Using the EVN ParselTongue Pipeline

Cormac Reynolds (reynolds@jive.nl)

7 August, 2006

Abstract

In January 2006, the old EVN pipeline (described in astro-ph/0205118) which uses a set of `POPS` procedures that run within `AIPS` was superceded by a new version that uses ParselTongue to provide access to the `AIPS` tasks from a Python script. The following is a quick guide to using the new ParselTongue pipeline to calibrate EVN experiments.

1 Introduction

This version of the pipeline requires 31DEC02 and 31DEC04 versions of `AIPS` to be installed. If you are running different versions of `AIPS` , you will probably need to tweak the scripts slightly as the input parameters for `AIPS` tasks evolve with time. A new version that runs solely within 31DEC05 should be available soon.

It is assumed that the file names (FITS files and ancillary calibration information) follow the normal conventions used by the EVN data archive.

2 Get the Calibration Files

Get the calibration files from `vlbeer`. These are *expname*`.uvflg and` *expname*`.antab`.

For globals you will also have to get the VLBA log file (*expname*`cal.vlba`) from `zia.aoc.nrao.edu/astronomy` (username `vlbiobs`) and concatenate this with the `vlba_gains.key` to make the file *expname*`.vlbacal`.

If the VLA or GBT were used, you should also get the relevant calibration files (*expname*`cal.y`, *expname*`cal.gb`) and concatenate them with the EVN ANTAB file (after checking the INDEX and changing the station name from Y to YY). Note that for the VLA you will also have to enter the source flux densities by hand using `setjy` before the amplitude calibration is done (future versions of the pipeline should simplify this).

By default the pipeline will delete the outer 1/16th of the channels in each `AIPS` IF. You can provide a customised *expname*`.chflag` file if you want to override this.

3 Fill in the Input File

A template input file is distributed with the pipeline: `template.inp`. The pipeline has only a very small number of compulsory inputs. The following gives examples and explanations for each of these.

- `tmask` = 1, 3 – the pipeline is split into a number of sub-parts (see Appendix A). The first and last parts of the pipeline which you wish to run are selected by the `tmask` parameter. This example would run parts 1 to 3 of the pipeline. The end tmask defaults to the end of the pipeline, e.g. `tmask` = 1 will run the whole pipeline starting from part 1.

- `experiment = n06c1` – the experiment name. If it is a multiple correlation experiment (i.e. has FITS files whose name differs in the number after the first underscore) then simply append the correlation number to the experiment name (e.g. n06c1_1, or n06c1_2).

  Note that if you pipeline a dataset with multiple freqids or if you pipeline different '`glue_pass`es' (i.e. if the FITS file names differ only in the number after the *second* underscore and would normally be combined with `VBGLU`) separately, then you will have to use different experiment names for each pipeline run (e.g. n06c1_1, n06c1_2) and set the `fits_file` parameter for each as you will no longer be following standard naming conventions.

- `userno = 100` – choose a new `AIPS` number for each experiment.

- `refant = Ef, Tr, Mc` – a prioritised list of reference antennas. Choose the first one carefully.

- `plotref = Ef` – a list of antennas, all baselines to which will be plotted by the pipeline. For globals, it is a good idea to choose a plot reference on each side of the Atlantic.

- `bpass = 3C345, J2005+7752` – a list of sources to be used for the bandpass calibration (no spectral line sources!)

The following inputs must be given in certain cases. If they are not required, simply comment out the parameter (with a #) from the input file.

- `phaseref = 3C345, J1722+5856`
  `target = NRAO512, 'weak2'` – A list of phase reference sources, with a corresponding list of target sources. The first source in `target` will be calibrated with the first source in `phaseref`, etc. The same source may appear more than once in the `phaseref` list.

- `solint = 3.5` – the solution interval for fringe fitting. If phase-referencing then this will default to the typical scan length on the phase calibrator. Must be set if not phase-referencing.

- `glue_pass = 1, 2` – For experiments in which some subbands were correlated in separate correlation passes, but you wish to join these subbands together for processing (using `VBGLU`), then set `passes` to the list of correlation passes you wish to join (max = 4). Defaults to 1.

There are several other optional inputs. The default values and a brief explanation of each of these follows. To get default values, simply comment out the parameter (with a #) from the input file.

- `fitsdir = $PIPEFITS/`*`expname`*`_*/fits` – directory to load the FITS files from.

- `indir = $IN/`*`expname`* – directory to load the calibration and flag files from.

- `outdir = $OUT/`*`expname`* – directory to write output (plots, logs, etc.) to.

- `sciter = 2` – number of iterations of selfcal (0 means no self-calibration)

- `nfits = all` – number of fits files to load. Default means load all fits files that match the experiment.

- `fits_file = experiment_P.IDI*` – name of the fits file. Default is to assume EVN archive standard names, where `experiment` is the input parameter. `experiment` will have `_1` appended if it does not already have the form `experiment_N`. *P* in the default will be replaced by the pass numbers given by the `glue_pass` parameter (1 by default).
- `avg = 0` – interval in seconds for averaging the data.
- `plotavg = 0.0` – averaging time in minutes for plots of data versus time (`VPLOT`).
- `freqid = 0` – select the freqid you want to process. You should only set this parameter if there are multiple freqids in the dataset.
- `fring_snr = 7` – SNR cut-off for fring (`aparm`[7])
- `doplot = 1` – 0 turns off plotting.
- `msgkill = 0` – sets the level of detail for `AIPS` messages.
- `sources = all` – specify sources to process with the pipeline. Default means to process all sources in dataset.

4 Run the Pipeline

Simply run the script `EVN.py` with the input file as an argument:
`> EVN.py` *expname*`.inp`

Check the log file in *outdir/expname*`.pypelog`. Pay particular attention to messages beginning "pype warning".

The pipeline will produce a large number of plots at various stages of the calibration process. The `feedback.pl` script distributed with the pipeline will format these into a web page:
`> feedback.pl -exp=`*expname* `-jss=`*email*
For a description of the plots, see `http://www.evlbi.org/pipeline/pipe_desc.html`

A Selecting Sub-parts of the Pipeline (`tmask`)

Often you will want to run only a part of the pipeline (e.g. when re-starting with a new ANTAB file, you may want to skip the loading of the data and just carry on from the point where the ANTAB file is loaded). To enable this, the pipeline script is split into a number of sub-parts which are selected using the `tmask` parameter. `tmask` takes two arguments – the first is the start point and the second is the end point. The end point defaults to the end of the pipeline. By checking the `.pypelog` log file you can see where any previous run started and finished. The various sub-parts which can be selected with `tmask` are listed below.

1. Load the data. Runs `FITLD` and `INDXR`. `MSORT` and `VBGLU` are also run if necessary.
2. Apply flagging file (*expname*`.uvflg`). The outer 1/16th of the channels in each subband will also be flagged unless there is an *expname*`.chflag` file in which case it will be applied instead. For global experiments, `VLOG` will also be run.
3. Plot the uncalibrated data.
4. Amplitude calibration. Runs `ANTAB` and `APCAL`.

5. Fringe fitting. Runs `FRING`. For spectral line experiments with too many channels `FRING` runs on an averaged version of the dataset (created with `AVSPC`) and the solution table is then copied back to the original dataset.

6. Bandpass calibration. `BPASS` is run using the sources given in the input file.

7. Plot the calibrated data.

8. `SPLIT` off the individual sources, calibrating and averaging in preparation for the mapping procedure.

9. Create `MULTI` files and produce initial maps of the sources.

10. Iterate on self-calibration and mapping, using tasks `CALIB` and `IMAGR`. Amplitude calibration results are clipped using `SNSMO`.

11. Plot the self-calibrated data in a variety of ways.

12. Calculate the antenna sensitivities.

13. Save useful data (fits images, calibration tables, etc.).