The Pre-Facet-Calibration Pipeline

Andreas Horneffer

November 25, 2015

1 Stand-Alone Documentation

The following part is intended as a sub-section of the Factor chapter in the LOFAR-Cookbook. Because of that the formatting ad phrasing is a bit strange in this stand-alone version, but I believe it might sill be helpful.

1.1 Data Preparation: Pre-Facet Calibration Pipeline[1]

Factor Requirements: The input data to Factor must have the average amplitude scale set and average clock offsets removed. Furthermore, the data should be corrected for the LOFAR beam towards the phase center. The data should be concatenated in frequency to bands of about 2 MHz bandwidth (so about 10–12 subbands). All bands (= input measurement sets) need to have the same number of frequency channels[2]. Also the number of channels should have many divisors to make averaging to different scales easy. The data should then undergo direction-independent, phase-only self calibration, and the resulting solutions must be provided to Factor.

The Pipeline: The pre-facet calibration pipeline in intended to prepare the observed data so that it can be used in the facet calibration pipeline. It is a parset for the genericpipeline, that first calibrates the calibrator, then transfers the gain amplitudes, clock delays and phase offsets to the target data, and finally does a direction independent phase calibration of the target.

Please have a look at the documentation for the genericpipeline at:

`http://www.astron.nl/citt/genericpipeline/`

You should be reasonably familiar with setting up and running a genericpipeline before running this pipeline parset.

1.1.1 Download and Set-Up

The pipeline parset and associated scripts can be downloaded from github:

```
git clone https://github.com/lofar-astron/prefactor.git
```

[1] This subsection was written by Andreas Horneffer (`ahorneffer@mpifr-bonn.mpg.de`) with a lot of help from Tim Shimwell

[2] Factor does lots of averaging by different amounts to keep the data-size and computing time within limits. If the input files have different numbers of channels then finding valid averaging steps for all files gets problematic.

It can also be copied from CEP3 from: `/cep3home/horneffer/Pre-Facet-Cal`

It contains the following directories:

`bin`	scripts that process date, e.g. do the clock- / TEC- fitting, generate plots etc.
`plugins`	scripts for manipulating mapfiles (They may move to the general LOFAR software at some time.)
`parsets`	parsets (mostly BBS) that are used by the pipeline
`skymodels`	skymodels that are used by the pipeline (e.g. for demixing or calibrating the calibrator)
`docs`	the pre-facet calibration pipeline section of the LOFAR cookbook

The main part is the pipeline parset: `Pre-Facet-Cal.parset`. This requires as its input data that has been pre-processed by the ASTRON averaging pipeline. It can work both with observations that had a second beam on a calibrator and with interleaved observations in which the calibrator was observed just before and after the target. It doesn't do any demixing, but does A-Team clipping for the target data.[3] It averages over all the calibrator solutions that it gets, so observations taken at different nights should be processed in different runs of the pre-facet calibration pipeline.

To run the genericpipeline you need a correctly set up configuration file for the genericpipeline, see `http://www.astron.nl/citt/genericpipeline/#quick-start`. In addition to the settings mentioned there in the Quick Start section, you need to ensure that the plugin scripts of the pre-facet calibration pipeline are found. For that you need to add the pre-facet calibration directory to the `recipe_directories` entry in the pipeline configuration file (so that the `plugins` directory is a subdirectory of one of the `recipe_directories`). The pre-facet calibration pipeline uses only standard genericpipeline tasks, all of which are already defined in the task.cfg file that is part of the LOFAR software release since version 2.12.

At the beginning of `Pre-Facet-Cal.parset` there is a section with the following parameters that you need to set:

`avg_timestep`	averaging step needed in NDPPP to average the data to 4 seconds time resolution
`avg_freqstep`	averaging step needed to average the data to 4 ch/SB frequency resolution
`cal_input_path`	path to the directory in which the calibrator data can be found
`cal_input_pattern`	pattern that matches the calibrator data files in `cal_input_path` (Can e.g. be used to restrict the amount of data for test runs.)xs
`calibrator_skymodel`	path to the skymodel for the calibrator
`target_input_path`	same as `cal_input_path` but for the target data
`target_input_pattern`	same as `cal_input_pattern` but for the target data
`target_skymodel`	path to the skymodel for the phase-only calibration of the target
`num_SBs_per_group`	how many subbands should be concatenated into one measurement set
`results_directory`	path to the directory into which the processed target data will be moved

Below that is a section with the paths to the parsets and scripts that the pipeline uses. You need to adjust the paths to your setup.

1.1.2 Running the Pipeline

The steps of the pipeline are described in at bit more detail at: `http://www.astron.nl/citt/facet-doc/prefacet.html`. **(ToDo: Update the prefacet webpage!)** Before running the pipeline we strongly

[3] More versions of the pipeline which also do demixing and / or work with raw (non averaged) data are planned and will be added when they are done. They will probably not explained in detail but work in an analog fashion to the basic version.

suggest to go through the pipeline parset and familiarize yourself at least roughly with how the pipeline is supposed to work. In particular the flagging parameters for `NDPPP` in the `ndppp_prep_cal` and `ndppp_prep_target` steps probably need to be adjusted to each observation.

Running the pipeline is done in two steps: first only the calibrator part of the pipeline is run. After that the inspection plots of the fitting routines should be inspected. The page at http://www.astron.nl/citt/facet-doc/prefacet.html#plots contains explanations of the plots in the description of the steps. It is rather likely that some additional flagging or so is needed after the first run, this can be included by modifying the pipeline parset. After that the calibration part needs to be re-run. This may need to be repeated a number of times. A full re-run of the pipeline can be done by removing the runtime and work directories of the previous pipeline run.[4]

Only after the calibrator processing was successful and the fits to the calibration values are satisfactory, should the target part of the pipeline be run. This can be done by either adding the steps for the target processing to the existing pipeline, or by copying the numpy (`*.npy`) files from the work directory of the calibrator part to the new work directory and running only the steps for the target processing.

After the pipeline was run successfully, the resulting measurement sets are moved to the specified directory. The instrument table of the direction-independent, phase-only calibration are inside the measurement sets with the names `instrument_directioninpendent`.

[4] Experts of the genericpipeline can re-run only part of a pipeline run by either modifying the statefile or making a new pipeline parset that pick up the data at an intermediate step and continues from there. But there is no easy way to do that yet.