V VLA Maintenance Tasks in *AIPS*

Most tasks in *AIPS* are designed to work for a generalized interferometer or single-dish. However, a number of tasks have been written specifically at the behest of scientists supporting the calibration and maintenance of the VLA and EVLA. These include a task to read the "telcal" text file to prepare an initial solution (SN) table, a task to solve for antenna position corrections from appropriate solutions in an SN table, a task to fit corrections to the antenna pointing model from a file produced by the on-line system in pointing mode, and several tasks to process holography data up to and including a map of the corrections to be applied to the antenna panels. This appendix is an attempt to document such tasks. Note that TLCAL is a task which may be of interest to general VLA users. The antenna location task LOCIT and the holography tasks may well be of interest to scientists maintaining interferometers other than the VLA. However, the pointing task PEEK will probably be of use only to people tasked with maintaining the VLA.

V.1 Real-time calibration tables: TLCAL

The VLA on-line system determines antenna-based complex gain and delay values for the average of every calibrator scan. These are written into a text file which is stored at the VLA site on a computer named mchammer. The actual file name is

/home/mchammer/evladata/telcal/*year*/*month*/*asdmfile*.GN

where *year* is the four-digit year, *month* is the two-digit month, and asdmfile is the long complicated name assigned to the particular scheduling block. It is possible to read such files directly from computers in Socorro and task TLCAL supports such access. The text file may also be copied by, for example, the analysts and made available to users in Socorro and elsewhere. The gains in the telcal file all assume that every source is 1.0 Jy in all spectral windows. If you have run SETJY on your data set before using TLCAL, then you can adjust the gains of the known flux calibrators using the fluxes stored in the source (SU) table. In this case, the SN table produced will be ready for use with GETJY.

To run TLCAL on a file in your data area, enter

> TASK 'TLCAL' C_R	
> INDI *n* ; GETN *m* C_R	to specify the data set.
> DOAPPLY 1 C_R	to use the fluxes in the source table.
> ASDMFILE(1) = 'MYAREA: C_R	two parts are concatenated.
> ASDMFILE(2) = 'TSUB0001.sb23643640.eb23666175.56454.14034020834 C_R	
> INP C_R	to check that you remembered to leave off the close quotes above.
> GO C_R	to run the task.

Substitute your file name parameters for those used above and leave off the .GN part of the name and the close quotes (to preserve your desired case). It would be wise to check the SN table produced with, *e.g.*, SNPLT . Bad RFI may render some telcal values bogus.

The Telcal file is text file with very long lines beginning with 3 header lines. Then there are columns for MJD (to high accuracy), UTC (sexagesimal), LST (days), LST (sexagesimal), IFID (5-character string), Sky frequency (MHz), Antenna ID (*e.g.*, ea03), gain amplitude, phase(degrees), residual, delay (ns), flagged (true/false), zeroed (true/false). hour angle (radians), azimuth (radians), elevation (radians), source name, and flag reason. TLCAL uses the MJD, antenna, amplitude, phase, delay, flagged, and source columns. A combination of the IFID (peculiar to the EVLA) and the sky frequency is used to identify the IFs to which these apply.

V.2 Correcting antenna positions: LOCIT

The data taken for pointing analysis after antennas have been moved is now also used to determine corrections to the assumed antenna locations. The data are loaded into *AIPS* using `BDF2AIPS` in the usual way, but `DOKEEP = 1` is required in order to load pointing data. Then `FRING` is run on one scan from a strong source with `SOLINT = 5` minutes and with no rates (`DPARM(9) = 1`). Then `CLCAL` is used to apply the `SN` table to all times in the `CL` table. Then the data set is reduced in size by running `SPLAT`, averaging in frequency and time:

> DEFAULT SPLAT C_R	to initialize the adverbs
> INDISK *n* ; GETN *m* C_R	to specify the data set.
> OUTDISK INDISK C_R	to keep on same disk.
> APARM 3, 1, 0, 0, 1 C_R	to average data in frequency, but not over IF. The input data are taken at 1-second interval.
> DOCAL 1 ; GAINU 0 C_R	to apply the calibration.
> CHANNEL 64 ; CHINC 0 C_R	to average all spectral channels together.
> SOLINT 5/60 C_R	to average over time to 5-second intervals.
> INP C_R	to check the inputs.
> GO C_R	to make a rather smaller data set.
> INDISK *n* ; GETN *j* C_R	to specify the averaged data set.
> INEXT 'NX' ; INVERS 0 ; EXTDEST C_R	to delete the index table which has each direction of the pointing as a separate scan.
> DEFAULT 'INDXR'	to initialize everything.
> INDISK *n* ; GETN *j* C_R	to specify the averaged data set.
> CPARM 0 , 10 C_R	to set maximum scan length to 10 minutes.
> GO C_R	to make the new index table.

It might be wise to make a backup copy of the data set at this point in case later machinations go wrong. Use `UVCOP`.

A solution table is now needed containing the phases of every scan in the data set. Use

> DEFAULT CALIB C_R	to initialize the task and adverbs.
> INDISK *n* ; GETN *j* C_R	to specify the averaged data set.
> SOLINT 10 C_R	to average full scans.
> APARM 3, 0 C_R	to require at least 3 antennas, but do *not* average polarizations or IFs.
> SOLTYPE 'L1R' ; SOLMODE 'P' C_R	to solve for phases only with recursive L1 methods.
> DOCALIB 1 ; GAINUS 0 C_R	to apply calibration, a no-op at this stage.
> SNVER 0 ; REFANT *nn* C_R	to create a new solution table and to select a reference antenna (*nn*)
> INP C_R	to double check.
> GO C_R	to make the solution table needed next.

Finally, fit the antenna positions. It is best to do each IF separately and also try the mode that uses the phase difference between IFs.

> DEFAULT LOCIT C_R	to initialize things including `DOTV` false, and `REFANT 0`.
> INDISK *n* ; GETN *j* C_R	to specify the averaged data set.
> BIF 1 ; EIF BIF C_R	to examine one IF at a time.
> STOKES 'RR' C_R	to examine one polarization at a time.

> DPARM 0, 1, 10, 120 C_R	to use each scan by itself (rather than differencing consecutive scans), do not fit the K term, and limit the fit to elevations between 10 and 120 degrees. All hour angles and declinations are used. The total phase is used.
> BPARM 3, 0, -180, 180, 3, 4, 1, 3 C_R	to plot 3 antennas per page between -180 and 180 degrees as boxes with connecting lines, and, if doing more than 1 IF or polarization, plot them in the same plot.
> DOOUTPUT -1 C_R	to ignore any values contained in pre-existing files named OUTPRINT.*arrayname*. If this file exists before running LOCIT and DOOUTPUT $>$ 0, it will be read and the corrections contained within it added to those found before they are written to a new OUTPRINT.*arrayname* file and also to the other output files. See HELP LOCIT C_R for details.
> OUTPRINT 'FITS:*date*' C_R	to set the output file base *name* used for output text files.
> INP C_R	to check that all is well.
> GO C_R	to run the task.

Then do

> STOKES 'LL' ; GO C_R	to redo with the other Stokes.
> BIF 2; EIF BIF C_R	to redo with the second IF.
> STOKES 'RR' ; GO C_R	to redo with the other Stokes.

Now compare the answers and uncertainties found in each of the four runs. Also it may be good to do

> BIF 1; EIF 2 C_R	to include both IFs.
> STOKES 'RR' C_R	to do one Stokes at a time.
> BPARM(9) 1 C_R	to select the mode which uses the phase difference between the 2 IFs.
> GO C_R	to get a fifth estimate of the antenna position corrections.

Repeat for STOKES 'LL'.

Now comes the hard part. LOCIT will have created a great many plot files, an example of which is shown in Figure V.1. It also writes text files named OUTPRINT with extensions:

- FIT This file contains all solutions with their uncertainties in a form that is easy to read. It may be used to accumulate corrections depending on the setting of DOOUTPUT.
- EVLA This file contains only significant corrections in a form that can be hand edited to put in the dates when the antennas move and when the database is fixed. The edited result may be edited to a system file on the web to enable VLANT to apply the appropriate corrections.
- PAR This file, containing only significant corrections, is read into Parminator to update the database.
- 001 This file contains only significant corrections in a script form to read into CLCOR. Do this, to make a new CL table and then rerun CALIB to see if the resulting phases really end up near zero.

You will need to review the plots (TVPL is helpful) and study the .FIT file to decide which antennas have significant corrections that need to be made. You will almost certainly need to edit the .001 file and then apply it to your averaged data set. Repeat CALIB and then examine the resulting SN table with EDITA or SNPLT. If things worked well, then edit the .PAR and .EVLA files to match.

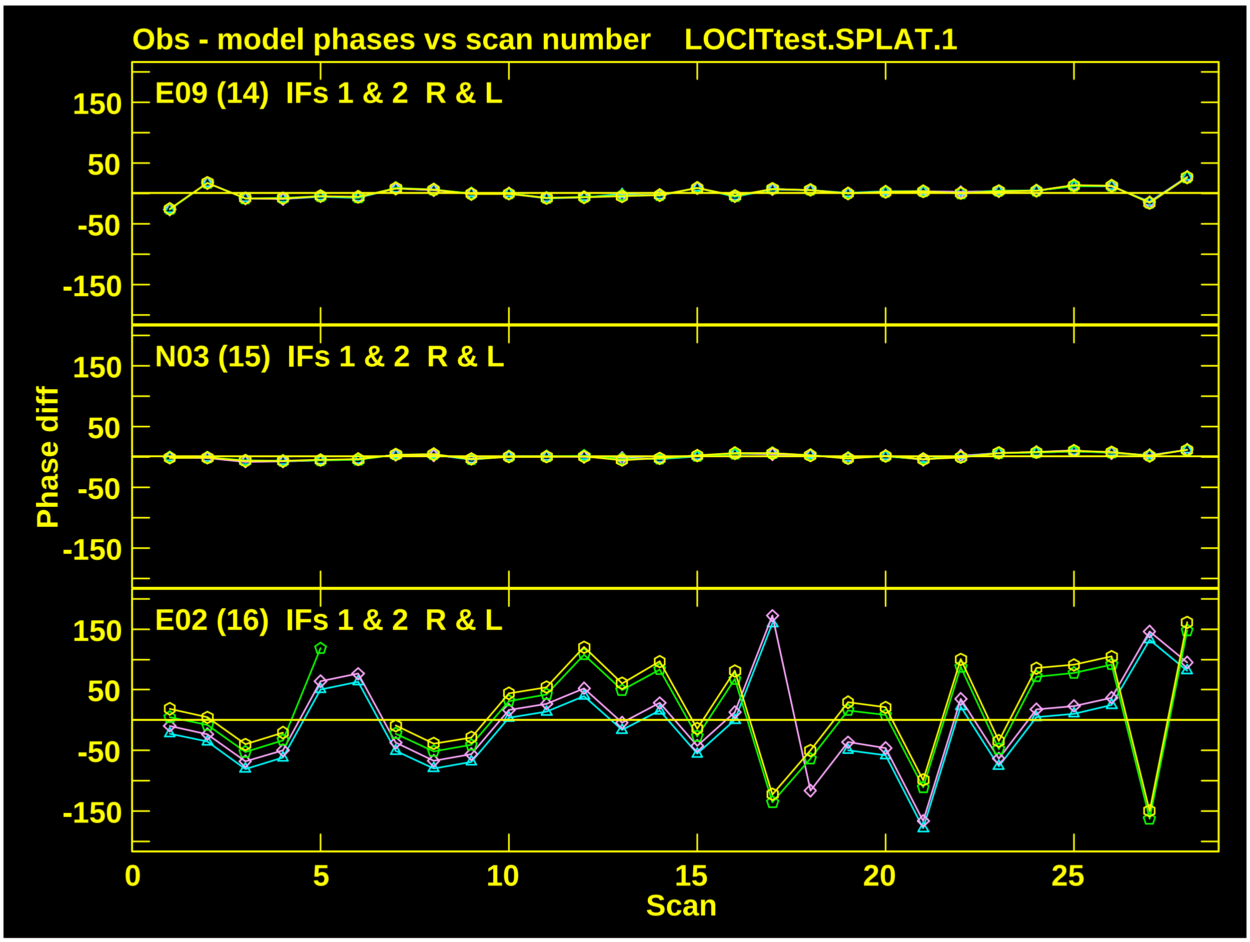

Figure V.1: Example of the residual phases after a `LOCIT` solution is applied to the input `SN`-table phases. The solutions for antennas 14 and 15 are fine, but antenna 16 did not converge to a correct solution. The data from both IFs and both polarizations were used in this run of `LOCIT` and are represented by different colors in the plot.

V.3 Improving antenna pointing: PEEK

When the EVLA on-line system conducts a pointing observation, be it for a long series of such observations after antenna moves or a single observation to be applied to the next target source, the software updates a text file with extension `.PNT`. These files may be copied from the computers at the VLA and then analyzed to determine increments to the parameters of the pointing model. This analysis was done with a program called `peek` available from Ken Sowinski's disk area. That program has been adapted to run as an *AIPS* task. The method of setting the input parameters was changed to be the familiar *AIPS* methods, the program was given the option to make *AIPS*-style plots in addition to the `gnuplot` files, and has the option to suppress some of the output text files.

An example usage would be

> DEFAULT PEEK C_R	to initialize the task name and all adverbs.
> INDISK *n* ; GETN *m* C_R	to specify an arbitrary catalog entry to which plot files are attached. It may be omitted if `DOPLOT`≤ 0 or `DOTV = 1`.
> ASDMFILE(1) = 'MYAREA:peek/ C_R	two parts are concatenated.
> ASDMFILE(2) = 'PX8262 C_R	file renamed to useful name size
> DOPLOT 1 ; DOTV 1 C_R	to make *AIPS* plots on the TV display.
> INP C_R	to check the inputs.
> GO C_R	to run the task.

Do not include the `.PNT` extension in `ASDMFILE`; the task will add it. Adverb `APARM` is used to set data and output selection parameters. The defaults are to include all dates, to include elevations between 7 and 122 degrees, to include all samples with amplitude > 0.01, to omit samples with wind speed > 5 m/s, to include normalized beamwidths between 0.95 and 1.15, and to write the filtered change file only when the change is > 0.1 arc minutes and > 3 times the uncertainty.

Adverb `FPARM` is used to determine which parameters the task determines, with the default being the usual set (Tilt, Azzero, Elzero, Colima, and Refrac). `FPARM` can be, but is normally not, used to attach the tilt to the antenna rather than the pad and to suppress the writing of various output text files. These text files are written to the disk area which is the "root" directory of the `ASDMFILE` adverb — `MYAREA:peek/` in the example above. The output files are named with `ASDMFILE` and extensions `.PTR` (large overall summary), `.gp` (`gnuplot` plotting instructions), `.changes` (all of the adjustments found), and `.filtchg` (those adjustments that are deemed significant). In addition, files are written named `ant`*nn*`.dat` that include all relevant parameters for all antennas which have been fit and which are used by `gnuplot`.

You can use the `.gp` file by typing `gnuplot` followed by the name of the `.gp` file. It will make a file named `plot.ps`. Adverb `DOPLOT` may be used to request *AIPS*-style plots with substantially better control, using `DPARM`, over symbol types, sizes, and colors. In both styles of plotting, six plots appear on each plot page showing measured and residual azimuth and elevation corrections plotted against azimuth, elevation, and relative time. You may plot these on the TV or make plot files, one for each antenna fit. If using the TV, button `D` tells the task to stop plotting but continue doing all the fits and text file outputs. A new procedure to set colors for `LWPLA` has been written called `OKCOLORS`. It sets the colors to useful values for plotting on a white background. Black backgrounds use a lot of printer ink and so are less desirable when printing many plots.

V.4 Improving antenna surfaces: holography

In "holography" data from the EVLA, reference antennas are pointed towards the nominal source position, as normal. All other antennas are pointed to an offset position controlled by the observer. The angular offset from the nominal position, for each antenna, is written to data sets in the same variables that are normally used for u, v, w. Normally, when UVCOP is used to select IFs, the values of u, v, w are scaled to the new header frequency. To prevent this, set UVCOPPRM(8) = 1 and, if any holography sources have names other than HOLORASTER, also set SRCALIAS.

In holography mode, only data on a baseline between a reference antenna and an offset (non-reference) antenna are normally used. (Data between two moving antennas may be plotted by UVHOL however.) The correlations observed on these baselines are stored in the multi-source file in the normal way. Data are calibrated as usual and then extracted with UVHOL for holographic analysis. This analysis includes determination of the shapes of the primary beams with PBEAM, Fourier analysis with HOLOG to make images of the dish surface including the deviations from the desired shape, and finally PANEL to prepare detailed prescriptions for adjusting the antenna panels. Verb TVLAYOUT plots an antenna panel layout on top of a TV holography image and has the layouts for the VLA and VLBA built in.

Holography data are taken by driving the moving antennas to a desired offset and then dwelling for a period of time, accumulating some number of samples. Then the moving antennas are driven to the next desired offset, usually horizontally or vertically, and the process is repeated. Holography samples are recorded with the u random parameter as the azimuth offset direction cosine, the v random parameter as the elevation direction cosine, and the w random parameter as a counter of the dwell interval. In the old VLA, the w had an additional 40000.0 added as a holography indicator. With the EVLA, the source name is the sole method of discriminating between regular visibility data taken for calibration and the holography data.

Beginning in late 2022 and supported only in 31DEC23 (and beyond), a new "on the fly" mode of holography observation has been introduced. In that mode, the moving antennas move continuously at the user-specified rate. This is much faster since there is not need to blank data while the telescopes start up and then settle at the new position. Of course, the data integration time at each offset position is considerably reduced. UVHOL and HOLOG have the new adverb OTFMODE to tell the tasks that the data were taken in this mode and to tell HOLOG to average the data in time if necessary. In this mode, the w parameter does not appear to have any reliable meaning, and UVHOL treats each new time as a new position. Otherwise, these tasks function as they always have.

In 31DEC23, the task HOLIS lists the groups of holography data showing which scans contain which range of coordinates. It tests for data samples out of order in OTF mode calling them "Backwards." It calls each scan at constant L or M a group in OTF mode and each dwell position a group in non-OTF mode. It can print a very detailed account of the latter or a summary. This listing should allow the user to find particular coordinates for plotting in UVHOL.

UVHOL has three main modes of operation: printing the data much like UVPRT, averaging and writing the data to text files for analysis by HOLOG, and plotting the data in *AIPS* plot files or on the TV display. UVHOL has all of the usual data selection and calibration adverbs. In addition it has a number of basic holography-specific adverbs. ANTENNAS is used to list the reference antennas and BASELINE is used to list the moving antennas. Since it takes some time for the antennas to settle down after each move, adverbs NPOINTS or APARM(1) through APARM(3) control which samples within each pointing are retained. APARM also controls whether the data are averaged over all moving antennas and/or all reference antennas and/or over time. Normally, holography data should not have the parallactic angle correction applied on DOPOL, but it can be done if desired. DPARM is used to advise the print routines on scaling of data and weights and also to invoke special plotting options (use db for amplitudes, plot phase or imaginary rather than amplitude/real). There are a number of the usual plot options, such as SYMBOL, DO3COL, LTYPE, and the like. DPARM(8)=1 has the

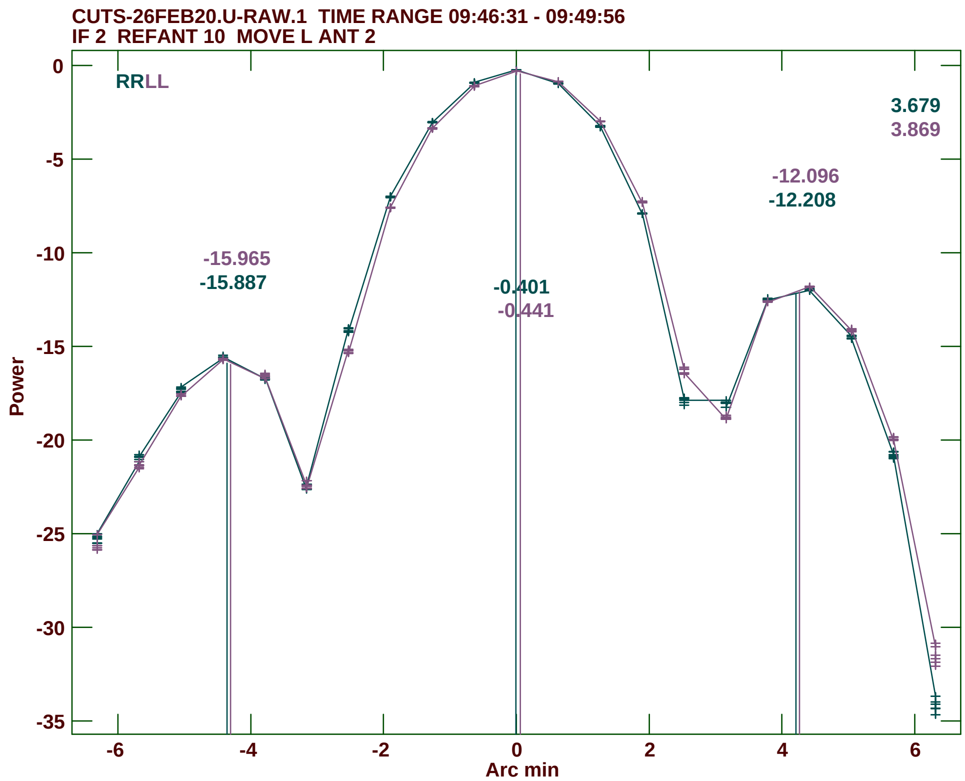

Figure V.2: Example of the amplitudes measured in a single holography scan as plotted by `UVHOL`. Antenna 2 moved in the `L` coordinate while antenna 10 remained pointed directly at the source. Both polarizations are plotted with multiple samples at each pointing (not averaged). The fit maxima are plotted along with their values in db and the sidelobe ratio in db.

task fit the main and sidelobe peaks and display the values on the plot. `DOCRT` controls where the print out is sent, either to the terminal or to `OUTPRINT`.

On `OPTYPE='HOLG'` mode, the holography data are written to text files, a number of files for each pair of antennas. Each line of these files, after a number of header lines, contains l, m, amplitude or real, phase or imaginary, error in amplitude or real, and error in phase or imaginary. In this mode, `OUTPRINT` specifies the logical directory name of the output files which are then named with an additional `HOLO`*nn*-*mmppii* where *nn* is the reference antenna (`BASELINE(`*j*`)` or 0 if multiple), *mm* is the antenna (or 0 if multiple), *pp* are 2 letters giving the polarization, and *ii* is the IF. These files are then given to `HOLOG`.

An example of plotting one short raster at 4.8 GHz has inputs

> DEFAULT UVHOL C_R	to initialize the task name and all adverbs.
> INDISK *n* ; GETN *m* C_R	to specify the holography data set.
> SOURCES 'HOLORASTER' ' C_R	to specify the standard holography source name. Required with EVLA data. Note the extra quote mark at the end which makes all the other values of `SOURCES` blank.
> STOKES 'HALF" C_R	to plot RR and LL.
> ANTENNAS 9,0 C_R	to specify one reference antenna.
> BASELINE 8,0 C_R	to specify one moving antenna.
> ICHANSEL 5, 60 C_R	to omit outer channels from the average.
> DOCALIB 1 ; DOBAND 1 C_R	to apply the calibration.

> APARM 1, 3, 2 CR	to include all but the first 3 samples and last 2 in each pointing with no averaging.
> DPARM 0 0 0 0 0 1 0 1 CR	to plot amplitudes in db and to find the center and sidelobe maxima and display and plot their locations, values, and ratio in db.
> DO3COLOR 1 CR	to plot the two polarizations in different colors.
> OPTYPE 'PLOT' CR	to make plots rather than output text.
> INP CR	to check the inputs.
> GO CR	to make the plot file shown in Figure V.2.

Now, to continue the holography analysis, we write out the data in appropriate text files. An example, would be

> TGET UVHOL CR	to get the previous parameters for `UVHOL`.
> OPTYPE 'HOLG' CR	to set the holography mode.
> OUTPRINT 'MYAREA:' CR	to set the logical directory for the output text files.
> ANTENNAS 1, 5, 9, 17 CR	to specify more of the reference antennas.
> APARM 1, 3, 2, 0, 1, 1 CR	to include all but the first 3 samples and last 2 in each pointing averaging over time within the pointing and over all reference antennas.
> INP CR	to double check the inputs.
> GO CR	to write out the data for `HOLOG`.

This will write a file named `MYAREA:HOLO00-08RR01` and one named `MYAREA:HOLO00-08LL01`. Note that we did not specify the one IF to be used, so IF number one was what we got.

`HOLOG` reads the visibility data from the text file produced by `UVHOL`. It can read either amplitude and phase or real and imaginary (`APARM(7)` in `UVHOL`). It then Fourier transforms the data to produce the grading in the aperture plane of the antenna. Images of the re-gridded observed amplitude and phase (`A_AMP`, `A_PHA`), of the weights in the re-gridding (`WGT`), of the derived amplitude and phase of the illumination in the antenna aperture (`V_AMP`, `V_PHA`), of the amplitude and phase of the point-spread function (`P_AMP`, `P_PHA`), of the phase corrections removed by the focus model (`MODEL`), of the surface deviations of the antenna (`V_DEV`) in meters, and of the interpolated antenna power pattern (`A_PWR`) may be produced under control of adverb `DPARM`. Task `PANEL` requires `V_AMP` (`DPARM(4)`) for the mask and `V_DEV` (`DPARM(9)` for the deviations. An example of the images produced by `HOLOG` is shown in Figure V.3.

Many of the control adverbs must be set to appropriate values. A sample set of inputs is

> DEFAULT HOLOG CR	to initialize everything.
> INFILE 'MYAREA:HOLO00-28RR01 CR	to point at the text file, 28 is the moving antenna with several reference antennas.
> OUTNAME '28-Q2-40935E' CR	to name the output images.
> OPTYPE 'SUBR' CR	to specify the type of antenna, sub-reflector, sub-reflector with reference pointing, or primary focus.
> FACTOR 13 CR	to set magnification factor for sub-reflector antennas; default 13.0.
> OFFSET 0.512 CR	to set the distance from the prime focus to the bottom of the subreflector in meters; 0 -¿ 0.512.
> APARM 0, -1, 25, 3.4, 9 CR	to set antenna outer diameter, inner diameter, and focal length in meters.
> BPARM 30, 256, 0, 1, 1, -2, -2.5, -12, 8100, -1 CR	to set data reduction parameters.
> DPARM 0 0 0 1 0 1 0 1 1 0 CR	to request `V_AMP`, `P_AMP` , `MODEL`, and `V_DEV` images.

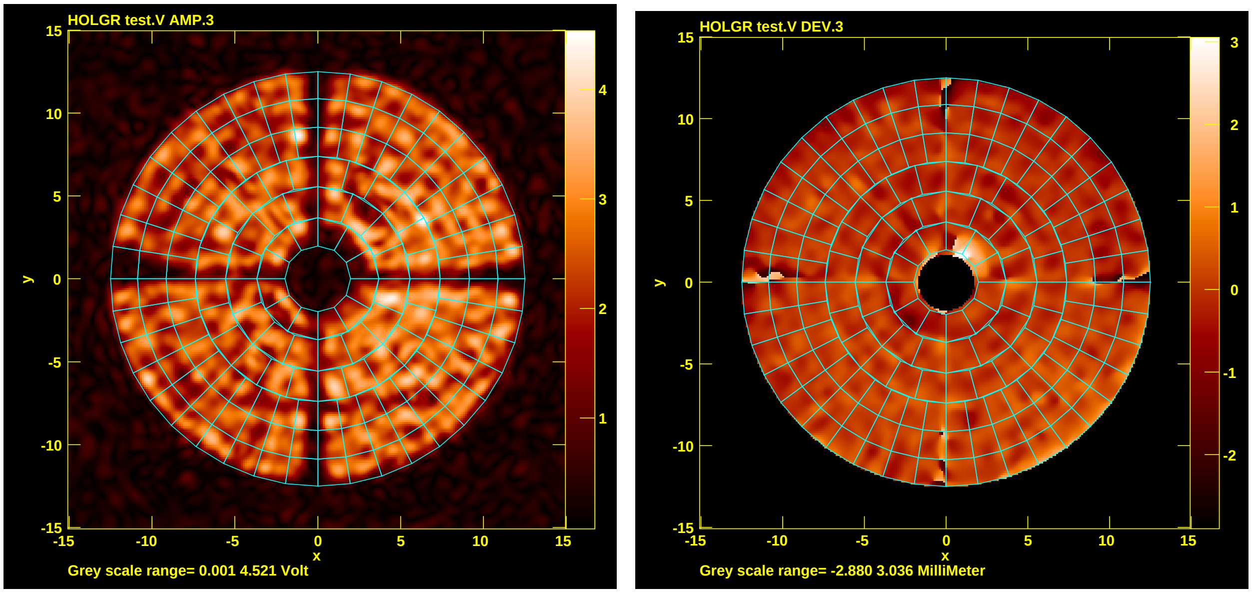

Figure V.3: Example of the mask (`V_AMP`) and deviation (`V_DEV`) images produced by `HOLOG`. Task `GREYS` can do both the pseudo-coloring with `OFMFILE = 'RYPHLAME'` and the panel layout with `INFILE = 'VLA'`.

> INP CR	to double check everything.
> GO CR	to make the images.

`BPARM` specifies the required image size in meters, the image size in pixels, minimum and maximum antenna scan angle, amplitude scaling factor, Fourier transform control (-2 says use DFT with VLA sign convention), minimum and maximum coordinate used in correcting for pointing, focus, and feed offset (negative says in the radial coordinate), a control parameter (100 inhibits local phase ambiguity correction in the `V_PHA` plane, 8000 turns on finding the Cassegrain offset), and type of data (-1 for VLA). The image size in meters is a function of frequency. To avoid this, `CELLSIZE` may be set to override `BPARM(1)` when analyzing multiple frequencies. `CPARM` all zero says to use the default spheroidal function in gridding on both `L` and `M` axes; highly recommended, but not relevant when using DFT imaging. Note that `HOLOG` can also generate images from a feed model using `VPARM`.

In `31DEC23`, an experimental task `HOCLN` was written to do a Clean operation on the images produced by `HOLOG`. Normally, data sampling in holography is essentially on a regular grid at or less than the critical sampling. As a result, this complex Clean operation usually is of little value.

The last task in the process of measuring corrections to the antenna surface is `PANEL`

> DEFAULT PANEL CR	to initialize everything.
> INDISK *n* ; GETN *m* CR	to specify the `V_DEV` image.
> IN2DI *i* ; GET2N *j* CR	to specify the `V_AMP` image used as a mask.
> OUTFILE 'MYAREA:PANEL.out CR	to specify the text file to receive the panel adjustment instructions.
> DOPLOT 7 CR	to request plots of the input deviations, the panel adjustments, and the residuals.
> APARM 2, 1, 0, 0, 42 CR	To set the panel model, the clip level in the mask image, and the number of pointings. The map size and frequency should be available from the image header.
> INLIST ' ' CR	to use the default VLA panel layout.
> INP CR	to review the inputs.

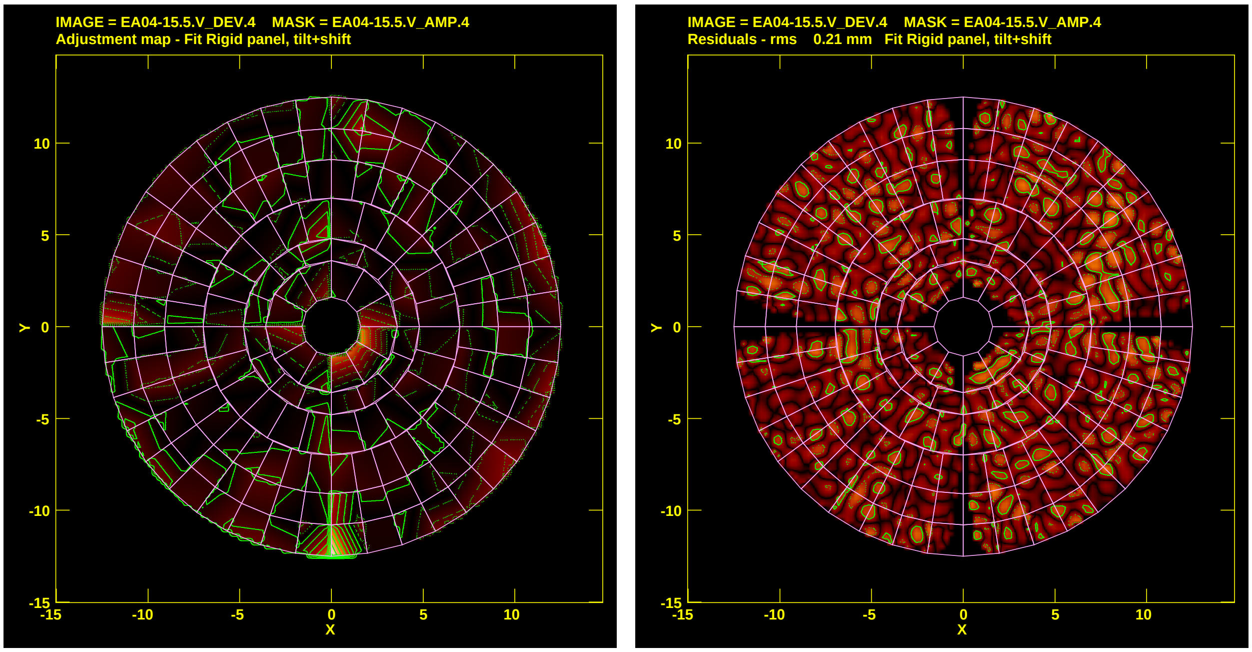

Figure V.4: Example of the panel adjustment image and the residual image computed and plotted by `PANEL`.

> GO C_R	to fit the data, write the adjustment instructions, and make plots.

The corrections are determined by fitting all samples in each panel to a model of a rigid panel with only shift or with tilt and shift or a flexible panel. There are a large number of samples in each panel, so in `31DEC23`, an option to omit samples near the edges of panels was added. This `APARM(6)` option allows the user to suppress effects spilling over from adjacent panels. Example plots of the panel adjustments and residual are shown in Figure V.4. Adding 8 to `DOPLOT` produces an image of the dish with all panels marked to show the corner definitions which rotate around the dish.

With appropriate holography data, the output of `UVHOL` may also be used to fit an analytic function to the primary beam. The task that does this is called `PBEAM` and it fits a polynomial function and makes contour plots of the input data, the fit model, the residual, and `IRING`-like plots comparing data and model in two colors, and then `IRING`-like plots of the residual image and residual image divided by the model. Plots against radius of the smooth model with the data points and of the residuals may also be generated. In `31DEC25`, `QBEAM` can construct and plot similar images without the beam fitting which may be useful for polarization data.

Sample inputs could contain

> DEFAULT PBEAM C_R	to initialize everything.
> OUTNAME 'PBEAM Cband C_R	to make output images of the data, model, and residual. Required if `DOTV` is false and `DOPLOT` is > 0.
> INFILE 'MYWORK:C-6052I_1' C_R	to provide the text file from `UVHOL`.
> OUTFILE 'MYWORK:CbandTxt C_R	to save the messages including answers to a text file.
> VPARM 0 C_R	to use all defaults in the fitting.
> DOPLOT 15 C_R	to get all possible plots.
> OPTYPE 'ELLI' C_R	to fit eccentricity and position angle.
> INP C_R	to double check everything.
> GO C_R	to do the fit and write out images and plots.

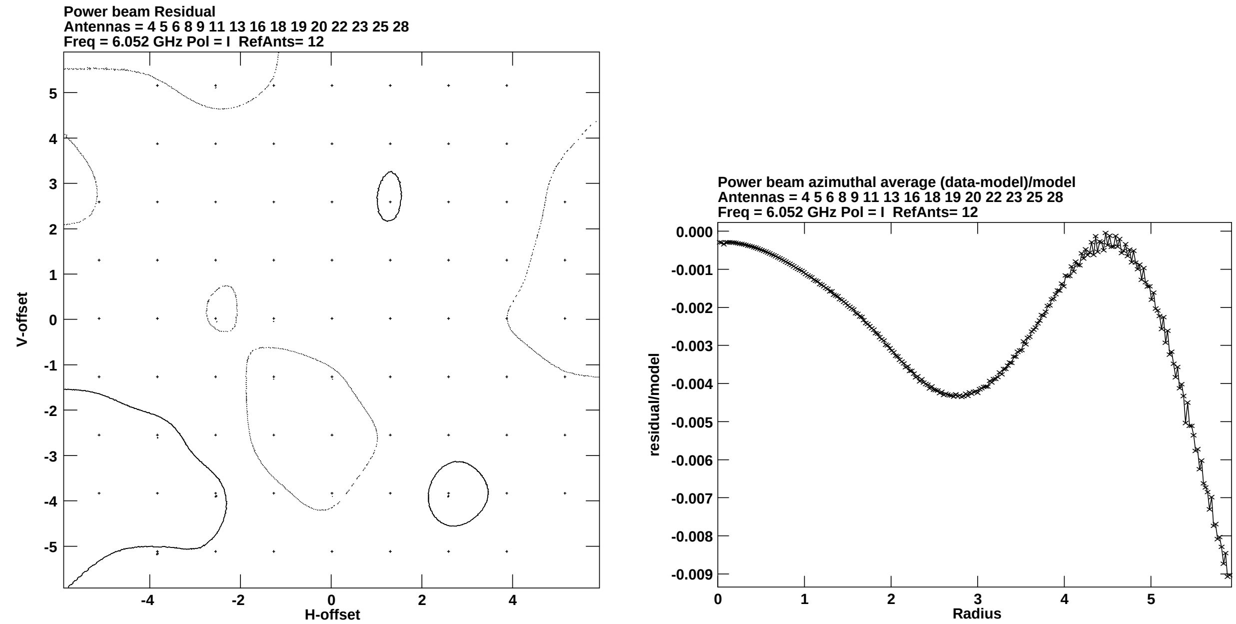

Figure V.5: Example of the plots from PBEAM: contour plot of the residual image and IRING-like plot of the residual image divided by the model.

Sample plots are shown in Figure V.5.

V.5 Antenna polarization: PDPLT

The basic calibration task PCAL will write channel-dependent polarization D-terms into a PD table (§ 4.3.16). POSSM can plot PD tables, one antenna at a time while BPLOT can plot all antennas on one plot, separated vertically. In 31DEC20 a new task PDLT was written to plot up to four PD tables on top of each other. This allows comparison of an antenna's D-term spectrum over time, *i.e.*, in multiple data sets with different antenna stations. Antennas may also be compared as in the example below.

To run DLT on one file to compare different antennas,

> DEFAULT 'PDPLT' CR	
> INDI *n* ; GETN *m* CR	to specify the data set.
> IN2DI *n* ; GET2N *m* CR	to specify the same data set.
> IN3DI *n3* ; GET3N *m3* CR	to specify a different data set.
> INVERS 3 CR	to set the PD version to use with the first data set.
> IN2VERS 1 CR	to set the PD version to use with the second data set.
> IN3VERS 3 CR	to set the PD version to use with the third data set.
> ANTENNAS 1, 6 CR	to select the antennas from the first PD table to appear in the two plots.
> BPARM 2, 3 CR	to select the antennas from the second PD table to appear in the two plots.
> CPARM 4, 5 CR	to select the antennas from the third PD table to appear in the two plots.

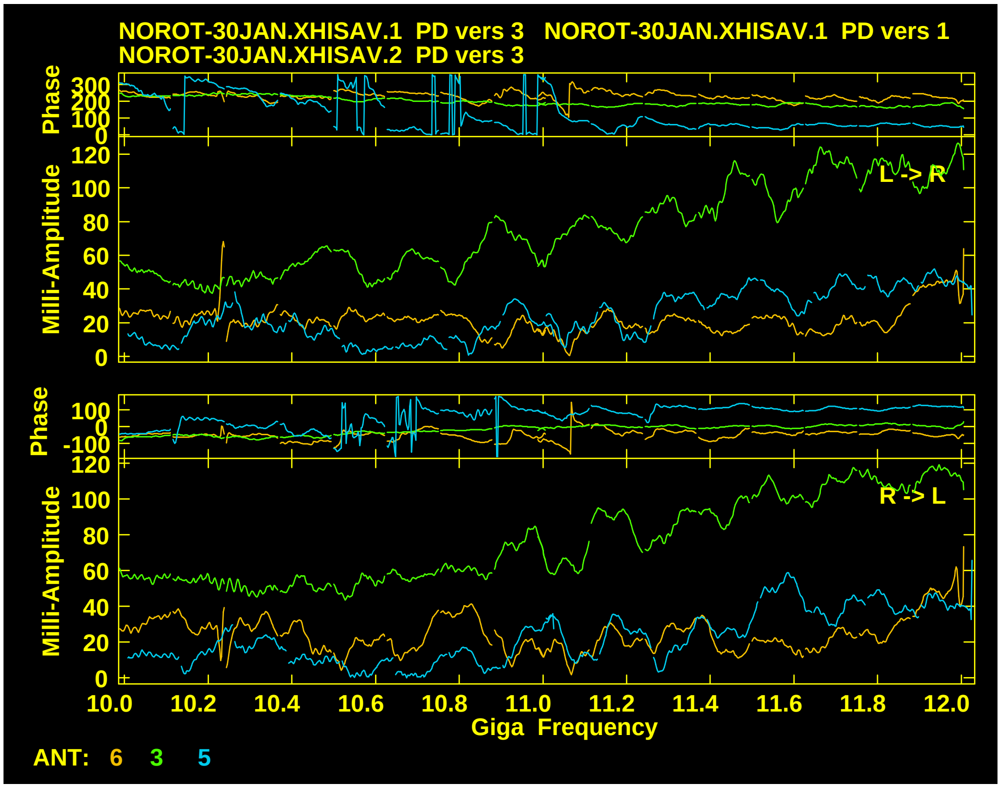

Figure V.6: The second plot from the example PDPLT.

> ECHAN 62 C_R	to omit 2 questionable channels in each IF.
> OPTYPE 'A&P' C_R	to lot both amplitude and phase.
> APARM 0 0 1 C_R	to label the x-axis with frequency.
> FACTOR -0.05 C_R	to use connected line plots only.
> DO3COLOR 1 C_R	to separate the plots with color.
> INP C_R	to check your adverb values.
> GO C_R	to run the task.

The second of the two plots made is shown in Figure V.6.