Comparison of Linear and Circular Products after PolConvert

L. V. Benkevitch[1]

[1]MIT Haystack observatory, Westford, MA 01886, USA.

September 4, 2024

Abstract

The VGOS data are collected with the use of linearly polarized receivers. However, circular polarization has a number of benefits. The PolConvert software allows VGOS data conversion from linear to circular polarization. In this study we do a statistical comparison of the pseudo-Stokes data before and after the conversion in order to characterize the errors PolConvert introduces.

Contents

1 VGOS Data

This study is based on the data acquired during the VLBI session ER2201 on September 8, 2022. The calibrator source 4C 39.25 was observed for 6 hours with the antennas of the six stations listed in Tab. 1.

Table 1: Stations in the Schedule for the Session ER2201

Code		Name	Description
1-ltr	2-ltr		
E	Ws	WETTZ13S	Wettzell 13-m antenna south, Germany
M	Mg	MACGO12 M	McDonald Geodetic Observatory (MGO), TX, USA
S	Oe	ONSA13NE	Onsala 13-m antenna north-east, Sweden
T	Ow	ONSA13SW	Onsala 13-m antenna south-west, Sweden
V	Wf	WESTFORD	Westford, MA, USA
Y	Yj	RAEGYEB	13-m at Yebes, Spain

The correlated data are located in the folder `/data-sc16/geodesy/3819` on the server `demi.haystack.mit.edu`. PolConvert has been applied to the data, and the pseudo-Stokes I results with the circular polarization were saved in the folder
`/data-sc16/geodesy/3819/polconvert/3819/scratch/pcphase_stokes_test/3819`.

2   Parameter Temporal Variations Before and After PolConvert

Converting linear polarization of the source data to the circular polarization with the PolConvert software introduces changes into its parameters. Here we consider the changes in multi-band delays (MBD), single-band delays (SBD), and signal-to-noise ratios (SNR) for the pseudo-Stokes polarization products, of which we only select the pseudo-Stokes I parameters as functions of time, t. From the original, linearly polarized data set we read I, which we denote $\mathrm{LinI}(t)$. PolConvert is applied to the original data set and generates another data set with circular polarization, from which we read I parameters denoted as $\mathrm{CirI}(t)$. Of course, the data are presented in the discrete, numerical format, so instead of the continuous time, t, integer indices are used, like $k = \overline{1..N}$, where N is the number of data points in the sample. The following Figures show temporal variations of the $\mathrm{LinI}[k]$ and $\mathrm{CirI}[k]$ parameters during the experiment before and after PolConvert, one graph for each baseline:

MBD variations in Fig. 1;
SBD variations in Fig. 4;
SNR variations in Fig. 7.

One can see that most of the graphs after PolConvert, $\mathrm{CirI}[k]$, are significantly biased with respect to $\mathrm{LinI}[k]$. The biases calculated as $\mathrm{LinI}[k] - \mathrm{CirI}[k]$ are plotted in the Figures below:

MBD bias variations in Fig. 2;
SBD bias variations in Fig. 5;
SNR bias variations in Fig. 8.

For each of the biases the bias mean, $\overline{\mathrm{Bias}}$, is computed.

However, removing the bias by subtracting the mean values of each curve makes the differences much less significant:

$$\text{LinI}[k] = \text{LinI}[k] - \text{mean}(\text{LinI}[k]) \tag{1}$$

$$\text{CirI}[k] = \text{CirI}[k] - \text{mean}(\text{CirI}[k]) \tag{2}$$

In order to compare the unbiased $\text{LinI}[k]$ with $\text{CirI}[k]$, the residual, $\text{res}[k]$, is computed as their difference:

$$\text{res}[k] = \text{LinI}[k] - \text{CirI}[k] \tag{3}$$

The graphs of residuals with the means subtracted are shown in the following Figures:

MBD residual variations in Fig. 3;
SBD residual variations in Fig. 6;
SNR residual variations in Fig. 9.

To assess the similarity of $\text{LinI}[k]$ and $\text{CirI}[k]$, the following parameters are computed:

- Pearson correlation coefficient r_{corr};
- Root Mean Square Error, RMSE;

Since $\text{LinI}[k]$ and $\text{CirI}[k]$ already have their means subtracted, the correlation formula is simplified:

$$r_{corr} = \frac{\sum \text{LinI}[k] \cdot \text{CirI}[k]}{\sqrt{\sum \text{LinI}[k]^2 \cdot \text{CirI}[k]^2}} \tag{4}$$

The correlation coefficient has its upper limit, unity, and if r_{corr} is very close to unity, the curves are almost identical, save their constant bias. MBD and SNR demonstrate excellent correlations between LinI and CirI.

The root mean square error, RMSE, in our case is actually the standard deviation of the residual, σ:

$$\text{RMSE} = \sigma = \sqrt{\frac{1}{N} \sum \text{res}[k]^2} \tag{5}$$

Lower RMSE indicates a better fit between the curves. However, RMSE is not an "absolute" indicator of the curves' proximity. It needs to be related to the absolute values of both LinI and CirI. They are shifted, however, to very roughly juxtapose RMSE with the absolute values of MBD, SBD, and SNR (here denoted as PRM) we have introduced their "absolute averages":

$$\overline{\text{PRM}} = (|\text{mean}(\text{LinI})| + |\text{mean}(\text{CirI})|) / 2. \tag{6}$$

To visually compare the observed values of the parameters LinI and CirI with their unbiased residual RMS error, we write it in the format $\overline{\text{PRM}} \pm \text{RMSE}$ or $\overline{\text{PRM}} \pm \sigma$. The relative error between the unbiased LinI and CirI can be expressed as

$$\text{relerr}(\%) = \text{RMSE}/\overline{\text{PRM}}. \tag{7}$$

These characteristics are summarized in Tables below:

MBD PolConversion parameters in Table 2;
SBD PolConversion parameters in Table 3;
SNR PolConversion parameters in Table 4.

Table 2: Pseudo-Stokes I MBD PolConversion Parameters

BL	$\overline{\text{MBD}} \pm \sigma \, (\mu s)$	relerr (%)	$\overline{\text{Bias}} \, (\mu s)$	r_{corr}
EV	508.9 ± 0.45	0.09	-10.7	0.999995
EY	1020.6 ± 3.75	0.37	160.7	0.999464
ME	898.9 ± 0.76	0.08	1.0	0.999989
MS	2052.0 ± 1.51	0.07	-3.5	0.999417
MT	168.7 ± 2.86	1.69	126.4	0.996943
MV	372.7 ± 1.11	0.30	-8.8	0.999805
MY	213.2 ± 5.74	2.69	147.4	0.999327
SE	1186.3 ± 0.68	0.06	7.9	0.999993
SV	1686.3 ± 0.65	0.04	-2.9	0.999924
SY	2279.0 ± 2.95	0.13	193.7	0.999898
TE	1035.2 ± 0.55	0.05	-122.3	0.999992
TV	533.7 ± 0.70	0.13	-132.6	0.999942
TY	70.9 ± 3.26	4.60	65.2	0.999860
VY	592.7 ± 1.91	0.32	177.7	0.999939

Table 3: Pseudo-Stokes I SBD PolConversion Parameters

BL	$\overline{\text{SBD}} \pm \sigma \, (\mu s)$	relerr (%)	$\overline{\text{Bias}} \, (\mu s)$	r_{corr}
EV	629.5 ± 46.4	7.4	797.2	0.993186
EY	332.2 ± 95.8	28.9	664.3	0.973586
ME	657.8 ± 41.4	6.3	-1315.6	0.985598
MS	1817.7 ± 67.1	3.7	-455.2	0.978372
MT	410.8 ± 48.9	11.9	-821.6	0.986706
MV	756.9 ± 46.6	6.2	-554.8	0.981644
MY	339.1 ± 48.1	14.2	-678.3	0.949062
SE	1708.4 ± 13.4	0.8	-870.2	0.997747
SV	1137.6 ± 14.8	1.3	-149.6	0.994778
SY	2287.7 ± 64.5	2.8	-360.1	0.966807
TE	276.0 ± 35.1	12.7	-476.9	0.992071
TV	265.0 ± 14.8	5.6	171.5	0.995829
TY	1110.7 ± 73.2	6.6	125.7	0.954197
VY	1013.6 ± 46.5	4.6	-214.3	0.952361

Table 4: Pseudo-Stokes I SNR PolConversion Parameters

BL	$\overline{\mathrm{SNR}} \pm \sigma$	relerr (%)	$\overline{\mathrm{Bias}}$	r_{corr}
EV	1611.5 ± 18.7	1.16	56.2	0.999468
EY	1508.8 ± 14.0	0.93	-32.0	0.998684
ME	1110.5 ± 5.1	0.45	-6.2	0.999997
MS	668.0 ± 9.9	1.49	-12.7	0.999991
MT	753.4 ± 9.0	1.19	-9.3	0.999986
MV	682.3 ± 16.2	2.37	25.4	0.999993
MY	634.2 ± 5.7	0.90	-10.3	0.999981
SE	1610.0 ± 12.4	0.77	-21.5	0.999420
SV	4275.0 ± 37.0	0.87	197.3	0.999660
SY	2982.7 ± 11.9	0.40	-80.4	0.999763
TE	1869.3 ± 15.1	0.81	-10.4	0.999200
TV	5007.9 ± 39.7	0.79	244.7	0.999630
TY	3371.2 ± 9.1	0.27	-54.1	0.999899
VY	3507.2 ± 13.3	0.38	91.1	0.999912

3 Parameter residual Statistics Before and After PolConvert

While in the previous Section MBD, SBD, and SNR pseudo-Stokes I parameter data were treated as functions of time, here we do their statistical study. In order to estimate the errors introduced by PolConvert, histograms of the unbiased parameter residuals (after the mean subtraction, see Eqs. (1) and (2)) are plotted. Below are the histograms for all of the baselines:

MBD residual histogram for all of the baselines in Fig. 10;
SBD residual histogram for all of the baselines in Fig. 12;
SNR residual histogram for all of the baselines in Fig. 14.

It is interesting to see which of the individual stations contribute to the errors. We plotted histograms of the parameter residuals for the baselines involving each particular station. They are shown in the following Figures:

MBD residual histograms for the baselines including one station in Fig. 11;
SBD residual histograms for the baselines including one station in Fig. 13;
SNR residual histograms for the baselines including one station in Fig. 15.

Note that everywhere the means of the residuals are very close to zero (actually, down to the machine epsilon!).

In order to evaluate the significance of the error we attempted to use the Pearson's chi-squared test comparing the histograms to the normal distributions:

$$f(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} . \tag{8}$$

All the histograms of residuals are created with 21 bins. To perform the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (frequencies less than 5) were grouped and the test

used fewer number of bins N_b (see it printed in each plot, Fig. 10 - Fig. 15). The Pearson's χ^2 test compares the observed data counts in the bins B_i with the expected (theoretical) counts E_i, where $i = \overline{1 \ldots N_b}$, obtained from the normal PDF (8) as $E_i = Np_i$. Here p_i is the probability that a random value sampled from $f(\mu, \sigma)$ is within the i-th interval. The value of the test-statistic is calculated as

$$X^2 = \sum_{i=1}^{N_b} \frac{(B_i - E_i)^2}{E_i}, \tag{9}$$

where N_b is the number of bins. The X^2 test statistic asymptotically approaches the χ^2_{df} distribution, where df is the number of degrees of freedom:

$$\mathrm{df} = N_b - k - 1, \tag{10}$$

where $k = 2$ is the number of parameters estimated from the data, mean μ and standard deviation σ. The additional -1 is due to the fact that the χ^2 test itself has a restriction imposed by the sum of the observed frequencies being equal to the sum of the expected frequencies.

If we want to be 95% confident that the data is distributed normally, X^2 cannot exceed the critical value χ^2_{cr} such that the probability for X^2 to appear within the interval $[0 \ldots \chi^2_{cr}]$ is $p = 0.95$. Hence we should set the level of significance $\alpha = 1 - p = 0.05$. The critical χ^2_{df} value can be calculated as the value of χ^2 Probability Point Function (PPF), the inverse of the χ^2 Cumulative Distribution Function or CDF. In Python this is done as `scipy.stats.chi2.ppf(1-alpha, df)`. Alternatively, the Inverse Survival Function (ISF) may be used as `scipy.stats.chi2.isf(alpha, df)`.

However, in our case the Pearson's χ^2 test does not work, because the residuals are grouped around their means substantially denser than the normal distributions with the same standard deviations. In most distributions, the calculated values of the histogram chi-squares are many times greater than the critical chi-square values for the p-values less than or equal to 0.05 (printed in each of the histogram plots). We have used another method. Since the residuals are mostly within $\pm\sigma$ (standard deviation), we compute their proportion and compare it to the expected proportion under a normal distribution, which is about 68.27%. The proportion is printed near each histogram. It is at the level of about 80%, which is better than the standard normal for most of the cases.

4   Conclusion

We have conducted a study to assess how well the PolConvert software makes VGOS data conversion from linear to circular polarization. The pseudo-Stokes data of the Multi-Band Delays (MBD), Single-Band Delays (SBD), and Signal-to-Noise Ratios (SNR) as functions of time before and after the conversion were compared. We also did a statistical comparison of the pseudo-Stokes data of MBD, SBD, and SNR before and after the conversion in order to characterize the errors PolConvert introduces. Two criteria were used to assess the residuals: the Pearson's χ^2 test and the proportions of residuals within $\pm\sigma$ compared to the normal distribution.

PolConvert introduces large (and random!) biases between the pseudo-Stokes data parameters, which, we believe, is unacceptable.

On the other hand side, after the bias removal, the average errors (computed as unbiased residual rms) are very small making tenths of percent for the most of baselines for MBD (see Table 2) and SNR (see Table 4). For SBD the relative error is an order of magnitude greater. The SBD conversion biases are generally larger, too (see Table 3).

The errors (the unbiased residual RMSE) generally have good distributions, denser than the normal with the same STD, as can be seen in Figs. 10 to 15.

5 Appendix: Software

5.1 make_sorted_idx.py: Saving VGOS data in Python dictionaries

The VLBI Global Observing System (VGOS) database is organized as a tree-like directory structure. For our purpose of statistical analysis of a small number of parameters scattered across many directories and files below the root directory of the experiment, this implies significant overhead in opening multiple files and accessing the parameters within each of them. The data files in their names only provide the station or baseline names, and no time or polarization information. For example, extraction of, say, SNR data for a particular polarization product and within a specific time range would require opening *all* the files and accessing their times and polarizations using HOPS API calls.

We wrote a script, `make_sorted_idx.py`, to extract the parameters for statistical analysis for the whole experiment and to put it in a Python dictionary, preserving the temporal order. We call such dictionaries "indices". The index can be "pickled" and saved on disk. Interestingly, these files are small, in the hundreds of kilobytes. The other data analisys and plotting scripts read the index files, unpickle them into the Python dictionaries, and use data from the dictionaries.

The script `make_sorted_idx.py` should be run on the `demi.haystack.mit.edu` server where the VGOS data are stored under the directory `/data-sc16/geodesy/`. Current version of the script works on the experiment 3819 data located under `/data-sc16/geodesy/3819`. It creates three dictionaries pickled in the files `idx3819l.pkl`, `idx3819c.pkl`, and `idx3819cI.pkl` in the directory where the script was run.

- `idx3819l.pkl`: linear polarization products immediately from the `/data-sc16/geodesy/3819/` directory;

- `idx3819c.pkl`: circular polarization products generated by PolConvertwithout the pseudo-Stokes 'I' data, only 'LL', 'LR', 'RL', 'RR'.
  The data are found in
  `/data-sc16/geodesy/3819/polconvert/3819/scratch/pol_prods1/3819` directory.

- `idx3819cI.pkl`: pseudo-Stokes 'I' only for the circular polarization products generated by PolConvert. The data are taken from
  `/data-sc16/geodesy/3819/polconvert/3819/scratch/pcphase_stokes_test/3819`

A pickle file can be unpickled into a dictionary using the `pickle.load()` function. For example:

```
import pickle
with open('idx3819c.pkl', 'rb') as finp:
    idx3819c_1 = pickle.load(finp)
```

The script `make_sorted_idx.py` is also a Python module defining the function

`make_idx(base_dir, pol='lin', max_depth=2)` with parameters:

   `base_dir`: the directory containing the VGOS data. For example, it may be

```
                    /data-sc16/geodesy/3819/.
        pol: polarization, 'lin' - linear, 'cir' - circular.
             This parameter is used for the data generated by PolConvert.
             It converts the polarization product names
                  'XX', 'XY', 'YX', 'YY', and the lists ['XX', 'YY'] into the correct names
                  'LL', 'LR', 'RL', 'RR', and 'I', respectively.
        max_depth: Limits the maximum depth of recursing into the subdirectories of base_dir.
```

`make_idx()` creates and returns the index dictionary with the data from `base_dir`.

The index dictionary has three dimensions: the baseline name, the polarization, and the data proper, including 'time', 'file', 'mbdelay', 'sbdelay', and 'snr'. Consider a particular index named `idx3819l_1` (experiment 3819, linear polarization). Its first dimension is indexed with the baseline names derived from the set of stations, {'E', 'M', 'S', 'T', 'V', 'Y'}.
The possible first indices are the baseline names:
```
idx3819l_1.keys()
dict_keys(['SE', 'VY', 'MV', 'MT', 'TV', 'EY', 'SY', 'TY', 'MS', 'SV',
           'TE', 'EV', 'MY', 'ME'])
```

Each of the baselines is associated with the cross-corellation products and the pseudo-Stokes I parameter. Thus the second index is one of the products. For example, for the 'ME' baseline:
```
idx3819l_1['ME'].keys()
dict_keys(['XX', 'XY', 'YX', 'YY', 'I'])
```

For example, the times, the full data file names, SNRs, multi- and single-band delays for the 'SY' baseline and the 'XY' polarization products from this baseline are contained in the index dictionary under
```
idx3819l_1['SY']['XY']:
```

`idx3819l_1['SY']['XY'].keys()` prints
```
dict_keys(['time', 'file', 'mbdelay', 'sbdelay', 'snr']).
```

For example, in order to access the multi-band delay data list in the ascending temporal order for the baseline 'SV' and the pseudo-Stokes I, one should issue the following command:
`mbd = idx3819l_1['SV']['I']['mbdelay']`.

5.2   group_tails.py: Group Tail Bins with Sparse Data

`group_tails.py` is a Python module defining the function `group_tails(ni, fni)`:

Usage:
```
from group_tails import group_tails
ni_grp, fni_grp = group_tails(ni, fni)
```

The input parameter `ni` is an array containing a distribution of observed frequencies. If its left-tail and right-tail have sparse data (i.e. frequencies smaller or equal 5), the values in tails are summed up and replaced with a single values on the either side. The second parameter, `fni`

array, must have the same size as `ni`. It is supposed to contain expected (theoretical) frequencies, calculated from a distribution to which `ni` has been fit. The function returns the grouped arrays: `ni_grp` and `fni_grp`, grouped similarly to `ni`.

5.3 Plot Parameter Temporal Variations

The scripts `plot_mbdelay.py`, `plot_sbdelay.py` and `plot_snr.py` create plots for MBD, SBD, and SNR, respectively, shown in Figures 1 to 9. The data are read from the dictionaries `idx3819l.pkl` and `idx3819cI.pkl`.

5.4 Plot Parameter Histograms

The scripts `plot_st_difmbd.py`, `plot_st_difsbd.py` and `plot_st_difsnr.py` create histogram plots for MBD, SBD, and SNR, respectively, shown in Figures 10 to 15. The data are read from the dictionaries `idx3819l.pkl` and `idx3819cI.pkl`.
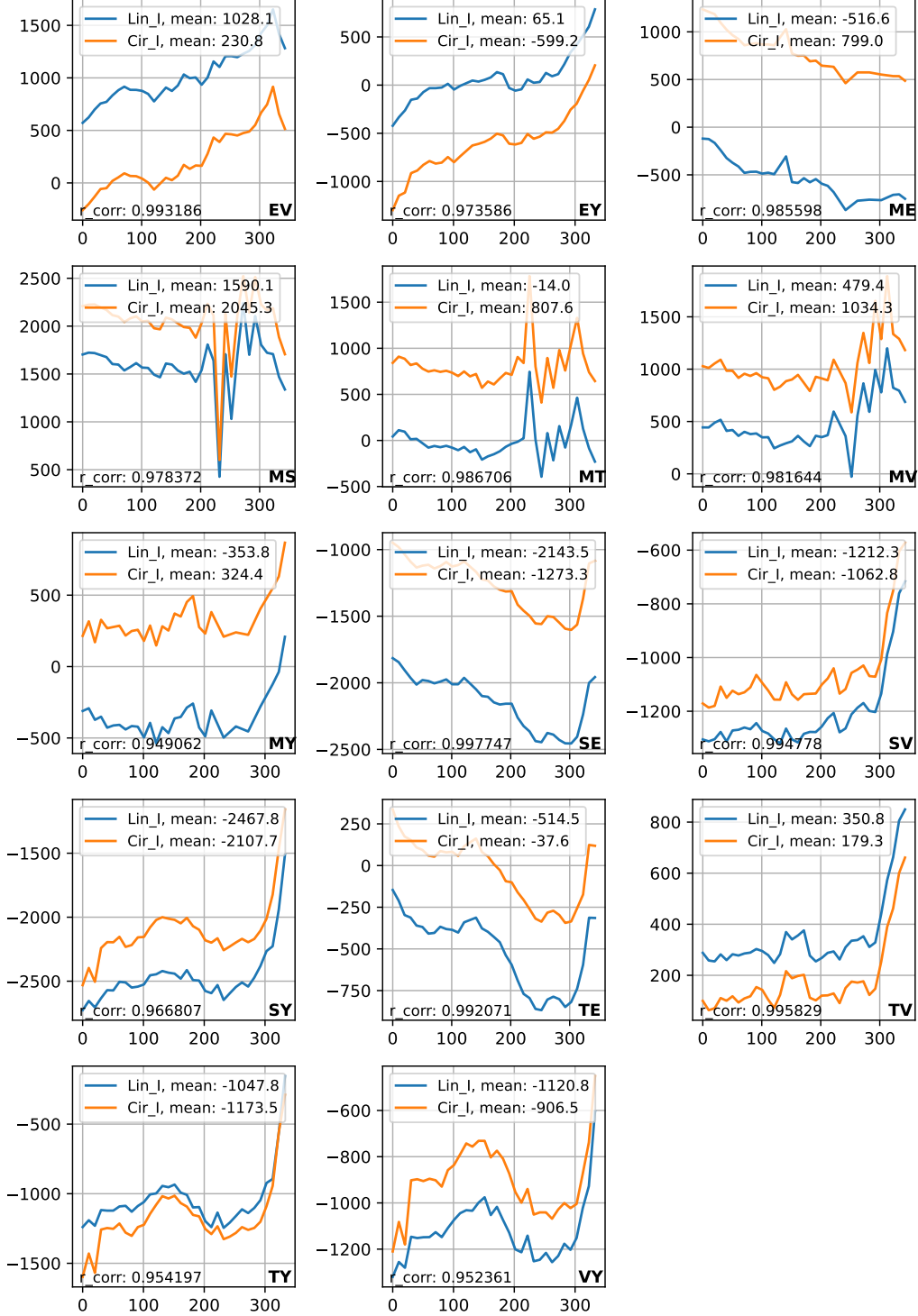
6 Appendix: Figures
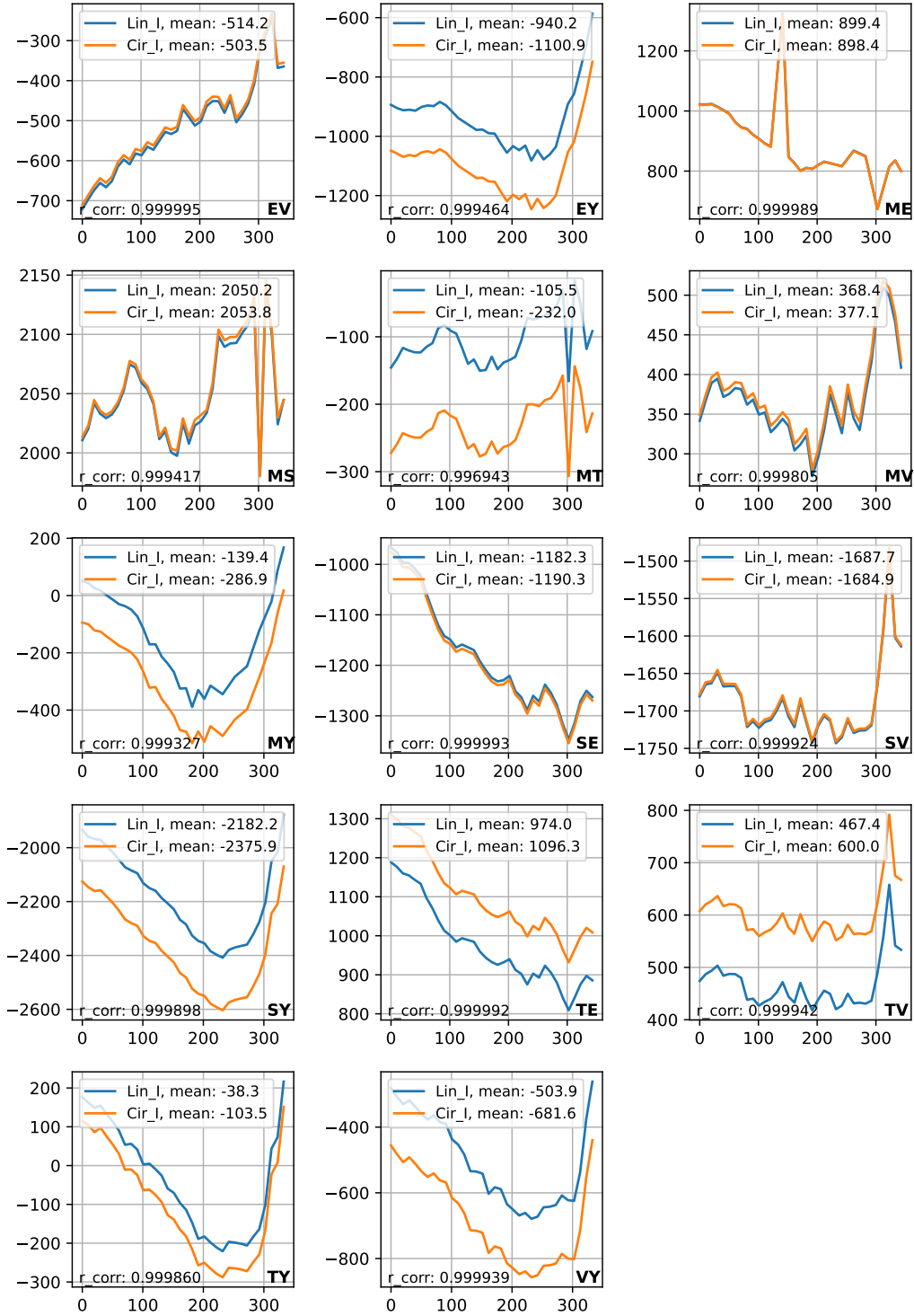
Figure 1: Evolution of Multi-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows two curves: LinI, I from the original, linearly polarized data set and CirI, I from the circularly polarized data set obtained by applying PolConvert to the original data. Graphs have large biases. However, the coefficients of correlation r_corr between LinI and CirI are so close to unity that the curves can be considered identical (except for the biases).
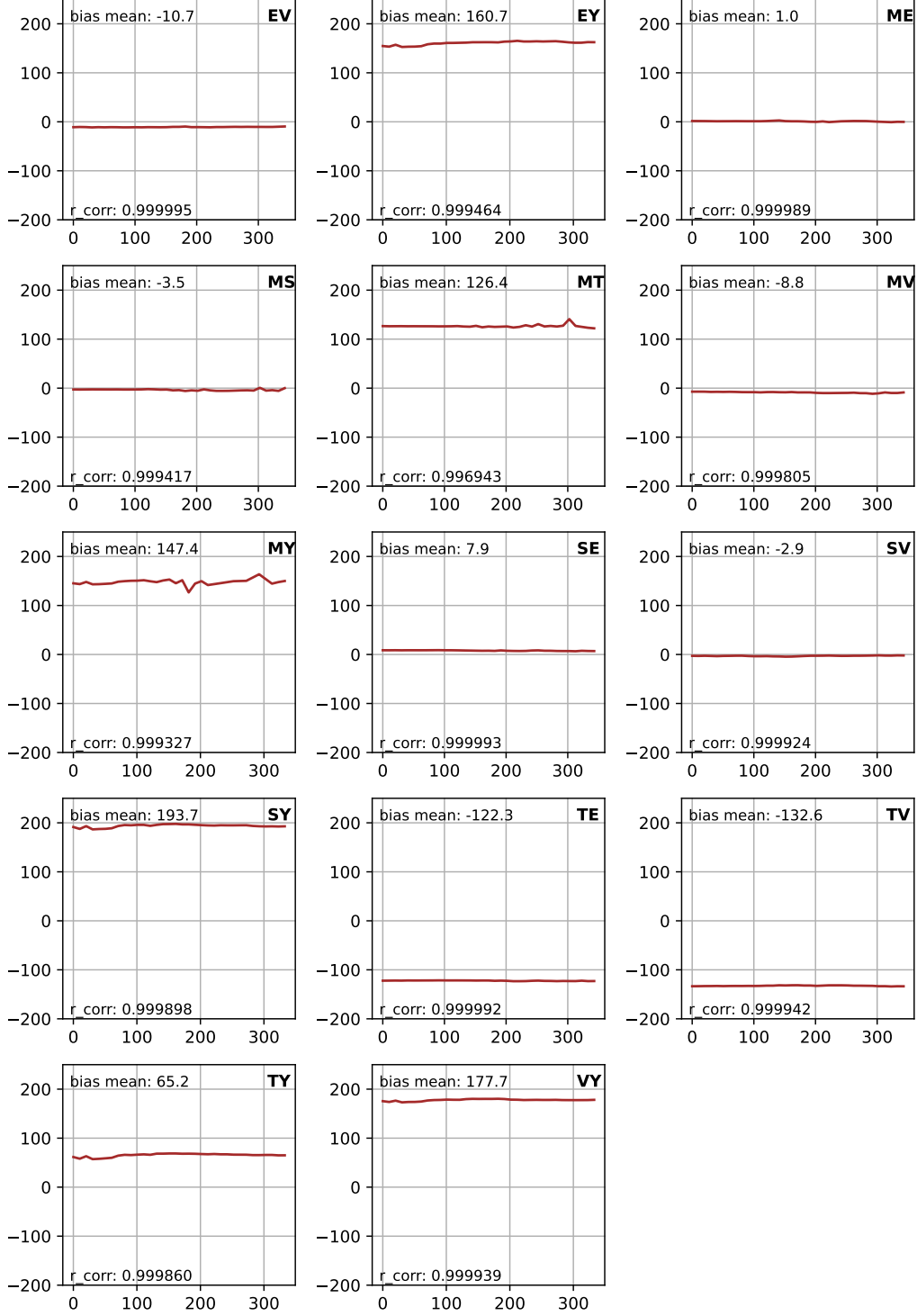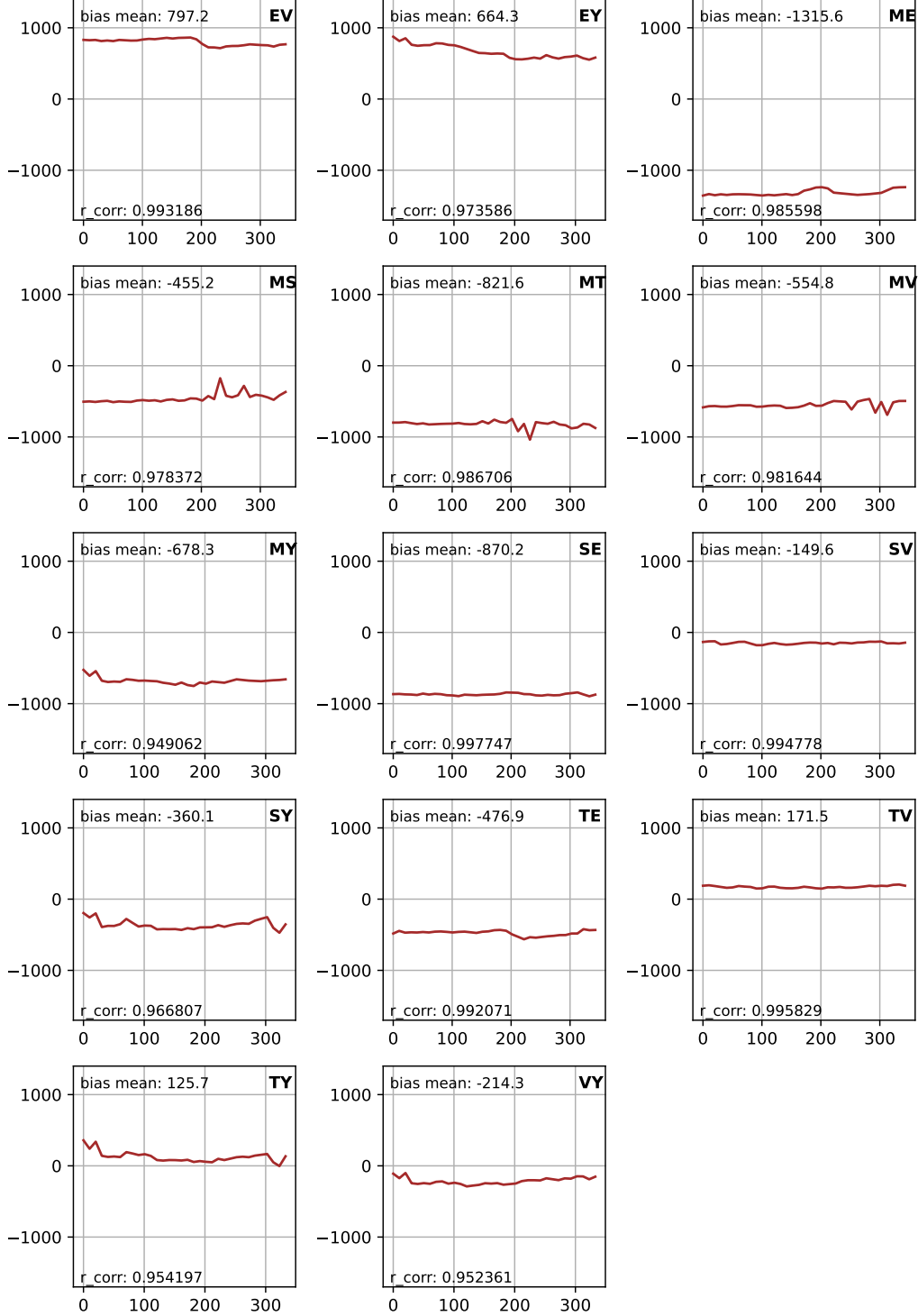
Figure 2: Evolution of the bias between LinI and CirI Multi-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows one curve of LinI - CirI, where LinI is I from the original, linearly polarized data set and CirI is I from the circularly polarized data set obtained by applying PolConvert to the original data. The means for each curve and the coefficients of correlation r_corr between LinI and CirI are given.
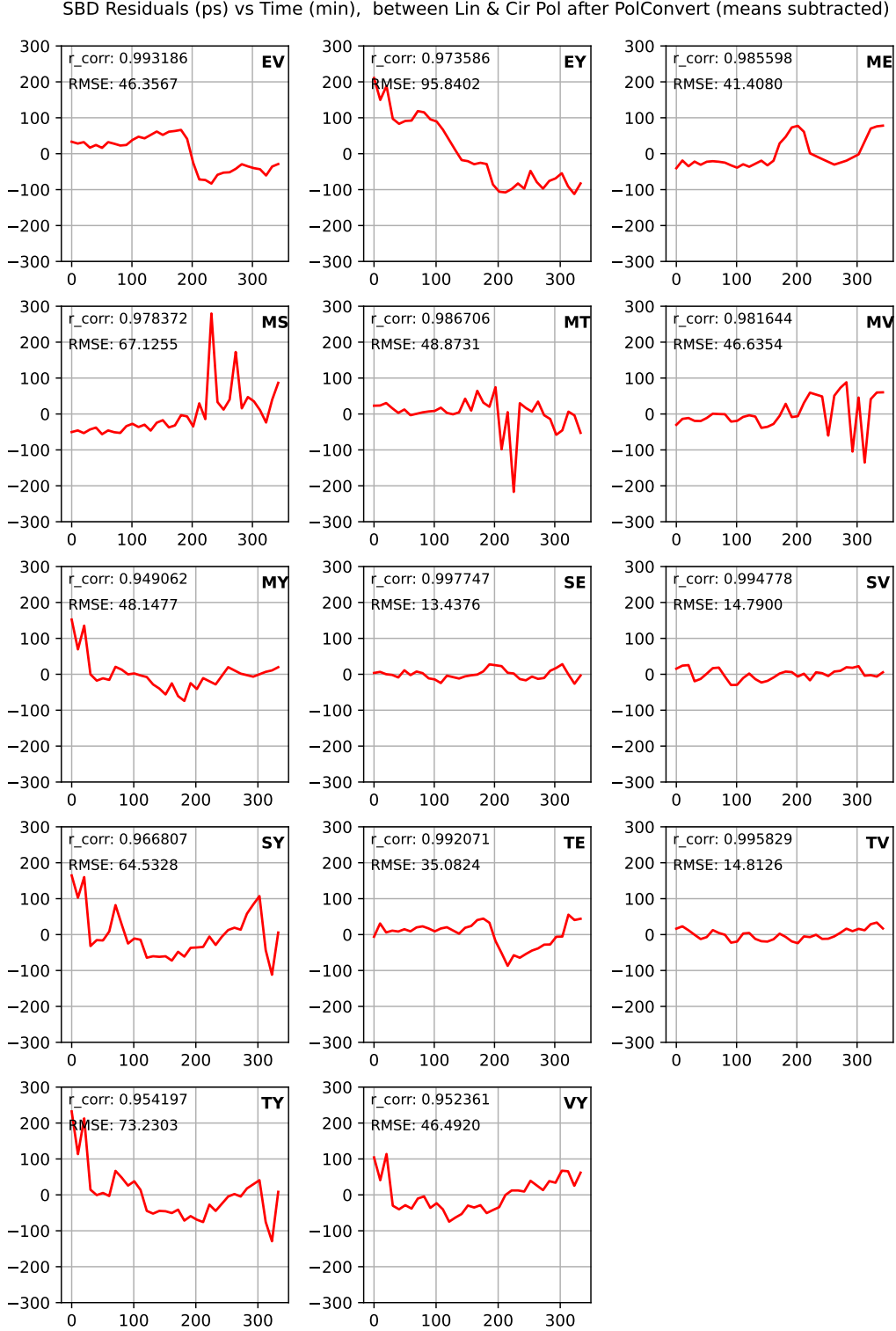
MBD Residuals (ps) vs Time (min), between Lin & Cir Pol after PolConvert (means subtracted)
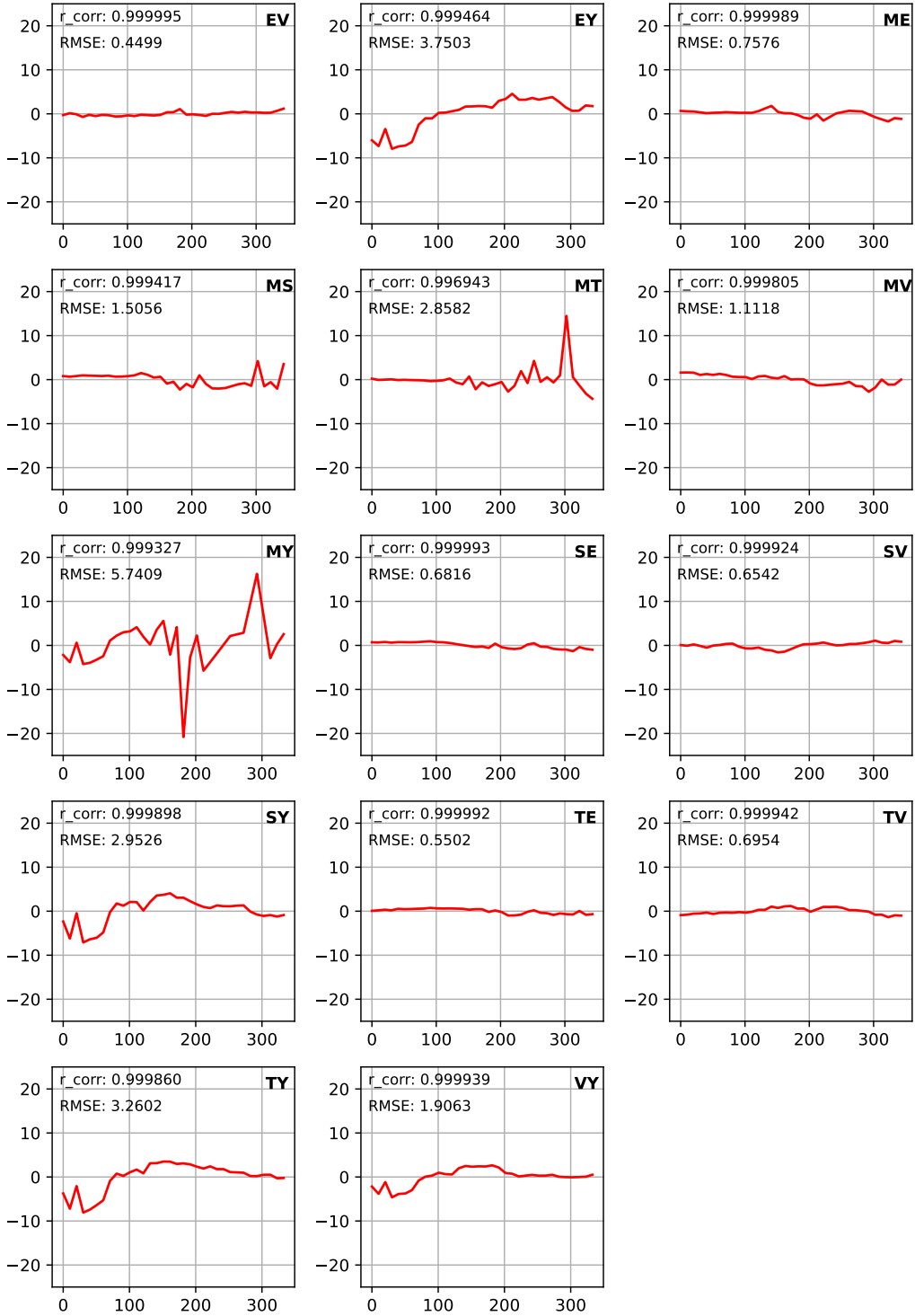
Figure 3: Evolution of the residuals between unbiased LinI and CirI Multi-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows a curve of LinI - CirI, both with subtracted means, where LinI is I from the original, linearly polarized data set and CirI is I from the circularly polarized data set obtained by applying PolConvert to the original data. The coefficients of correlation r_corr between LinI and CirI are very close to unity. The Root Mean Square Error (RMSE) shows the standard deviation of the residuals.

14

Figure 4: Evolution of Single-Band Delays (inpicoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows two curves: LinI, I from the original, linearly polarized data set and CirI, I from the circularly polarized data set obtained by applying PolConvert to the original data. Graphs have large biases. The coefficients of correlation r_corr between Lin_I and CirI for most of the baselines are close to unity. If it were not for the biases, the curves could be considered close to identical.

15

Figure 5: Evolution of the bias between LinI and CirI Single-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows a curve of LinI - CirI, where LinI is I from the original, linearly polarized data set and CirI is I from the circularly polarized data set obtained by applying PolConvert to the original data. The means for each curve and the coefficients of correlation r_corr between LinI and CirI are given.

Figure 6: Evolution of the residuals between unbiased LinI and CirI Single-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows a curve of LinI - CirI, both with subtracted means, where LinI is I from the original, linearly polarized data set and CirI is I from the circularly polarized data set obtained by applying PolConvert to the original data. The coefficients of correlation r_corr between LinI and CirI are reasonably close to unity. The Root Mean Square Error (RMSE) shows the standard deviation of the residuals.
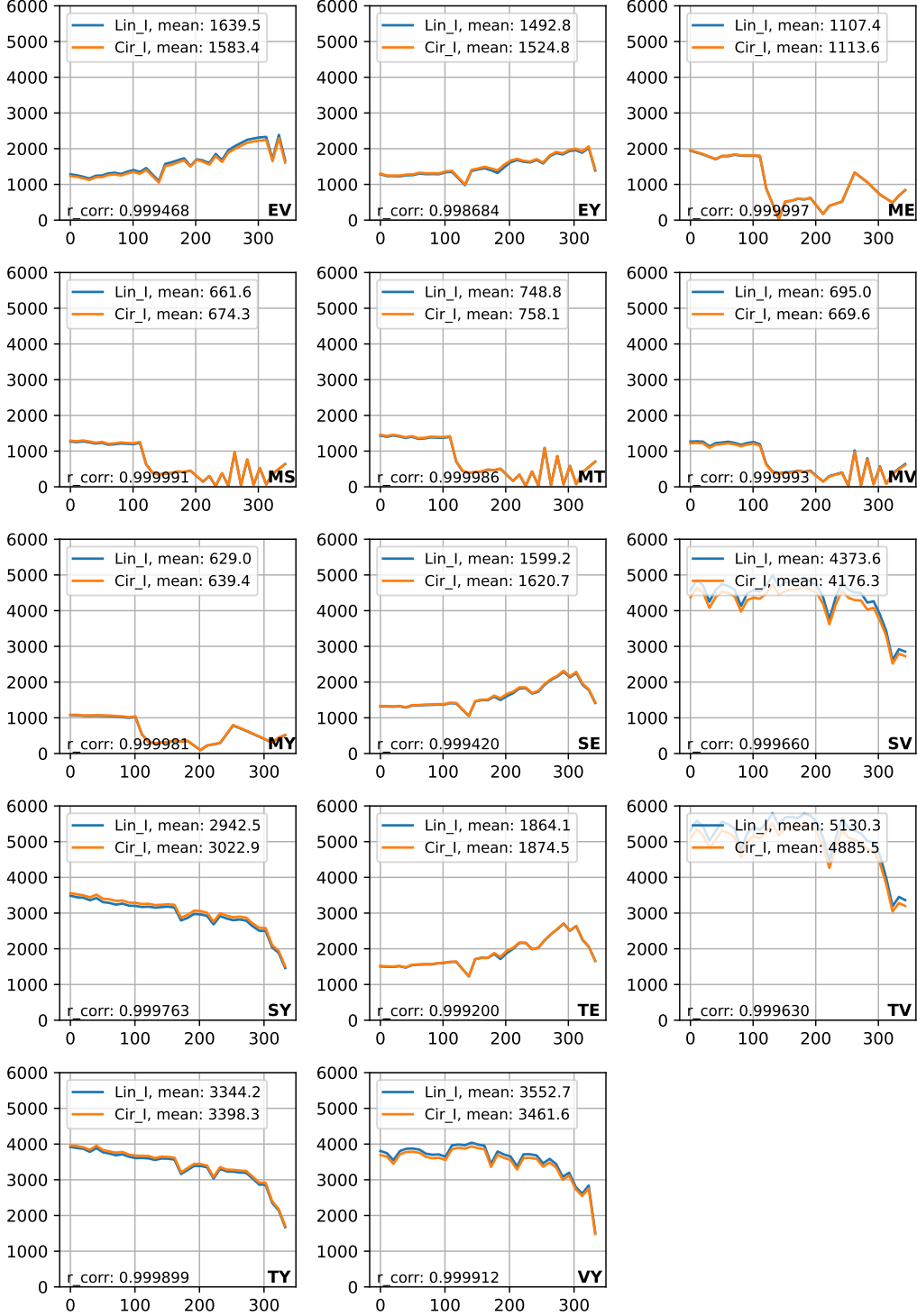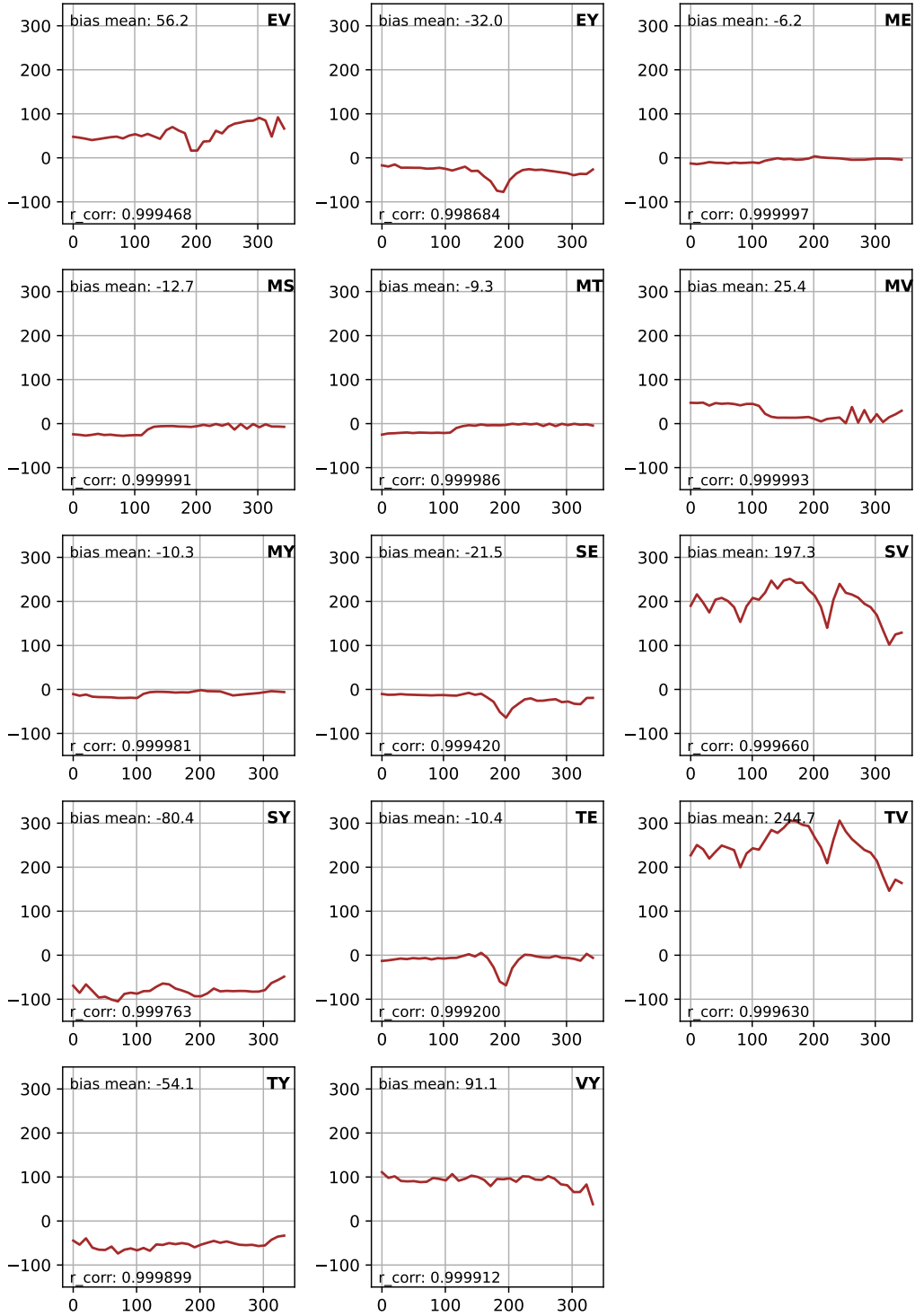
Figure 7: Evolution of Signal-to-Noise Ratios during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows two curves: LinI, I from the original, linearly polarized data set and CirI, I from the circularly polarized data set obtained by applying PolConvert to the original data. Graphs have large biases. However, the coefficients of correlation r_corr between LinI and CirI are so close to unity that the curves can be considered identical (except for the biases).
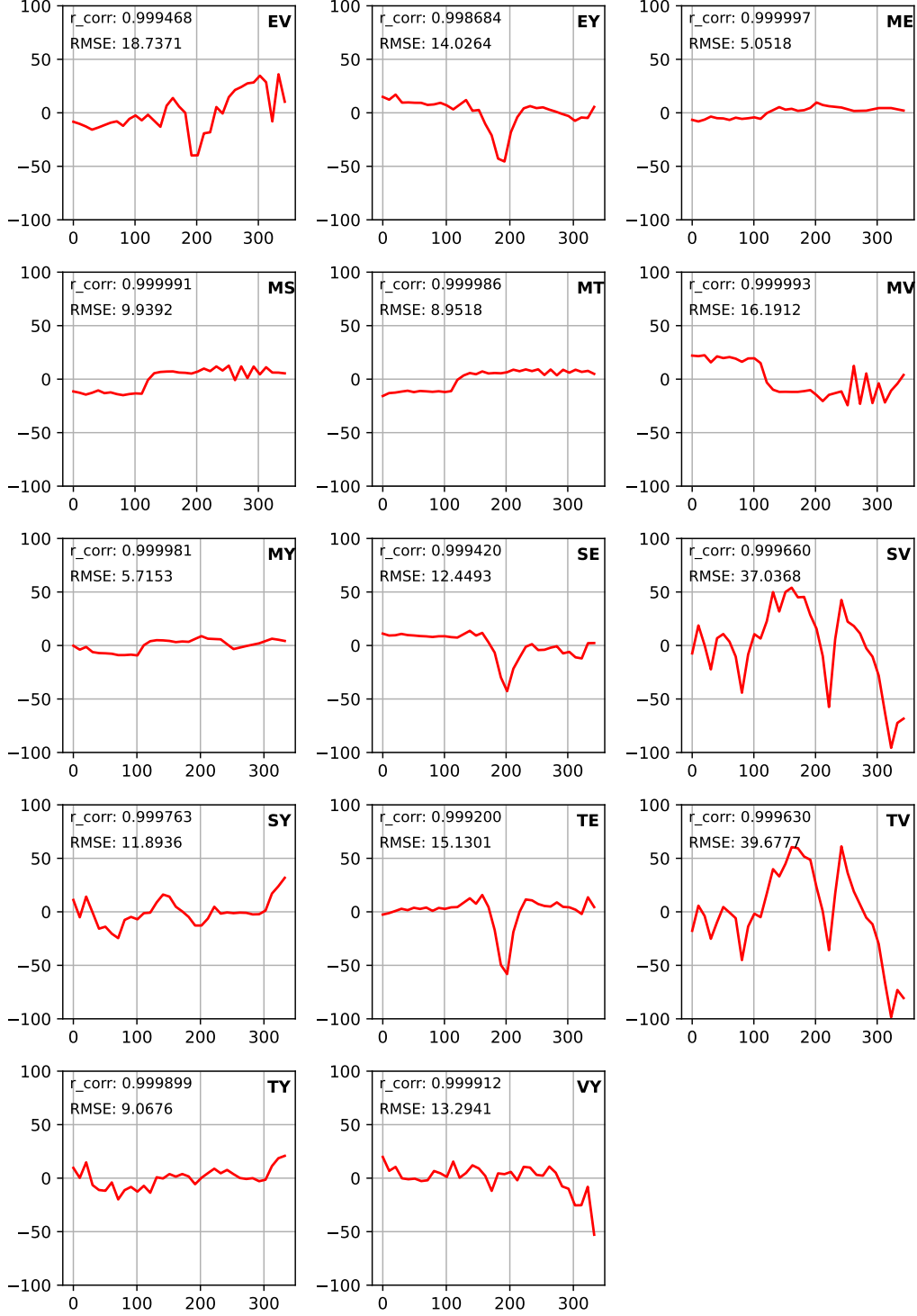
Figure 8: Evolution of the bias between LinI and CirI Signal-to-Noise Ratios during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows a curve of LinI - CirI, where LinI is I from the original, linearly polarized data set and CirI is I from the circularly polarized data set obtained by applying PolConvert to the original data. The means for each curve and the coefficients of correlation r_corr between LinI and CirI are given.

SNR Residuals vs Time (min), between Lin & Cir Pol after PolConvert (means subtracted)

Figure 9: Evolution of the residuals between unbiased LinI and CirI Signal-to-Noise Ratios during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows a curve of LinI - CirI, both with subtracted means, where LinI is I from the original, linearly polarized data set and CirI is I from the circularly polarized data set obtained by applying PolConvert to the original data. The coefficients of correlation r_corr between LinI and CirI are very close to unity. The Root Mean Square Error (RMSE) shows the standard deviation of the residuals.
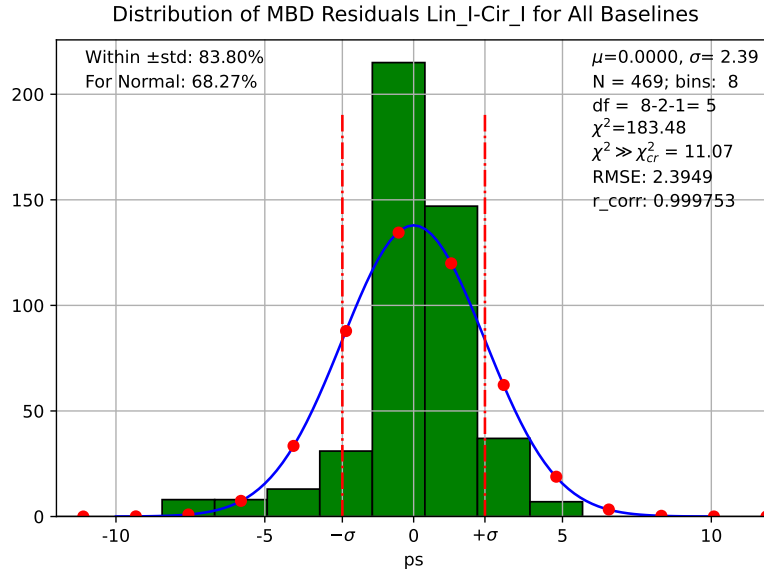
Figure 10: Histogram of the Multi-Band Delay residuals (see Eq. (3)) for all of the baselines. N is the total number of residuals spread over 21 bin. For the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (less than 5 counts in each) were grouped to make only $N_b = 8$ bins. The number of degrees of freedom df is 5 (i.e. $N_b - 3$). With this df, the critical value, χ^2_{cr} is 11.07. The χ^2 observed, 183.48, greatly exceeds χ^2_{cr}, so the test fails. However, the proportion of residuals within $\pm\sigma$, 83.8%, is much better than 68.27% of a normal distribution.
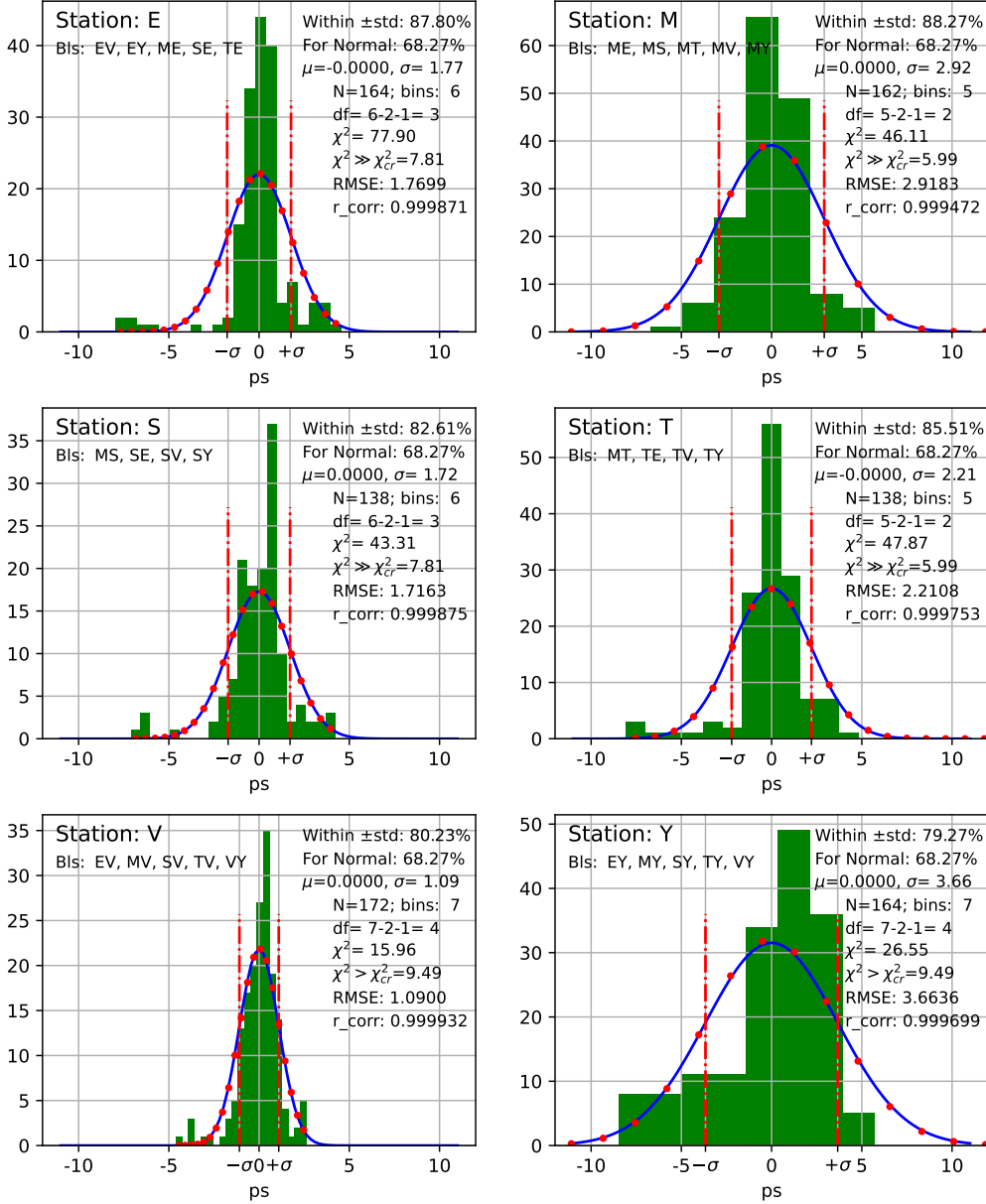
Figure 11: Histograms of the Multi-Band Delay residuals (see Eq. (3)) for the groups of baselines including an individual station. N is the total number of residuals spread over 21 bin. For the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (less than 5 counts in each) were grouped to make smaller number of bins N_b. The number of degrees of freedom df is $N_b - 3$. With these df, the critical values χ^2_{cr} are calculated. For all the distributions, the χ^2 observed exceed χ^2_{cr}, so the tests fail. However, the proportions of residuals within $\pm\sigma$ are much better than 68.27% of a normal distribution.
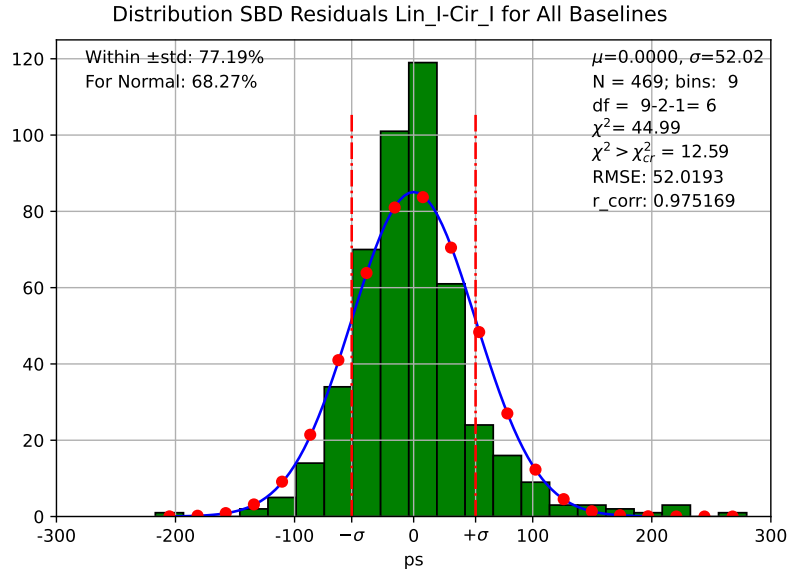
Figure 12: Histogram of the Single-Band Delay residuals (see Eq. (3)) for all of the baselines. N is the total number of residuals spread over 21 bin. For the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (less than 5 counts in each) were grouped to make only $N_b = 9$ bins. The number of degrees of freedom df is 6 (i.e. $N_b - 3$). With this df, the critical value, χ^2_{cr} is 12.59. The χ^2 observed, 44.99, significantly exceeds χ^2_{cr}, so the test fails. However, the proportion of residuals within $\pm\sigma$, 77.19%, is much better than 68.27% of a normal distribution.
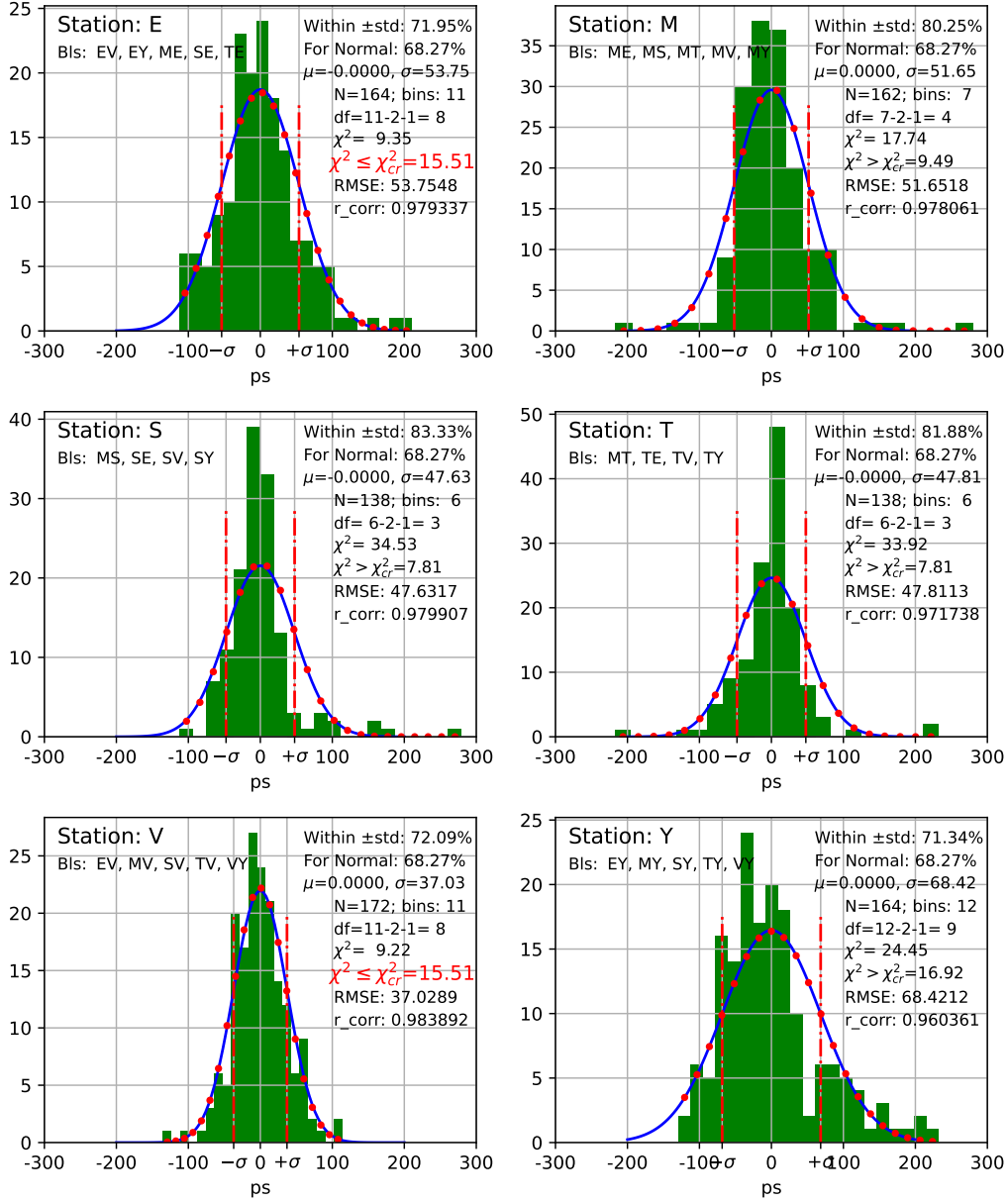
Figure 13: Histograms of the Single-Band Delay residuals (see Eq. (3)) for the groups of baselines including an individual station. N is the total number of residuals spread over 21 bin. For the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (less than 5 counts in each) were grouped to make smaller number of bins N_b. The number of degrees of freedom df is $N_b - 3$. With these df, the critical values χ^2_{cr} are calculated. For most of the distributions (except those including stations E and V), the χ^2 observed exceed χ^2_{cr}, so the tests fail. Stations E and V have normally distributed residuals with 95% confidence. However, for all of the stations, the proportions of residuals within $\pm\sigma$ are better than 68.27% of a normal distribution.
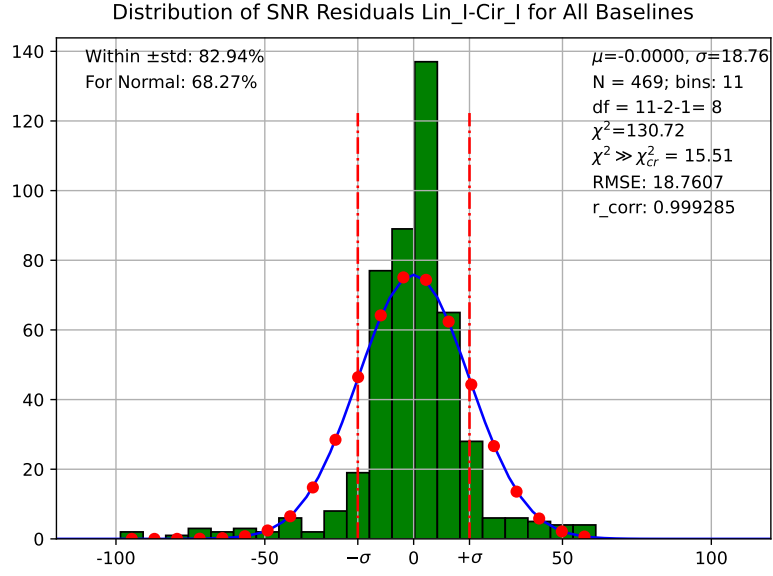
Figure 14: Histogram of the Signal-to-Noise Ratio residuals (see Eq. (3)) for all of the baselines. N is the total number of residuals spread over 21 bin. For the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (less than 5 counts in each) were grouped to make only $N_b = 11$ bins. The number of degrees of freedom df is 8 (i.e. $N_b - 3$). With this df, the critical value, χ^2_{cr} is 15.51. The χ^2 observed, 130.72, greatly exceeds χ^2_{cr}, so the test fails. However, the proportion of residuals within $\pm\sigma$, 82.94%, is much better than 68.27% of a normal distribution.
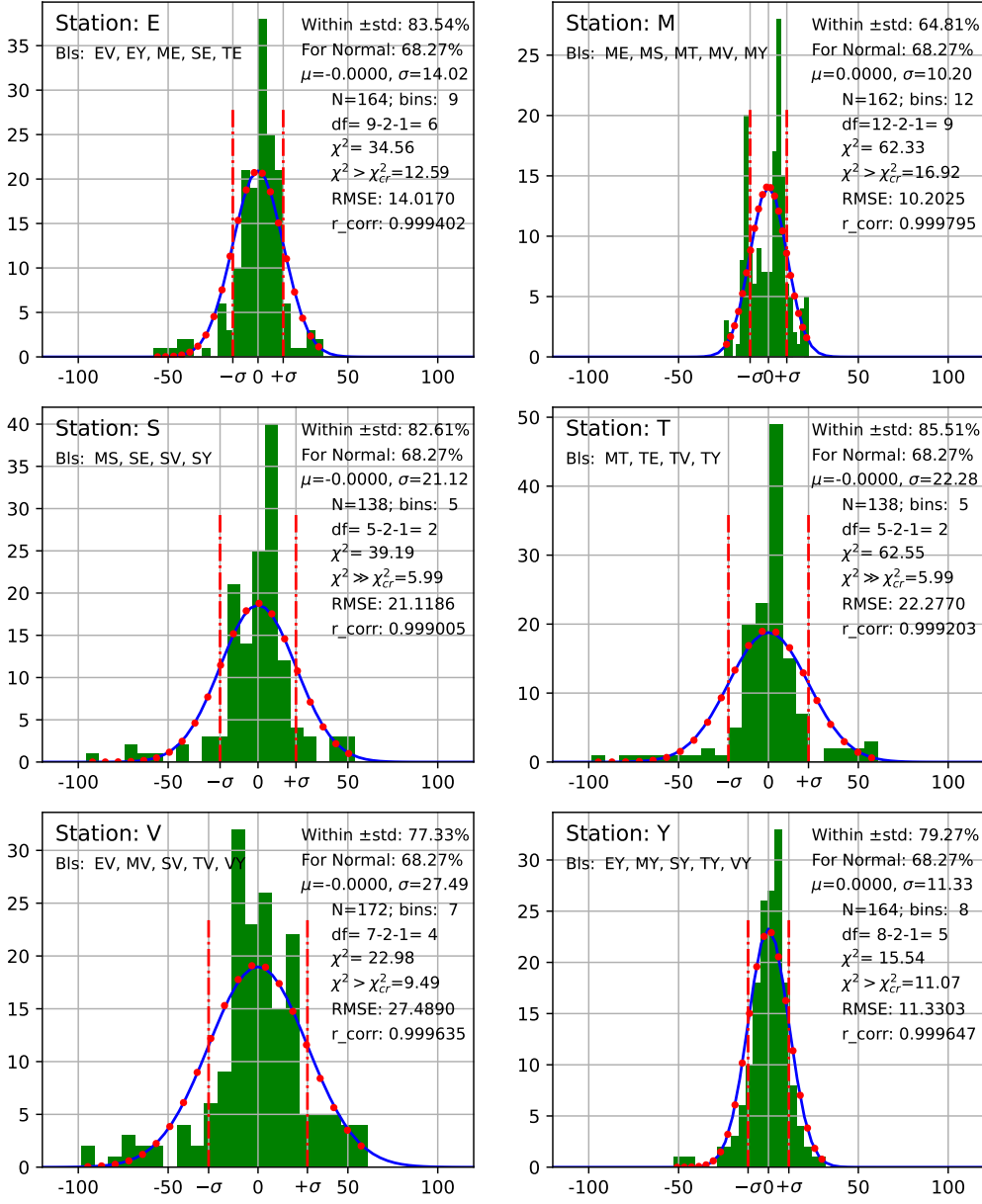
Figure 15: Histograms of the Signal-to-Noise Ratio residuals (see Eq. (3)) for the groups of baselines including an individual station. N is the total number of residuals spread over 21 bin. For the Pearson's χ^2 test, the left-tail and right-tail bins with sparse data (less than 5 counts in each) were grouped to make smaller number of bins N_b. The number of degrees of freedom df is $N_b - 3$. With these df, the critical values χ^2_{cr} are calculated. For all the distributions, the χ^2 observed exceed χ^2_{cr}, so the tests fail. However, the proportions of residuals within $\pm\sigma$ (except station M with its 64.81%) are much better than 68.27% of a normal distribution.