Online EVN Mini-Symposium & Users Meeting, 12-14 July, 2021

VLBI data reduction with CASA

M. Janssen[1], C. Goddi[2], M. Kettenis[3], D. Small[3], I. van Bemmel[3]

[1]MPIfR (Germany), [2]Radboud University (Netherlands), [3]JIVE (Netherlands)

mjanssen@mpifr-bonn.mpg.de

Why CASA?

- Future-proof software with scalability.
- Easy-to-use Python interface & pipelines are quickly written.
- Connect VLBI & connected element interferometer communities.

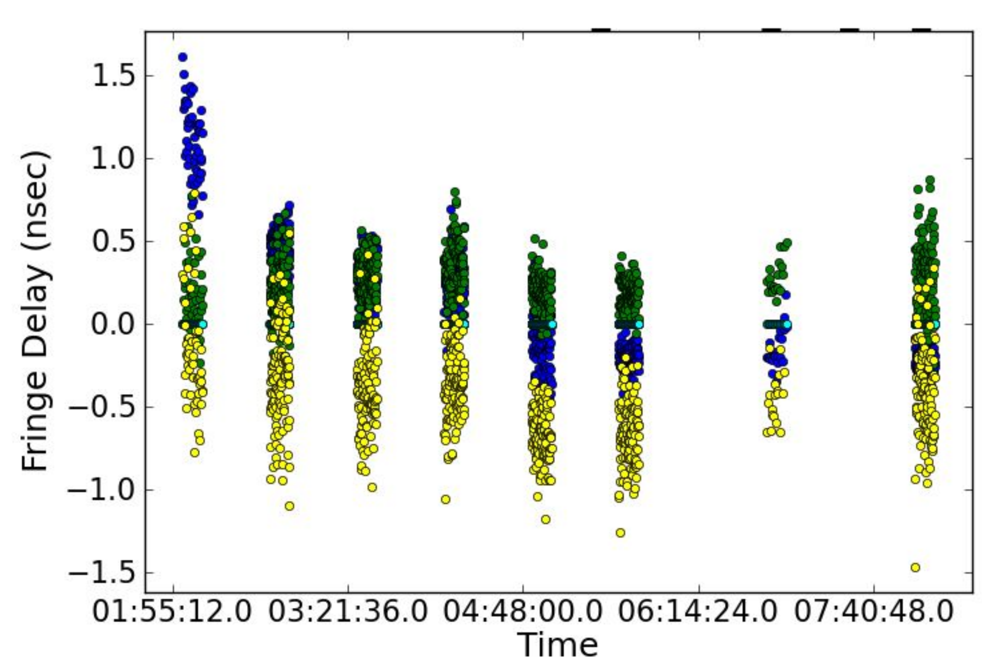
Installation and setup

- https://casa.nrao.edu/casa_obtaining.shtml
  - Extract tarball & set path to casa-release-##version##/bin (Linux).
  - No root or sudo permission needed (Linux).
- Supporting VLBI scripts can be obtained from https://github.com/jive-vlbi/casa-vlbi
  - E.g., attaching Tsys data to FITS-IDI files.

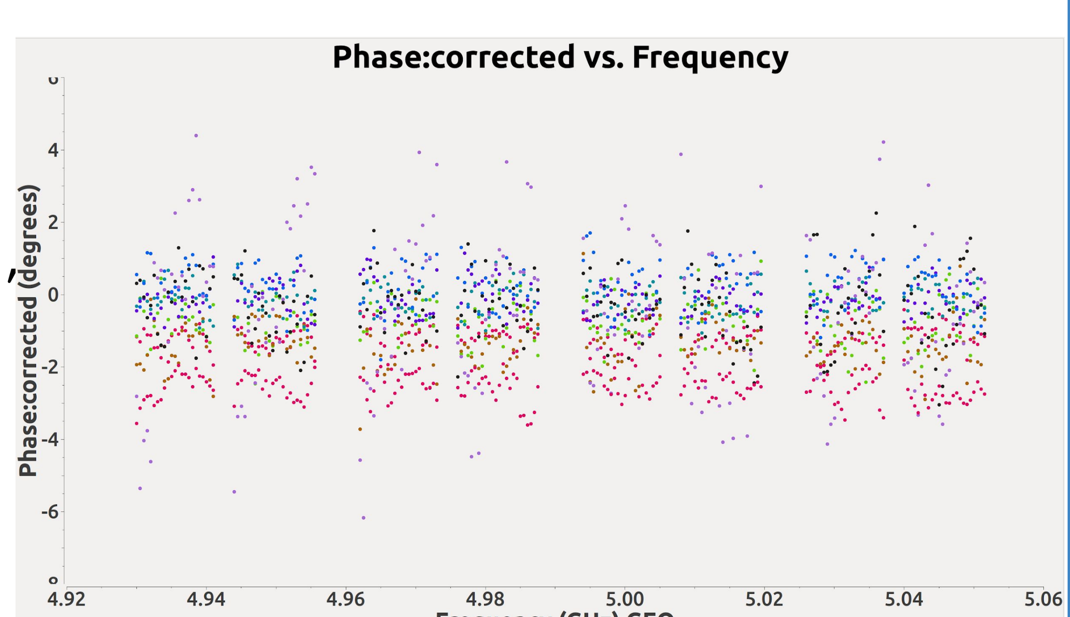
Interactive data calibration and imaging

- Open a CASA session: `$ casa`

```
importfitsidi(vis='VLBI.ms', fitsidifile=['<FITS-IDI file
#1>', '<FITS-IDI file #2>', '<FITS-IDI file #3>', ...],
constobsid=True, scanreindexgap_s=15) #Create a
MeasurementSet from your data

flagdata(vis='VLBI.ms', mode='list', inpfile='flags.txt',
flagbackup=True) #Gather ranges of bad data in a flags.txt
file and apply the flags

gencal(vis='VLBI.ms', caltable='tsys.t', caltype='tsys',
uniform=False) #Create Tsys calibration table

fringefit(vis='VLBI.ms', solint='inf', scan='1',
refant='<refant-name>', zerorates=True, minsnr=10,
caltable='sb.t', gaintable=['tsys.t'], interp=['linear'],
parang=True) #Solve for single-band phase and delay offsets
using scan #1 with the Tsys calibration applied on-the-fly

fringefit(vis='VLBI.ms', solint='60s',
refant='<refant-name>', minsnr=7, combine='spw',
caltable='mb.t', gaintable=['tsys.t', 'sb.t'],
interp=['linear', 'nearest'], parang=True) #Solve for
multi-band phases, delays, and rates every 60s of all the
data

plotcal(caltable='mb.t',
xaxis='time',
yaxis='delay', poln='R')
#Check fringefit RCP
delay results for all
antennas
```




```
bandpass(vis='VLBI.ms', field='<bandpass-calibrator>',
caltable='bp.t', gaintable=['tsys.t', 'sb.t', 'mb.t'],
interp=['linear', 'nearest', 'linear'], parang=True)
#Bandpass calibration using all the data from one source

N = … #Specify the number of spectral windows in your data
here

applycal(vis='VLBI.ms', gaintable=['tsys.t', 'sb.t',
'mb.t', 'bp.t'], interp=['linear', 'nearest', 'linear',
'linear'], spwmap=[[], [], N*[0], []], parang=True) #Apply
calibration tables

plotms(vis='VLBI.ms',
xaxis='frequency',
yaxis='phase',
ydatacolumn='corrected',
correlation='LL',
antenna='EF',
coloraxis='baseline',
scan='10')
#Plot some calibrated
Data
```




```
tclean(vis='VLBI.ms', field='<sience-target>',
imagename='vlbi.im', imsize=[200], cell=['0.1mas'],
niter=100, interactive=True) #Image the science target
```

⇒ See also https://code.jive.eu/bemmel/EVN_CASA_pipeline

rPICARD: Pipeline-based VLBI data reduction

- Specify a set of input parameters and run the full data reduction automatically, skipping the tedious manual steps.
- Better diagnostics and improved calibration methods compared to standalone CASA.
- MPI speedup.
- Extensive documentation.
- Get reproducible results.
- Low barrier to entry.
- Able to restart at any point.
- Janssen et al. A&A, 626, A75.
- Philosophy
  - Let it run with the default parameters.
  - Inspect diagnostics.
  - Adjust input parameters accordingly, identify bad data, and run it again to produce science-ready data.

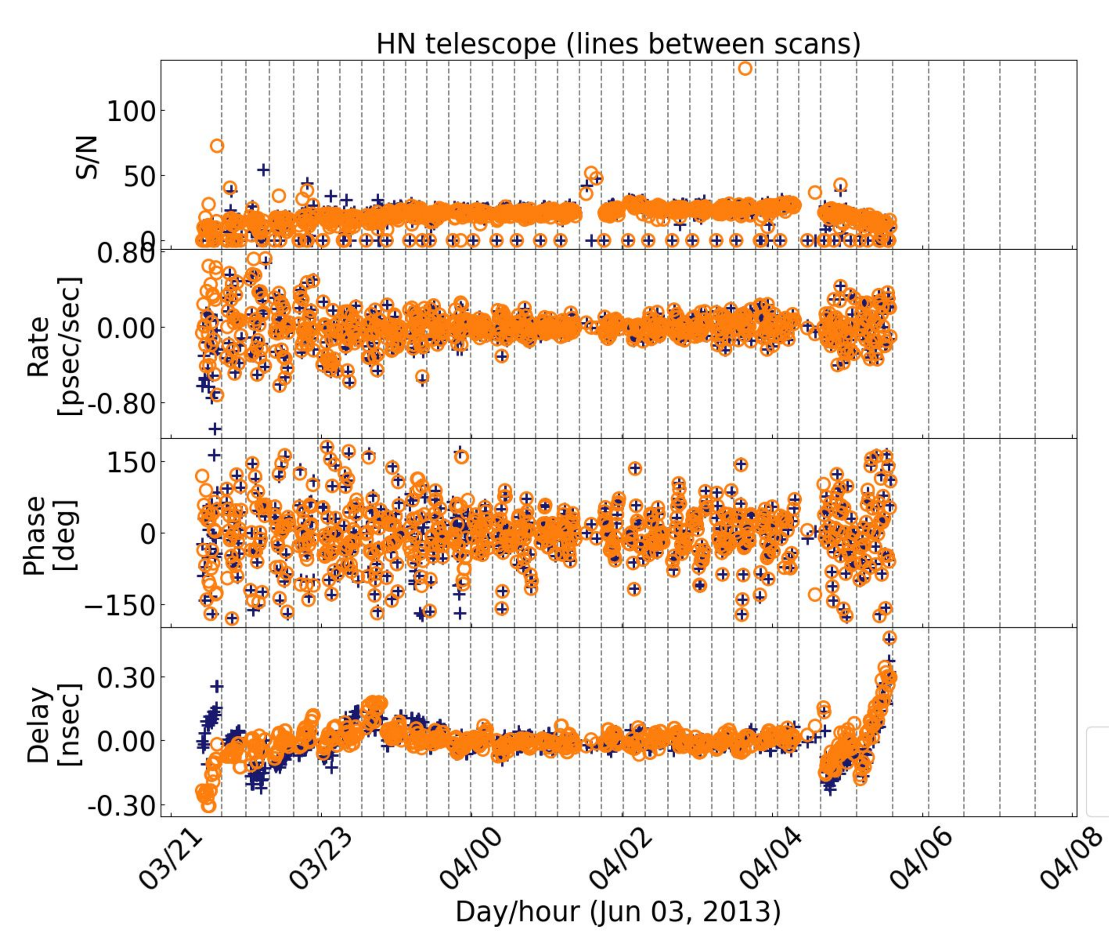
rPICARD instllation

- `$ git clone https://bitbucket.org/M_Janssen/picard`
  - Run setup script to link rPICARD to a CASA installation.
  - Set PATH and PYTHONPATH.

Running rPICARD

- `$ cd /path/to/your/working/directory`
- `$ cp -r /picard/installation/input_template/ input`
- Set a few non-default input parameters:
  - Name of science target(s) and calibrator(s) in `input/observation.inp`
  - *array_type* & *refant* in `input/array.inp`
- **`$ picard -p`**

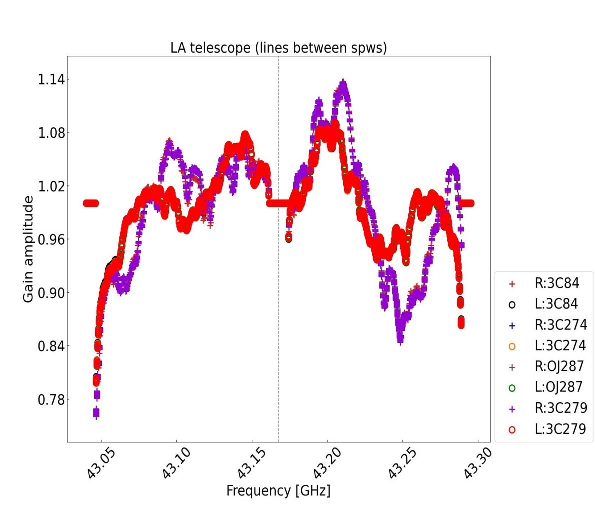
rPICARD special calibration steps & diagnostics

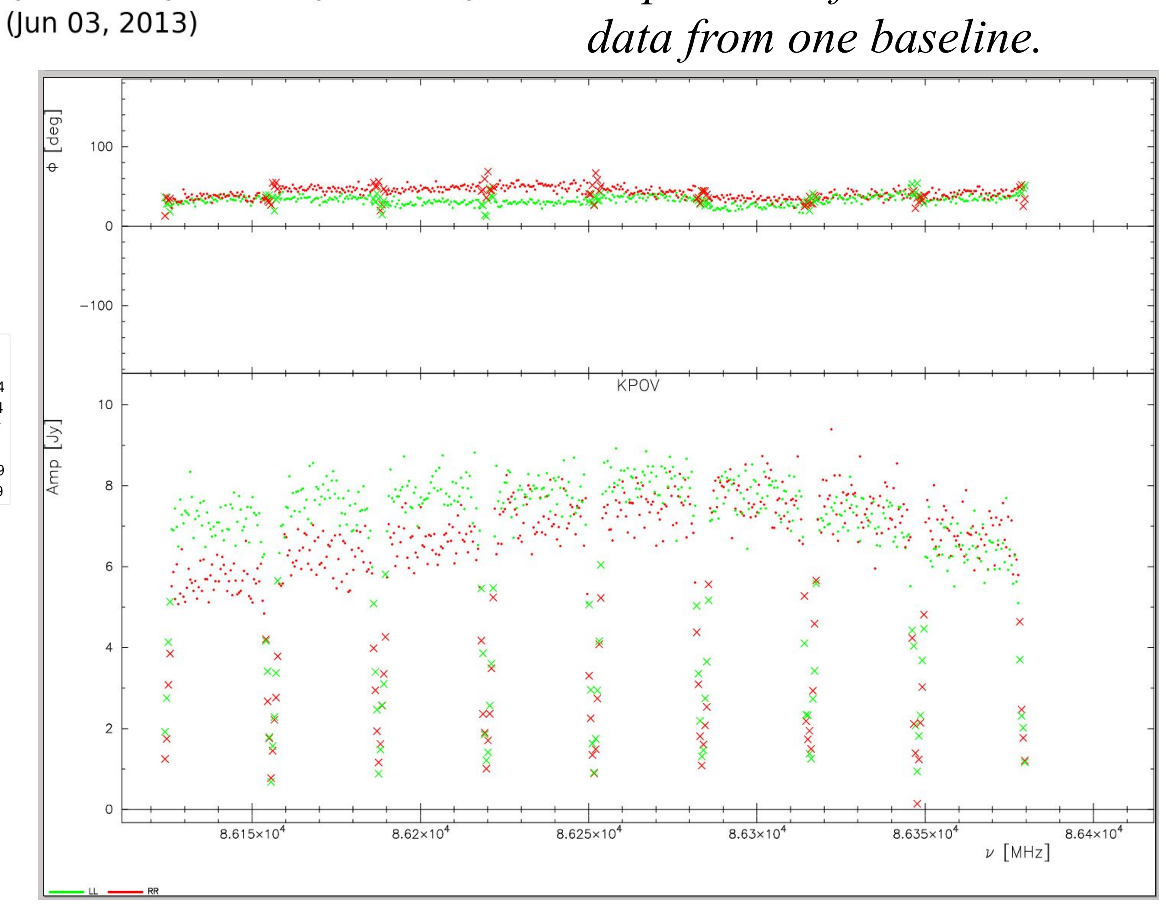


Example results from rPICARD based on Q-band VLBA data (BW0106) are shown here. These are all automatically generated diagnostics.

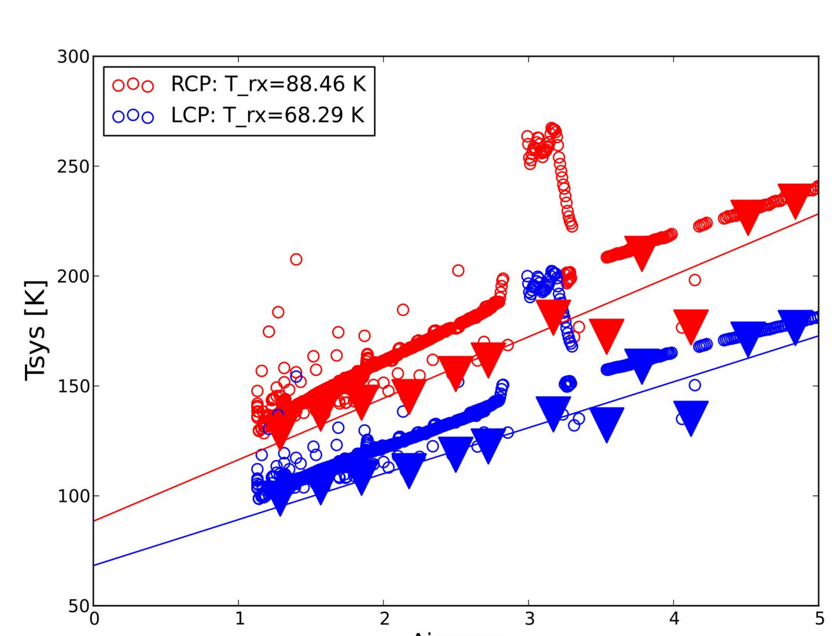
Left: *Fringe-fit results for one telescope obtained at a short cadence.*,

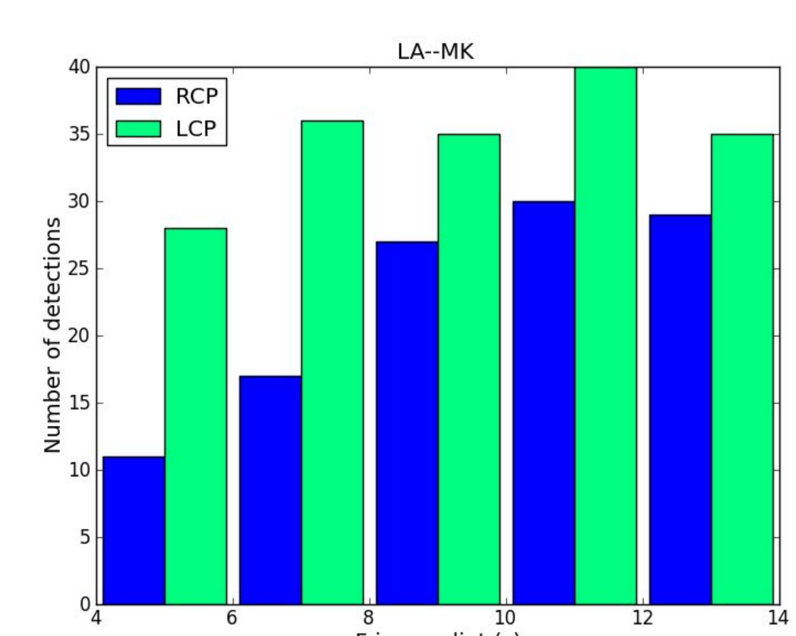
Bottom: *Phases and amplitudes of the calibrated data from one baseline.*





Top: *Scalar bandpass calibration solutions, which are estimated with a custom rPICARD routine.*





Two more special rPICARD calibration routines: *A fit for a receiver temperature used for a correction of atmospheric opacity (left) and a fringe-fit solution interval optimization that extends the coherences time and maximizes the number of source detections (right).*

Additional information

- http://jive.nl/~small/FringeFitting/n14c3_tutorial.html
- https://www.evlbi.org/evn-data-reduction-guide (AIPS/CASA comparison)
- https://www.chalmers.se/en/researchinfrastructure/oso/events/ERIS2019