(1.28 \pm 0.06) \times 10^{-11}$	$(1.27 \pm 0.09) \times 10^{-11}$
Intrinsic Noise ch C	$(7.2 \pm 0.3) \times 10^{-12}$	$(7.2 \pm 0.6) \times 10^{-12}$
Intrinsic Noise ch D	$(2.4 \pm 0.1) \times 10^{-11}$	$(2.4 \pm 0.2) \times 10^{-11}$
Correlated Noise	$(1.15 \pm 0.05) \times 10^{-11}$	$(1.2 \pm 0.1) \times 10^{-11}$

We see that the fitted values are in agreement with the known values.

3.2 Performance With Real Data

Applying this algorithm to real data, we can see that the sum of the correlated and uncorrelated parts of the noise spectrum perfectly match the data. Furthermore, we see that the results agree with intuition from looking at the correlation coefficients. From the correlations, we would assume that the increased low frequency noise below 1kHz is due to correlated noise.

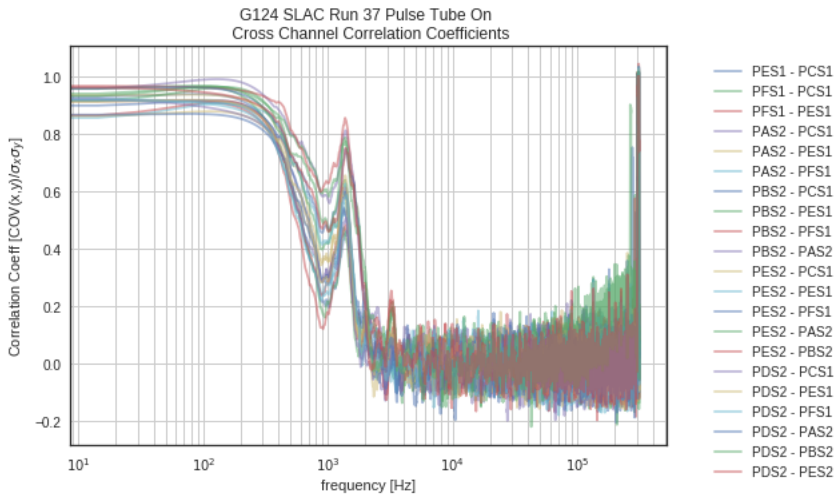

Figure 1: Correlation Coefficients of detector G124 from Run 37 at SLAC

We see that Once the correlated noise is removed, our spectrum becomes mostly flat below the τ_{eff} pole (The knee around 1-3kHz for this detector). This implies that this detector is behaving as we would expect, we are dominated by thermal fluctuation noise on the TES.

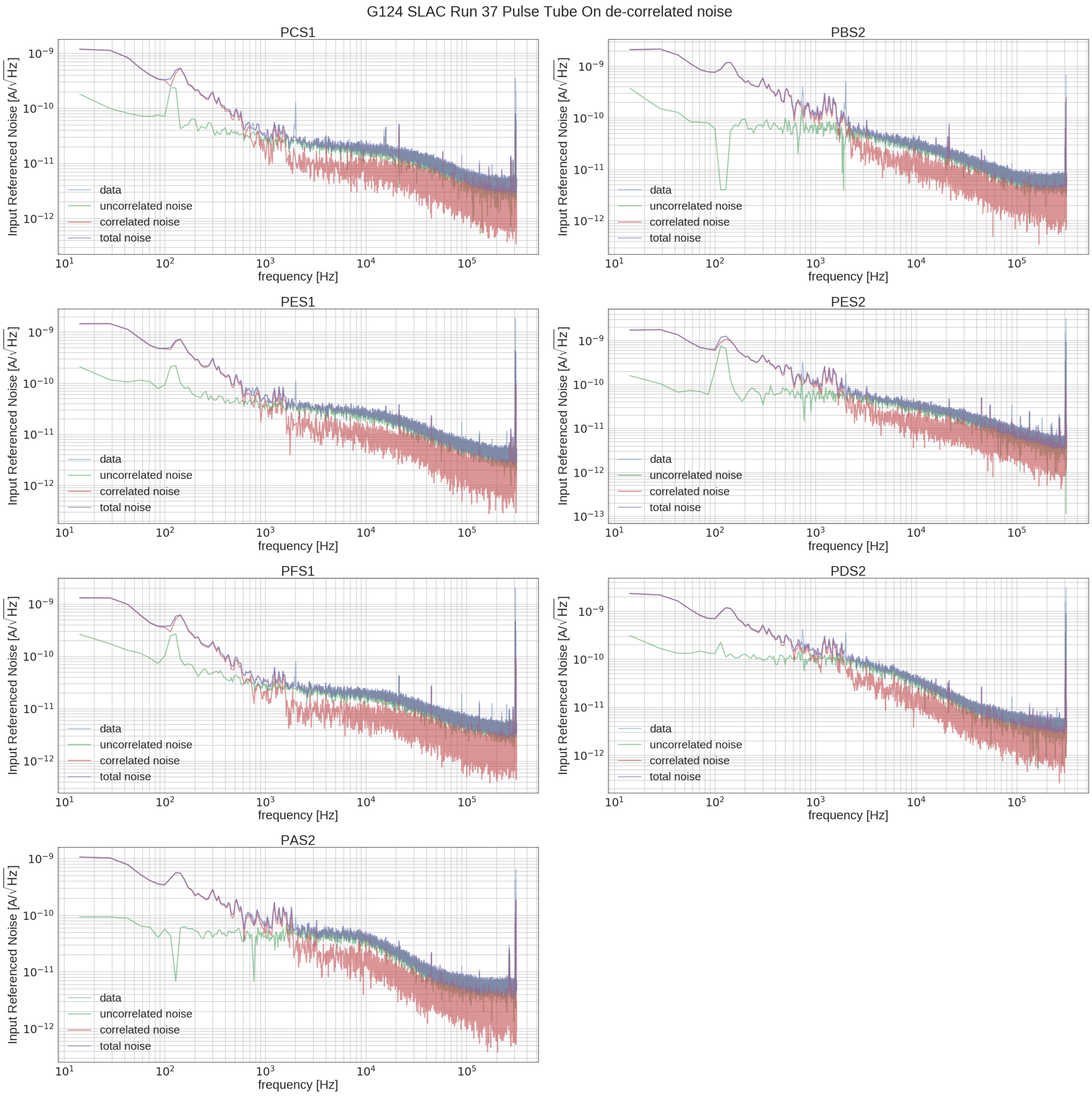

Figure 2: Plot of correlated and uncorrelated component of the noise spectrum from detector nG124 from Run 37 at SLAC