LBWG memo 16

Loop 3

Neal Jackson, 2018.11.17, update 2018.12.31

Purpose of Loop 3

Loop 3 is the inner loop of the pipeline, implemented in the script `loop3B.py` which, together with other required files, is on my github `github.com/nealjackson/lofar-lb`. It is partly based on initial scripts by Joe Callingham and Frits Sweijen. Its purpose is to perform hybrid mapping (imaging/selfcalibration) of an input measurement set, with an optional starting model. It produces as output:

- either one or two `h5parm` files containg phase and amplitude corrections, in files of the form `(visibility)_X_c0.h5`
- a final output image, in the intermediate data products directory
- a large plot of the phase/amplitude corrections and intermediate images at each stage of the loop, `(visibility)_output.png`
- a log file, `(visibility)_proc.log`
- a large number of intermediate data products in a directory `(visibility)_processing`
- if amplitude calibration is being done, an intermediate measurement set `(visibility)_A` with all the phase corrections applied

The main routine within the loop is `hybridloops`, which calls all the others. It has one compulsory argument, which is the file containg the measurement set, *which must previously have had the delays removed*. There are three optional/defaulted arguments:

- The strategy. This is a description of the calibration steps to be carried out. Each step consists of a letter (P for phase calibration and A for amplitude calibration) followed by the solution interval in seconds. Note that these are starting points for the algorithm rather than instructions to be rigidly followed: in the phase calibration, if the solutions are not coherent, longer solutions will be attempted (see below), and steps will be skipped if the algorithm has converged (i.e. if the image stops improving). The default strategy is three 32-second phase calibrations, followed by three amplitude+phase steps with solution intervals decreasing from 30 to 5 minutes. `The visibility sampling time is currently hardwired as 8 seconds, since the input to NDPPP is the number of samples in a solution interval - should be modified to work out the sampling time first.`
- A starting model. This is a string which can be specified as blank (make an image first) or as the name of a model image on disk. `Only the first currently works.`
- A threshold for the masking (currently defaulted to 5). This is important because it controls the parts of the image which are included in the calibration steps: a higher threshold will be more conservative in this inclusion.

It should be runnable multiple times - i.e. it overwrites previous iterations on the same measurement set without crashing. (It will, however, append to the logfile rather than overwriting it).

Requirements

- the two program files `loop3B.py` and `loop3B_service.py`
- the plotting program `aplpy_makeplot.py`, which takes a FITS file and produces a contour/grey-scale plot with automatically chosen zoom and contour level
- the Python packages `numpy`, `astropy`, `aplpy`, `scipy`, `h5parm`, `pyrap.tables`, `h5py`, `bdsf` as well as standard Python packages
- SExtractor (used for identifying islands of flux to calibrate the zoom of the plotting)
- `NDPPP`
- `wsclean` (in principle the `wsclean` dependence is factored into a single service routine which can be removed and replaced if needed)

Loop 3 uses low-level `h5` manipulation libraries, but does not use LoSoTo (see detailed description).

Algorithm

Loop 3 runs a number of hybrid mapping loops (imaging/selfcal). Phase selfcal is normally done first, with each loop having the following elements:

- If this is the *first loop*, either run `wsclean` to obtain an initial model image, or accept a starting model in the form of a model file or image. `Only the first of these currently works.` On subsequent loops, use `wsclean` to generate an image with the current corrected data, using the current coherence length for the maximum u-v distance to be imaged. `wsclean` is run using a set of standard defaults: 100-mas cellsize, and robust-zero weighting.
- Use `pybdsf` to make a mask of bright regions in the current image, with the `ith` parameter controlling the mask. Higher values allow only smaller regions with more certain signal to be allowed into the mask region.
- Calculate a goodness statistic in the image as minimum/maximum flux value. Exit at this point if this has not significantly improved since the last iteration.
- Predict the current image into the MODEL column of the MS.
- Selfcalibrate using this model column to form the corrected data. The selfcalibration is performed using successively greater solution intervals, beginning with the value specified by the current loop of the `strategy` argument and increasing by factors of 3, until all stations have coherent solutions or until an interval of 1 hour is reached,

whichever happens first. A final solution table is formed using the shortest interval for each station with a coherent solution, interpolated as necessary to a 30-second grid. Judgement of what constitutes a "coherent" solution is made using a metric involving the square of the gradient of the difference of X and Y phase solutions (see previous memos), with incoherence being declared if more than 10% of the solutions are `NaN`. An array of coherence values is returned, one per telescope. The baseline length over which solutions are coherent is also returned; this is calculated by finding the telescope with the furthest distance from Exloo over which the solutions are coherent. This coherence length is then used to set the uv distance over which subsequent imaging will be performed, thus ensuring that imaging is not done at a resolution for which no significant correlated signal exists.

The phase-calibration loops are all run on the same data, so that the phase calibration, in principle, improves as the image gets better. In other words, each phase calibration is not incremental, and each phase calibration represents a mapping from raw to "ideal" data. Once all phase calibrations specified by `strategy` have been done, or the image has stopped getting better, the final phase calibration is applied to the MS, giving a second MS with the suffix `_A` appended. This is done because amplitude solution times are generally much longer than phase solution times (otherwise long VLBI experience is that the data will be forced very easily in the direction of the model). The amplitude solutions then proceed on this corrected MS in the same way, but in a separate series of loops.

After any amplitude calibration loops, the `_A` measurement set is left on the disk, along with the final phase-calibration `h5` file and the final amplitude calibration `h5` file. These two files should both be applied in subsequent analysis (e.g. to correct the visibility measurement set to map nearby objects). An output png file is produced which plots all of the phase and amplitude calibration solutions, together with the image obtained in each loop. Plots of the images are made using the `aplpy_makeplot` script, which calls SEXTRACTOR to locate regions with significant signal and zoom the plot as needed, and automatically sets the contour levels based on noise in the image.

There are two workarounds in the pipeline. The first is that `NaN` values are excluded from the calibration solutions before calculation of the coherence metric, because `np.unwrap` behaves oddly with `NaN` values: if the first item of an array is `NaN`, all subsequent values in the unwrapped array are `NaN`. The second issue is one of h5 file locking: after NDPPP `gaincal` is run, the resulting file is locked and cannot be re-opened. This process is therefore done in a separate python instance; the calculation of the coherence metric is also done in this way. For this reason also, loop3 does not use LoSoTo.

Testing of loop 3 on the 2015 test field data.

I have tested loop 3 using the source 1344+5503, which is a bright source close to the centre of the field (less than 1 degree away), and the fainter source 1350+5447. The best map I can make by hand of 1344+5503 (in difmap) is shown in Fig. 1. This is a source a few arcseconds across, with a jet structure and a more extended lobe. It is pretty obvious by hand that all these components are real. 1350+5447 is harder because there is less long-baseline flux, but as far as I can see this is a slightly extended point source with overall size less than about 1 arcsecond. Note: AIPS has been used for the pre-processing - see memo 20 for further details.

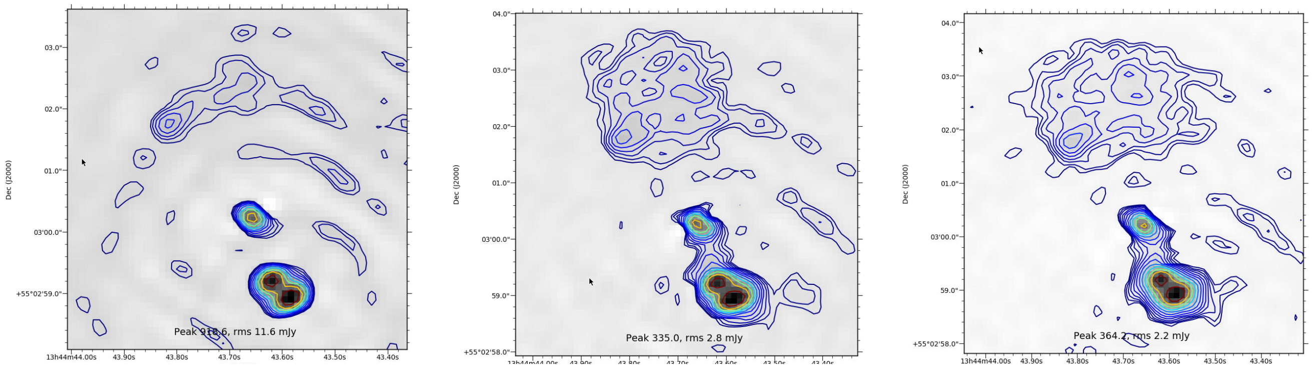

Figure 1: Handmade maps of 1344+5503 with difmap. Left: phases only. Middle: with a simple amplitude scaling. Right: with amplitude calibration.

Loop 3 tests on 1344+5503

The loop 3 output for 1344+5503 is shown in Fig. 2. It is based on a measurement set in which delay calibration and initial phase calibration has been done on the LBCS calibrator 1327+5504, which is about 2.5 degrees away at the edge of the field. The top three rows of the figure are the phase calibration loops, and the bottom three are amplitude calibration with successively shorter solution intervals. Phase calibration recovers the lobe structure, but also gives a lot of artefacts and a spurious component appears to the southeast. Amplitude calibration removes these structures and reveals all of the structure in the hand map, but there is additional structure in a halo around the source which I think is not real. Opinion: this will probably be removed with masking at higher threshold, but this needs to be tested.

Loop 3 tests on 1350+5447

Figs. 3, 4 and 5 show tests of loop 3 run on the fainter source 1350+5447, which hand-mapping (using initial calibration from 1344+5503) shows to be something close to a point source. There are three runs on this source: the first with only delay calibration; the second with delay calibration, and initial phase calibration performed on the source 1327+5504, 2.5 degrees away; and the third, with initial calibration performed on the much closer source 1344+5503. It appears that convergence can be achieved on relatively weak sources, provided that an initial phase solution is applied from a nearby calibrator first. Opinion: (1) I think that the convergence will be considerably faster if an initial amplitude calibration is also applied; (2) in this case we may very well be able to get away without initial closure modules.

Comments on further testing

Useful tests to do:

- needs to be done on weaker sources with structure; maybe 1350 will approach hand-map with more iterations of amplitude selfcal
- needs to be tested with different mask thresholds
- needs to be tested with weak sources with initial phase *and* amplitude calibration applied from nearby sources
- needs to be done on empty bits of sky to check that sources are not selfcalibrated into existence.
- needs to be tried on some LBA data

Notes on practical details. I have a lot of files of LBCS sources on my AIPS area (340) on lof020 - please use MOVE to copy them if you want them! By default they have *not* been fringe-fitted, although there are some SN tables attached to 1327+5504 with fringe fits and phase calibration. See memo 20 for details of how to process them.

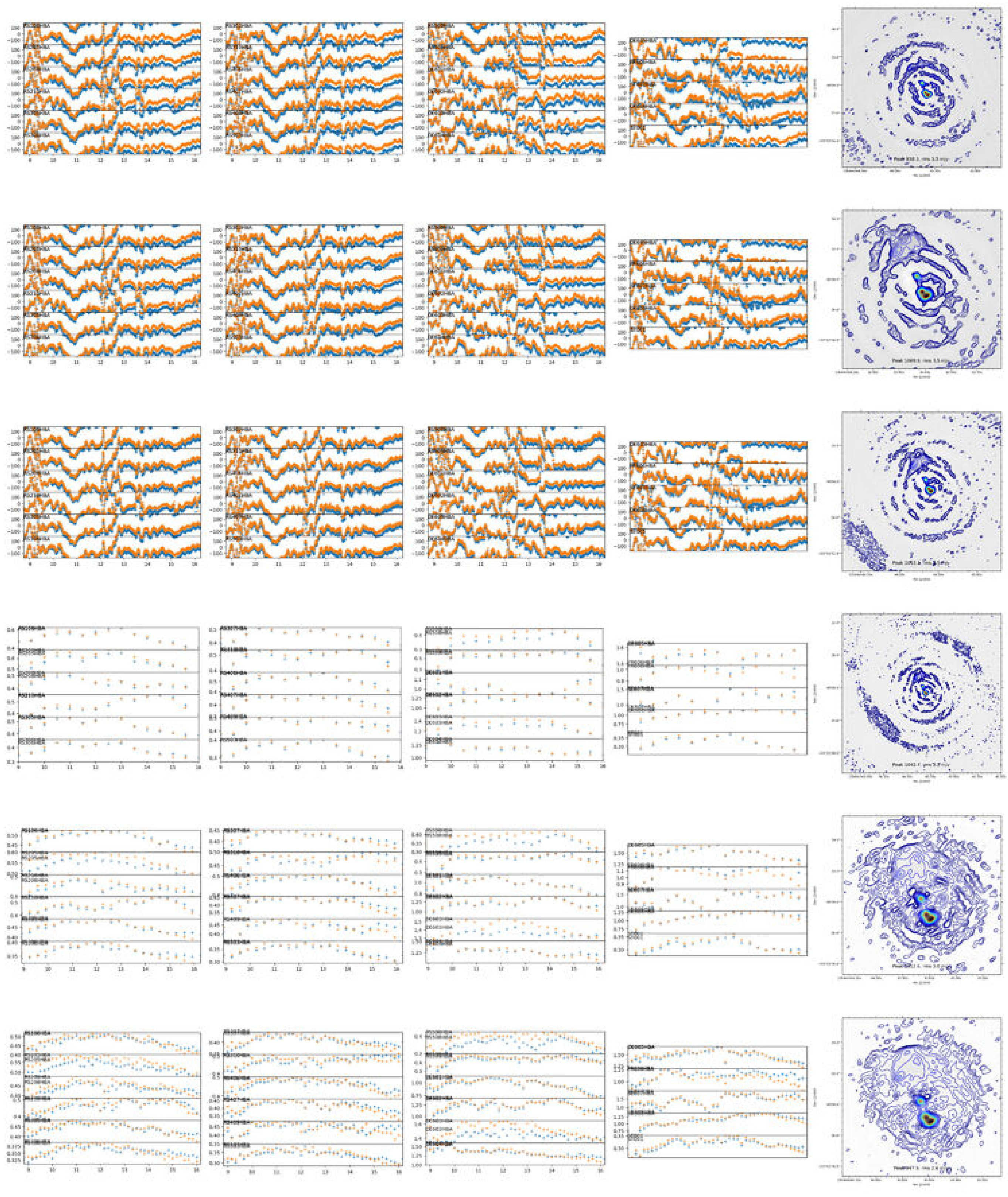

Figure 2: Loop3 test on 1344+5503. Rows represent successive iterations. In each row the phase solution and resulting map is shown.

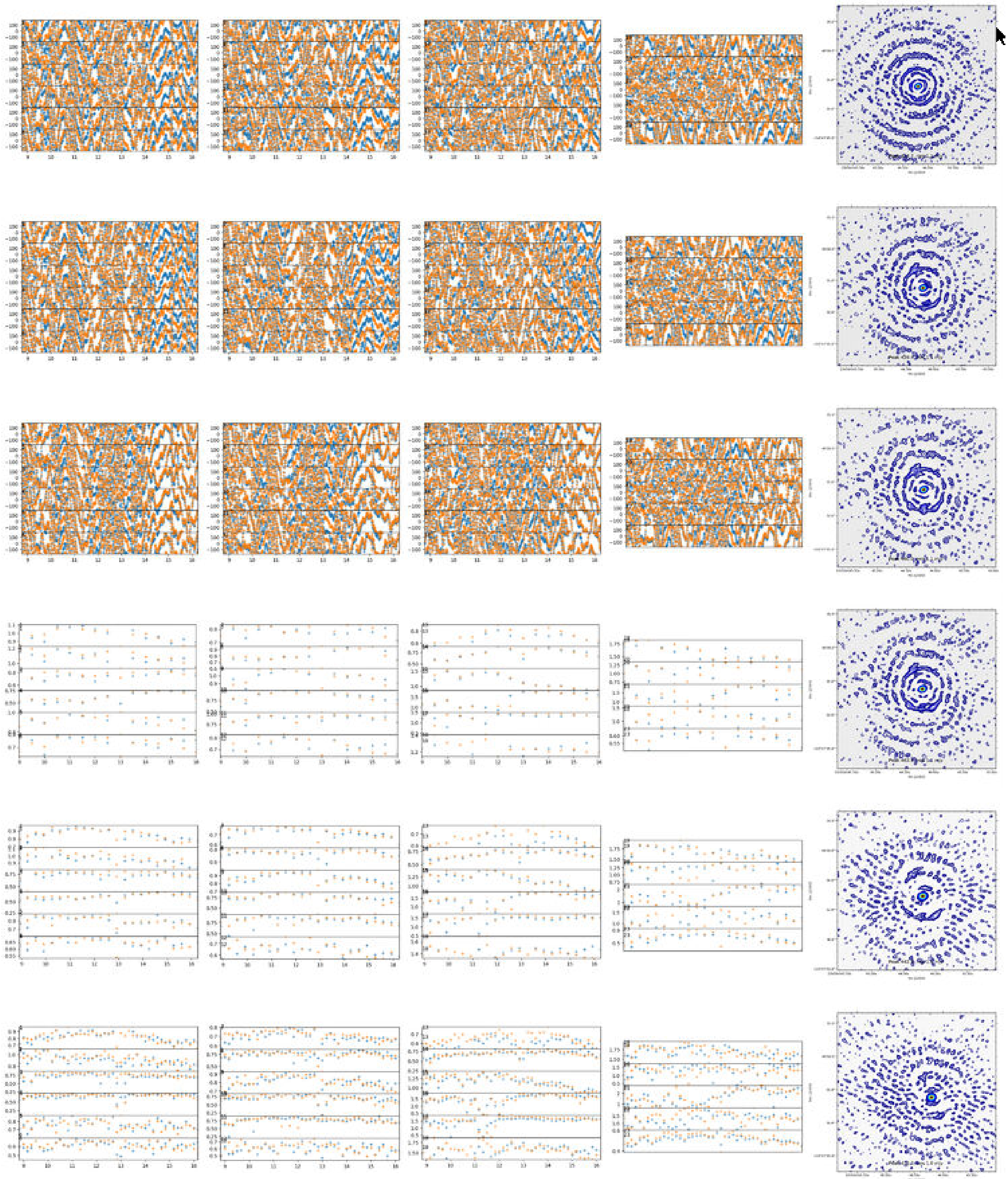

Figure 3: Loop3 test on 1350+5447 using a measurement set with delay-only calibration. Rows represent successive iterations. In each row the phase solution and resulting map is shown.

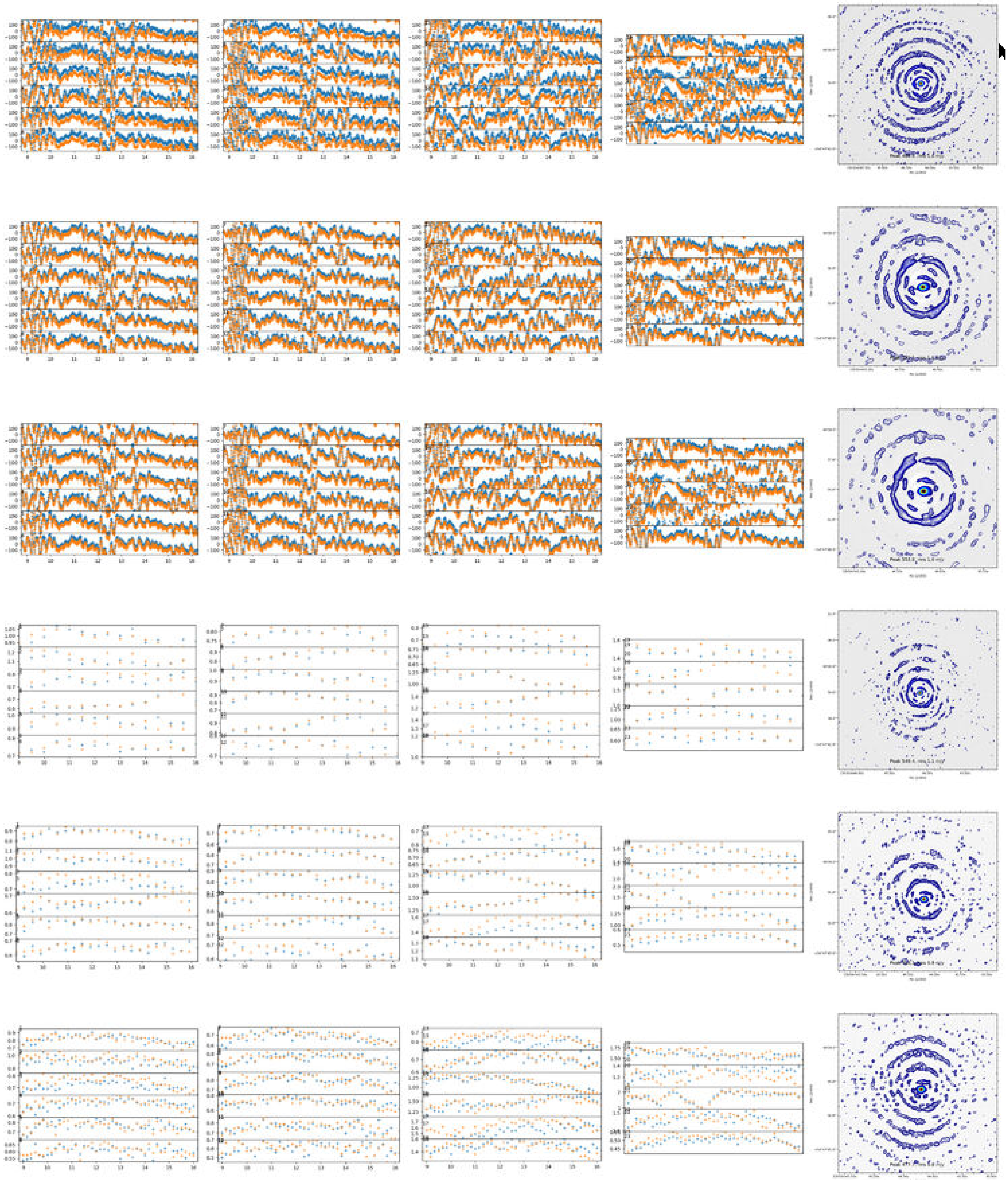

Figure 4: Loop3 test on 1350+5447 using a measurement set with delay calibration, and initial phase calibration based on 1327+5504, 2.5 degrees away. Rows represent successive iterations. In each row the phase solution and resulting map is shown.

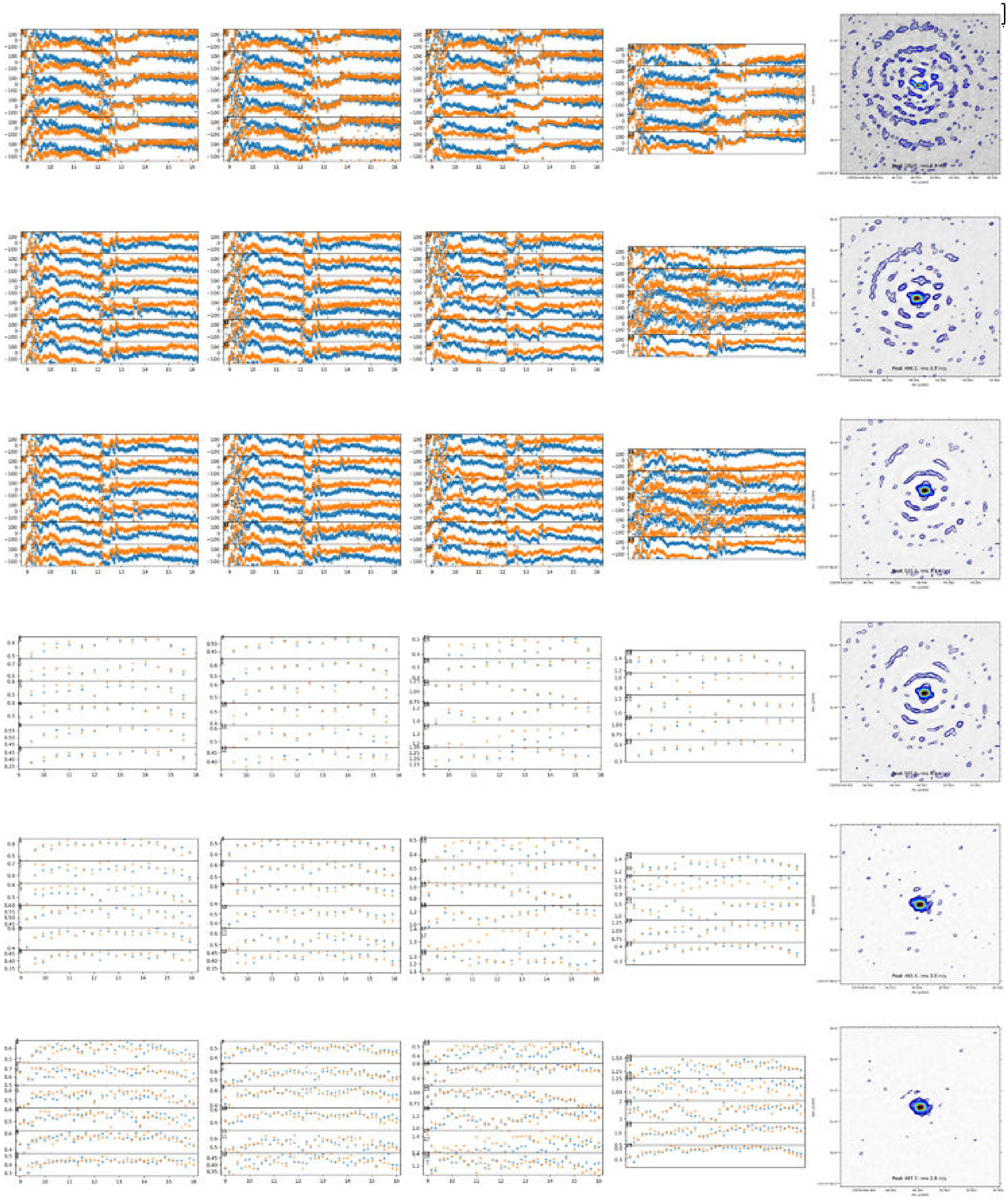

Figure 5: Loop3 test on 1350+5447 using a measurement set with delay calibration, and initial phase calibration based on 1344+5503, about 1 degree away. Rows represent successive iterations. In each row the phase solution and resulting map is shown.