Treating Correlated Noise in SHARP

Nicholas Chapman

31 May 2012

1 Introduction

Since the early days of SHARP, the instrument has been plagued by correlated optical noise. The strength of this noise varies in time but manifests as higher measured Q and U errors in the output _int.fits files produced by `sharpinteg`. This higher noise roughly appears in rows 9-12 and also more on the right side of the array. See Figure 1 for an example of this noise in file #44718.

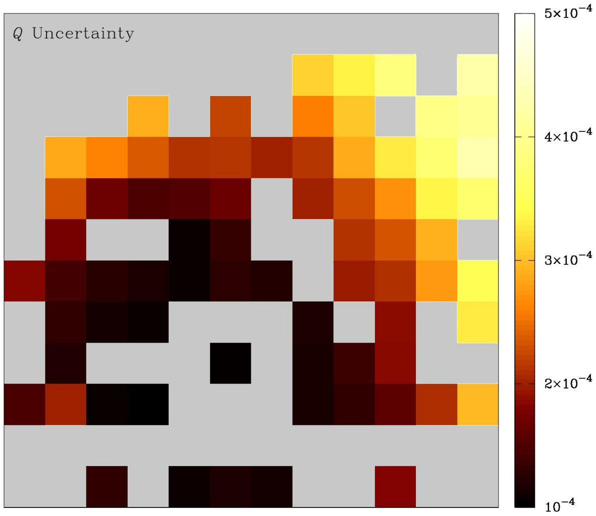

Figure 1: Example of the correlated noise for file #44718. Shown is the uncertainty in Stokes Q. Light grey pixels are bad polarimetry pixels.

When multiple files are combined in `sharpcombine`, adjacent pixels are averaged together. Because the noise in adjacent pixels is correlated in time, the propagation of errors into the final combined map needs to account for these correlated errors. Versions of `sharpcombine` up to and include 5.30 do *not* account for these correlated errors. The result is that the polarization vectors obtained from the combined map have error bars that are too small. This is why we have performed χ^2 analysis and inflated the errors on output maps until now.

1.1 Cause of the Correlated Noise

In May 2011 Giles Novak, Darren Dowell, and Nicholas Chapman performed a series of tests at the CSO to diagnose the cause of the correlated noise. They tested turning the chopper on and off, heating the HWP, removing the polarizing grids, banging on SHARP, having a fan blow air on SHARP, and finally replacing the cold load mirrors on SHARP with warm absorbers. This last test proved to eliminate the correlated noise.

The correlated noise can be seen in the raw timestreams plotted in Figure 2. Panel (a) has the cold load mirrors installed while in (b) the mirrors were replaced with absorber. The larger scatter in the data in panel (a) compared to (b) is evidence of the correlated noise. The overall slope to the data in panel (b) is not relevant to the discussion of correlated noise.

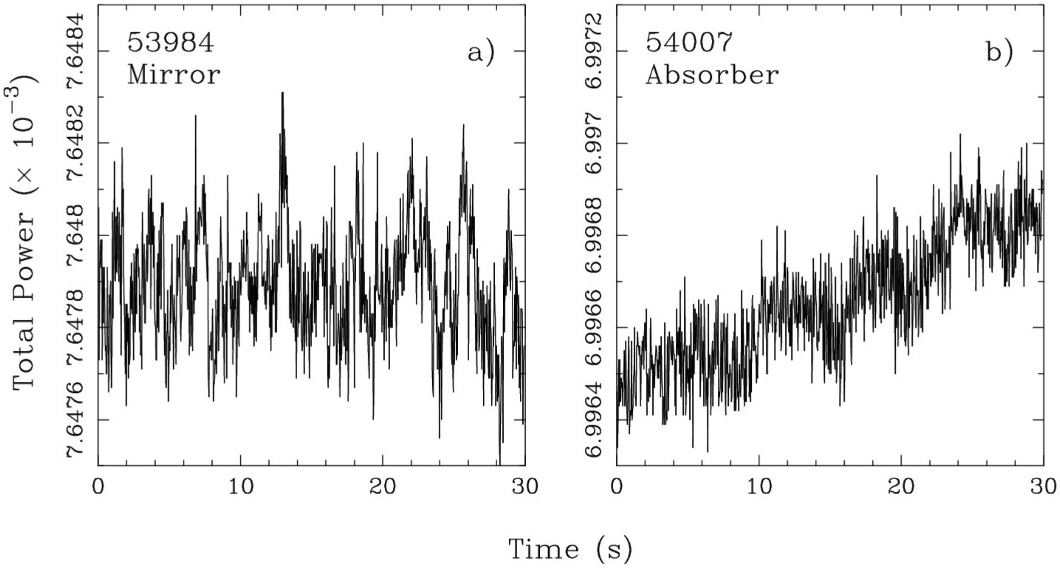

Figure 2: Timestream of the raw data in pixel (11,10) for two files. a) File #53984 with the cold load mirrors installed. b) File #54007 with the mirrors replaced by absorber. The vertical axis range is identical in each panel, even though the absolute values differ.

The correlated noise can be seen more clearly by taking the Fast Fourier Transform (FFT) of each timestream, as shown in Figure 3. Clearly, the presence of the cold load mirrors in panel (a) adds additional noise in the $0-2$ Hz range. This noise is called optical noise because it arises from the optical path within the instrument. For the December 2011 SHARP run, the mirrors were replaced by warm absorbers. This eliminated the correlated noise but reduced the sensitivity somewhat.

2 Covariance in `sharpinteg`

Because the correlated noise is a random noise, it cannot be easily removed in software. However, it is possible to correctly account for the noise when reducing data. The first step is to compute the covariance between pixels in `sharpinteg`. The correlated noise between pixels is really just another way of talking about the covariance between pixels. Using the basic

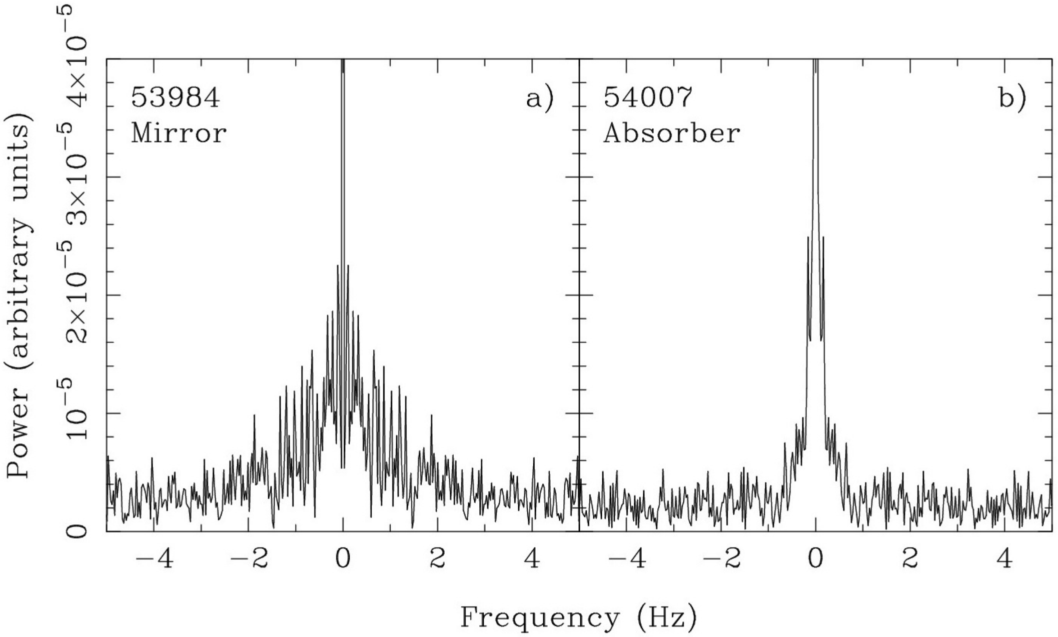

Figure 3: FFT of the data from Figure 2.

properties of covariances and making some simple assumptions we can derive the equations needed. Stokes Q is defined as $Q = 0.5 \times (HWP_1 - HWP_3)$, where HWP_1 is the average chopper- and nod-demodulated value of $H - V$, computed for half-wave plate angle 1. The covariance in Q for two pixels x and y can be expanded:

$$\begin{aligned} cov(Q_x, Q_y) &= \frac{1}{4}\, cov(HWP_{1x} - HWP_{3x}, HWP_{1y} - HWP_{3y}) \qquad (1) \\ &= \frac{1}{4}\Big[cov(HWP_{1x}, HWP_{1y}) - cov(HWP_{1x}, HWP_{3y}) - \\ &\quad -cov(HWP_{3x}, HWP_{1y}) + cov(HWP_{3x}, HWP_{3y})\Big]. \qquad (2) \end{aligned}$$

Since the four HWP angles are observed consecutively, there is no reason to assume that the HWP signal at the beginning of a file is correlated with the HWP signal later in the file. Therefore, the 2nd and 3rd terms are assumed to be zero. As shown in § 1.1 the correlated optical noise is a random noise, justifying this assumption. HWP_{1x} is the average value of the left nod minus the average value of the right nod for the demodulated $H - V$ data for HWP angle 1. Therefore, the term $cov(HWP_{1x}, HWP_{1y})$ can also be expanded:

$$\begin{aligned} cov(HWP_{1x}, HWP_{1y}) &= cov(L_{1x} - R_{1x}, L_{1y} - R_{1y}) \qquad (3) \\ &= cov(L_{1x}, L_{1y}) - cov(L_{1x}, R_{1y}) - \\ &\quad cov(R_{1x}, L_{1y}) + cov(R_{1x}, R_{1y}) \qquad (4) \end{aligned}$$

where L and R are the left and right nods, respectively. Again we can assume the 2nd and 3rd terms are zero since the samples comprising the left and right beams are completely different and they occur at different times within a file. A similar expansion for $cov(HWP_{3x}, HWP_{3y})$ can also be derived.

The final equation for the covariance in Q is then:

$$cov(Q_x, Q_y) = \frac{1}{4}\Bigg[cov(L_{1x}, L_{1y}) + cov(R_{1x}, R_{1y}) + cov(L_{3x}, L_{3y}) + cov(R_{3x}, R_{3y})\Bigg]. \tag{5}$$

The variables L_{1x}, L_{1y}, etc are themselves average quantities of N individual chopper-demodulated data points ($\sim$ 1 Hz rate). To clarify how one computes terms like $cov(L_{1x}, L_{1y})$ consider two generic data sets x_i and y_i with N points each. These are meant to represent data time streams, e.g., for adjacent pixels. The variance of the sample mean $\bar{x}$ is:

$$var(\bar{x}) = \frac{1}{N}\frac{1}{N-1}\sum(x_i - \bar{x})^2 \tag{6}$$

Simiarly, the covariance of the sample means $\bar{x}$ and $\bar{y}$ is:

$$cov(\bar{x}, \bar{y}) = \frac{1}{N}\frac{1}{N-1}\sum(x - \bar{x})(y - \bar{y}) \tag{7}$$

where $\bar{x}$ and $\bar{y}$ are the mean x and y. Note that in the limit that $x = y$, this formula for covariance reduces to the familiar formula for variance of the mean. The sum in Equation 7 is over all the 'good' samples in the left or right beam. A 'good' sample is one which is not rejected by Chauvenet's criterion in either pixel x or y (see memo [1]). In this way, terms like $cov(L_{1x}, L_{1y})$ in Eq. 5 can be computed in `sharpinteg`. An equation similar to Equation 5 can also be derived for $cov(U_x, U_y)$.

We modified `sharpinteg` to output another extension to the standard _int.fits file. The extension is a binary FITS data table and has the name COVAR. The data table simply lists the covariance in Q and U for every pair of pixels x, y. By definition, $cov(x, y) = cov(y, x)$. Since a `SHARP` file is a 12×12 array, the data table has $144 \times 144 = 20,736$ lines. Figure 4a shows the median covariance in each pixel for file #44718. Comparing it to Figure 1 shows that the covariance tends to be higher in the same regions where the noise in higher.

2.1 Covariance in I

Why do we not consider the covariance in the I map? I is the horizontal plus vertical polarization data. We know that this quantity varies with time because the sky brightness varies in time. Furthermore, the sky is much brighter than our sources. The resultant I maps made by `sharpinteg` looks sensible because we compute the left nod - right nod, $L-R$, which helps to remove the variations in sky brightness. The background subtraction algorithm in `sharpcombine` further corrects for the varying sky brightness. In `sharpinteg` the variation in sky brightness is stronger than the optical noise, so computing the covariance in I leads to a roughly uniform map that is similar to the standard deviation in the I value computed from the four HWP angles. Therefore, we ignore the covariance in I.

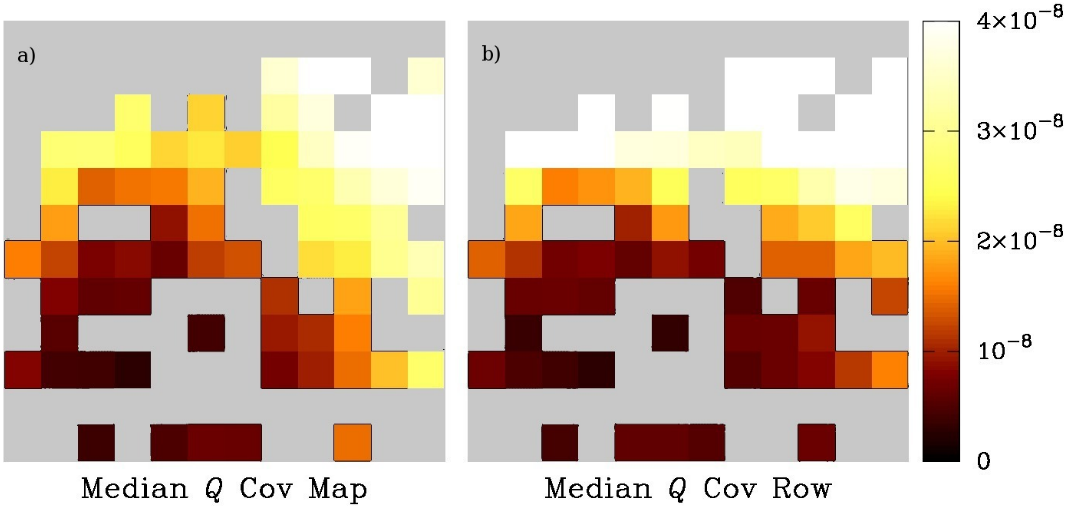

Figure 4: Two maps of the median covariance in Q. (*Left*) Covariance of each pixel with respect to every other pixel, where the median value is plotted. (*Right*) Covariance of each pixel with respect to its row. Light grey pixels are bad polarimetry pixels.

3 Covariance in `sharpcombine`

`sharpcombine` creates an output image from a stack of input `sharpinteg` files. These input files do not all completely overlap, so `sharpcombine` registers them onto a larger grid with higher resolution. For each output grid point, `sharpcombine` computes the weighted average of nearby `sharpinteg` pixels, where the weighting factors take into account both the variance and the distance from the grid point [4]. The output image, I, at any grid point (u, v) and its variance are:

$$I(u,v) = \frac{\sum_n \sum_i g_{in} w_{in} I_{in}}{\sum_n \sum_i g_{in} w_{in}} \tag{8}$$

$$\sigma_I^2(u,v) = \frac{\sum_n \sum_i g_{in}^2 w_{in}}{\left[\sum_n \sum_i g_{in} w_{in}\right]^2} \tag{9}$$

where g_{in} are the gaussian weighting factors, w_{in} are the inverse of the variances, and I_{in} is the value of the input image for each pixel, i, and each file, n. The sum is performed over all input images n and all pixels i within a given radius of (u, v). These equations are correct, *assuming the pixels i are not correlated with each other.* Since we know such correlations exist, they need to be accounted for to properly compute the weighted average and error. We modified `sharpcombine` to accept a `-cov` flag that will read and utilize the covariances computed by `sharpinteg`. How this happens is described below.

3.1 Theory of Optimal Estimators

Assume we have a number of measurements $\boldsymbol{x} = x_1, x_2, \ldots, x_n$, each of which is an estimate of a quantity f. The value of f can then be estimated by the linear combination $f =$

$b_1x_1 + b_2x_2 + \ldots + b_nx_n$. We further have the constraint that $b_1 + b_2 + \ldots + b_n = 1$. What are the factors $\boldsymbol{b} = b_1, b_2, \ldots, b_n$ that will produce the optimal (minimum variance) estimate of f? This is a known problem in statistics with a solution (see, e.g. § 3.4.3 of [8]). It has also been solved for the case of extinction mapping using the method of Lagrange multipliers [5]. For two measurements the solution is:

$$(b_1, b_2) = \left(\frac{\sigma_{22} - \sigma_{12}}{\sigma_{11} + \sigma_{22} - 2\sigma_{12}}, \frac{\sigma_{11} - \sigma_{12}}{\sigma_{11} + \sigma_{22} - 2\sigma_{12}} \right) \tag{10}$$

where σ_{11} is the variance of x_1, σ_{22} is the variance of x_2, and σ_{12} is their covariance. The generalized Gauss-Markov theorem writes the solution in matrix form for an arbitrary number of parameters [2]:

$$f = \frac{W^T C^{-1} \boldsymbol{x}}{W^T C^{-1} W}, \tag{11}$$

where C^{-1} is the inverse of the covariance matrix (defined below), $\boldsymbol{x}$ is a vector of the parameters $x_1, \ldots, x_n$, and W and W^T are the design matrix and its transpose respectively. For our purposes the design matrix is simply a vector of ones.

It is straightforward to derive Equation 10 from Equation 11. Starting with the covariance matrix and its inverse:

$$C = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} \tag{12}$$

$$C^{-1} = \frac{1}{\sigma_{11}\sigma_{22} - \sigma_{12}^2} \begin{bmatrix} \sigma_{22} & -\sigma_{12} \\ -\sigma_{12} & \sigma_{11} \end{bmatrix} \tag{13}$$

where we have taken advantage of the fact that $\sigma_{21} = \sigma_{12}$ (another property of covariances) and just used σ_{12}. The determinant, $\sigma_{11}\sigma_{22} - \sigma_{12}^2$, appears in the numerator and denominator of Eq. 11, thus it will cancel out (unless your determinant is zero!). Therefore, we have neglected this factor in the derivations below. The numerator of Eq. 11 can be written:

$$W^T C^{-1} \boldsymbol{x} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{22} & -\sigma_{12} \\ -\sigma_{12} & \sigma_{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{14}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{22}x_1 - \sigma_{12}x_2 \\ \sigma_{11}x_2 - \sigma_{12}x_1 \end{bmatrix} \tag{15}$$

$$= (\sigma_{22} - \sigma_{12})x_1 + (\sigma_{11} - \sigma_{12})x_2 \tag{16}$$

Similarly, the denominator of Eq. 11 is:

$$W^T C^{-1} W = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{22} & -\sigma_{12} \\ -\sigma_{12} & \sigma_{11} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{17}$$

$$= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{22} - \sigma_{12} \\ \sigma_{11} - \sigma_{12} \end{bmatrix} \tag{18}$$

$$= \sigma_{11} + \sigma_{22} - 2\sigma_{12} \tag{19}$$

Eq. 11 can thus be written:

$$f = \frac{\sigma_{22} - \sigma_{12}}{\sigma_{11} + \sigma_{22} - 2\sigma_{12}} x_1 + \frac{\sigma_{11} - \sigma_{12}}{\sigma_{11} + \sigma_{22} - 2\sigma_{12}} x_2 \tag{20}$$

which are the same coefficients given in Eq. 10.

3.1.1 Variance

For a generic linear equation $f = b_1x_1 + b_2x_2 + \ldots + b_nx_n$, the variance on f can be written as:

$$\sigma_f^2 = \boldsymbol{b}^T C \boldsymbol{b}, \tag{21}$$

where C is the covariance matrix for the x_i. For example, when $n = 2$, $f = b_1x_1 + b_2x_2$ and the variance is:

$$\begin{aligned} \sigma_f^2 &= \boldsymbol{b}^T C \boldsymbol{b} && (22) \\ &= \begin{bmatrix} b_1 & b_2 \end{bmatrix} \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} && (23) \\ &= b_1^2\sigma_{11} + b_2^2\sigma_{22} + 2b_1b_2\sigma_{12} && (24) \end{aligned}$$

Note that the familiar result could also be obtained by propagation of errors on f.

3.2 Application to `sharpcombine`

Implementing the Gauss-Markov theorem into `sharpcombine` would be straightforward except for the use of the gaussian weighting factors, g. Because these gaussian factors weight by distance instead of statistical uncertainty, it is not obvious how to include them in the formalism from the previous section. However, we can make a reasonable guess for how to compute the errorbars with gaussian weighting by distance. For simplicity we will consider the case of two measurements. If we make the substitutions $\sigma_{11} \rightarrow \sigma_{11}/g_1$, $\sigma_{22} \rightarrow \sigma_{22}/g_2$, and $\sigma_{12} \rightarrow \sigma_{12}/\sqrt{g_1g_2}$, Eq. 10 can be written as:

$$(b_1, b_2) = \left(\frac{g_1\sigma_{22} - \sqrt{g_1g_2}\sigma_{12}}{g_2\sigma_{11} + g_1\sigma_{22} - 2\sqrt{g_1g_2}\sigma_{12}}, \frac{g_2\sigma_{11} - \sqrt{g_1g_2}\sigma_{12}}{g_2\sigma_{11} + g_1\sigma_{22} - 2\sqrt{g_1g_2}\sigma_{12}} \right), \tag{25}$$

where we multiplied both the numerator and denominator by g_1g_2. Notice that in the limit $\sigma_{12} = 0$, this is identical to the coefficients in Eq. 8 (after also dividing the numerator and denominator by $\sigma_{11}\sigma_{22}$).

Equation 25 can alternatively be derived by changing the design matrix to be $W^T = [\sqrt{g_1}\sqrt{g_2}]$. I am not as familiar with the theory of the design matrix, but I think this is equivalent to setting the constraint $g_1b_1 + g_2b_2 = 1$, where before the constraint was $b_1 + b_2 = 1$.

The variance is still given by Equation 21, but note that the gaussian weighting factors g_i are incorporated into $\boldsymbol{b}$ and do not appear in the covariance matrix.

3.2.1 Implementation

There are five functions in `sharpcombine` that need to be altered to work with the covariances in addition to the image errors. They are:

`atten()`	This function corrects for opacity
`linearbgnd2()`	Background subtraction (errors scaled by normalized slope)
`instrumpol2()`	Subtract i.p. (errors in Q, U inflated by error in I)
`Rotate()`	Rotate images from sky frame to raw frame
`gauss_smooth()`	Gaussian smoothing

The first four are simple bookkeeping where changes in errors are also applied to the covariances. The last function involves using the equations above. For output grid point (u, v), the nearby pixels within a given radius are found. A covariance matrix, C, is created from these pixels which is then inverted using LU decomposition to create C^{-1} [7]. Because we assume that the covariance between pixels in different files is zero, we can treat each file separately and sum the individual C. Equations 8 and 9 can now be written as:

$$I(u,v) = \frac{\sum_n \sum_i \sum_j \sqrt{g_{in} g_{jn}} C^{-1}_{ijn} I_{jn}}{\sum_n \sum_i \sum_j \sqrt{g_i g_j} C^{-1}_{ijn}} \tag{26}$$

$$\sigma_I^2(u,v) = \sum_n \sum_i \sum_j b_{in} b_{jn} C_{ijn} \tag{27}$$

where there are n files, and double summing over pixels i, j in each file.

4 Testing

4.1 Consistency Check on L1527

To test the new procedure for covariance we reprocessed the L1527 data from November 2007 and September 2008. These are the same data used in the [3] paper. As closely as possible, we processed these data the same way they were processed for the published paper. Specifically, we used the same files, custom RGM masks, pointing corrections, smoothed tau values, and mask flag cutoff used for the paper (see also [6]). However, we used the newest version of `sharpinteg`, version 3.1.3, which contains a few minor improvements over the older versions (see [1]). We temporarily disabled the checking for nod length to ensure we used the same files as in the Davidson paper. The command line flags we used are:

```
   sharpinteg sharc2-0#####.fits -r rgmfile -f 1 -w -sil -em -c -m 150
```

Then we ran `sharpcombine` version 5.40 both with and without the `-cov` flag:

```
   sharp_combine list combine.fits -ip 0.0034 0.00017 0.0036 0.0 -l 51 51
-hwp 91 -bg 10 0 -sm 2 -ma 5 -ps 9.5 -pm 12
```

We performed a χ^2 analysis by dividing the data into 7 bins where each bin had 10-14 files. The bins were temporal, i.e. the first bin contained the first 12 files, the second bin contained the next 12, etc. We computed the average value of the reduced χ^2 over the entire map and inflated the errorbars on extracted polarization vectors by the square root of this χ^2 except for the case where we used the `-cov` flag, when no inflation was necessary because the reduced χ^2 was near unity.

The results of this 'before-and-after' test are shown in Figure 5. The polarization vectors are similar between the three panels, giving confidence that the method is not introducing spurious vectors.

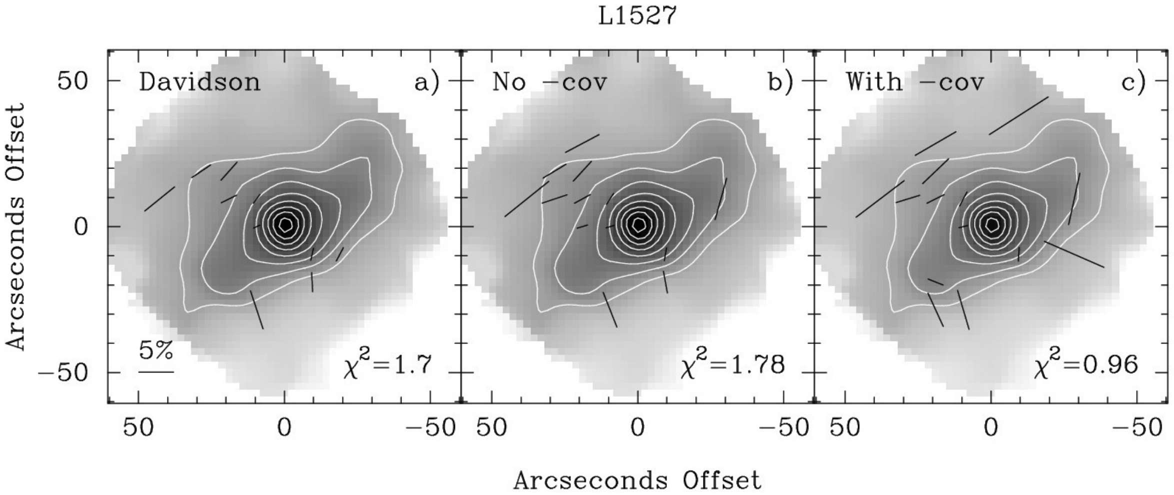

Figure 5: $E-$vectors for L1527. All vectors are $\geq 2\sigma$ before debiasing. a) Original data from [3] b) Data processed without the `-cov` flag. The errors are inflated by the square root of the χ^2 before plotting vectors. c) Data processed with the `-cov` flag. No inflation of the errors is performed.

4.2 New Processing Techniques

As a second test we processed several different data sets (including L1527) using newer techniques than before. A future memo will describe these techniques in detail, but we summarize them below.

We used `sharpinteg` version 3.1.3 (nothing disabled):

```
sharpinteg sharc2-0#####.fits -r rgmfile -f 1 -w -sil -em -c
```

Note that no masking was performed (`-m` flag). Instead, we computed the median covariance over the entire map (Fig. 4a) and within each row (Fig. 4b) to identify optical noise and electronic row noise. We set a cutoff level of 2×10^{-8}. Stokes Q and U pixels where the whole map covariance was greater than or equal to this cutoff were flagged and discarded. All pixels (I, Q, and U) where the single row covariance was greater than or equal to this cutoff were flagged and discarded.

We estimated the sky noise in each raw file by computing the Lomb-Scargle periodogram (basically an FFT for non-uniformly sampled data). We summed the power over the frequency range $0.01 - 0.1$ Hz and discarded any file with an integrated power $\geq 5 \times 10^{-11}$.

For the remaining files we computed smoothed tau values using the standard `sharctau` program and computed pointing corrections by polynomial interpolation over the values returned by `fitgauss`.

Lastly, we combined the data using version 5.40 of `sharpcombine` both with and without the `-cov` flag:

```
    sharp_combine list combine.fits -ip 0.0027 -0.0018 0.0001 0.0 -l 51 51
-hwp 91 -bg 10 0 -sm 2 -ma 5 -ps 9.5 -pm 12
```

The glass M3 mirror was replaced with aluminum in 2009. For data taken after that time (L483, L1448-IRS2, SERP-FIR1) the newer instrumental polarization (IP) above was used. For older data (L1527, L1157), we used the older IP (`-ip 0.0034 0.00017 0.0036 0.0`).

We divided the data for each source into temporal bins then ran the `chi2` program on the data. Our results are summarized in Table 1. Using the `-cov` flag dramatically lowered the reduced χ^2 in all five sources.

Source	#	#	Mean Reduced χ^2 in Q,U	
	Files	Bins	Regular	Covariance
L483	58	9	2.07	1.04
L1157	136	17	2.40	1.19
L1448-IRS2	113	16	2.29	0.99
L1527	72	7	1.87	0.97
SERP-FIR1	16	3	2.40	1.21

Table 1: χ^2 results both with and without the `-cov` flag.

References

[1] Chapman, N. 2012 May 14, Memo "Removing Electronic Row Noise in SHARP"

[2] Cox, M. G., Eiø, C., Mana, G., & Pennecchi, F. 2006, Metrologia, 43, S268

[3] Davidson, J. A., Novak, G., Matthews, T. G., et al. 2011, ApJ, 732, 97

[4] Houde, M. & Vaillancourt, J. E. 2007, PASP, 119, 871

[5] Lombardi, M. & Alves, J. 2001, A&A, 377, 1023

[6] Novak, G. 2009 May 14, Memo "First Attempt at Combining 2007 and 2008 Data For L1527"

[7] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, Numerical Recipes in C (Cambridge: University Press)

[8] Serfling, Robert J. 1980, Approximation Theorems of Mathematical Statistics (New York, NY: John Wiley & Sons)