Comparison of Linear and Circular Products after PolConvert

L. V. Benkevitch[1]

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

August 12, 2024

Abstract

Contents

1 Parameter Variations Before and After PolConvert

Converting liear 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. The following Figures show temporal variations of the 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 are significantly shifted in values. However, subtracting means of each curve makes the differences less significant. The graphs of differences with the means subtracted are shown in the following Figures:

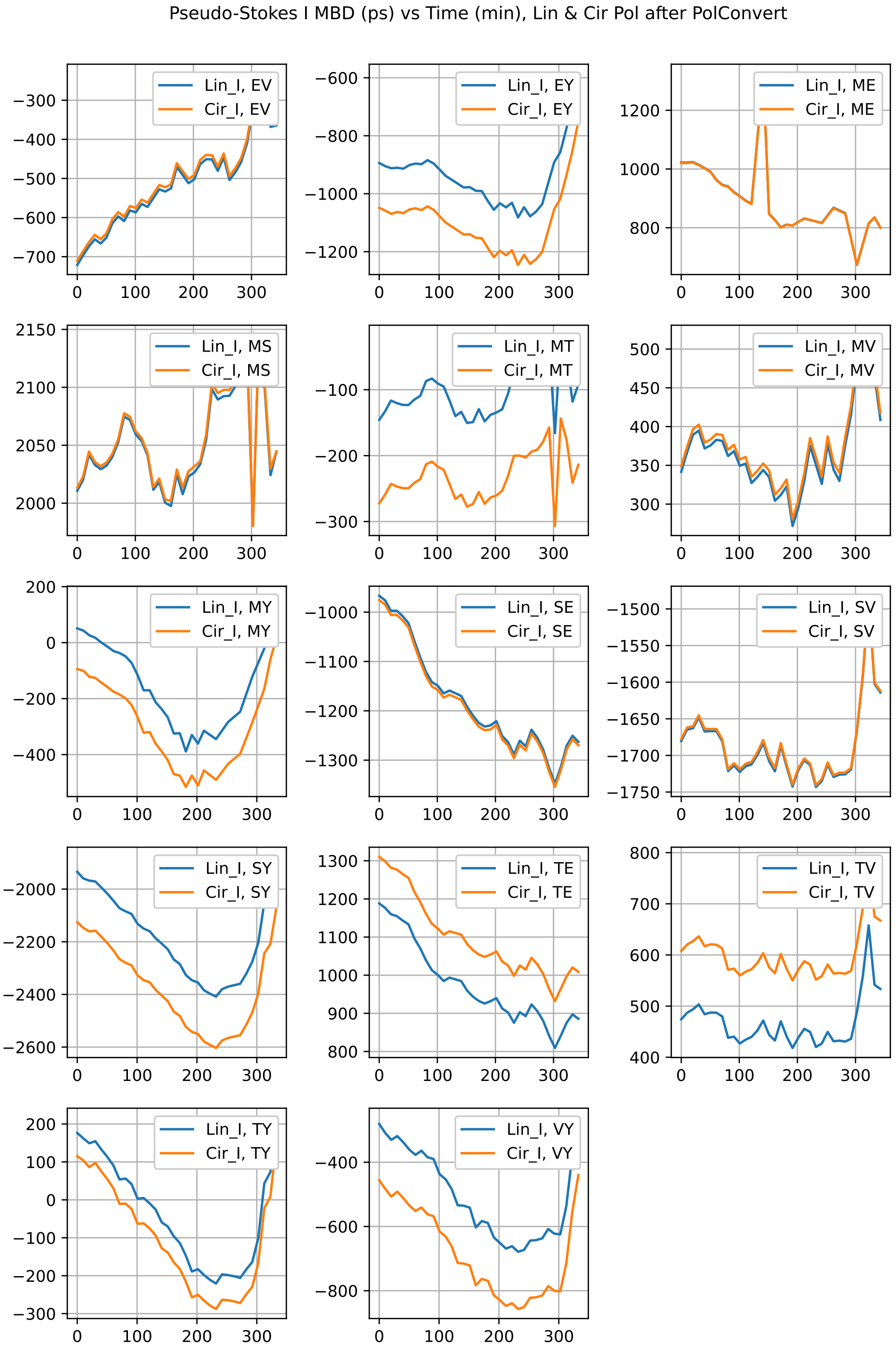

Figure 1:

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

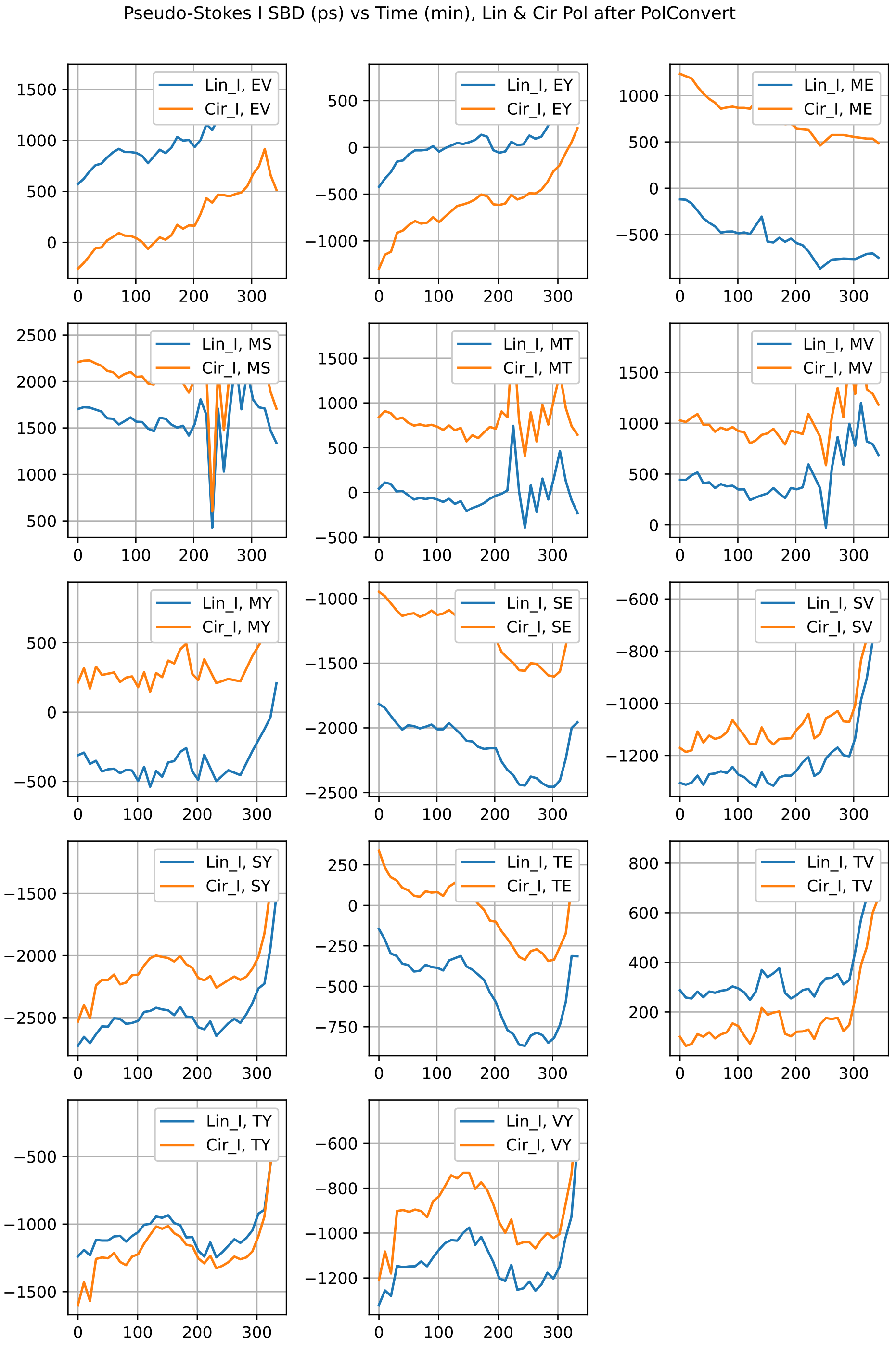

Figure 2:

2 Parameter Difference Statistics Before and After PolConvert

In order to estimate the errors introduced by PolConvert the histograms of the parameter differences (after the mean subtraction) are plotted. The following are the histograms for all of the

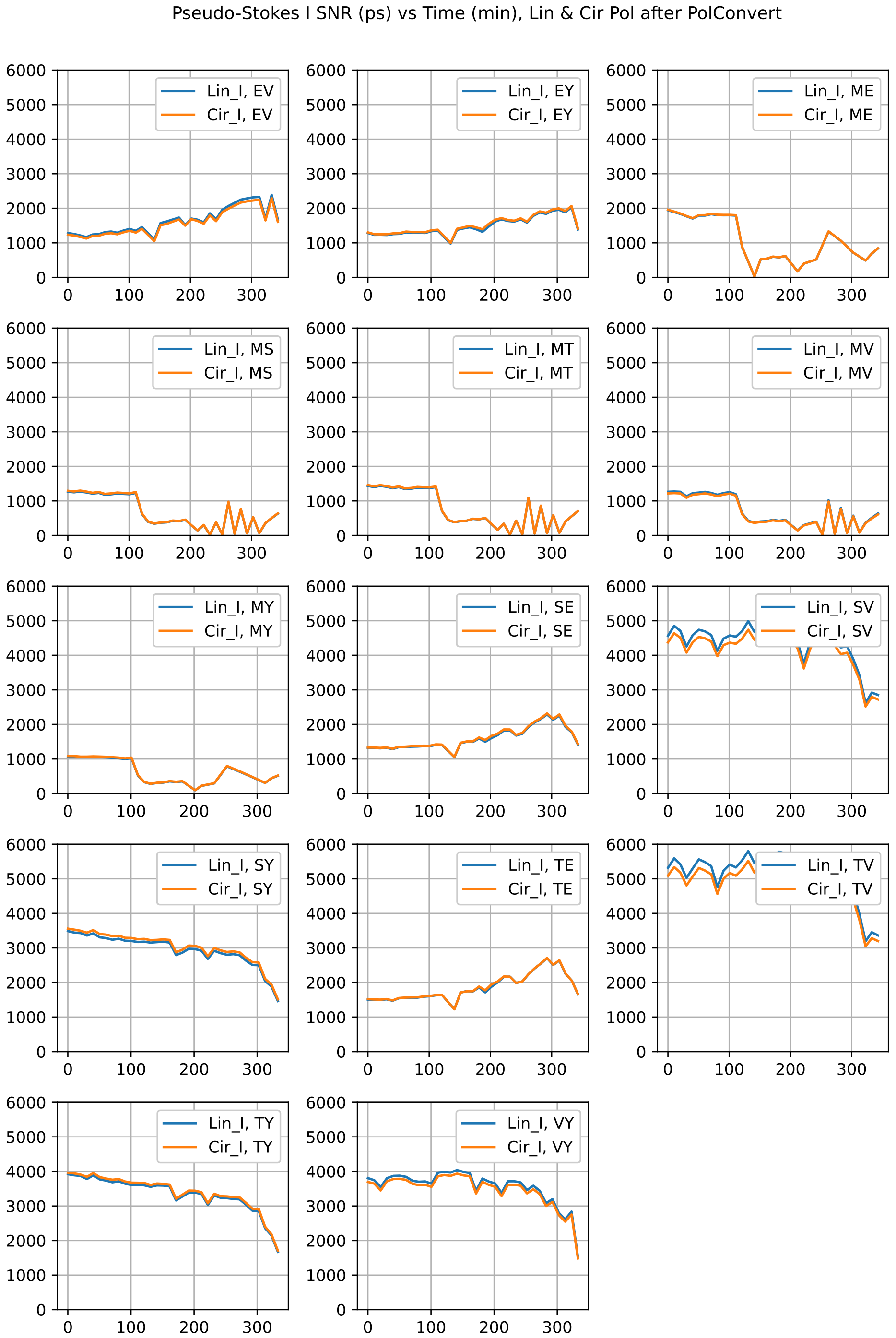

Figure 3:

baselines:

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

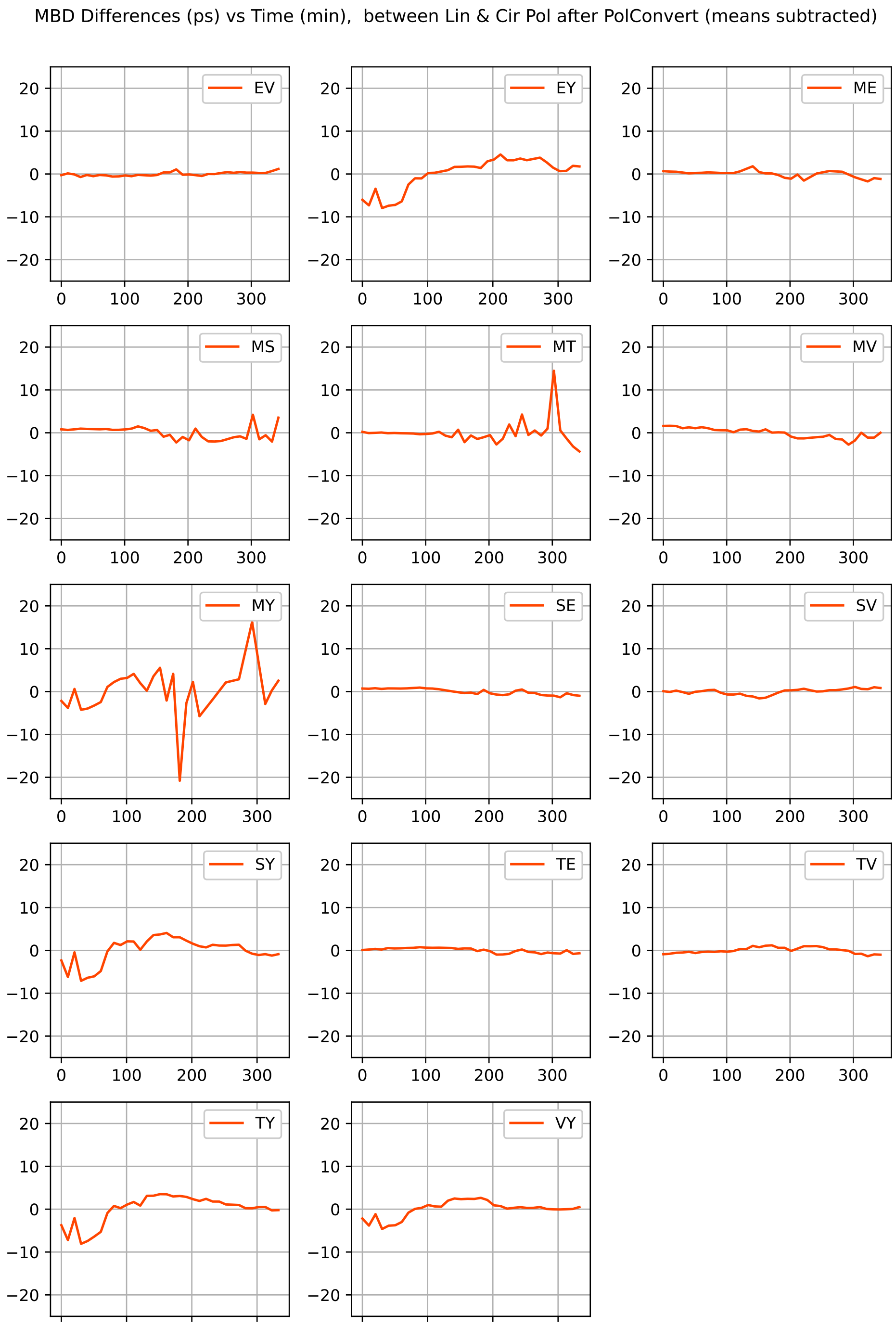

Figure 4:

Note that the means of the differences 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

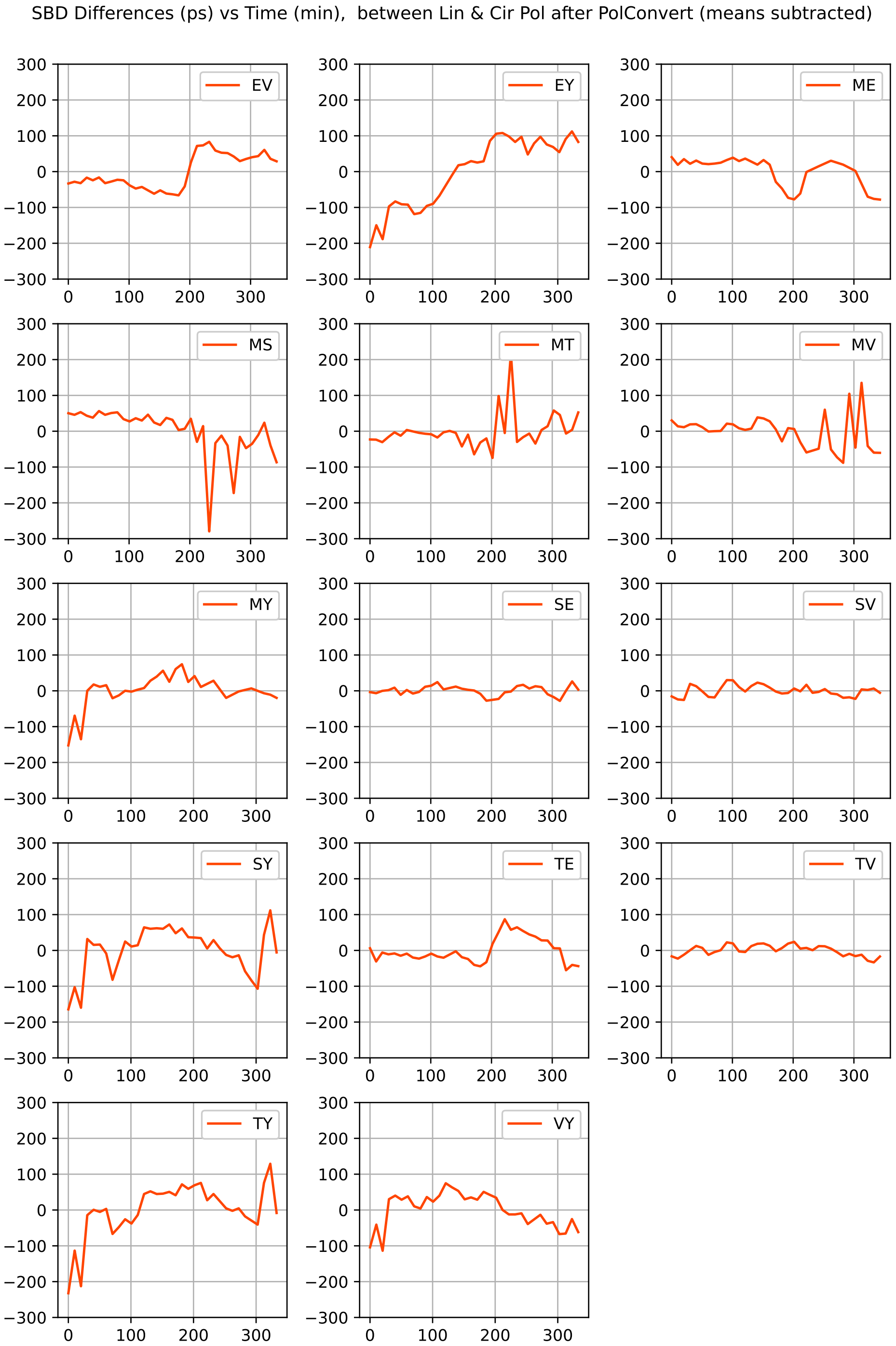

Figure 5:

(printed in each plot).

However, this method did not work: the differences 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).

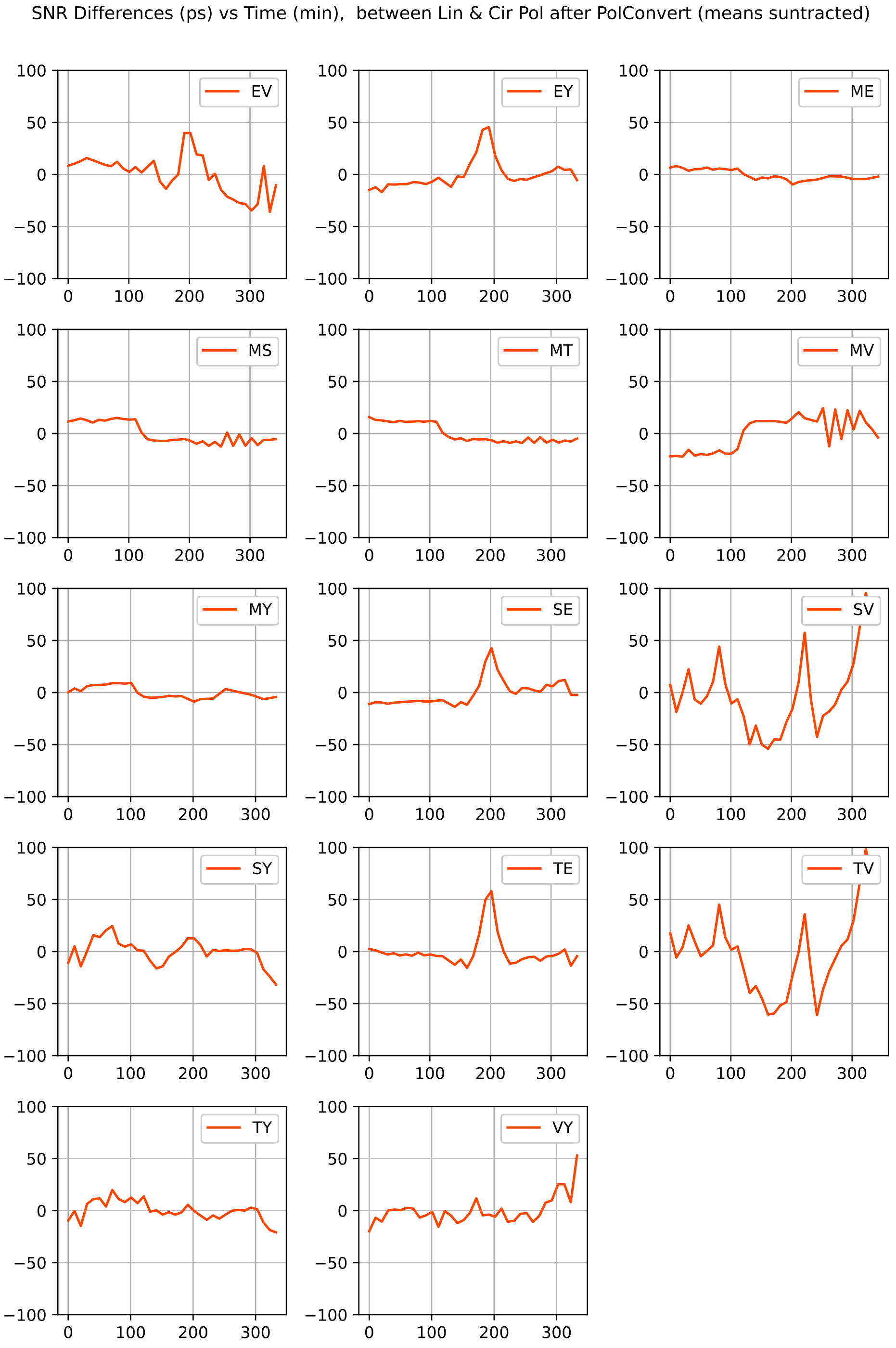

Figure 6:

We have used another method. Since the differences are mostly within ±σ (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

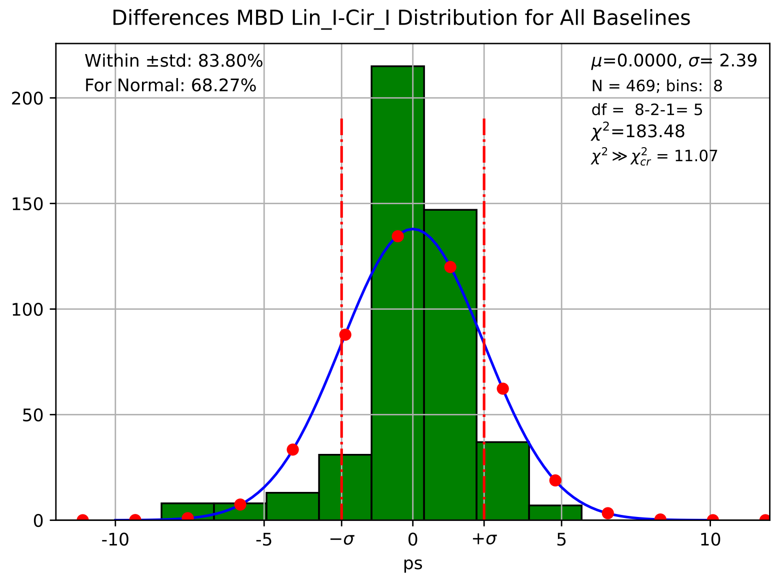

Figure 7:

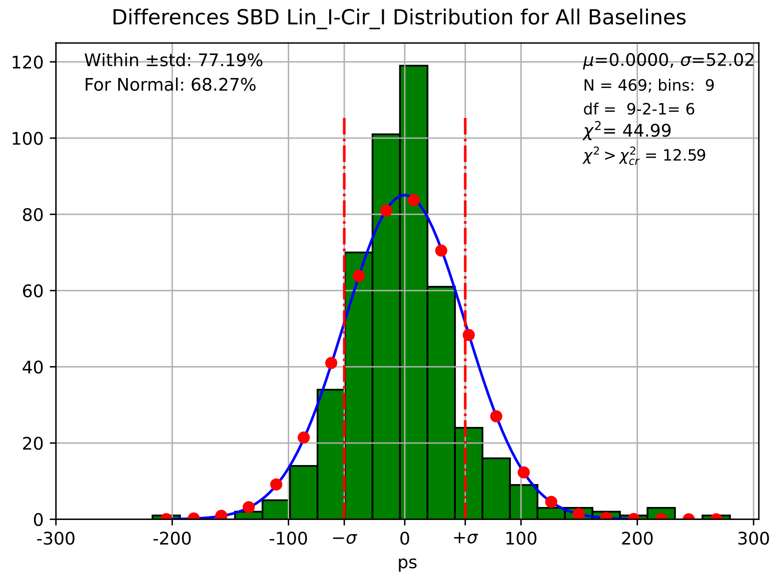

Figure 8:

histograms of the parameter differences for the baselines involving each particular station. They are shown in the following Figures:

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

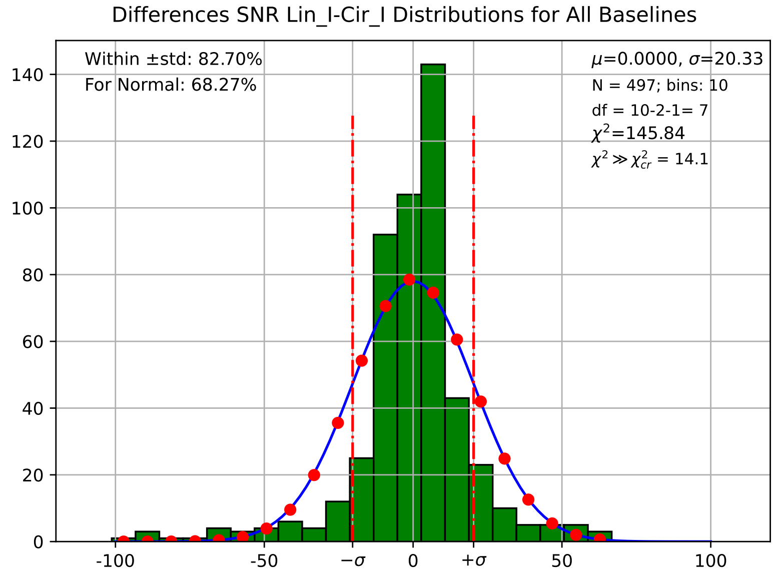

Figure 9:

3 Convenience 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'`.

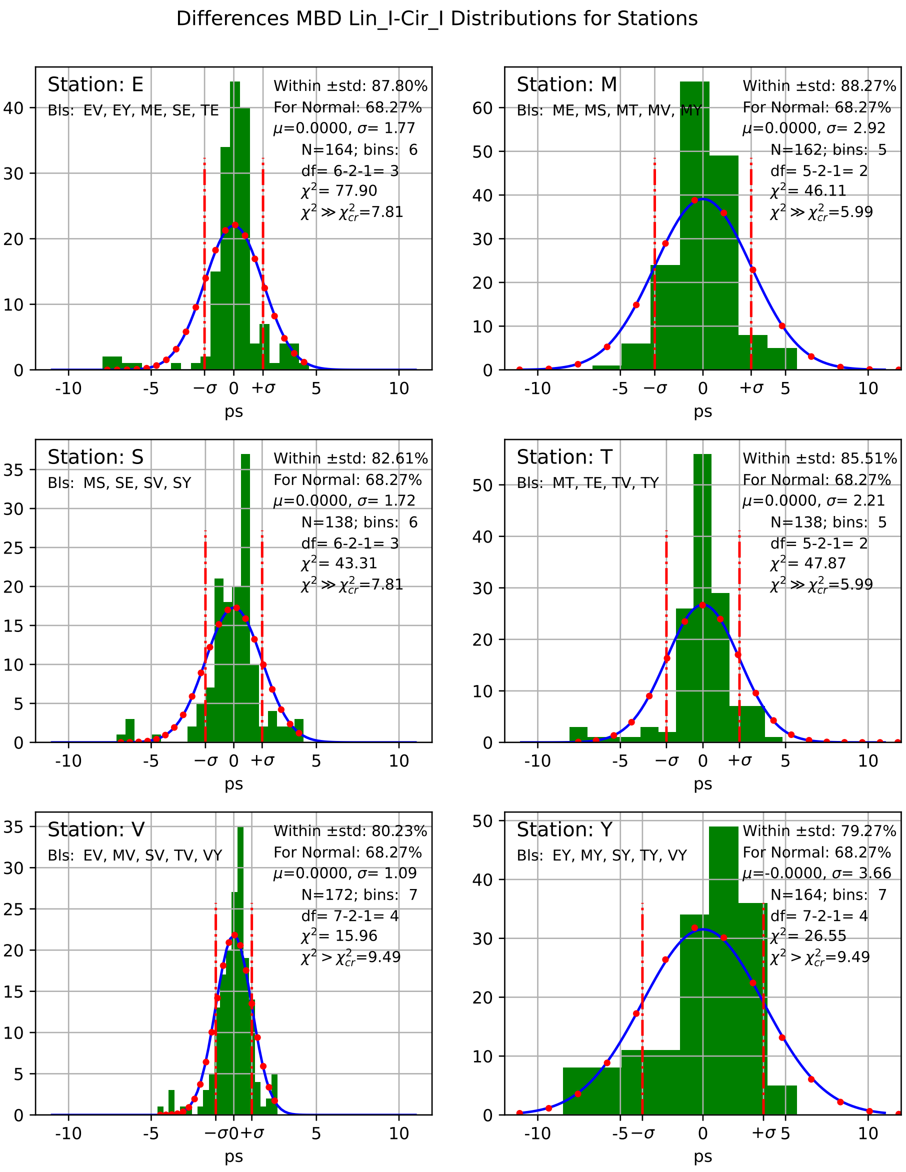

Figure 10:

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:
```

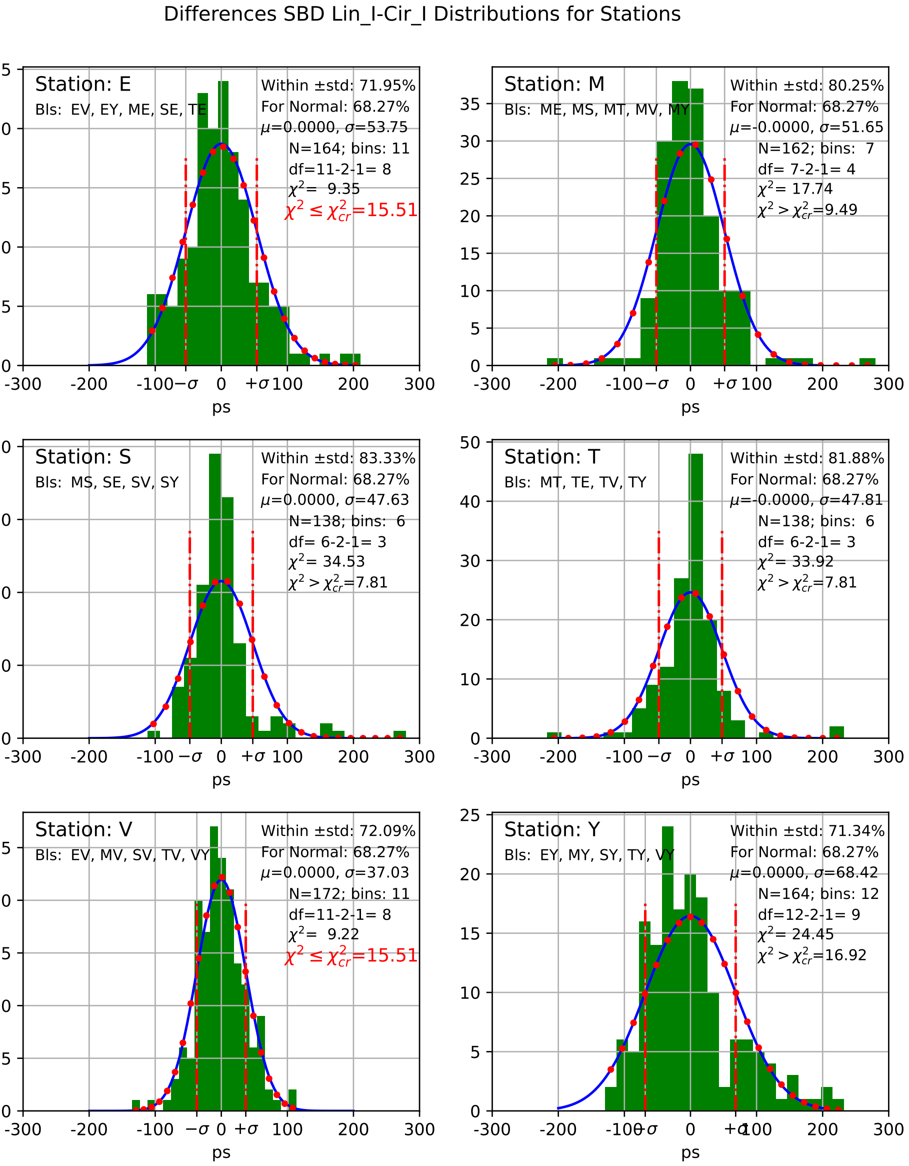

Figure 11:

```
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

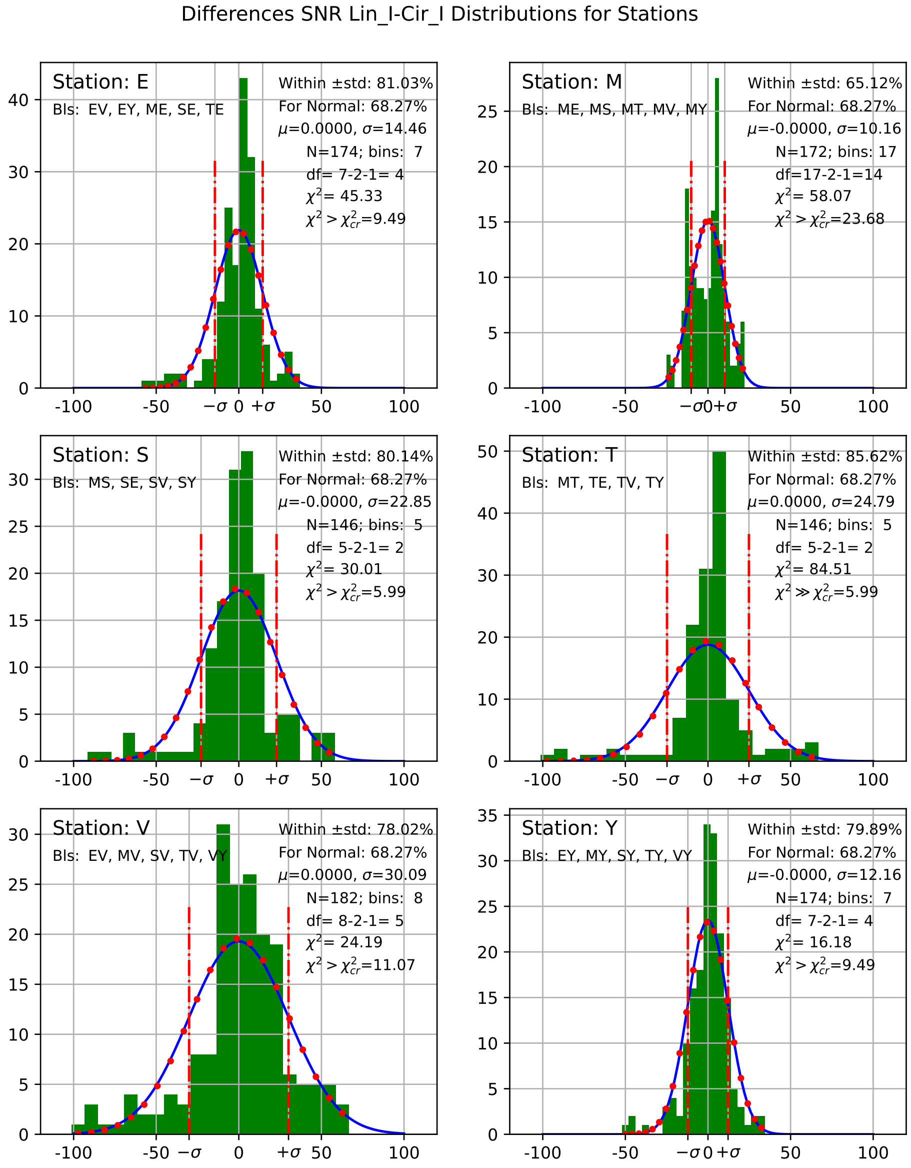

Figure 12:

'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