Obit Development Memo 90
MeerKAT Continuum Scripts:
Outline of Data Reduction and Heuristics

Bill Cotton
May 16, 2025

1 Introduction

1.1 Scope

The scope of the present version of the MeerKAT continuum scripts is to perform standard calibration and editing of MeerKAT data and produce continuum, wide-band, images of target sources. Logs, reports and numerous diagnostic plots help evaluate the results of the processing. If default processing parameters are adequate, the scripts will start from MeerKAT archive UVTAB format FITS files and result in FITS images, calibrated data, reports, etc. The scripting is also capable of being highly tuned to a particular project and can be rerun in whole or part with user specified parameters.

1.2 Software

The calibration and imaging scripts are implemented in the Obit package [1] [1] using Obit and AIPS software. AIPS tasks are used for diagnostic plots only and not for any calibration or editing steps. Failure of AIPS diagnostics plots generates only a warning and the processing continues. Thus, the script can be run without an installation of AIPS but the diagnostic plots will not be available if needed.

The MeerKAT Obit scripts are:

- Written in python, and
- Use Obit and AIPS tasks to do the data processing, and
- Use AIPS data structures for intermediate data, and
- Write FITS images and (AIPS FITAB format) calibrated datasets.

[1] http://www.cv.nrao.edu/∼bcotton/Obit.html

AIPS (http://www.aips.nrao.edu/index.shtml) and Obit (http://www.cv.nrao.edu/ bcotton/Obit.html) are available for installation via download to Linux computers. The MeerKAT scripts are in the $OBIT/python directory with the template parameter script in $OBIT/share/scripts. A binary Linux distribution is available.

2 Execution

Several steps are needed to execute the MeerKAT scripts.

2.1 Generate parameter scripts

The processing is guided by values from python parameter scripts. These scripts can be created and initialized by information provided by the user or gleaned from the MeerKAT uvtab files using routine MeerKATCal.MKPrepare (see section 6). This will create a parameter file which can be used to drive subsequent processing.

Alternatively, the parameter file can be derived manually using the template file $OBIT/share/scripts/MKTemplateParm.py and making the substitutions described in the file.

2.2 Modify parameter scripts

If the default values in the automatically generated parameter script(s) are not appropriate, they can be changed. The details of each processing step and the parameters used are described in Section 6. Default parameters and control switches can be overridden in the parameter scripts. Additional calibrator model information can be entered as described in section 3. The end of the parameter script contains switches to turn on and off various stages. The parameter script is executed after all default parameters are set and can override any of them.

2.3 AIPS and Obit setup scripts

A script needs to be created giving the details of the AIPS and Obit installations. This script is described in detail in Section 4.

2.4 Execute scripts

The calibration pipeline script can be executed from the Unix shell by

```
> ObitTalk3 MeerKATPipeline.py AIPSSetup.py \
     MeerKATContParm_myProject.py
```

where MeerKATContParm_myProject.py is the name of your parameter script.

This procedure should start a uvtab format dataset extracted from the SARAO archive and result in a set of calibrated data, images, reports, logs and various diagnostic plots, see Section 7 for details. if Polarization calibration/imaging is required, the DelayCal data associated with the main dataset is also required. The final section of the parameter script contains a number of switches which can be used to turn on or off various steps.

3 Calibrator models

The standard MeerKAT flux density calibrators (1934-638 and 0408-65) are both unresolved but due to the other sources in the field, calibrator models are required. Calibrator source models using the Obit CLEAN components with spectra are distributed in $OBIT/share/data (0408-65LModel.fits.gz, 0408-65UHFModel.fits.gz, 1934-638LModel.fits.gz, 1934-638UHFModel.fits.gz) and should be copied to the first FITS directory and gunzipped.

The MeerKAT calibration scripts operate on arrays of calibrator dict structures with the following entries:

- `Source:` Source name as given in the SU table.
- `CalName:` Calibrator model Cleaned AIPS map name
- `CalClass:` Calibrator model Cleaned AIPS map class
- `CalSeq:` Calibrator model Cleaned AIPS map seq
- `CalDisk:` Calibrator model Cleaned AIPS map disk
- `CalNfield:` Calibrator model No. maps to use for model
- `CalCCVer:` Calibrator model CC file version
- `CalBComp:` Calibrator model First CLEAN comp to use, 1/field
- `CalEComp:` Calibrator model Last CLEAN comp to use, 0=>all
- `CalCmethod:` Calibrator model Modeling method, 'DFT','GRID',' '
- `CalCmodel:` Calibrator model Model type: 'COMP','IMAG'

- `CalFlux`: Calibrator model Lowest CC component used
- `CalModelSI`: Calibrator Spectral index
- `CalModelFlux`: Parameterized model flux density (Jy)
- `CalModelPos`: Parameterized model position offset (asec)
- `CalModelParm`: Parameterized model parameters (maj, min, pa, type)
- `useSetJy`: Standard spectrum (from SetJy) flux density used.

These dicts are created in the parameter script by routine MeerKATCal.MKCalModel for the various types of calibrators. MeerKATCal.MKStdModel is then used to fill in the details about calibrators it knows about and can find in the first FITS directory. Information not known to these scripts may be entered into the calibrator dict structure in the parameter script.

4 AIPS and Obit Setup

These scripts use data in AIPS format and some AIPS tasks; the location of the AIPS data directories and other details as well as the Obit initialization are given in the AIPSSetup.py file. An example is given in $OBIT/share/scripts/AIPSSetup.py. Note: An AIPS installation is not needed to use AIPS format data. The items that need to be specified are:

- `adirs`
  A list of the AIPS data directories as a tuple, the first element is the URL of the ObitTalkServer or None for local disk. The second element is the directory path. These can be any directory but for AIPS use needs a "SPACE" file:

  ```
  % touch aips_directory_path/SPACE
  ```

- `fdirs`
  A list of the FITS data directories as a tuple, the first element is the URL of the ObitTalkServer or None for local disk. The second element is the directory path.

- `user`
  The AIPS user number to be used.

- `AIPS_ROOT`
  The root of the AIPS system directories. An environment variable of this name is set by the AIPS startup scripts. Python None will default to your AIPS setup.

- `AIPS_VERSION`
  The AIPS version. An environment variable of this name is set by the AIPS startup scripts. Python None will default to your AIPS setup.

- `DA00`
  The AIPS DA00 directory (TDD000004; file needed). An environment variable of this name is set by the AIPS startup scripts. Python None will default to your AIPS setup.

- `OBIT_EXEC`
  The root directory of your Obit directories. Python None will default to your system installation on NRAO Linux machines.

- `noScrat`
  A list of AIPS disks to avoid for scratch files, max. 10.

- `nThreads`
  The maximum number of threads allowed to be used. This generally should not be more than the number of cores available.

- `disk`
  The AIPS disk number to use for temporary storage of the data and images.

An example AIPSSetup.py file follows, items which may need to be modified are marked by `<====`.:

```
# <====  Define AIPS  and FITS disks
# On Smeagle
adirs = [ \
    (None, "/export/ssd/bcotton/SSD"),
    (None, "/export/raid_1/aips/DATA/SMEAGLE_1"), \
    (None, "/export/raid_1/aips/DATA/SMEAGLE_2"), \
]
fdirs = [ \
    (None, "/export/raid_1/bcotton/fits"),
    (None, "/export/ssd/bcotton/FITS")]
```

```
############################## Initialize OBIT ##########################################
err     = OErr.OErr()
user    = 105                        # <==== set user number
ObitSys = OSystem.OSystem ("Pipeline", 1, user, 0, [" "], \
                          0, [" "], True, False, err)
OErr.printErrMsg(err, "Error with Obit startup")

# Setup AIPS, FITS
AIPS.userno = user

AIPS_ROOT    = "/home/aips/"                  # <==== set root of AIPS
AIPS_VERSION = "31DEC22/"                     # <==== set AIPS version
DA00         = "/export/raid_1/bcotton/DA00/" # <==== set AIPS DA00 directory
# <====  Define OBIT_EXEC for access to Obit Software
OBIT_EXEC    = None  # (def /usr/lib/obit/bin)
OBIT_EXEC    = "/export/raid_2/bcotton/ObitBase/src/Obit/" # OBITBASE
OBIT_EXEC    = "/export/ssd/bcotton/Git/Obit/ObitSystem/Obit/" # GPU

# setup environment
ObitTalkUtil.SetEnviron(AIPS_ROOT=AIPS_ROOT, AIPS_VERSION=AIPS_VERSION, \
                        OBIT_EXEC=OBIT_EXEC, DA00=DA00, ARCH="LNX64", \
                        aipsdirs=adirs, fitsdirs=fdirs)

# List directories
ObitTalkUtil.ListAIPSDirs()
ObitTalkUtil.ListFITSDirs()

# Make sure AIPS Tasks enabled
if 'LD_LIBRARY_PATH' in os.environ:
    os.environ['LD_LIBRARY_PATH']+=':'+os.environ['AIPS_ROOT']+\
    os.environ['AIPS_VERSION']+os.environ['ARCH']+'/LIBR/INTELCMP/'
else:
    os.environ['LD_LIBRARY_PATH'] = os.environ['AIPS_ROOT']+\
    os.environ['AIPS_VERSION']+os.environ['ARCH']+'/LIBR/INTELCMP/'


# Disks to avoid
noScrat     = [0]           # <==== AIPS disks to avoid
nThreads    = 24            # <==== Number of threads allowed
disk        = 2             # <==== AIPS disk number
```

5 The Process Overview

The scripted processing uses the following steps. Several of the default processing parameters are frequency dependent and may be overridden and the various steps may be turned on or off. The parameter script discussed in Section 2 is executed after all standard, default parameters are set can can be used to override any of them.

5.1 Steps

The general approach to calibration and editing is to first run editing steps which can be applied to uncalibrated data to remove the most serious RFI and equipment failures. Then, an initial pass at calibration is done and a pass at the editing needing calibrated data. Calibration aids in the editing as calibrator data with no detections are effectively removed and calibration with deviant amplitude solutions are also removed. Once the first pass at editing and calibration is completed, the initial calibration tables are deleted and the calibration repeated. This procedure removes the bulk of the RFI infected and other bad data.

Diagnostic plots at various stages of the processing are generated. These include plots of calibration results as well as sample spectra.

The calibrated data are then imaged. Images, calibrated data and calibration tables are saved to FITS files. A processing log is kept containing most details of the processing.

Following is a summary of the processing. Details and parameters which may be modified are described in a section 6.

1. Generation of parameter scripts
2. Data is converted to AIPS format using Hanning.
3. The end channels are trimmed (in Hann) and the data divided into 8 IFs
4. Clear previous calibration
5. Copy initial FG table
6. Apply any Special Editing

7. Shadow Flagging
8. Initial Time domain flagging
9. Initial Frequency domain flagging
10. Initial RMS flagging of calibrators
11. Noise diode calibration (if doPol)
12. Find reference antenna
13. Plot raw sample spectra
14. Delay calibration
15. Bandpass calibration
16. Amp & phase Calibration
17. Flagging of calibrated data
18. Recalibration
    (a) Clip calibrators in I, XY, YX
    (b) Noise diode calibration (if doPol)
    (c) Delay calibration
    (d) Bandpass calibration
    (e) Amp & phase Calibration
    (f) Flagging of calibrated data
19. Calibrate and average data
20. Instrumental polarization calibration (if doPol)
21. X-Y phase/delay calibration (if doPol)
22. Plot final calibrated spectra
23. Image targets
24. Generate source report
25. Save images, calibrated data
26. Cleanup AIPS directories

5.2 Polarization Calibration

To first order, the interferometric response for an interferometer between antennas j and k using linear detectors of an unresolved source is [5]:

$$\begin{aligned}
v_{XX} =& \frac{1}{2} g_{jX} g^{*}_{kX} (I + Q\cos 2\chi + U\sin 2\chi, \\
v_{XY} =& \frac{1}{2} g_{jX} g^{*}_{kY} [(d_{jX} + d^{*}_{kY}) I - Q\sin 2\chi + U\cos 2\chi + iV)], \\
v_{YX} =& \frac{1}{2} g_{jY} g^{*}_{kX} [(d_{YX} + d^{*}_{kX}) I - Q\sin 2\chi + U\cos 2\chi - iV)], \\
v_{YY} =& \frac{1}{2} g_{jY} g^{*}_{kY} (I - Q\cos 2\chi - U\sin 2\chi),
\end{aligned} \tag{1}$$

where g_{jp} is the complex gain of the electronics for polarization p on antenna j, "*" denotes the complex conjugate, I, Q, U, and V are the Stokes parameters of the source emission, i is $\sqrt{-1}$ and, χ is the parallactic angle given by:

$$\chi \ = \ \tan^{-1}\Big(\frac{\cos\lambda\ \sin h}{\sin\lambda\ \cos\delta\ -\ \cos\lambda\ \sin\delta\ \cos h}\Big) \tag{2}$$

where δ is the source declination, λ is the latitude of the antenna and h is the hour angle of the source. For linearly polarized detectors with detectors rotated from the local horizontal and vertical, this rotation needs to be added to the value of χ given above. All four correlations are needed to calibrate and image polarized emission.

Characterizing the feeds in terms of their ellipticities and orientations is equivalent to using the "d" terms and has the advantage that the orientations can, in principle, be measured with a plumb bob and protractor. Linear feed polarization in Obit is further discussed in [2, 3].

5.2.1 Constraints on parallel–hand calibration

The general scheme for calibrating the parallel–hand systems (XX, YY) is to make independent calibrations of them; solving for the "g" terms in Equation 1. This allows for systematic offsets between the systems of calibration which are only significant for the cross–hand correlations (XY,YX). The parallel–hand systems are degenerate in phase, only phase differences are measured. The traditional solution is to pick a "reference antenna" and call its phase zero. If the same reference antenna is used for both the XX and YY systems, the relationship between the XX and YY systems is that of the reference antenna.

If polarization calibration and imaging are to be done, there are several constraints on the parallel–hand calibration. As seen in Equation 1, Stokes correlations XX and YY are I $\pm$ the same function of Q, U and χ, hence generally sensitive to linear polarization. The independent calibration of the XX and YY systems are thus allowed, if unconstrained, to disturb any relationships between the two systems which can corrupt the cross–hand (XY,YX) data. In order to prevent this, the scheme adopted here is to use an unpolarized (Q=U=0) calibrator for both delay and bandpass calibration and thereafter use 0.5*(XX+YY) = I for calibration and the same reference antenna for XX and YY. This preserves the X-Y relationships and leaves them at that for the reference antenna. Calibrators for delay and bandpass are checked aginst the list of known unpolarized sources for MeerKAT (1934-638, 0408-65, with aliases). Fortunately, at the low frequency of MeerKAT, unpolarized calibrators are available and the primary flux density calibrators are essentially unpolarized.

5.2.2 Noise diode calibration

The DelayCal measurement at the beginning of each observing session involves injecting a noise diode signal into the data streams for both polarizations in each antenna. The measured autocorrelation functions can be used to measure the X-Y phase function downstream of the injection point (just after the Orthomode Transducers = OMTs). This removes the most variable portion of the X-Y phase function. The data for the DelayCal (separate dataset) is specified to MKPrepare as DCalFile (see section 6). This is discussed in more detail in [4].

5.2.3 Instrumental polarization

The imperfections in the feeds, i.e. "d" terms, result in a spurious polarized response to an unpolarized signal. To remove this effect, the feeds need to be characterized ("d" terms or ellipticity/orientation) and corrected. Strong, unpolarized calibrators are useful for this can can be specified as UnPolCal in MKPrepare.

As is apparent from Equation 1, the effect of parallactic angle, χ, is different on the instrumental ("d" terms) and source (Q, U) contributions to the signal. If a calibrator of unknown polarization is observed over a range of parallactic angle, this effect can be used to separate the contributions of instrumental and source polarization. Such sources are specified to MKPrepare as GainCals. These two types of calibrators are used jointly to characterize the antenna feeds.

5.2.4 Residual X-Y phase/delay

Errors in the X-Y phase difference rotate a parallactic angle dependent component of linear polarization into circular polarization. The signal path upstream of the noise diode injection point, especially the OMTs, contribute to the X-Y phase difference variation with frequency. This residual X-Y phase/delay must be determined from a calibrator of known polarization. Such calibrators are specified to MKPrepare as PolCal.

Equation 1 shows the real part of the instrumental polarization corrected cross–hand data to be a function of linear polarization ($Q\sin 2\chi + U\cos 2\chi$) and an imaginary part which is $\pm$V. Most calibrators have very weak Stokes V so the X-Y phase can be directly derived from the cross–hand data assuming V=0.

However, this part of the system is relatively stable, the correction may be determined from a separate, completely calibrated, dataset using the same reference antenna. The critical part of the calibration is the AIPS SN table derived by XYDly which may be obtainable from another dataset. Such substitution is not implemented in the present scripts.

One complication to the residual X-Y phase/delay calibration is that there are parallactic angles, and possibly frequencies, where the real part of the cross–hand correlations go through zero, see Equation 1, and cannot be used for this calibration. A case of this is illustrated in [3].

6 Script Stage Details

Details of the various processing steps and the parameters are described in the following. Processing parameters are stored in a python dict named parms and may be specified in the parameter script as

```
parms["parameter"] = value
```

Tables in the following give the parameter name, default value and a description. The contents of parms is given in the log file at the start of each run and saved in a pickle file (project_band.Parms.pickle).

1. Generation of parameter scripts This step is performed from ObitTalk to generate the parameter script(s).

```
# Example
>>> from MeerKATCal import MKPrepare
>>> datafile='/bcotton/MeerKAT/PipeTestData/G4.8+6.2_Raw.uvtab'
>>> dcalfile='/bcotton/MeerKAT/PipeTestData/SNR_4_DelayCal_Raw.uvtab'
>>> uf=getFITS(datafile,0)
>>> MKPrepare(uf,err,project='G4.8+6.2', Targets=['G4.8+6.2'], \
      DataFile=datafile, DCalFile=dcalfile, doPol=True, \
      BPCal=['J1939-6342'], DlyCal=['J1939-6342'], \
      GainCal=['J1833-2103','J1830-3602'], AmpCal=['J1939-6342'], \
      PolCal=['J1331+3030'], UnPolCal=['J1939-6342'], MKrefAnt='m058')
```

This will parse the datafile indicated and, with user intent input, generate a parameter script needed to process the data. The template can be in the current working directory or the default in $OBIT/share/scripts.

project	??	Project name(+session should be 12 or fewer characters) used as AIPS file name
session	??	session code
datafile		Archive data in uvtab format
dcalfile		DelayCal Archive data in uvtab format, if doPol
doPol		True if polarization calibration/imaging desired
Targets		List of the names of the target fields to be calibrated and imaged.
BPCal		List of bandpass calibrators
DlyCal		List of group delay calibrators
GainCal		List of gain calibrators
AmpCal		List of flux density calibrators
PolCal		List of known polarized calibrators, if doPol
UnPolCal		List of known unpolarized calibrators, if doPol
template	MKTemplateParm.py	name of the parameter template
MKrefAnt	None	MeerKAT name of reference antenna to use.
parmFile		Name of desired parameter file, generated if not given

2. General control parameters
   These parameters control the naming of files and script debugging control.

project	??	Project name
session	??	session code
band	??	Observing band code, derived from input data
check	False	Only check script, don't execute tasks
debug	False	run tasks debug

3. Data converted to AIPS format
   The bulk of the processing uses AIPS format UV data and images. The static RFI flagging and trimming of the end channels is done in Hanning (task Hann) which is used to convert the data into AIPS format. The details of the AIPS configuration are given in the AIPS-Setup.py file provided to the processing script. The output data file name is the Project+Session and the class is "UVDa"+band.

doHann	True	Load archive data into AIPS by Hanning
datafile		Archive data in uvtab format
dcalfile		DelayCal Archive data in uvtab format, if doPol

4. Drop end channels
   The first and last few channels of the band are dropped where the gain it too low. This can be turned off by setting begChanFrac and endChanFrac to 0.

begChanFrac	0.035	Fraction of bandpass to drop at the beginning
endChanFrac	0.035	Fraction of bandpass to drop at the end

5. Hanning
   At lower frequencies and compact arrays, RFI signals are frequently sufficiently strong and narrow to cause serious "Gibbs" ringing due to the truncation of the lag spectra. Hanning smoothing can be used to suppress this effect. The Hanning is also used to convert the data to AIPS format and permanently flag data affected by persistent strong RFI sources.

doHann	True	Apply Hanning smoothing, convert to AIPS? Always used
doDescm	True	If True, drop every other channel after smoothing
datafile		Name of the archive data (uvtab)
dcalfile		Name of the DelayCal archive data (uvtab)

6. Clear previous calibration
   If the script is restarted it is frequently desirable that previous attempts at calibration and editing be removed.

doClearTab	True	Clear cal/edit tables?
doClearGain	True	Clear SN and CL tables > 1?
doClearFlag	True	Clear FG tables > 1?
doClearBP	True	Clear BP tables?

7. Copy initial FG table
   To allow restarting of the flagging, the on-line flags which are in FG table 1 are copied to table 2 and new flags added there. This should be turned off if the script is restarted except at the beginning.

doCopyFG	True	Copy FG 1 to FG 2?

8. Apply Special Editing
   If some data are known to be bad, e.g. no receiver, then this information can be passed to the script. if doEditList is True, each entry in editList is a python dict with the following:

   - `timer:` The affected time range as a pair of strings of the form day/hour:min:sec.
   - `Ant:`A baseline specification as a pair of antenna numbers, if the second is zero, then all baselines to the first antenna number is flagged. If the first is zero, then all antennas are flagged
   - `IFs:` Range (1-rel) of IFs (spectral windows) to flag. If the second is zero then all IFs higher than the first are flagged.
   - `Chans:` Range (1-rel) of channels to flag. If the second is zero then all channels higher than the first are flagged.
   - `Stokes:` Array of flags, 1=>flag, 0 => not flag; in order XX, YY, XY, YX.
   - `"Reason:` Up to 24 characters giving reason.

   an example:

```
parms["doEditList"]  = True        # Edit using editList?
parms["editList"] = [
    {"timer":("0/00:00:0.0","5/00:00:0.0"),"Ant":[1,0],
     "IFs":[1,0],"Chans":[1,0],  "Stokes":'1111',"Reason":"No Rcvr"}
]
```

doEditList	False	Edit using editList?
editFG	2	Table to apply edit list to
editList	[ ]	List of data to flag

9. Shadow Flagging
   In the compact MeerKAT configuration, some antennas may shadow others at times. The affected data may be flagged using Obit/UVFlag.

doShad	True	Do shadow flagging?
shadBl	18.0	Minimum shadowing baseline (m)

10. Initial Time domain flagging
    Obit task MednFlag can be used to flag data by amplitudes deviant from a running median by more than a specified amount. This is performed independently on each data stream (baseline, channel, IF, poln). At this point the data are uncalibrated.

doMedn	True	Median editing?
mednSigma	10.0	Sigma clipping level
mednTimeWind	8.0	Window width (min) for median flagging
mednAvgTime	8.0/60.	Averaging time (min)
mednAvgFreq	1	1=>avg mednChAvg chans, 2=>avg all chan, 3=>avg chan and IFs
mednChAvg	8	Number of channels to average

11. Initial Frequency domain flagging
    The uncalibrated data can be examined for impulsive signals in frequency by comparison with a running median in each spectrum and deviant data are flagged using Obit task AutoFlag. Since bandpass corrections have not been determined and applied at this stage, structure in the instrumental bandpass will increase the apparent RMS in the baseline reducing the sensitivity of this test. Note: this should NOT be used for data expected to have prominent spectral features.

doFD1	True	Do initial frequency domain flagging?
FD1widMW	55	Width of the initial FD median window
FD1maxRes	10.0	Clipping level in sigma
FD1TimeAvg	1.0	time averaging (min). for initial FD flagging

12. Initial RMS flagging of calibrators
    Calibrators are expected to be simple and have significant SNR so can be edited by having an RMS/average amplitude of less than some amount. Discrepant calibrator data can be flagged in this step using Obit task AutoFlag.

doRMSAvg	True	Edit calibrators by RMS/Avg?
RMSAvg	5.0	Max RMS/Avg for time domain RMS filtering
RMSTimeAvg	1.0	Time averaging (min)

13. Noise diode calibration
    If polarization calibration is to be applied, the "DelayCal" at the beginning of the observing session is used to correct the X-Y phase function of each antenna downstream of the noise diode signal injection, just after the Orthomode transducer (OMT).

doNDCal	True	Do Noise diode calibration?
dcalfile		Name of the DelayCal archive Data (uvtab)

14. Reference antenna
    If the reference antenna is unspecified (0), this step runs Obit task Calib on the bandpass calibrator(s) (assumed to give good fringes) using the middle half of each spectrum. The resultant SN table is then examined for the antenna with the maximal amount of valid solutions and with the highest average SNR; this antenna is used as the reference antenna. Once a reference antenna is determined its value is saved in a pickle file and will be recovered on subsequent runs.

refAnt		Reference antenna, if None then determine
BPCals		Specified to MKPrepare
bpsolint1	0.25	Bandpass first solution interval

15. Plot Raw spectra
    At this point plots of sample spectra can be made to display residual RFI and other problematic data. Note: The data are mislabeled as RR and LL (really XX and YY) to trick AIPS task POSSM into plotting them.

doRawSpecPlot	True	Plot diagnostic raw spectra?
plotSource		Default is first bandpass calibrator
plotTime		List of start and end time in days.
		Default is first bandpass calibrator scan
refAnt		Reference ant., baselines to refAnt are plotted

16. Delay calibration
    Parallel hand group delays are solved for using the list of calibrator models in DCals. Obit task Calib solves for the delays which are then smoothed and applied to all sources in a new CL table using Obit task CLCal. Solutions can be plotted. If doPol, these calibrators must all be unpolarized.

doDelayCal	True	Determine/apply delays?
DCals		The list of delay calibrators given to MKPrepare.
		using and standard calibrator models.
delayBChan	1	first channel to use in delay solutions
		max(2, 0.05*nchan)
delayEChan	0	highest channel to use in delay solutions
		0=>all
delaySolInt	2.0	Solution interval (min)
delaySmoo	0.0	Delay smoothing time (hr)
delayZeroPh	False	Zero phase in Delay solutions?
delayAvgIF	False	Average the IFs?
delayAvgPol	False	Average the HH and VV pols?
refAnts	[refAnt]	Delay reference ant., baselines to refAnt are plotted
gainUVRange	[0.0,0.0]	Range of baseline used in kilolambda, zeros=all
doTwo	True	Use two baseline combinations in delay cal
doSNPlot	True	Plot calibration solutions?

17. Bandpass calibration
    Bandpass calibration uses Obit task BPass and calibrator model list BPCals. BPass does a two pass calibration, the first doing a phase only calibration to straighten out the phases followed by a longer amplitude and phase calibration using blocks of channels. The resultant solutions are then combined into a BP table. If doPol, these calibrators must all be unpolarized.

doBPCal	True	Determine/apply bandpass calibration?
BPCals		The list of bandpass calibrators given to MKPrepare. using standard calibrator models.
bpsolint1	15/60	BPass phase correction solution in min.
bpsolint2	60	BPass bandpass solution interval (min)
bpsolMode	'A&P'	Bandpass type 'A&P', 'P', 'P!A'
bpBChan1	1	Low freq. channel, initial cal
bpEChan1	0	Highest freq channel, initial cal, 0=>all
bpBChan2	1	Low freq. channel for BP cal
bpEChan2	0	Highest freq channel for BP cal, 0=>all
bpChWid2	1	Number of channels in running mean BP soln
specIndex	-0.7	Spectral index of BP Cal
bpDoCenter1	None	Fraction of channels in 1st, overrides bpBChan1, bpEChan1
bpdoAuto	False	Use autocorrelations rather than cross?
bpUVRange	[0.0,1e5]	UV range for bandpass cal zeroes=> all
refAnt		BP reference ant., baselines to refAnt are plotted
doAmpEdit	True	Edit/flag on the basis of amplitude solutions
ampSigma	10.0	Multiple of median RMS about median gain to clip/flag

18. Amp & phase Calibration
    Standard flux density calibrators have their flux densities entered into the SU table using Obit task SetJy, other calibrators have their flux density entries set to 1.0. All the amplitude and phase calibrators have Obit/Calib run using their models and doing amplitude and phase solutions. Solutions are then median window smoothed using Obit/SNSmo to time solSmo clipping really wild points. Obit task GetJy then solves for the flux densities for non flux density calibrators and adjusts the SU and SN tables. If doAmpEdit is True, solutions in each IF (spectral window) more than ampSigma from the mean are flagged both in the SN table and in FG table ampEditFG. Finally solutions are applied to the previous CL table to create a new CL table. Calibrator sources are "self-calibrated" and targets use the smoothed solutions. Solution plots are written into file
    parms["project"]+"_"+parms["session"]+"_"+parms["band"]+"APCal.ps".

doAmpPhaseCal	True	Do amplitude and phase calibration?
ACals		The list of amplitude cals (AmpCal) given to MKPrepare, using standard calibrator models.
PCals		The list of phase (gain) calibrators given to MKPrepare.
refAnt		Reference antenna
solInt	2	Solution interval (min).
ampBChan	1	first channel to use in A&P solutions max(2, 0.05*nchan)
ampEChan	0	highest channel to use in A&P solutions 0=>all
doPol		For polarization calibration the solutions average XX and YY.
solSmo	0.0	Smoothing interval (min)
ampScalar	False	Ampscalar solutions?
doAmpEdit	True	Edit/flag on the basis of amplitude solutions
ampSigma	20.0	Multiple of median RMS about median gain to clip/flag
ampEditFG	2	FG table for editing
doSNPlot	True	Plot calibration solutions?

19. Flagging of Calibrated Data
    Calibrated calibrator data are then edited using Obit/AutoFlag. Data with amplitudes outside of a given range are flagged and data overly discrepant from a running median in frequency is flagged.

doAutoFlag	True	Autoflag editing after first pass calibration?
minAmp	1.0e-6	Minimum allowable amplitude
timeAvg	0.5	AutoFlag time averaging in min.
doFirstAFFD	False	do AutoFlag frequency domain flag
FDmaxAmp	IClip	Maximum average amplitude (Jy)
FDwidMW	51	Width of the median window
FDmaxRMS	[10.0,.1]	Channel RMS limits (Jy)
FDmaxRes	7.0	Max. residual flux in sigma
FDmaxResBL	7.0	Max. baseline residual
FDbaseSel	[1,0,1,0]	Channels for baseline fit

20. Recalibration
    If doRecal is true then the previous calibration tables are deleted and the calibration redone using the flag table from the first pass.

    (a) Calibrator Clipping
        If all four Stokes polarization products are in the data:

doClipCals	True	Clip Calibrator Data?
IClip	50	AutoFlag Stokes I clipping (Jy)
XClip	3	AutoFlag XY, YX clipping (Jy)

    (b) Noise diode calibration
        as before if doNDCal2=True.
    (c) Delay calibration
        As before if doDelayCal2 = True.
    (d) Bandpass calibration
        As before if doBPCal2 = True.
    (e) Amp & phase Calibration
        As before if doAmpPhaseCal2 = True.
    (f) Flagging of calibrated data
        As before if doAutoFlag2 = True.

21. Calibrate and average data
The calibration and editing files are then applied with possible averaging in time and/or frequency. This uses Obit/Splat or Obit/UVBlAvg which write a multi-source file.

doCalAvg	?	Calibrate and average?
		"Splat" => Calibrate data with no averaging.
		"BL " => Calibrate data with baseline dependent averaging.
		to leave FOV=FOV undistorted and time <= solPInt (min).
avgClass	"UVAv"	+band, AIPS class of calibrated/averaged UV data
seq	1	AIPS sequence
CalAvgTime	0	Time for averaging ("Splat") calibrated UV data (min)
avgFreq	0	0 => no averaging, 1 =>avg chAvg chans, 2 =>avg all chan,
		3 =>avg chan and IFs
FOV	1.0	Desired Field of view.
solPInt	0.5	phase self cal solution interval (min)
chAvg	1	Number of channels to average
CABChan	1	First channel to copy
CAEChan	0	Highest channel to copy, 0 => all higher than CABChan
CABIF	1	First IF to copy
CAEIF	0	Highest IF to copy, 0 => all higher than CABIF

22. Instrumental Polarization Calibration (if doPol)
Determine instrumental polarization from a list of known unpolarized and gain calibrators. In MKPrepare these are specified as 'UnPolCal' and 'GainCal'. The polarization state of the GainCals is determined and UnPolCal are specified as unpolarized. Calibration uses Obit task PCal which determines antenna (and source?) polarization parameters on blocks of channels in a running window. The antenna parameters are the ellipticity and orientation of the feed; see Obit Development Memo 30 for details.

doPolCal		Determine instrumental polarization?
UnPolCal		as specified to MKPrepare.
GCalList		as specified to MKPrepare.
PCSolInt	2.0	Instrumental solution interval (min), 0 => scan average(?)
PCRefAnt	0	Reference antenna, 0 => absolute solution
PCSolType	” ”	Solution type, ”LM ” (better), ” ” (faster)
PCSolInt	2.	Data averaging time prior to solution.(min)
PCChInc	17	Channel step in spectrum
PCChWid	17	Number of channels to average

23. X-Y delay calibration (if doPol)
Determine right–left (X-Y) delays from a list of known polarized calibrators. Sources specified to MKPrepare as PolCal are searched for in a standard list of know calibrators and, if found, the Rotation measure and EVPA at the reference frequency and RM are entered into the XYCals structure. If a source is not in the standard list, the appropriate values can the entered into the parameter script. The solution uses task XYDly.

doXYDelay	??	Do X-Y delay calibration?
XYCals	??	Array of triplets of (name, EVPA (deg at ref Freq), RM (rad/m**2)
xyBChan	1	First (1-rel) channel number
xyEChan	0	Highest channel number. 0 => high in data.
xyUVRange	[3,200]	Range of baseline used in kilo wavelengths, zeros=all
xytimerange	[0.0,1000.0]	Time range of data (days)
xyFitType	0	Fitting type in XYDly, 0=>both, 1=>XY, 2=>YX
xyChAvg	8	Number of channels to average
xyTimeAvg	2.0	Data Averaging time (min)
xySolnInt	1000.	Solution interval (min)
xyrefAnt	refAnt	Reference antenna, defaults to refAnt
xyminSNR	5	Minimum SNR in XYDly

24. Plot final calibrated spectra
    At this point, plots of sample spectra can be made to display calibrated data. Note: The data are mislabeled as RR and LL (really XX and YY) to trick AIPS task POSSM into plotting it.

doSpecPlot	True	Plot diagnostic spectra?
plotSource		Default is first bandpass calibrator
plotTime		List of start and end time in days.
refAnt		Reference ant., baselines to refAnt are plotted

25. Plot calibrated polarized spectra
    Plots of sample polarized spectra can be made to display calibrated data. Note: The data are mislabeled as RL and LR (really XY and YX) to trick AIPS task POSSM into plotting it.

doPolSpecPlot	?	Plot polarized spectra?
plotSource		Default is first polarized calibrator (PolCal)
plotTime		List of start and end time in days.
refAnt		Reference ant., baselines to refAnt are plotted

26. Image targets
All targets are imaged and deconvolved using Obit/MFImage. Phase only and amp and phase calibration may be applied if sources exceed given thresholds. The resultant images are cubes having planes:
    (a) Total intensity at reference frequency.
    (b) Spectral index at reference frequency
    (c) any higher order planes
    (d) One plane for each of the sub-bands.

doImage	True	Image targets?
targets	[?]	Specified to MKPrepare as Targets.
seq	1	AIPS sequence for images
doPol	True	Apply polarization cal?
PDVer	1	PD table to apply
outIClass	"IClean"	Image AIPS class
Stokes	"I"	Stokes to image, if doPol "IQUV"
Robust	0.0	Weighting robust parameter
FOV	1.0	Field of view radius in deg.
Niter	50000	Max number of CLEAN iterations
UVRange	[0.0,0.0]	Range of baseline used in kilo wavelengths, zeros=all
minFlux	0.0	Minimum CLEAN flux density (Jy)
minSNR	4.0	Minimum Allowed SNR in self cal
maxPSCLoop	0	Max. number of phase self cal loops
minFluxPSC	0.01	Min flux density peak for phase self cal
solPInt		Phase self cal solution interval (min)
solPMode	"P"	Phase solution for phase self cal
solPType"	" "	Solution type for phase self cal
maxASCLoop	0	Max. number of Amp+phase self cal loops
minFluxASC	1.0	Min flux density peak for amp+phase self cal
solAInt		amp+phase self cal solution interval (min)
solAMode	"A&P"	Amp and phase self cal
solAType	" "	Solution type for Amp and phase self cal
refAnt		Reference ant., for self cal, def. determined by script
avgPol	True	Average poln in self cal?
avgIF	False	Average IF in self cal?
nTaper	0	Number of additional imaging multi-resolution tapers
Tapers	[20.0,0.0]	List of tapers in pixels
do3D	False	Make ref. pixel tangent to celest. sphere for each facet
noNeg	False	Allow negative components in self cal model?
BLFact	1.01	Baseline dependent time averaging for > 1.0?
BLchAvg	True	Baseline dependent frequency averaging?
MBnorder	1	Order of wide-band imaging
MBmaxFBW	0.05	max. MB fractional bandwidth
CleanRad	None	CLEAN radius about center or None=autoWin
PBCor	False	Make primary beam correction when finished
antSize	24.5	Antenna diameter for primary beam correction (m)

27. Save images, calibrated data
Images and calibrated/averaged data and calibration tables are written to FITS files. File names begin with parms["project"]+parms["session"]+parms["band"] followed by <source_name>+<Stokes>+"Clean.fits" for images and ".uvtab" for calibrated data and "CalTab.uvtab" for calibration tables from the original data.

doSaveImg	True	Save target images to FITS?
targets	[?]	Specified to MKPrepare as Targets.
doSaveUV	True	Save calibrated UV data for AIPS/FITAB format?
doSaveTab	True	Save calibration tables for AIPS/FITAB format?

28. Generate report

doReport	True	Generate source report?
targets	[?]	Specified to MKPrepare as Targets.
seq	1	AIPS sequence for images
outIClass	"IClean"	Image AIPS class
Stokes	"I"	Stokes imaged, "IQUV" if do Pol

29. Cleanup
AIPS data and image files are zapped.

doCleanup	True	Clean out AIPS directories?

7 The Products

- Calibrated (u,v) dataset with calibration and flagging tables in AIPS FITAB format – Tables from initial data and calibrated, averaged visibilities. These files are named:
parms["project"]+parms["session"]+parms["band"]+"CalTab.uvtab" and parms["project"]+parms["session"]+parms["band"]+".uvtab"
NB: This is not "uvfits" format.

- FITS Images – for each target object in files
parms["project"]+"_"+parms["session"]+"_"+parms["band"]+ source_name+".IClean.fits".
If wide-band imaging is used, then the resultant images are cubes having planes:

1. Total intensity at reference frequency.
2. Spectral index at reference frequency
3. any higher order planes
4. One plane for each of the coarse frequency samples.

- Diagnostic plots – calibration and several per source. The project plots have prefix parms["project"]+"_"+parms["session"]+"_"+parms["band"] and are
  - `RawSpec.ps:` AIPS/POSSM plots of sample spectra with initial editing but no calibration applied.
  - `Spec.ps:` AIPS/POSSM plots of sample spectra with final editing and calibration applied.
  - `PolSpec.ps:` AIPS/POSSM plots of sample XY and YX spectra with final editing and calibration applied of the first polarized calibrator.
  - `DelayCal.ps:` AIPS/SNPLT plots of delay calibration. One per pass through the calibration.
  - `APCal.ps:` AIPS/SNPLT plots of amplitude and phase calibration. One per pass through the calibration.
- Reports and logs created during the process
  The logfile is
  parms["project"]+"_"+parms["session"]+"_"+parms["band"]+".log".

The file set comprising all files and the meta-data are stored in a single directory.

References

[1] W. D Cotton. Obit: A Development Environment for Astronomical Algorithms. *PASP*, 120:439–448, 2008.

[2] W. D Cotton. On–axis Instrumental Polarization Calibration for Linear Feeds. *Obit Development Memo Series*, 32:1–11, 2015.

[3] W. D Cotton. Cross Polarized Delay Calibration of Linear Feeds with XYDly and General Polarization Calibration. *Obit Development Memo Series*, 89:1–8, 2025.

[4] A. Plavin, W. D Cotton, and T. Mauch. MeerKAT L Band Polarimetric Calibration. *Obit Development Memo Series*, 62:1–9, 2025.

[5] A. R. Thompson, J. M. Moran, and G. W. Swenson, Jr. *Interferometry and Synthesis in Radio Astronomy, 2nd Edition.* Wiley-Interscience, 2001.