The NxM Optimal Filter

Yan Liu, Scott Oser

July 6, 2022
v2.2

Revision History

Revision	**Date**	**Author(s)**	**Description**
1.0	2022Apr11	YL, SO	created
2.0	2022May02	YL, SO	updated derivation, restructuring
2.1	2022May06	YL, SO	minor corrections
2.2	2022Jul06	YL, SO	correcting history

Abstract

Optimal filtering is at the core of data processing for SuperCDMS, and its performance has direct implications on many aspects of data analysis including energy resolution, position sensitivity, and others. The NxM optimal filter fits pulses from N channels simultaneously with M templates (N and M are the numbers of channels and templates, respectively). In this document, we start with an overview of the optimal filtering theory and many of its algorithms that have been used in the collaboration. Then, the motivations for the NxM optimal filter are discussed, followed by a new derivation of the algorithm. Finally, we note some technical peculiarities in its implementation in CDMSBats.

1 Introduction to the Optimal Filter

Optimal filtering in the context of SuperCDMS data processing is a means of non-adaptive extraction of a weak desired signal in the presence of noise. Mathematically speaking, it deals with the generic problem of $D(t) = aT(t-t_0) + N(t)$, where $D(t)$ is the data, a linear combination of signal and noise, $T(t)$ is the template that has approximately the same pulse shape as the true signal, and N(t) is the noise. An important assumption here is that we have some prior knowledge of the signal shape, which differs from most signal processing considerations such as the Wiener filter in electrical engineering [1]. The name "optimal" comes from the fact that, if the given template perfectly describes the signal and if the noise is *stationary* and *Gaussian*[1], the filter method will render theoretically the best possible fit results (amplitude a and time delay t_0).

A SuperCDMS event is defined as an instance of energy deposition in a detector. These deposited energies transverse the detector in the form of electrons/holes and phonons, which in turn are collected in either the charge or phonon channels. As a result, a raw event consists of multiple different copies of digitized pulses, embedded with information about the same event. For details, refer to chapter 4 of [2].

The default and simplest Optimal Filter (OF) used by the collaboration is the single-channel, single-template, stationary Optimal Filter (1x1OF). In addition to applying it to individual channels, most often the OF is used on the sum of all individual (phonon) channels, fitted as a virtual "PT" channel. In this case, only one trace needs to be dealt with, making $S(t)$, $A(t)$, and $N(t)$ effectively vectors. Appendix B of [3] and Appendix E of [5] present two different flavors of 1x1OF construction that arrive at the same formulae. Appendix E of [5] emphasized more explicitly the issue at hand as a χ^2 minimization problem. This will continue to be the rhetoric for deriving the NxM formula.

The actual application of the 1x1OF in the SuperCDMS processing software (CDMSBats) inserted two additional caveats. The first is that instead of searching for the minimum of the χ^2, we are looping through amplitude as a function of time (bins) to find the maximum amplitude[2]. The second is the use of the Inverse Fast Fourier Transform (IFFT) to convert amplitudes from frequency space to time. Both have proved to be significant at improving the processing speed but cannot be generalized to multiple-channel, multiple-template cases. Appendix A of [6] briefly touched on this topic and we will come back to these two caveats when discussing the NxM optimal filter as well.

Over the years many variants of OF were developed with different focuses on some particular characteristics of the data, with strong contributions from Pyle and others. One natural extension of the 1x1OF is the single-channel, two-templates, stationary Optimal Filter (2TOF, sometimes also referred to as 1x2OF). Equipped with both primary and residual templates, it excels at pulses with varying shapes as was observed in the CDMSlite dark matter search. Additionally, it was demonstrated that a special combination of the two fitted amplitudes is indicative of the event position [7]. Derivation and performance evaluation of the 2TOF can be found in [8] and Appendix E of [5].

[1] *Stationary* noise is defined as the noise whose power spectral density does not change with time. A consequence of noise being stationary is that it will be uncorrelated between different frequencies. *Gaussian* means a sample at any given time is drawn from a Gaussian distribution. Note that they differ distinctively from *white*. *White* noise refers to noise whose Fourier transform (and thus the power spectral density) has the same amplitude at all frequencies. It is sometimes also referred to as *Stochastic*.

[2] The equivalence holds in some special cases, but generally there is a small systematic uncertainty associated with it.

Another logical extension is to include more than one channel. This is of special importance in charge channel processing, where the cross-talk between different channels is significant. The two-channel, two-template[3], stationary Optimal Filter(2x2OF), combining two charge channels with the aid of two additional cross-talk templates, is used as the default OF for processing charge data. Appendix A of [6] gave the derivation and performance evaluation. In this derivation, matrix notations were first used and several complications similar to the NxM case were recognized and discussed for the first time.

The single-channel, single-template, non-stationary optimal filter (NSOF), as its name suggests, deals with the case of non-stationary signals, i.e., signals with varying pulse shapes. Because only one template is used, the non-stationarity effectively translates from non-stationary signals to non-stationary noises. This breaks the underlining assumption that noise at different frequencies is independent, and therefore can contribute significantly to the statistical uncertainty of fitted parameters when using the 1x1OF. The NSOF was first designed to deal with the position-dependent pulse shape variation in the Soudan iZIP detectors, with the general idea of de-weighting the initial rising part of the pulse. The construction can be found in Appendix E of [5] and Chapters 6&7 of [9].

2 NxM Optimal Filter: Motivations

Since almost the very start of the CDMS era, it was noticeable that pulses of the same channel for different events can have different amplitudes, time delays, and perhaps most importantly pulse shapes. These sometimes intricate differences originate from the fact that events can occur at different positions in the detector. Intuitively, the closer the event is to one channel, the bigger the amplitude, the smaller the time delay (relative to other channels), and the sharper the rise time for that pulse in the channel. Similarly, pulses in different channels of the same event also have different amplitudes, time delays, and pulse shapes. Without prior knowledge of the event position and thus the *correct* template to use for the channels, fitting for amplitudes and time delays proves to be tricky. As stated in the foregoing section, the OF becomes *non-optimal* when the template can not perfectly describe the signal. This is reflected as a bias and a worsening of resolution in the fitted parameters. The historical solution to this problem has been to try to wash out the position dependence effects by fitting the summed pulse of all channels instead of individual channels. The 2TOF and NSOF also contribute to mitigating this effect, through either providing more than one template or including the variation in pulse shape as a systematic uncertainty contribution to the covariance matrix (resulting in a non-diagonal covariance matrix).

In light of the bigger size of the SNOLAB-type detectors, these effects are expected to be enhanced. Viewed from a different perspective, however, things are

[3]It is sometimes also referred to as four-template, because all four templates (two for each channel) are different from each other

not that grim. If dealt with properly, these amplitudes, time delays, and pulse shape variations can not only be correctly accounted for to improve the energy resolution, but also provide a valuable probe sensitive to the event position.

The NxM optimal filter (NxMOF), processing multiple (N) pulses of a detector event with multiple (M) templates, was proposed precisely for this reason. With this algorithm, all the (phonon) channels are processed simultaneously so that the correlated noise such as cross-talk between channels is correctly taken into account. To achieve this, a covariance matrix (the equivalent of the PSD in the case of a single-channel fit such as the 1x1OF) needs to be calculated, with the diagonal elements being the auto-PSDs of each channel and the off-diagonal elements the cross-PSDs between different channels. Additionally, each channel is fitted with multiple templates, so that the pulse shape variation can be correctly represented. The amplitudes of different templates adjust as the pulse shape varies. Together with the best fitted time delays for each channel, a figure-of-merit may be constructed to indicate the event position.

The idea of the NxMOF was not new. In fact, it has been a long appealing concept within the collaboration. Like most of the advanced OFs beyond the 1x1OF, Pyle first developed the NxM in Matlab in 2012 [4]. By 2015, Thakur, with the help of Pyle, had documented the first mathematical derivation on the NxMOF [5]. In 2018, Asamar wrote some python package [10] that includes both the algorithm itself as well as a template generation scheme. Simulated data were used to study a potential bias in the reconstructed energy. By the end of 2020, one of the authors (Liu) implemented the first workable version of the NxMOF in CDMSBats as part of the official data processing.

However, as more NxMOF studies were carried out, it became clear that the previous NxMOF derivation had several substantial limitations, and sometimes misconceptions. First and foremost, the notion of the complex covariance matrix is problematic. One of the intrinsic properties of the (complex) covariance matrix is that it is Hermitian (or symmetrical if the matrix is real). This means that the complex covariance matrix becomes non-invertible[4] if the size of the matrix is odd, which can potentially place some quite stringent limit on the application of the NxMOF[5]. It is also hard to interpret a complex χ^2, or derive a real χ^2 from a complex covariance matrix. Finally and perhaps more importantly, the complex covariance matrix intrinsically miscounts some important correlations. This can be shown by calculating the covariance between two samples of complex numbers, z_1 and z_2. Suppose $z_1 = a + bi$ and $z_2 = c + di$, where a, b, c, and d are real random variables, and μ_1 and μ_2 are the mean values of z_1 and z_2, respectively (e.g. $\langle z_1 \rangle = \langle a \rangle + \langle b \rangle i$). The complex covariance can be

[4]The determinant of an odd Hermitian matrix can be shown to be zero.

[5]One of the examples is that the NxMOF cannot be applied to detectors with odd-number channels.

written as:

$$
\begin{aligned}
cov(z_1, z_2) &= \frac{1}{N}\sum_{j=1}^{N}(z_{1,j} - \mu_1)(z_{2,j} - \mu_2)^* \\
&= cov(a, c) + cov(b, d) - [cov(a, d) - cov(b, c)]i,
\end{aligned}
\tag{1}
$$

where the real and imaginary parts of the complex covariance are linear functions of the covariances between the real and imaginary components of the complex samples themselves. Importantly, the imaginary component of this complex covariance is the difference between two covariances, which means that in some extreme cases, these correlations can cancel each other and are not accounted for at all. In fact, as long as $cov(a, d)$ or $cov(b, c)$ is not sufficiently small, the complex covariance shown above (and used in the previous derivation) miscalculates these correlations, namely the covariances between the real component of one channel and the imaginary component of the other channel. The size of these covariances is reflective of the time delay difference between the two channels, which is usually non-negligible.

Another drawback in the previous derivation is in the final amplitude expression. The amplitudes were given per template, whereas sometimes it is useful to have one amplitude for every channel and every template[6]. Depending on the input templates, the relative per-channel amplitude could potentially carry rich information about the event position. Furthermore, it is unclear how external constraints on the amplitude parameters should be applied. We formulate our new derivation so that different schemes of fit parameters can be accommodated.

The previous derivation also did not allow different channel time delays. Its NxMOF χ^2 only admits a single time delay, t_0, and thus inherently assumes that all pulses of the same event start at the same time. This could turn out to be a significant limitation to the application of NxMOF, especially when it comes to large detectors. When fitting a multiple-channel detector with 1x1OF, the difference in time delays for individual channels can arise from two sources: the time it takes for an event to traverse the detector to arrive at different sensors, and the "spillover" effect from (not correctly accounting for) the pulse shape variation. Using a single time delay is only justifiable in the NxMOF when the effect from the first source is negligible. For large detectors, this is generally not true. Take a SuperCDMS SNOLAB Ge HV detector (100 mm diameter, 33.33 mm thick) as an example. The phonon propagation velocity is $\sim$4000 m/s[7]. The time delay difference between channels contributed solely from the first source can be as high as $\frac{100\text{mm}}{4000\text{m/s}} = 25\mu$s, which is $\sim$16 time bins (the DAQ sampling rate is 625 kHz)[8]. In fact, the time delay difference between channels

[6]These amplitudes would be correlated, and are ultimately determined by the position and energy of the energy deposit through a possibly non-linear mapping.

[7]It is 5324.2077 m/s for longitudinal speed and 3258.7879 m/s for transverse speed. Si generally has faster phonon velocities, at 9000 m/s longitudinal and 5400 m/s transversal.

[8]Coincidentally, the downsampled pulses for SuperCDMS SNOLAB has a downsampling factor of 16, which would have justified the use of only a single overall time delay. However, the on-pulse region is not downsampled (for good reasons).

proves to be one of the most sensitive tools for inferring the event position.

In the following section, a new derivation for the NxMOF is laid out. In addition to discussing how the concerns listed above are addressed, we intend it to be a comprehensive, self-contained mathematical derivation. Perhaps one of the most distinctive differences from the previous derivation is that we resort to using matrix notation for our entire derivation.

Finally, we consider the template generation, as well as the question of how to calculate a figure-of-merit for event position, outside the scope of this document. Although a position indicator is one of the ultimate goals of NxMOF, extensive simulation and calibration are required to better understand the detector response and relate template/pulse shape to event position. As of now, neither is in place for detailed studies. We therefore constrain the discussion to the NxMOF algorithm itself: a method of retrieving the best fitted amplitudes and time delays given a collection of templates.

3 Derivation of NxMOF

3.1 The NxM χ^2

Suppose that we have N channels to which we fit M template shapes. Following the logic in 1x1OF, we expect an amplitude and a time delay for each template of each channel. Presumably we can then write the prediction for the data $\hat{d}_c$ in channel c in the time domain as:

$$\hat{d}_c = a_{c1}T_{c1}(t_{c1}) + a_{c2}T_{c2}(t_{c2}) + ... + a_{cM}T_{cM}(t_{cM}) = \sum_{i=1}^{M} a_{ci}T_{ci}(t_{cM}). \quad (2)$$

Here we suppressed any index for the time sample. T_{ci} represents the i^{th} template for channel c and a_{ci} (t_{ci}) is the amplitude (time delay) of that template in the model prediction. As is the case for the 1x1OF, the χ^2 has a nonlinear dependence on t_{ci} and therefore it is a good idea to discuss separately how to deal with time delays and amplitudes.

Eqn. 2 is written in most general terms and allows for $N \times M$ different time delays, one for each template of each channel. This, however, potentially introduces some significant degeneracy (with amplitudes for different templates of the same channel) and can result in multiple local minimums and eventually unstable fits. We therefore restrain all templates from the same channel to have the same time delay[9]. On the other hand, as was shown previously, it is necessary to have channel-dependent time delay terms. Notwithstanding, using only a single time delay term across the fit (as was the case in the previous derivation) is just a special case and is automatically included in this setup.

Searching for the best fitted time delay(s) is a non-linear problem for all OFs. The NxMOF is further complicated by the absence of an equivalent IFFT in the

[9]This differs from 1x2OF, where the fast and slow templates have their own time delays.

matrix format, which prohibits the application of the trick used in the 1x1OF to calculate amplitudes as a function of time, $a(t)$. Without such a shortcut, one of the few viable methods for obtaining the best fitted time delay is to calculate and compare χ^2 using brute force. This is likely to be computationally very expensive. Take the example of a 12-channel detector. In order to search through a time window of ± 5 bins for each channel, we need to calculate the χ^2 for $12^{10} \approx 10^{11}$ times. In this document we do not cover the methods to obtain the best fitted time delays as more studies are evidently needed to optimize this process[10]. For now, we simply assume that the best fitted time delays are known *a priori* (from fit results of other OFs such as the 1x1OF)[11].

Before we dive into the amplitudes, let us first note down some characteristics about the NxMOF templates. In Eqn. 2, we allow different channels to have different templates, by attaching two indices to T_{ci}. This is not necessarily needed, especially when taking into consideration the physics in SuperCDMS. In this case, using the same set of templates for all channels would reduce T_{ci} to T_i. However, it is not entirely impossible that future applications find it beneficial to have different templates assigned to different channels. Channels of the same detector are hardly "identical". Aside from electronic arrangement, the surface area of each channel covers a different shape, which may show up as a non-negligible effect in the pulse shapes. The cross-talk templates, if included, can also be different for different channels. Therefore, we try to be inclusive by denoting T_{ci} in the derivation. Whether the templates are the same for all channels is therefore purely a decision at the application stage (i.e., what templates are provided), not the derivation.

Eqn. 2 also dictates that all channels have the same number (M) of templates. This places hardly any constraint, though. In practice, one can always first identify the maximum number of templates needed for any channel and make it equal to M. All other channels with fewer numbers of templates can then be padded out with some random templates with their amplitudes fixed to zero (see below).

There are also $N \times M$ amplitude coefficients a_{ci} in Eqn. 2, but they are not all independent. In fact, the correlation between a_{ci} can be very complicated, and the ultimate energy or position estimators will be expressed in a non-linear mapping of a_{ci}. Nonetheless, external constraints can also be placed to reduce the number of degrees of freedom before running the fit. For example, we could require that a template has the same amplitude on every channel (e.g. $a_{ci} = C_i$), or we may not want to fit every template to every channel, in which case some of the coefficients can be set to zero.

Assume these external constraints are linear and suppose that we have P free parameters. We define a mapping from these P parameters to the $N \times M$

[10]Some interesting methods include the inverse Hartley transform and the Levenberg-Marquardt algorithm. One can also search through a carefully chosen small subset of time delay combinations.

[11]There is of course a systematic uncertainty associated with this assumption.

amplitude coefficients a_{ci}:

$$a_{ci} = \sum_{k=1}^{P} B_{c,ik} P_k. \qquad (3)$$

Here P_k is the value of the k^{th} parameter, and $\mathbf{B_c}$ is a matrix of dimension with M rows and P columns. There can be a separate such matrix for each channel, defining how the P free parameters map to the M template coefficients for that channel.

We can rewrite Eqn. 2 in matrix form as a function of $\vec{P}$, which is a column vector containing all the free parameters. Let matrix $\mathbf{T}_c(t_c)$ represent the values of all templates of channel c, where t_c is the time delay for channel c (remember templates of the same channel share the same time delay). The data prediction for channel c in the time domain is then,

$$\hat{d}_c = \mathbf{T}_c(t_c) \cdot \mathbf{B}_c \cdot \vec{P}. \qquad (4)$$

If we take a Fourier transform to the frequency domain, then at a given frequency f we have,

$$\hat{\tilde{d}}_{c,f} = \mathbf{R}(t_c) \cdot \tilde{\mathbf{T}}_{c,f} \cdot \mathbf{B}_c \cdot \vec{P}. \qquad (5)$$

Now $\hat{\tilde{d}}_{c,f}$ and $\tilde{\mathbf{T}}_{c,f}$ are complex vectors/matrices, and we have explicitly written their frequency dependence. $\mathbf{R}(t_c)$ is a rotation matrix that rotates the template $\tilde{\mathbf{T}}_{c,f}$ in the complex plane by a phase $\phi = 2\pi f t_c$. This is equivalent to doing a mapping such as:

$$\mathrm{Re}(\tilde{T}_{ci}) \rightarrow \cos\phi\ \mathrm{Re}(\tilde{T}_{ci}) + \sin\phi\ \mathrm{Im}(\tilde{T}_{ci})$$

$$\mathrm{Im}(\tilde{T}_{ci}) \rightarrow -\sin\phi\ \mathrm{Re}(\tilde{T}_{ci}) + \cos\phi\ \mathrm{Im}(\tilde{T}_{ci}).$$

for every template of channel c. $\mathbf{R}(t_c)$ is not dependent on either templates nor frequency.

We want to calculate the contribution to the χ^2 at frequency f for a single channel c, which we'll denote $\chi^2_{c,f}$. Because the data $\tilde{d}_{c,f}$ is complex, we need to include both the real and complex components in the χ^2. Since we only have a single channel to deal with, we can write:

$$\chi^2_{c,f} = \frac{|\tilde{d}_{c,f} - \mathbf{R}(t_c) \cdot \tilde{\mathbf{T}}_{c,f} \cdot \mathbf{B_c} \cdot \vec{P}|^2}{S_{c,f}}. \qquad (6)$$

where $S_{c,f}$, a scalar, is the (variance) noise of channel c at frequency f. Note that all of $\tilde{d}, \tilde{\mathbf{T}}$, and S are functions of channel and frequency.

This is equivalent to treating the real and the imaginary parts of the data residual as two independent data points, each of which must be fit to the data. So if we define the residual between data and model production as $\Delta_{c,f} = \tilde{d}_{c,f} - \hat{\tilde{d}}_{c,f} = \tilde{d}_{c,f} - \mathbf{R}(t_c) \cdot \tilde{\mathbf{T}}_{c,f} \cdot \mathbf{B_c} \cdot \vec{P}$, and writing out explicitly, we get

$$\chi^2_{c,f} = \frac{(\mathrm{Re}(\Delta_{c,f}))^2 + (\mathrm{Im}(\Delta_{c,f}))^2}{S_{c,f}} \qquad (7)$$

Eqn. 7 gives the χ^2 for the single-channel, multiple-template, and single-frequency case.

Let us first consider expanding the χ^2 to include multiple channels, where the correlations between channels are to be taken into account. These correlations can be expressed in a covariance matrix in place of $S_{c,f}$. Note that the noise covariance matrix is itself a function of frequency.

For a given frequency, we need a covariance matrix of size $2N \times 2N$ to encode the covariances between the real and imaginary components of the data from the N channels. Note that some off-diagonal elements of this matrix may be zero, especially between the real and imaginary terms of the same channel (if the noise is stationary)[12]. Let us denote this covariance matrix as $\mathbf{V}_{j\alpha,k\beta}$ where α and β can represent either the real or imaginary parts of the data for the channels j and k, respectively. Likewise we denote the real and imaginary parts of Δ_c with a Greek subscript. We then get this expression for χ_f^2:

$$\chi_f^2 = \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{\alpha=\mathrm{Re,Im}} \sum_{\beta=\mathrm{Re,Im}} \Delta_{j\alpha,f} \Delta_{k\beta,f} (\mathbf{V}^{-1})_{j\alpha,k\beta,f}. \tag{8}$$

The summation is over channel and real/imaginary components. Note that this is really just the inverse covariance matrix $\mathbf{V}_f^{-1}$ (dimension $2N \times 2N$, an entirely real matrix) multiplied on either side by a vector of length 2N representing the real and complex components of the residuals Δ_f.

To write Eqn. 8 in matrix format, let us first (re)define some terms. Let $\tilde{d}_f$ be a column vector of the size $2N$. $\tilde{d}_f$ contains data from all channels at frequency f in sequential order, alternating between real and imaginary components:

$$\tilde{d}_f = \begin{pmatrix} \mathrm{Re(chan1\ FFT)} \\ \mathrm{Im(chan1\ FFT)} \\ \mathrm{Re(chan2\ FFT)} \\ \mathrm{Im(chan2\ FFT)} \\ \vdots \\ \mathrm{Re(chanN\ FFT)} \\ \mathrm{Im(chanN\ FFT)} \end{pmatrix} \tag{9}$$

Similarly, we can redefine terms in Eqn. 5 to accommodate multiple-channel cases. $\mathbf{R}(\vec{t})$, now a function of the time delay vector $\vec{t}$, is extended into a block diagonal matrix where each block is a 2×2 rotation matrix, and all other entries

[12]The general observation from data is that these terms are close to zero but not exact - a reminder that the stationary noise assumption is not strictly met.

outside these 2×2 blocks are zero:

$$\mathbf{R}(\vec{t}) = \mathbf{R}(t_1, t_2, \cdots, t_N)$$
$$= \begin{pmatrix} \cos\phi(t_1) & \sin\phi(t_1) & & & & & & \\ -\sin\phi(t_1) & \cos\phi(t_1) & & & & & \multicolumn{2}{c}{0} \\ & & \cos\phi(t_2) & \sin\phi(t_2) & & & & \\ & & -\sin\phi(t_2) & \cos\phi(t_2) & & & & \\ & & & & \ddots & \ddots & & \\ & & & & \ddots & \ddots & & \\ & & & & & & \cos\phi(t_N) & \sin\phi(t_N) \\ \multicolumn{2}{c}{0} & & & & & -\sin\phi(t_N) & \cos\phi(t_N) \end{pmatrix} \quad (10)$$

Each 2×2 block is determined by a t_i, which is the time delay for that channel.

$\tilde{\mathbf{T}}_f$ is the template matrix consisting all templates (Fourier transformed) and $\mathbf{B}$ again is a (larger) transformation matrix, now for all channels. $\tilde{\mathbf{T}}_f$ and $\mathbf{B}$ are closely coupled and in principle can have some flexibility in their exact form as long as they combine to express the same underlining physics. One of the configurations for $\tilde{\mathbf{T}}_f$ is as such,

$$\tilde{\mathbf{T}}_f = \begin{pmatrix} \mathrm{Re}(\tilde{T}_{11}) & \cdots & \mathrm{Re}(\tilde{T}_{1M}) & & & & & & & \\ \mathrm{Im}(\tilde{T}_{11}) & \cdots & \mathrm{Im}(\tilde{T}_{1M}) & & & & & & 0 & \\ & & & \mathrm{Re}(\tilde{T}_{21}) & \cdots & \mathrm{Re}(\tilde{T}_{2M}) & & & & \\ & & & \mathrm{Im}(\tilde{T}_{21}) & \cdots & \mathrm{Im}(\tilde{T}_{2M}) & & & & \\ & & & & & & \ddots & & & \\ & & & & & & \ddots & & & \\ & & & & & & & \mathrm{Re}(\tilde{T}_{N1}) & \cdots & \mathrm{Re}(\tilde{T}_{NM}) \\ & 0 & & & & & & \mathrm{Im}(\tilde{T}_{N1}) & \cdots & \mathrm{Im}(\tilde{T}_{NM}) \end{pmatrix} \quad (11)$$

which has a size of $(2N) \times (N \times M)$. In this case, $\mathbf{B}$ becomes a $(N \times M) \times (N \times M)$ identity matrix if $\vec{P}$ is a column vector of the $N \times M$ amplitudes in the following form:

$$
\begin{aligned}
\vec{P} &= \vec{a} \\
&= \begin{pmatrix} a_{11} \\ \vdots \\ a_{1M} \\ \vdots \\ \vdots \\ a_{N1} \\ \vdots \\ a_{NM} \end{pmatrix}
\end{aligned}
\tag{12}
$$

Note that both $\mathbf{B}$ and $\vec{P}$ (a linear mapping of the $N \times M$ amplitudes) are independent of frequency.

With the above definitions, we can extend Eqn. 5 to the multiple-channel cases:

$$
\begin{aligned}
\hat{\tilde{d}}_f &= \mathbf{R}(\vec{t}) \cdot \tilde{\mathbf{T}}_f \cdot \mathbf{B} \cdot \vec{P} \\
&= \mathbf{A} \cdot \vec{P}
\end{aligned}
\tag{13}
$$

where we further define $\mathbf{A} \equiv \mathbf{R}(\vec{t}) \cdot \tilde{\mathbf{T}} \cdot \mathbf{B}$, which is effectively a transformation matrix that projects fitted parameters to data. $\mathbf{A}$ has the size of $2N \times P$, and has the form,

$$
\mathbf{A} = \begin{pmatrix} \rotatebox{90}{data prediction for all channels due to P_1} & \rotatebox{90}{data prediction for all channels due to P_2} & \dots & \rotatebox{90}{data prediction for all channels due to P_P} \end{pmatrix}
\tag{14}
$$

Fig. 1 is a schematic of Eqn. 13, showing additionally the size of the matrices.

Now we can write out the χ^2 for multiple-channel, multiple-template, and single-frequency cases in matrix form:

$$
\begin{aligned}
\chi_f^2 &= (\tilde{d}_f - \hat{\tilde{d}}_f)^T \; \mathbf{V}_f^{-1} \; (\tilde{d}_f - \hat{\tilde{d}}_f) \\
&= (\tilde{d}_f - \mathbf{A}_f \cdot \vec{P})^T \; \mathbf{V}_f^{-1} \; (\tilde{d}_f - \mathbf{A}_f \cdot \vec{P})
\end{aligned}
\tag{15}
$$

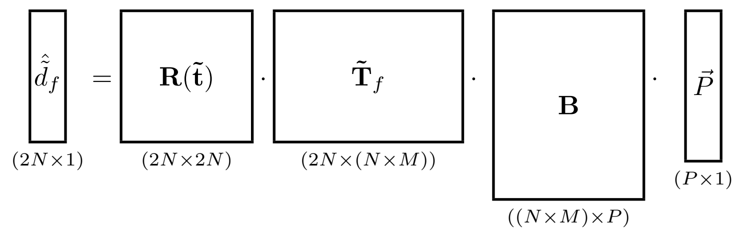

Figure 1: Schematic showing the size of various matrices for the single-frequency case.

We've written explicitly the dependence on frequency. Note that Eqn. 15 is equivalent to Eqn. 8.

To extend Eqn. 15 to multiple-frequency case, in principle we need to increase the size of the vectors and matrices in Eqn. 13 accordingly. Suppose the pulses have F sampling points/frequencies, the covariance matrix $\mathbf{V}$(including all frequencies), for example, will have the following form:

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{f=1} & \mathbf{cov}_{\mathrm{f}=1,2} & \dots & \mathbf{cov}_{\mathrm{f}=1,F} \\ \mathbf{cov}_{\mathrm{f}=1,2} & \mathbf{V}_{f=2} & \dots & \mathbf{cov}_{\mathrm{f}=2,F} \\ \dots & \dots & \ddots & \dots \\ \mathbf{cov}_{\mathrm{f}=1,F} & \mathbf{cov}_{\mathrm{f}=2,F} & \dots & \mathbf{V}_{f=F} \end{pmatrix} \tag{16}$$

where the diagonal elements are the $N\times N$ covariance matrices of all channels for a given frequency, and $\mathbf{cov}_{f=i,j}$ (i, j are natural numbers) is the $N \times N$ covariance matrix between frequency i and j. $\mathbf{V}$ in this case has the size of $(2N \times F) \times (2N \times F)$.

In reality $\mathbf{V}$ (and other vectors and matrices) can be extremely huge. The typical physics pulses in SuperCDMS have 32768 time sampling points ($N_f \approx 32768$), and for a 12-channel detector, this means a size of about 800000×800000 for $\mathbf{V}$. Inverting such a matrix is impractical, let alone the rest of the matrix multiplications.

An important assumption can come in handy. If the noise is stationary, then the frequencies are not correlated. In this case, $\mathbf{cov}_{f=i,j} = 0$ for all i, j. The χ^2_f is also independent of frequencies, and we can simply sum Eqn. 15 over all frequencies to get the total χ^2,

$$\begin{aligned} \chi^2(\vec{t},\vec{P}) &= \sum_f \chi^2_f(\vec{t},\vec{P}) \\ &= \sum_f (\tilde{d}_f - \mathbf{A}_f \cdot \vec{P})^T \ \mathbf{V}_f^{-1} \ (\tilde{d}_f - \mathbf{A}_f \cdot \vec{P}) \end{aligned} \tag{17}$$

Eqn. 17 gives the χ^2 expression as a function of time delays $\vec{t}$ ($\mathbf{A}$ is dependent on

$\vec{t}$) and amplitudes $\vec{P}$ for the multiple-channel, multiple-template, and multiple frequency case.

3.2 Minimizing the χ^2

Before we embark on minimizing the χ^2 for NxMOF, it is helpful to draw a comparison with the 1x1OF. Both OFs minimize the $\chi^2(t, a)$ separately for time delay $t(\vec{t})$ and amplitude $a(\vec{P})$ due to the non-linear dependence on the time delay. For the 1x1OF, the best amplitude is first calculated for every time delay via an analytical solution, which is derived by calculating $\frac{\partial}{\partial a}\chi^2 = 0$. Similarly for the NxMOF, we will derive an analytical solution for $\vec{P}$, which turns out to follow the same form as the 1x1OF.

For the 1x1OF, searching for the best fitted time delay rests on two important assumptions. First, we can make use of the IFFT transformation to convert amplitudes in frequency space to time. This saves a significant amount of computation required to construct such a function. Without this technique, different time delays mean effectively different templates, and quantities that otherwise could be calculated one time now have to be re-calculated for every time delay. Second, it so happens that $\frac{\partial}{\partial t}\chi^2 = \frac{\partial}{\partial t}a$, so that minimizing the χ^2 is equivalent to maximizing the amplitude.

Neither assumption holds for the NxMOF. There is no IFFT equivalence because we changed to use real covariance matrices to properly account for the correlations. It is also unclear what "maximum amplitude" means in the NxMOF. Maximizing any specific amplitude will disproportionally increase that amplitude in the overall fit; and maximizing the sum of all amplitudes does not guarantee a minimum χ^2 either. This is because each amplitude carries different weights (information on the pulse shape) and in general can not be arithmetically summed. It may be possible to figure out the weights and put them in $\mathbf{B}$, but we defer that to further studies. As a compromise, for now we use the best fitted time delay from the 1x1OF in the NxMOF as a fixed input, and turn our focus to minimizing the NxM χ^2 with respect to $\vec{P}$.

To do so, the most straightforward way is to use a minimizer and a good choice would be the MINUIT2 package in ROOT [11]. However, this is time-consuming and soon becomes impractical when dealing with a large number of channels and templates. On the other hand, because $\tilde{d}_f$ depends linearly on $\vec{P}$ if $\vec{t}$ is fixed (Eqn. 13), there exists an analytical solution for $\frac{\partial}{\partial \vec{P}}\chi^2 = 0$.

Starting with Eqn. 17, we can write $\frac{\partial}{\partial \vec{P}}\chi^2$ as,

$$\begin{aligned}\frac{\partial}{\partial \vec{P}}\chi^2 &= \frac{\partial}{\partial \vec{P}}\sum_f (\tilde{d}_f - \mathbf{A}_f \cdot \vec{P})^T\ \mathbf{V}_f^{-1}\ (\tilde{d}_f - \mathbf{A}_f \cdot \vec{P}) \\ &= \frac{\partial}{\partial \vec{P}}\sum_f (\tilde{d}_f^T \mathbf{V}_f^{-1}\tilde{d}_f - \tilde{d}_f^T \mathbf{V}_f^{-1}(\mathbf{A}_f \cdot \vec{P}) - (\mathbf{A}_f \cdot \vec{P})^T \mathbf{V}_f^{-1}\tilde{d}_f + (\mathbf{A}_f \cdot \vec{P})^T \mathbf{V}_f^{-1}(\mathbf{A}_f \cdot \vec{P}))\end{aligned} \tag{18}$$

Since $\mathbf{V}_f^{-1}$ is a symmetric matrix, the transpose of $\mathbf{V}_f^{-1}$ is itself. Therefore, we have,

$$
\begin{aligned}
\frac{\partial}{\partial \vec{P}}\chi^2 &= \frac{\partial}{\partial \vec{P}}\sum_f (\tilde{d}_f^T \mathbf{V}_f^{-1}\tilde{d}_f - 2\tilde{d}_f^T \mathbf{V}_f^{-1}(\mathbf{A}_f \cdot \vec{P}) + (\mathbf{A}_f \cdot \vec{P})^T \mathbf{V}_f^{-1}(\mathbf{A}_f \cdot \vec{P}))\\
&= -2\sum_f \tilde{d}_f^T \mathbf{V}_f^{-1}\mathbf{A}_f + \sum_f (\mathbf{A}_f \cdot \vec{P})^T (\mathbf{V}_f^{-1} + (\mathbf{V}_f^{-1})^T)\mathbf{A}_f\\
&= 2(\sum_f (\mathbf{A}_f \cdot \vec{P})^T \mathbf{V}_f^{-1}\mathbf{A}_f - \sum_f \tilde{d}_f^T \mathbf{V}_f^{-1}\mathbf{A}_f)
\end{aligned}
\tag{19}
$$

Setting the derivatives to be zero, and providing that the matrix $\sum_f \mathbf{A}_f^T \mathbf{V}_f^{-1}\mathbf{A}_f$ is not singular, we arrive at the analytical solution,

$$
\vec{P} = (\sum_f \mathbf{A}_f^T \mathbf{V}_f^{-1}\mathbf{A}_f)^{-1}\sum_f \mathbf{A}_f^T \mathbf{V}_f^{-1}\tilde{d}_f \tag{20}
$$

Following Chapter 7 of [12], it can be proven that this estimator is unbiased and has the minimum possible variance of any **linear** estimators (thus the name "optimal").

The covariance matrix for the parameters $\vec{P}$ is given by,

$$
\mathbf{U} = (\sum_f \mathbf{A}_f^T \mathbf{V}_f^{-1}\mathbf{A}_f)^{-1} \tag{21}
$$

4 Implementation of NxMOF

CDMSBats [13] is the official SuperCDMS software for data processing. As is the case for any data processing software in a collaboration, CDMSBats is under constant development and the implementation of NxMOF may change. The first CDMSBats master version that includes NxMOF is v?.?.? (TO BE FILLED). In this section we describe the NxMOF implementation for v?.?.? (TO BE FILLED).

The implementation of NxMOF is mostly in two main modules in CDMSBats: BatNoise and BatRoot. In BatNoise, we calculate two quantities related to Eqn. 20, which, accordingly to CDMSBats jargon, have the names "fSignalToNoiseRatioNxM" and "fOptimalFilterNxM":

$$
\begin{aligned}
\text{fOptimalFilterNxM} &= \mathbf{A}_f^T \mathbf{V}_f^{-1}\\
\text{fSignalToNoiseRatioNxM} &= U^{-1} = \sum_f \mathbf{A}_f^T \mathbf{V}_f^{-1}\mathbf{A}_f
\end{aligned}
\tag{22}
$$

Note that fOptimalFilterNxM is a function of frequency (i.e. for each frequency there is such a matrix). These quantities are calculated with noise pulses (and templates) for a given series in BatNoise, and stored in an intermediate "noise" ROOT file. BatRoot then reads in the two quantities and applies them on the *shifted* pulses to calculate the best fitted amplitudes via Eqn. 20. Since it is still unclear how template generation will work for NxMOF, we exert no external constraints on the amplitude parameters a_{ci} so that $\vec{P} = \vec{a}$ (see Eqn. 12) and $\mathbf{B}$ is an identity matrix of the size $(N \times M) \times (N \times M)$.

The NxMOF as is implemented does not try to search through time delays. Instead, it pulls the fitted results from 1x1OF (1x1OF is therefore a prerequisite for running NxMOF), and shifts the pulses by the amount of fitted 1x1OF time delay for that channel. The reason for shifting the pulses instead of the templates is evident. We don't know by what amount to shift the template before the pulses are processed using 1x1OF in BatRoot. An alternative is to calculate fOptimalFilterNxM and fSignalToNoiseRatioNxM for every shifted template within a specified time window, but that would mean a significant increase in the size of the intermediate noise file, which is unfavourable.

The time-delay-adjusted pulses do create a discontinuity in the raw traces. The way we shift the pulses is to cut a length of the raw pulse equal to the fitted 1x1OF time delay and attach it to the end. This discontinuity in time gets dispersed into all frequencies via Fourier transform. However, this effect is expected to be small as long as both the start and end of the pulses are pure noise.

Finally, the χ^2 is calculated using Eqn. 17, now that $\vec{P}$ is known. The output of the NxMOF includes a χ^2, the $N \times M$ amplitudes, and N time delays as calculated from 1x1OF.

References

[1] B. Anderson, *Optimal Filtering*, Dover Publications, 2005.

[2] SuperCDMS Collaboration, *SuperCDMS SNOLAB Technical Design Report*, SuperCDMS Internal Document, 2018.

[3] S. Golwala, *Exclusion Limits on WIMP-Nucleon Elastic Scattering Cross-Section from the Cryogenic Dark Matter Search*, Ph.D. Thesis, 2020.

[4] M. Pyle, *Original Source Code for NxM Optimal Filter*, SuperCDMS Internal Archive, 2012. `https://gitlab.com/supercdms/archive/cdmstools/-/blob/master/generalUtility/fitting/OF_NxM.m`

[5] R. Basu Thakur, *The Cryogenic Dark Matter Search Low Ionization-Threshold Experiment*, Ph.D. Thesis, 2015.

[6] J. Filippini, *A Search for WIMP Dark Matter Using the First Five-Tower Run of the Cryogenic Dark Matter Search*, Ph.D. Thesis, 2008.

[7] SuperCDMS Collaboration, *Low-Mass Dark Matter Search with CDMSlite*, arXiv:1707.01632, 2017.

[8] W. Rau, *New Position Estimators Based On 2-template Quantities*, SuperCDMS Internal Document, 2015. http://titus.stanford.edu/cdms_restricted/Soudan/R133/ebook/150707_wr/

[9] M. Pyle, *Optimizing the Design and Analysis of Cryogenic Semiconductor Dark Matter Detectors for Maximum Sensitivity*, Ph.D. Thesis, 2012.

[10] E. Asamar, *Description of A Complete Reconstruction Algorithm for the Phonon Signal of SuperCDMS SNOLAB Based On the NxM Filter*, SuperCDMS Internal Document, 2018. http://titus.stanford.edu/cdms_restricted/elias/2018_12_06-nxm_filter/

[11] F. James and M. Winkler, *Minuit 2*, public webpage, 2022. https://root.cern.ch/root/htmldoc/guides/minuit2/Minuit2.html

[12] G. Cowan, *Statistical Data Analysis*, Oxford University Press, 1998.

[13] SuperCDMS Collaboration, *CDMSBats documentation*, SuperCDMS Internal Document, 2022. https://supercdms.gitlab.io/Reconstruction/cdmsbats/