Comparison of Linear and Circular Products after PolConvert

L. V. Benkevitch[1]

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

August 22, 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 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 properties. 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 Lin_I(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 Cir_I(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 Lin_I[k] and Cir_I[k] parameters during the experiment before and after PolConvert, one graph for each baseline:

MBD variations in Fig. 1;
SBD variations in Fig. 2;
SNR variations in Fig. 3.

One can see that most of the graphs after PolConvert, Cir_I[k], are significantly shifted in values with respect to Lin_I[k]. However, subtracting means of each curve makes the differences far less significant:

Lin_I[k] = Lin_I[k] - mean(Lin_I[k])
Cir_I[k] = Cir_I[k] - mean(Cir_I[k])

In order to compare Lin_I[k] with Cir_I[k], the residual (res[k]) is computed as their difference:

$$\mathrm{res}[k] = \mathrm{Lin\_I}[k] - \mathrm{Cir\_I}[k] \tag{1}$$

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

MBD residual variations in Fig. 4;
SBD residual variations in Fig. 5;
SNR residual variations in Fig. 6.

To assess the similarity or even identity of Lin_I[k] and Cir_I[k], the following parameters are computed:

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

Since Lin_I[k] and Cir_I[k] already have their means subtracted, the correlation formula is simplified:

$$r_{corr} = \frac{\sum \mathrm{Lin\_I}[k] \cdot \mathrm{Cir\_I}[k]}{\sqrt{\sum \mathrm{Lin\_I}[k]^2 \cdot \mathrm{Cir\_I}[k]^2}} \tag{2}$$

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

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \sum \mathrm{res}[k]^2} \tag{3}$$

Lower RMSE indicates a better fit between the curves. However, RMSE is not an "absolute" indicator of the curves' proximity. The correlation coefficient, though, has its upper limit, unity, and if r_{corr} is very close to unity, the curves are almost identical, save their co0nstant bias. MBD and SNR demonstrate excellent correlations between Lin_I and Cir_I.

2 Parameter residual Statistics Before and After PolConvert

In order to estimate the errors introduced by PolConvert, histograms of the parameter residuals (after the mean subtraction) are plotted. Below are the histograms for all of the baselines:

MBD residual histograms for all of the baselines in Fig. 7;
SBD residual histograms for all of the baselines in Fig. 8;
SNR residual histograms for all of the baselines in Fig. 9.

Note that the means of the residuals are very close to zero. 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. Initially, all the histograms had 21 bins. The left-tail and right-tail bins with sparse data (frequencies smaller than 5) were grouped and the test used fewer number of bins (printed in each plot).

However, this method did not work: the residuals are grouped around their means substantially denser than the normal distributions with the same standard deviations. 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 the standard normal.

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. 10;
SBD residual histograms for the baselines including one station in Fig. 11;
SNR residual histograms for the baselines including one station in Fig. 12.

3 Software

3.1 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']`.

4 Conclusion

5 Appendix: Figures

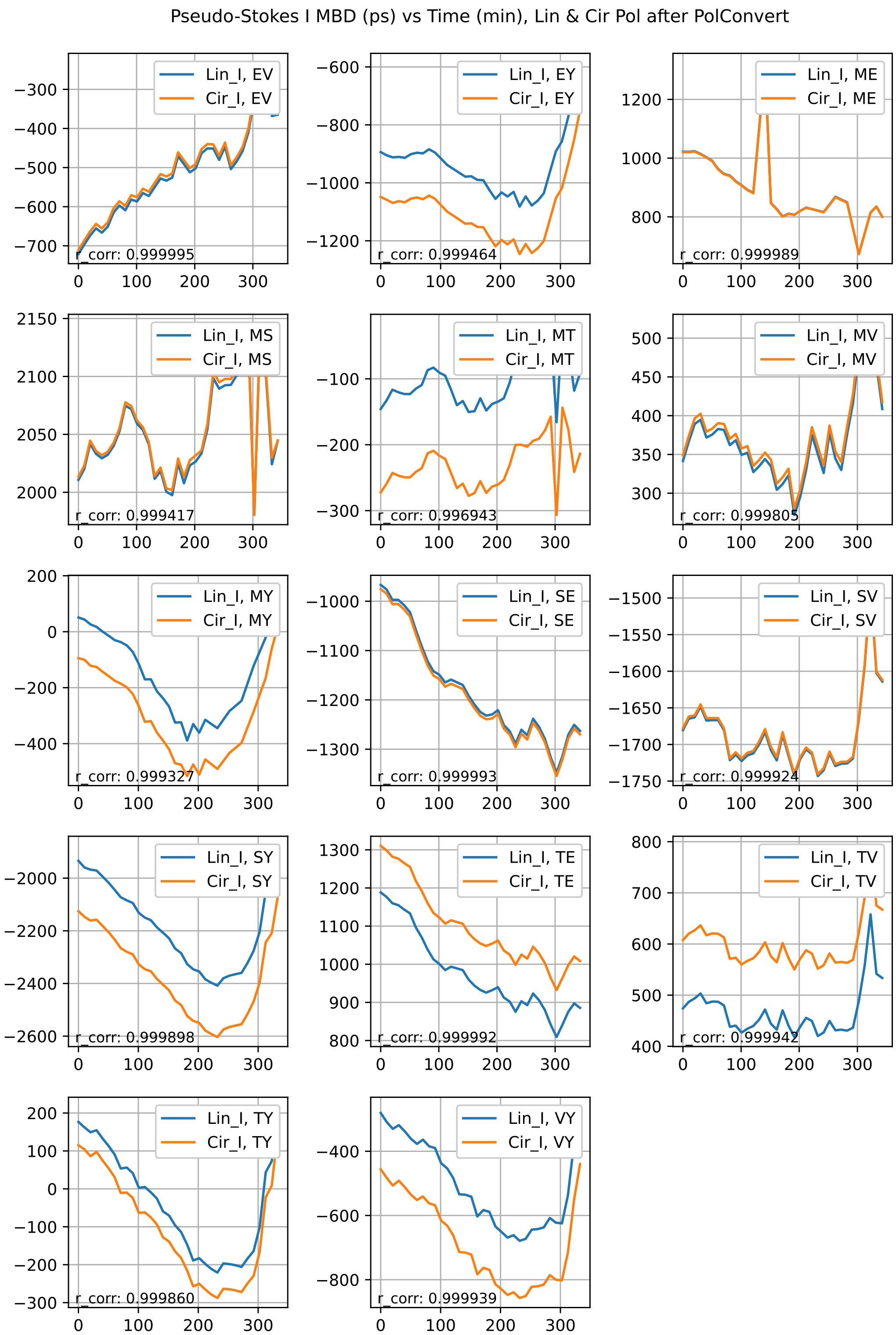

Figure 1: Evolution of Multi-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows two curves: Lin_I, I from the original, linearly polarized data set and Cir_I, I from the circularly polarized data set obtained by applying PolConvert to the original data. Graphs have large biases. However, the coefficients of correlation r_corr between Lin_I and Cir_I are so close to unity that the curves can be considered identical (except for the biases).

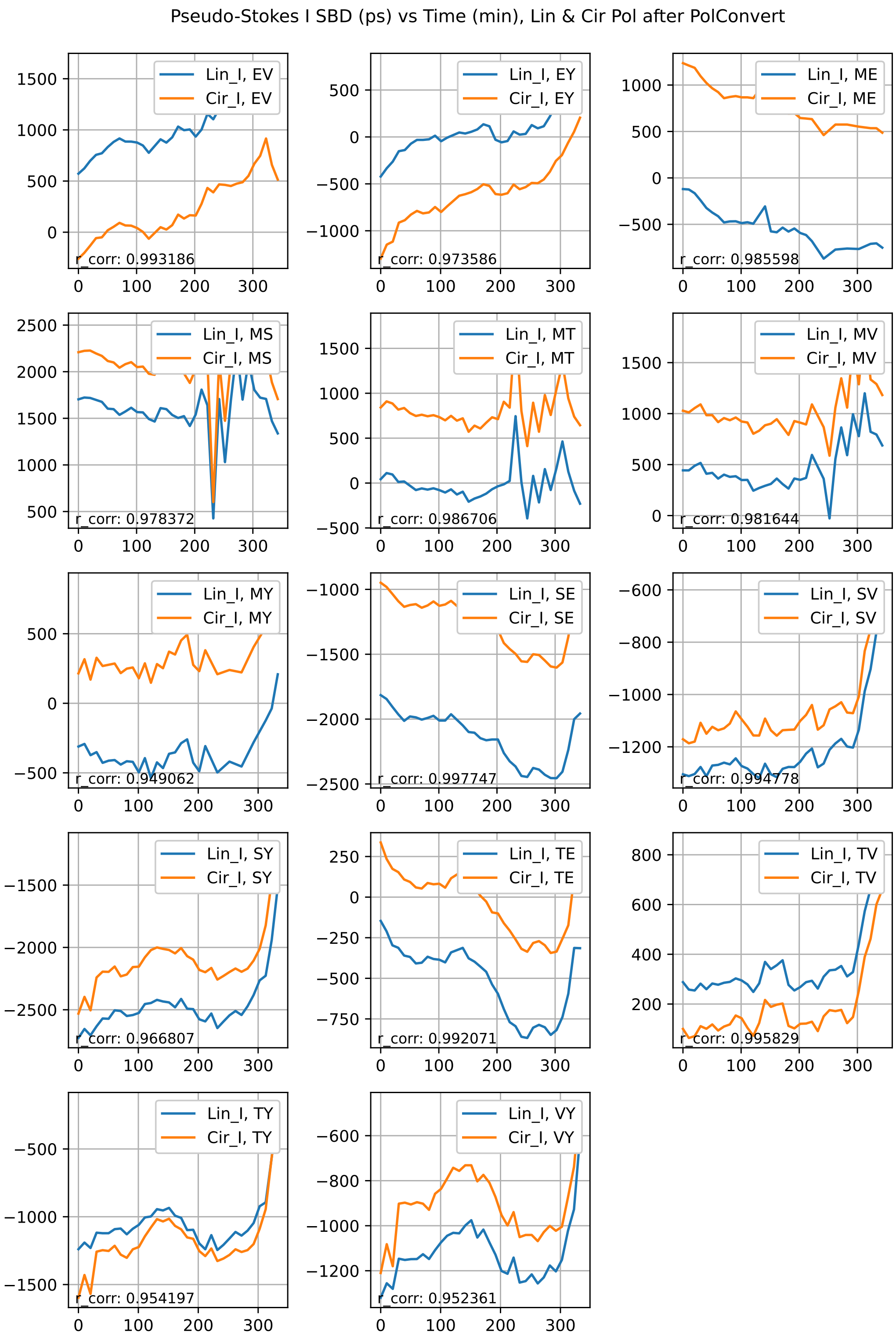

Figure 2: Evolution of Single-Band Delays (inpicoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows two curves: Lin_I, I from the original, linearly polarized data set and Cir_I, I from the circularly polarized data set obtained by applying PolConvert to the original data. Graphs have large biases. The coefficients of correlation r_corr between Lin_I and Cir_I for most of the baselines are close to unity. If it were not for the biases, the curves could be considered close to identical.

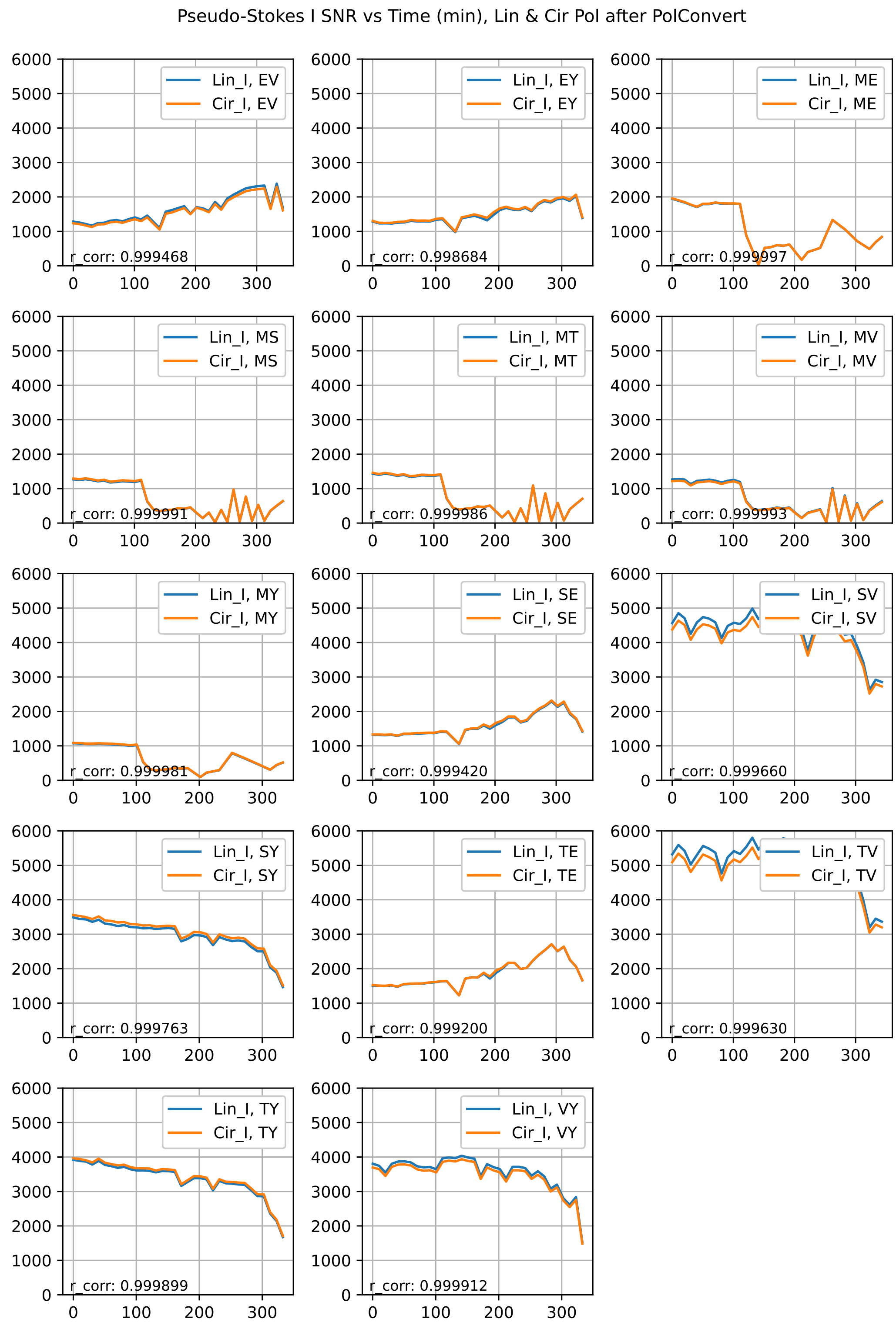

Figure 3: Evolution of Signal-to-Noise Ratios during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows two curves: Lin_I, I from the original, linearly polarized data set and Cir_I, I from the circularly polarized data set obtained by applying PolConvert to the original data. Graphs have large biases. However, the coefficients of correlation r_corr between Lin_I and Cir_I are so close to unity that the curves can be considered identical (except for the biases).

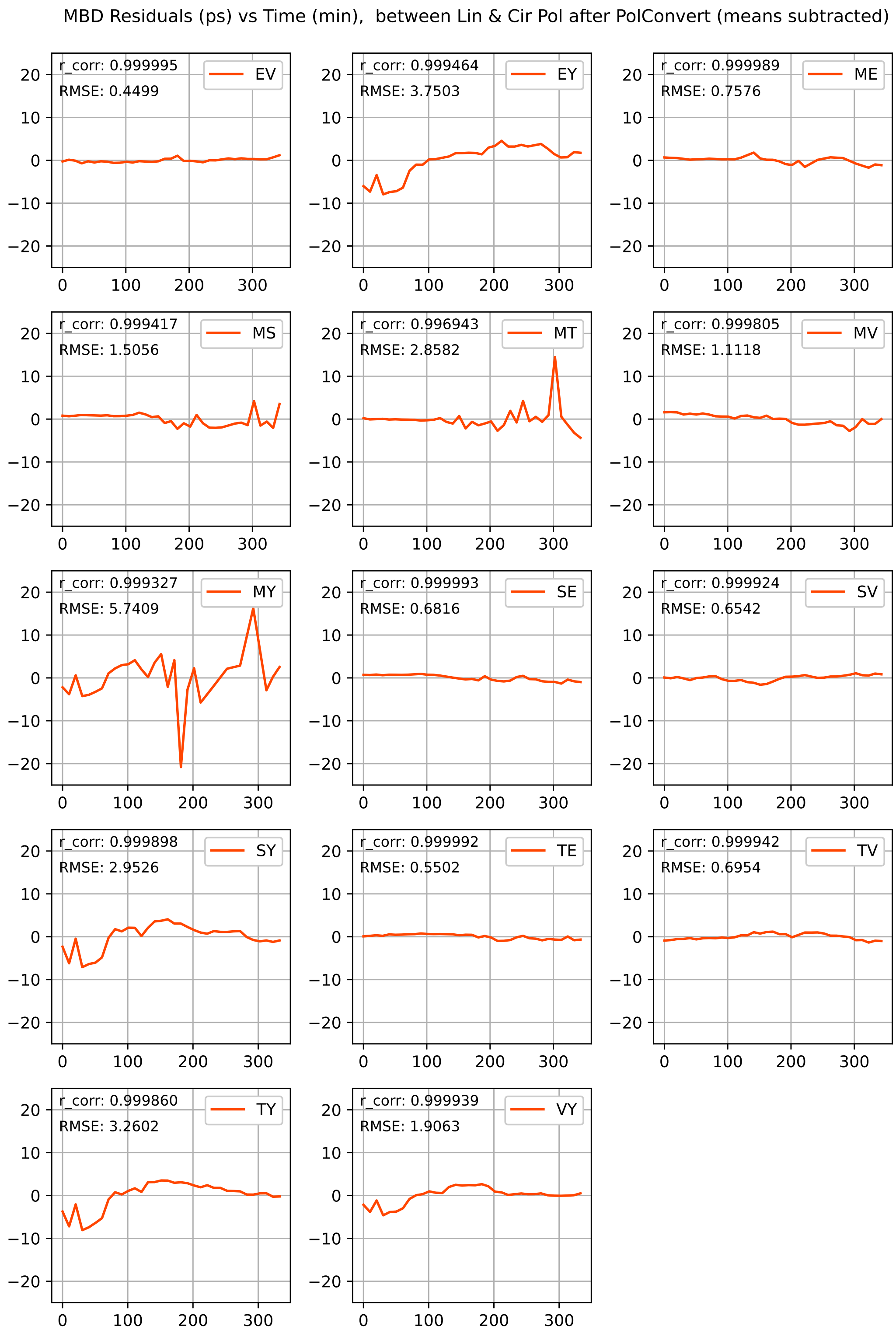

Figure 4: Evolution of the residuals between Lin_I and Cir_I Multi-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows one curve of Lin_I - Cir_I, where Lin_I is I from the original, linearly polarized data set and Cir_I is I from the circularly polarized data set obtained by applying PolConvert to the original data. The coefficients of correlation r_corr between Lin_I and Cir_I are very close to unity. The Root Mean Square Error (RMSE) shows the standard deviation of the residuals.

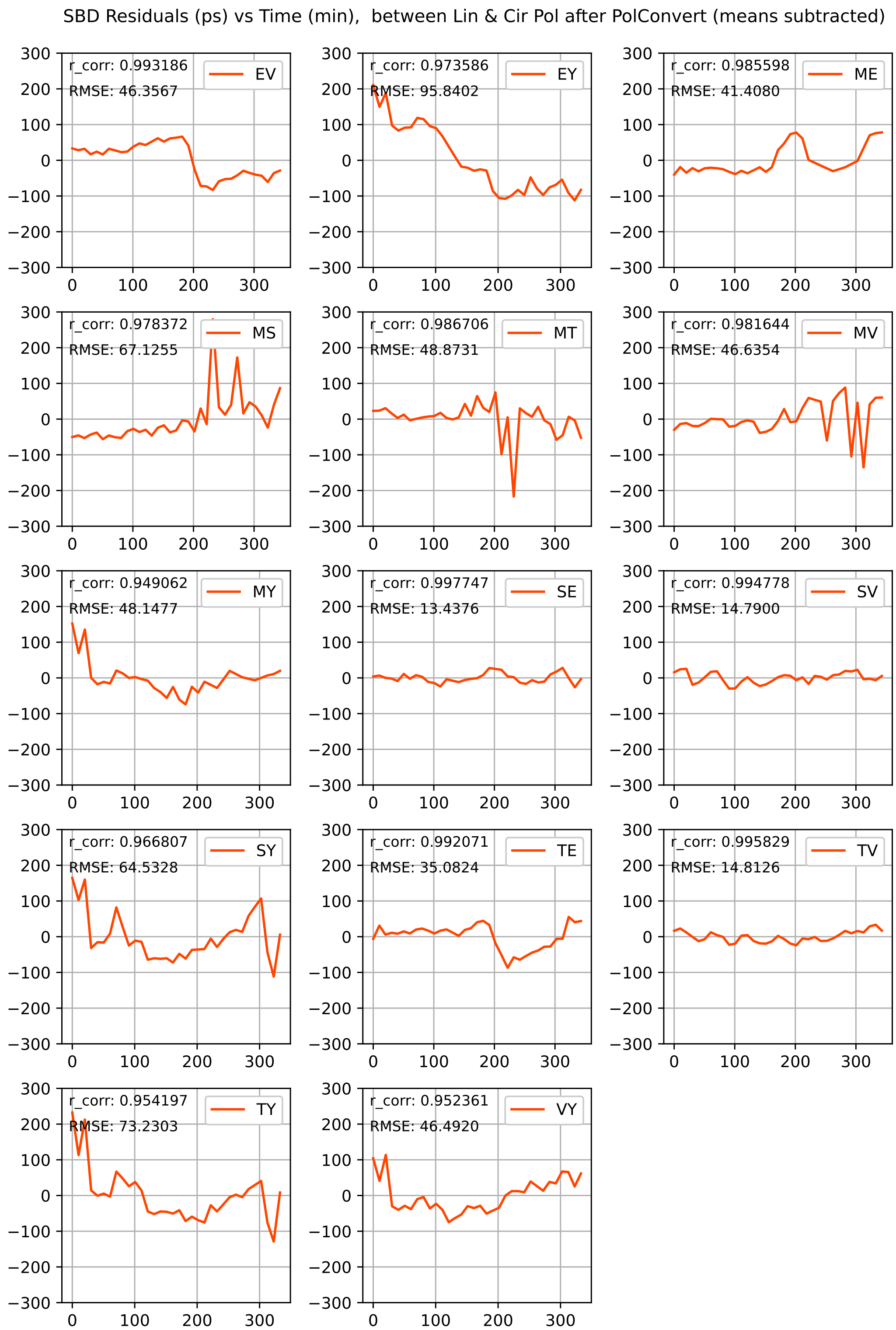

Figure 5: Evolution of the residuals between Lin_I and Cir_I Single-Band Delays (in picoseconds) during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows one curve of Lin_I - Cir_I, where Lin_I is I from the original, linearly polarized data set and Cir_I is I from the circularly polarized data set obtained by applying PolConvert to the original data. The coefficients of correlation r_corr between Lin_I and Cir_I are reasonably close to unity. The Root Mean Square Error (RMSE) shows the standard deviation of the residuals.

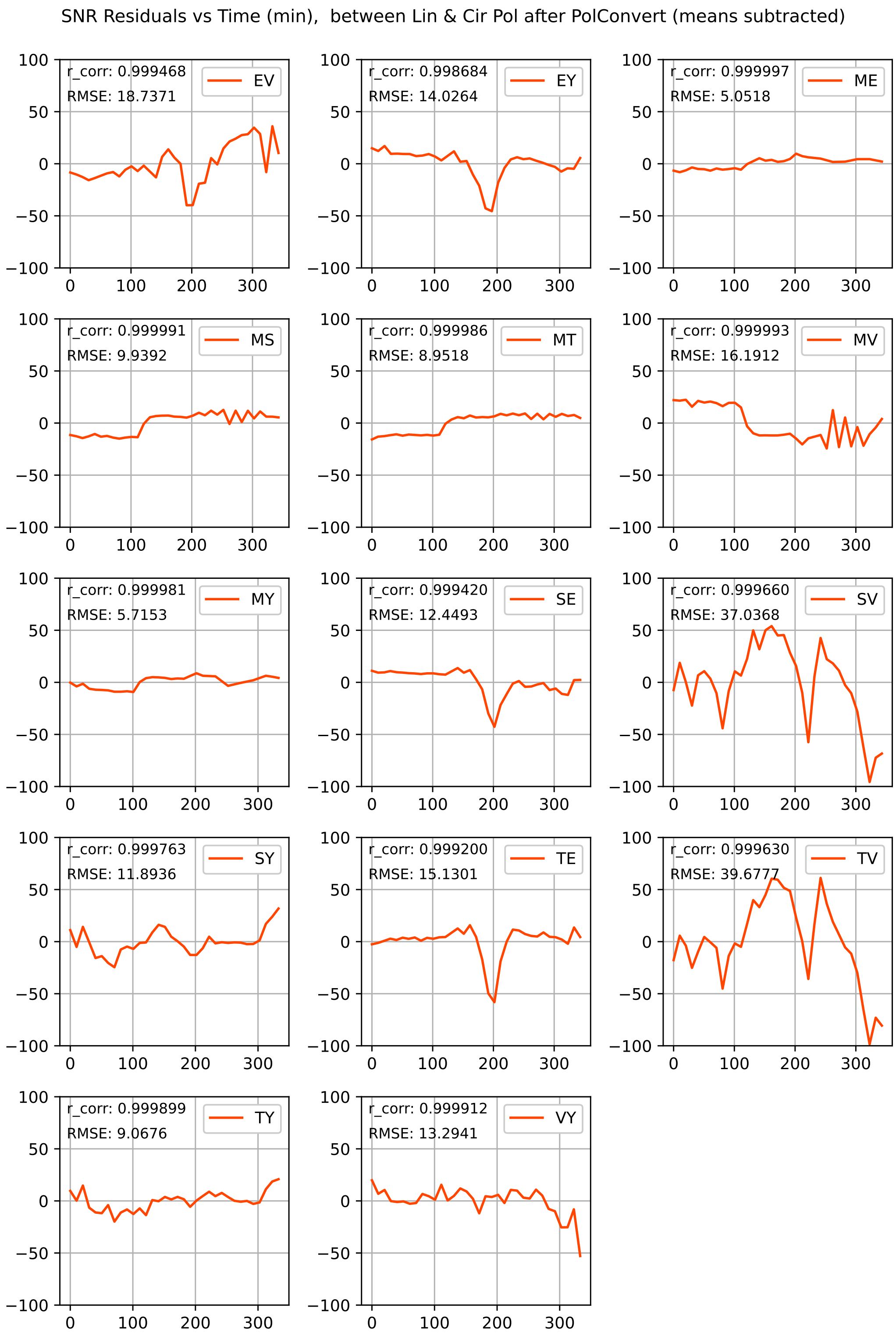

Figure 6: Evolution of the residuals between Lin_I and Cir_I Signal-to-Noise Ratios during the experiment (time in minutes) for the pseudo-Stokes I parameter for every baseline. Each panel shows one curve of Lin_I - Cir_I, where Lin_I is I from the original, linearly polarized data set and Cir_I is I from the circularly polarized data set obtained by applying PolConvert to the original data. The coefficients of correlation r_corr between Lin_I and Cir_I are very close to unity. The Root Mean Square Error (RMSE) shows the standard deviation of the residuals.

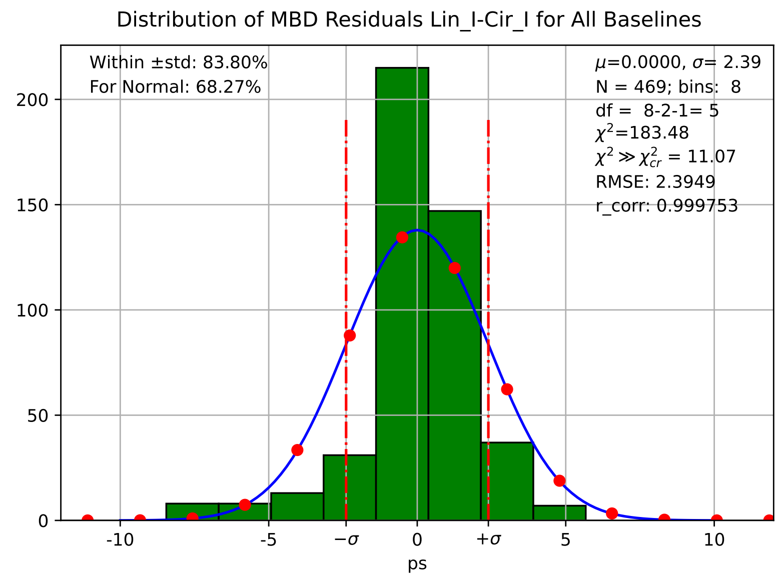

Figure 7:

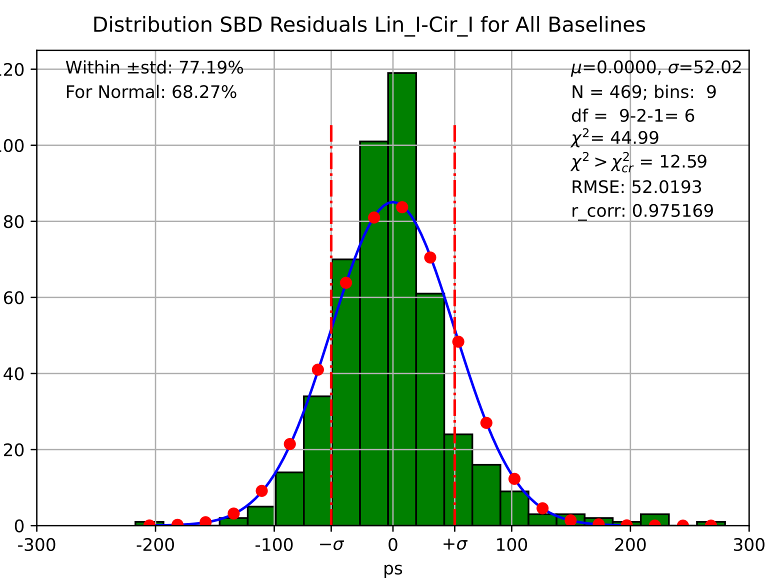

Figure 8:

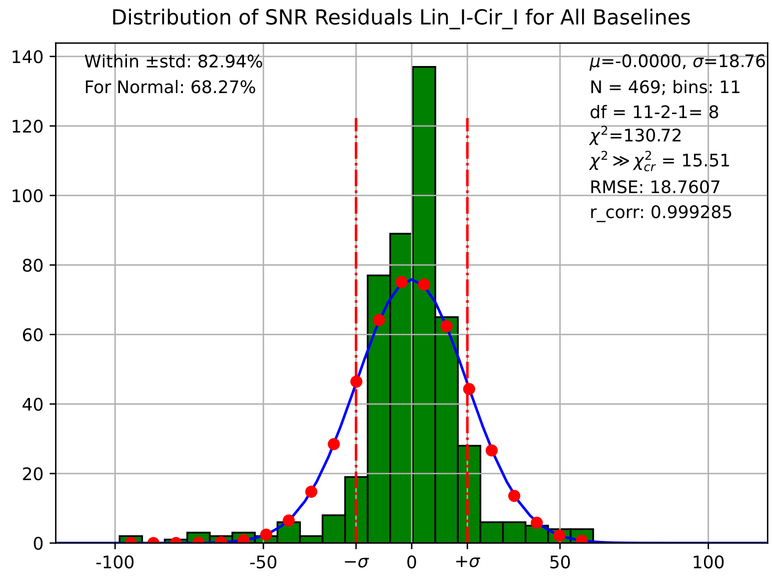
Distribution of SNR Residuals Lin_I-Cir_I for All Baselines
Within ±std: 82.94%
For Normal: 68.27%
μ=-0.0000, σ=18.76
N = 469; bins: 11
df = 11-2-1= 8
χ²=130.72
χ² ≫ χ²_cr = 15.51
RMSE: 18.7607
r_corr: 0.999285
140
120
100
80
60
40
20
0
-100
-50
−σ
0
+σ
50
100

Figure 9:

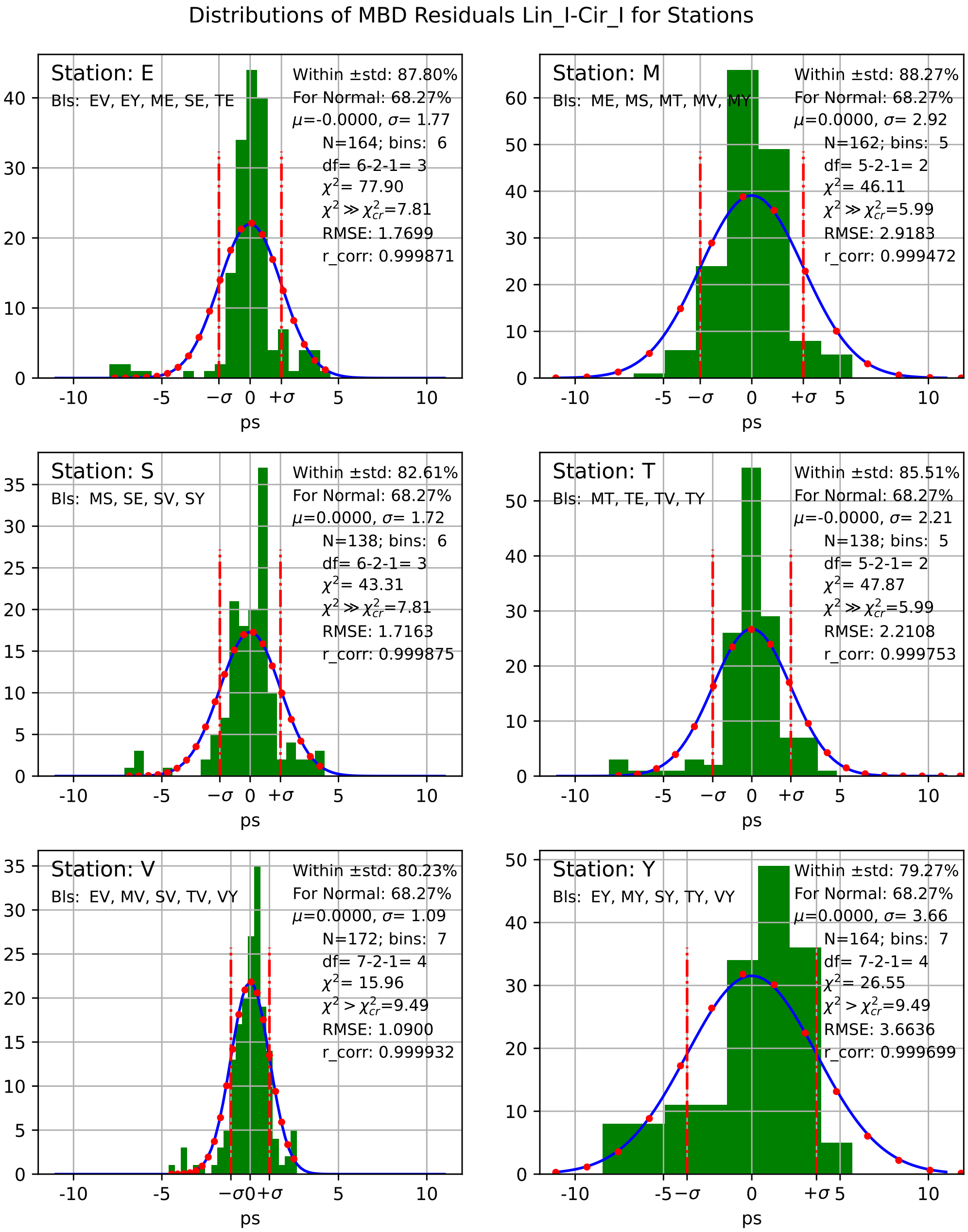
Distributions of MBD Residuals Lin_I-Cir_I for Stations
Station: E
Bls: EV, EY, ME, SE, TE
Within ±std: 87.80%
For Normal: 68.27%
μ=-0.0000, σ= 1.77
N=164; bins: 6
df= 6-2-1= 3
χ²= 77.90
χ² ≫ χ²cr=7.81
RMSE: 1.7699
r_corr: 0.999871
Station: M
Bls: ME, MS, MT, MV, MY
Within ±std: 88.27%
For Normal: 68.27%
μ=0.0000, σ= 2.92
N=162; bins: 5
df= 5-2-1= 2
χ²= 46.11
χ² ≫ χ²cr=5.99
RMSE: 2.9183
r_corr: 0.999472
Station: S
Bls: MS, SE, SV, SY
Within ±std: 82.61%
For Normal: 68.27%
μ=0.0000, σ= 1.72
N=138; bins: 6
df= 6-2-1= 3
χ²= 43.31
χ² ≫ χ²cr=7.81
RMSE: 1.7163
r_corr: 0.999875
Station: T
Bls: MT, TE, TV, TY
Within ±std: 85.51%
For Normal: 68.27%
μ=-0.0000, σ= 2.21
N=138; bins: 5
df= 5-2-1= 2
χ²= 47.87
χ² ≫ χ²cr=5.99
RMSE: 2.2108
r_corr: 0.999753
Station: V
Bls: EV, MV, SV, TV, VY
Within ±std: 80.23%
For Normal: 68.27%
μ=0.0000, σ= 1.09
N=172; bins: 7
df= 7-2-1= 4
χ²= 15.96
χ² > χ²cr=9.49
RMSE: 1.0900
r_corr: 0.999932
Station: Y
Bls: EY, MY, SY, TY, VY
Within ±std: 79.27%
For Normal: 68.27%
μ=0.0000, σ= 3.66
N=164; bins: 7
df= 7-2-1= 4
χ²= 26.55
χ² > χ²cr=9.49
RMSE: 3.6636
r_corr: 0.999699
ps

Figure 10:

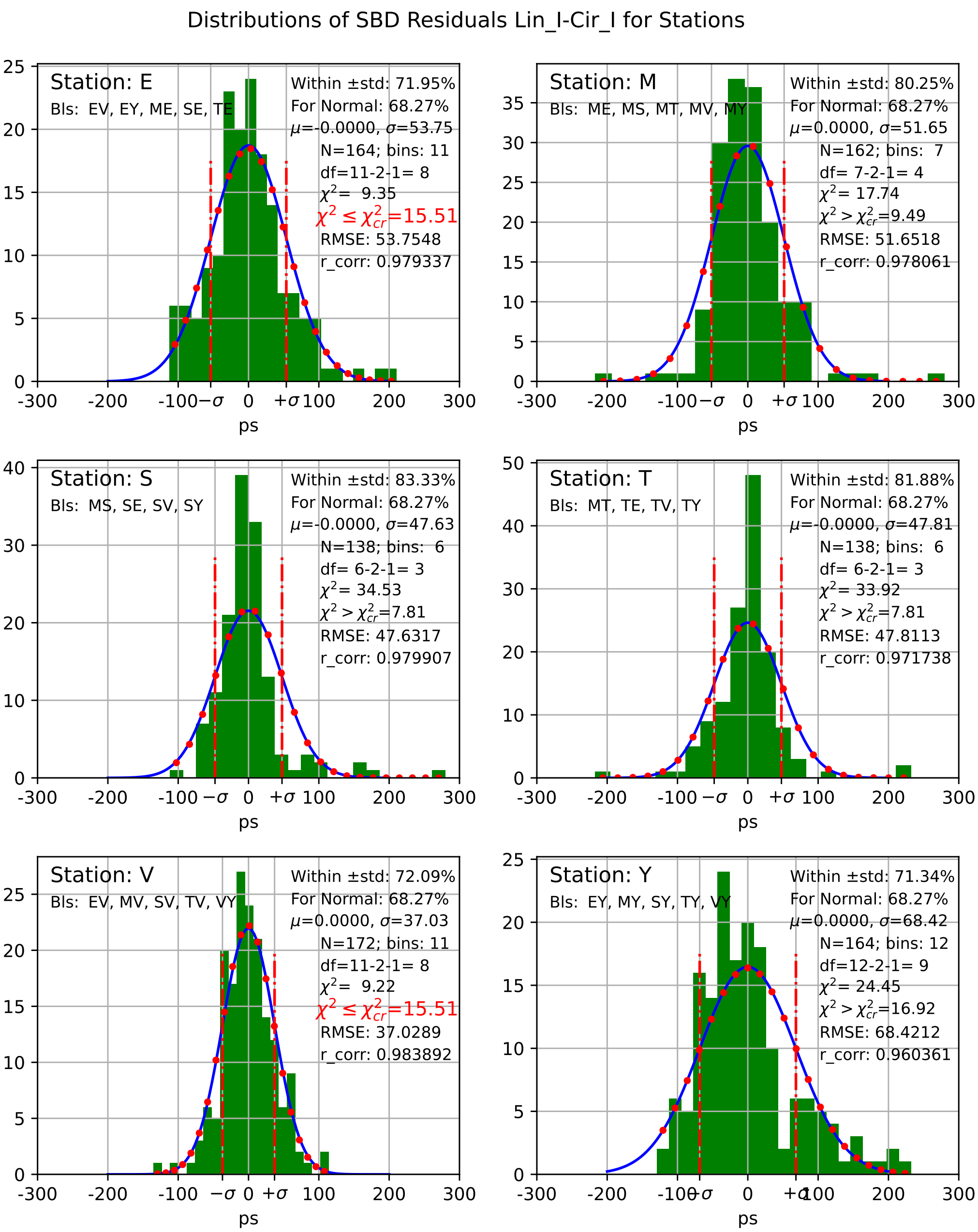
Distributions of SBD Residuals Lin_I-Cir_I for Stations
Station: E
Bls: EV, EY, ME, SE, TE
Within ±std: 71.95%
For Normal: 68.27%
μ=-0.0000, σ=53.75
N=164; bins: 11
df=11-2-1= 8
χ2= 9.35
χ2 ≤ χ2cr=15.51
RMSE: 53.7548
r_corr: 0.979337
Station: M
Bls: ME, MS, MT, MV, MY
Within ±std: 80.25%
For Normal: 68.27%
μ=0.0000, σ=51.65
N=162; bins: 7
df= 7-2-1= 4
χ2= 17.74
χ2 > χ2cr=9.49
RMSE: 51.6518
r_corr: 0.978061
Station: S
Bls: MS, SE, SV, SY
Within ±std: 83.33%
For Normal: 68.27%
μ=-0.0000, σ=47.63
N=138; bins: 6
df= 6-2-1= 3
χ2= 34.53
χ2 > χ2cr=7.81
RMSE: 47.6317
r_corr: 0.979907
Station: T
Bls: MT, TE, TV, TY
Within ±std: 81.88%
For Normal: 68.27%
μ=-0.0000, σ=47.81
N=138; bins: 6
df= 6-2-1= 3
χ2= 33.92
χ2 > χ2cr=7.81
RMSE: 47.8113
r_corr: 0.971738
Station: V
Bls: EV, MV, SV, TV, VY
Within ±std: 72.09%
For Normal: 68.27%
μ=0.0000, σ=37.03
N=172; bins: 11
df=11-2-1= 8
χ2= 9.22
χ2 ≤ χ2cr=15.51
RMSE: 37.0289
r_corr: 0.983892
Station: Y
Bls: EY, MY, SY, TY, VY
Within ±std: 71.34%
For Normal: 68.27%
μ=0.0000, σ=68.42
N=164; bins: 12
df=12-2-1= 9
χ2= 24.45
χ2 > χ2cr=16.92
RMSE: 68.4212
r_corr: 0.960361
ps

Figure 11:

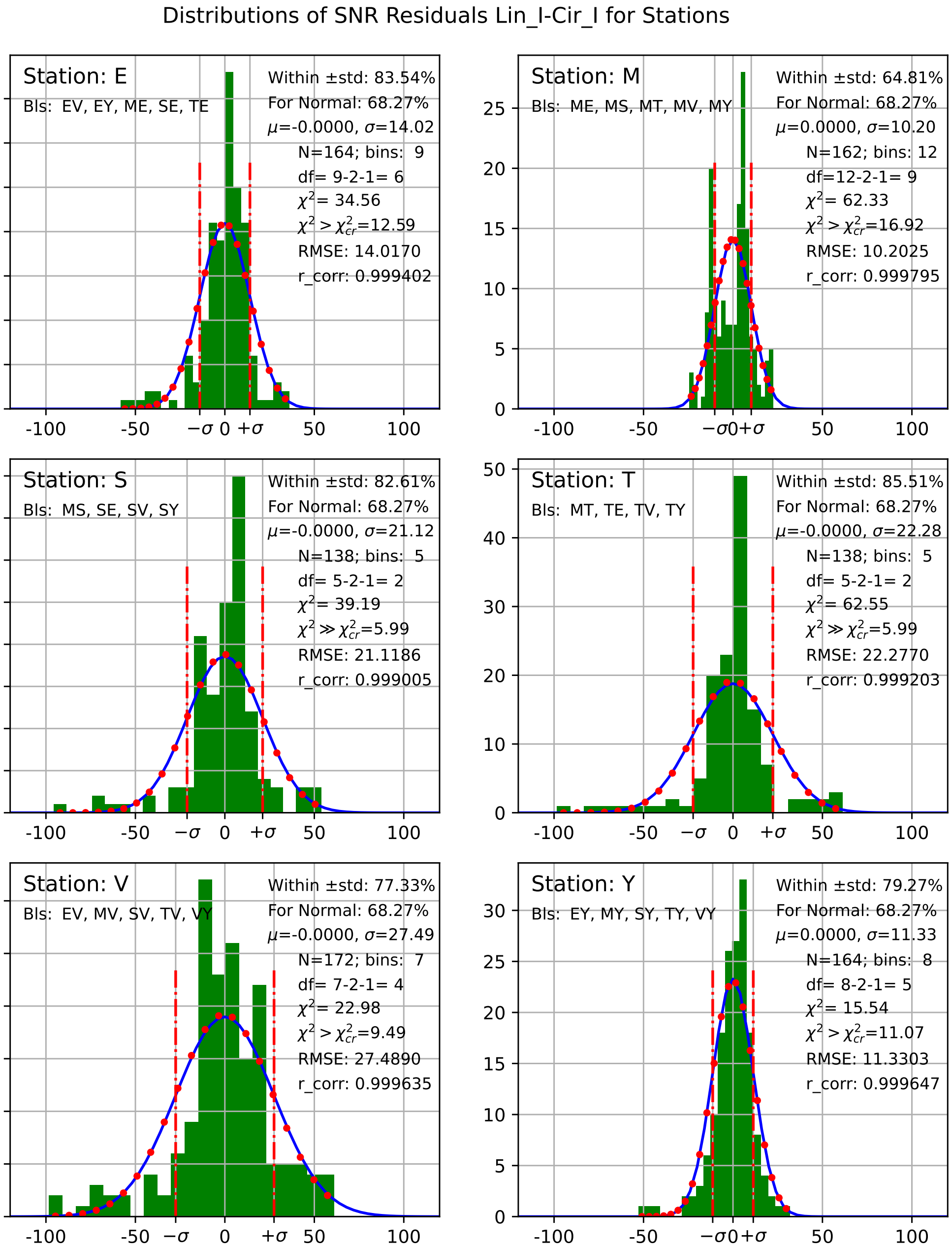
Distributions of SNR Residuals Lin_I-Cir_I for Stations
Station: E
Bls: EV, EY, ME, SE, TE
Within ±std: 83.54%
For Normal: 68.27%
μ=-0.0000, σ=14.02
N=164; bins: 9
df= 9-2-1= 6
χ²= 34.56
χ² > χ²cr=12.59
RMSE: 14.0170
r_corr: 0.999402
Station: M
Bls: ME, MS, MT, MV, MY
Within ±std: 64.81%
For Normal: 68.27%
μ=0.0000, σ=10.20
N=162; bins: 12
df=12-2-1= 9
χ²= 62.33
χ² > χ²cr=16.92
RMSE: 10.2025
r_corr: 0.999795
Station: S
Bls: MS, SE, SV, SY
Within ±std: 82.61%
For Normal: 68.27%
μ=-0.0000, σ=21.12
N=138; bins: 5
df= 5-2-1= 2
χ²= 39.19
χ² ≫ χ²cr=5.99
RMSE: 21.1186
r_corr: 0.999005
Station: T
Bls: MT, TE, TV, TY
Within ±std: 85.51%
For Normal: 68.27%
μ=-0.0000, σ=22.28
N=138; bins: 5
df= 5-2-1= 2
χ²= 62.55
χ² ≫ χ²cr=5.99
RMSE: 22.2770
r_corr: 0.999203
Station: V
Bls: EV, MV, SV, TV, VY
Within ±std: 77.33%
For Normal: 68.27%
μ=-0.0000, σ=27.49
N=172; bins: 7
df= 7-2-1= 4
χ²= 22.98
χ² > χ²cr=9.49
RMSE: 27.4890
r_corr: 0.999635
Station: Y
Bls: EY, MY, SY, TY, VY
Within ±std: 79.27%
For Normal: 68.27%
μ=0.0000, σ=11.33
N=164; bins: 8
df= 8-2-1= 5
χ²= 15.54
χ² > χ²cr=11.07
RMSE: 11.3303
r_corr: 0.999647

Figure 12: