CASA VLBI Requirements

Mark Kettenis *

May 7, 2020

1 Introduction

Over the last three years, a software development team from JIVE, in collaboration with the CASA team at NRAO has worked on adding VLBI capabilities to CASA. Basic VLBI data reduction is now possible in CASA and this functionality has been successfully used in the rPICARD[1] pipeline that was used for verification of the recent Event Horizon Telescope results. But while a `fringefit` task is now available in CASA, some of the bells and whistles available in older packages like AIPS are still missing. In addition to that there still is other functionality that is used for more advanced VLBI data reduction that is not yet available in CASA. As such CASA is not yet a fully functional replacement for AIPS.

This document is an attempt to identify what functionality is still missing. It collects stakeholder input collected at various training events and discussions with the rPICARD devloper with requirements from an earlier NRAO document written by Walter Brisken, Vincent Fish, Jeff Kern and Shep Doeleman. In fact a large fraction of the requirements in this document and some of the motivations for those requirements were taken verbatim from that document. Some of the requirements have been dropped even if the functionality has not been implemented in CASA because it was felt that those requirements were better addressed in a pipeline that uses CASA as a basis. In fact a lot of the functionality related to incoherent fringe fitting has been implemented already in the rPICARD pipeline. Requirements specific to geodesy have also been dropped for now as we consider them out of scope for the current devlopment effort at JIVE. These requirements may be brought back at a later stage.

Our intention is to share this document with stakeholders and use their input to prioritize future development efforts. We intend to update this

*Based on an earlier document written by Walter Brisken, Vincent Fish, Jeff Kern and Shep Doeleman

[1] M. Janssen et al, rPICARD: A CASA-based calibration pipeline for VLBI data, A&A 626, A75 (2019)

document with that input.

2 Generic

This section identifies some requirements that are related to the general calibration infrastructure in CASA.

2.1 Interpolation during application of various calibration tables (including, but not limited to fringe fitting) should have the capability of respecting scan boundaries.

2.2 Visibility data should be flagged in a polarization-independent fashion if desired by the user (currently CASA flags all polarization products if one of them ends up being explicitly or implicitly flagged).

2.3 Geometry-dependent calibration (parallactic angle, gaincurve, opacity, etc.) should use the POINTING.DIRECTION instead of FIELD.DELAY_DIR / PHASE_DIR / REFERENCE_DIR.

Question: How does mean vs. apparent position factor into this?

Answer: These two coordinates only differ substantially when pointing near the horizon. For dish antennas the gain curves tend to be very flat at low elevations so this isn't important in that case. However for aperture arrays the effect may be considerable.

2.4 It shall be possibe to add intents to an existing Measurement Set.

3 Amplitude calibration

Amplitude (gain) calibration for VLBI is done quite differently from other types of arrays since typically there are no usable flux calibrators at VLBI resolution. In particular the traditional gain calibration method using the `setjy` task cannot be used for VLBI.

3.1 System temperatures

3.1.1 It shall be possible to use gain constrained self calibration to solve (at least approximately, on bright sources) the appropriate gains on a specified set of antennas to work around cases where system temperature data is either not available or is not reliable for all antennas. The ANTUSE parameter in AIPS tasks CALIB is used for this purpose.

Note: selfcal is available in CASA but there is no equivalent of the ANTUSE parameter.

3.2 Opacity correction

3.2.1 It shall be possible to specify a zenith opacity.

Note: This may already be possible in `gencal`.

3.2.2 It shall be possible to use system temperature measurements made over a specified period as input to solve for zenith opacity.

3.2.2.1 For antennas with characterized spill-over and metrology data, the ground temperature should be estimated and removed before solving for zenith opacity.

3.2.2.2 It shall be possible to specify an elevation range for data to fit.

3.2.2.3 Both least squares and a robust fitting algorithm shall be available.

3.2.2.4 Given zenith opacity, possibly as a function of time, correction for opacity based on sec(z) shall be possible.

Note: This functionality is present in rPICARD. Integrating that code into CASA is probably desirable.

3.3 Antenna gain

3.3.1 Elevation gain curve tables naturally imported with visibility data should be usable.

3.3.2 Application of different elevation gain curves on different sub-bands should be supported. This is especially important in cases where dual-band observing (e.g., S- band and X-band simultaneously) is employed.

Note: This can already be done in CASA because gain curves are realized and resolved per spectral window. There may be ambiguities that need to be resolved though.

3.3.3 It should be possible to plot gain curves as function of elevation.

3.3.4 It should be possible to plot gain curves as function of time (following antenna pointing).

Note: There is a wish to support such a feature generally, plotting the calibration as a function of the relevant interpolation axes in the MS (time, freq). The option of plotting the calibration parameters or their effect (the elements of the Jones matrix) may be desirable.

Currently gain curves are not automatically imported when using the `importfitsidi` task. Instead helper scripts are available to create gain curve tables from GAIN_CURVE tables included in FITS-IDI files or from text files in ANTAB/VLOG format. A challenge here is that gain curves need to be converted from "total power" representation into "voltage" representation, which isn't always possible for (ill-defined) gain curves that aren't positive across the entire elevation range. When applying such a gain curve, visibilities outside the region where the gain curve is valid should be flagged.

3.4 Decorrelation correction

3.4.1 Correlator averaging parameters, possibly as a function of baseline, should be stored with the visibility database.

3.4.2 It shall be possible to optionally correct for decorrelation without requiring an additional pass through the data (on-the-fly application).

3.4.3 This correction shall be based on the cumulative delay/rate calibrations being applied. This correlation should only be applied once as it can't be done properly in a cumulative fashion.

3.5 Calibration editing

3.5.1 It shall be possible to manually edit input calibration data.

Note: The `plotms` and `pltocal` tasks already support manual flagging of calibration tables.

4 Pulse Cal data handling

Phase (or equivalently, pulse) calibration is a VLBI technique used to correct the sampled data for instrumental effects. For example, in geodesy one would like the observations to measure the baseline delay to a fixed point on each antenna, typically the intersection of axes. However, the data are sampled on the ground after passing through cables, connectors, downconverters, and filters. By injecting a series of pulses as close to the front end as is practical, a series of tones is produced in the frequency domain, which can be used to solve for both delay and phase effects between the front end and the samplers.

For example, in the broadband system used for the NASA Space Geodesy Program, pulses are injected at a 5MHz rate, producing tones (or rails) at 5MHz intervals. These tones are extracted in the correlation software and written to a calibration file that accompanies the visibility data, one per antenna. The phase cal data consist of triplets of frequency, amplitude, and phase, tabulated every second.

The fringe-fitting software (e.g., HOPS FOURFIT) finds via FFT the best-fit line of phase as a function of frequency, using all of the tones in a channel or any desired subset thereof (in the case where known RFI corrupts tones). The slope determines a delay, which is then differenced on the baseline and applied to the complex visibility data. The visibilities are also adjusted by the differential phase (at mid-band) of the two fits.

This process allows data that have passed through different anti-aliasing filters and samplers to be registered with one another, thus allowing phase-coherent delay solutions across multiple wide IFs. This technique has been applied with success to group delay extraction over a frequency span of about 6 GHz.

The comb frequency structure causes ambiguities in measured delay; any delay measurement solely determined by a comb with frequency interval Q can only determine delay modulo 1/Q. Resolution of this ambiguity can come from fringe fitting some data. Usually only a very small amount of data for an entire experiment is required as delays typically don't change by more than 10s of nanoseconds and the ambiguities are typically 200 or 1000 nanoseconds. Continuity of delay through time can be used to extend the period of ambiguity resolution.

4.1 Data Structure

4.1.1 Import of Pulse Cal data shall be supported

4.1.1.1 Pulse cal data attached to a FITS file shall be importable.

4.1.1.2 Each pulse cal measurement contains a real and imaginary value and the time interval corresponding to that measurement.

4.1.1.3 A pulse cal set is the collection of all pulse cal measurements made over one time interval at one antenna.

4.1.2 Between 0 and B+1 pulse cal tones per sub-band must be supported where B is sub-band bandwidth in MHz.

4.1.3 A cadence as fast as one pulse cal set per visibility integration time should be supported.

Note: This requirement contradicts the example system given above and probably needs to be changed. Maybe "faster than one pulse cal set per visibility"?

4.1.4 Time averaged pulse cal data should be supported; averaging intervals may be integer multiples of the visibility integration time or not.

4.1.5 An optional "cable cal" value, containing an additional instrumental delay correction, should be handled along with pulse cal data.

4.1.6 Different antennas may have different pulse cal intervals and/or number of tones.

4.1.7 Single precision floating point is sufficient for the real and imaginary parts of each pulse cal measurement; time should be accurately representable to at least 1ms.

4.2 Pulse Cal Data Selection

4.2.1 It should be possible to plot a time series of pulse cal amplitude or phase as a function of time for a selection of tones. Similarly the cablecal values should be plottable.

4.2.2 It should be possible to view the amplitude or phase of the time series of a pulse cal set as a raster image.

4.2.3 It should be possible to flag certain pulse cal values based on a priori information or interactive editing processing; flagged values should be ignored in computations involving the pulse cal data.

4.3 Calculations to perform

4.3.1 It shall be possible to determine the delay as a function of time based on the Pulse Cal data.

4.3.1.1 Solutions should be determined separately for each antenna and separately for each spectral window.

4.3.1.2 It shall be possible to determine a single delay value from multiple subbands.

4.3.2 In cases where cable cal data is present there should be the option to include the cable cal correction in the computed delay.

4.3.3 The determined delays should be stored in a table that can be further edited and applied as necessary.

4.3.4 It shall be possible to time average pulse cal values.

4.3.5 It shall be possible to form a bandpass calibration table based on pulse cal sets.

4.3.6 It shall be possible to form a gain calibration table by extracting the amplitude and/or phase of a single tone of each spectral window.

4.3.7 It shall be possible to use fringe-fit determined delays to resolve pulse cal delay ambiguities.

Note: This has been partly prototyped in Python by Des Small.

5 Fringe fitting

5.1 General Delay Fitting Requirements

5.1.1 It shall be possible to determine and correct cross-polarized delays and phases.

5.1.1.1 It shall be possible to include in the fit a dispersive multi-band delay component proportional to $1/\nu^2$. This is relevant for ionospheric calibration.

Note: Implemented. Should be available in CASA 5.7/6.1.

5.1.1.2 It shall be possible to solve for any combination of delay, dispersive delay, rate and phase, where unsolved parameters are assumed to be zero. This should include the possibility to solve for rate if only a single frequency channel is present.

Note: This is different than zeroing solutions after solving.

Note: Des is of the opinion that excluding the phase makes no sense, Should be available in CASA 5.7/6.1.

5.1.2 It shall be possible to plot fringefit solutions using the `plotms` task.

5.2 Modes of operation

Additonal modes of operation shall be implemented for the `fringefit` task.

5.2.1 It shall be possible to overlap time ranges by a specified amount (see the SOLSUB parameter for AIPS task `FRING`).

5.2.2 In cases where the cadence does not evenly span the data valid period of a scan an intelligent algorithm to shift the intervals shall be invoked.

5.2.3 It shall be possible for delay and rate windows, both center and width, to be specified by the user on a per-antenna basis.

5.2.4 It should be possible to smooth calibration solutions obtained using the CASA `fringefit` task.

5.2.4.1 It should be possible to smooth parameters of these solutions (phase, delay, rate) individually.

5.2.5 It should be possible to combine data by correlation products prior to determination of the delay paramters.

Question: Should this combine just the parallel hands? Stacking cross-hands could be helpful when working on data with poor polarization leakage. Should this be a simple stacking or is this better done as a simultaneous fit? The goal here is additional S/N and Walter Brisken suggests averaging visibilities across polarisations.

5.2.6 It shall be possible to use alternative weighting schemes in the CASA `fringefit` task (see the `WEIGHTIT` parameter for AIPS task `FRING`). At least the standard weighting by $1/\sigma^2$, weighting by $1/\sigma$ and no weighting (weighting by 1) shall be supported.

5.2.7 It shall be possible to use baseline stacking techniques to assist fringe detection.

5.2.8 It shall be possible to combine spectral windows (in order to do a multi-band fit) even if there are large gaps (in frequency) between spectral windows.

5.2.9 It shall be possible to combine spectral windows even if the channels of different spectral windows do not align on a single frequency grid.

Note: A possible approach has been prototyped in Python by Des Small.

5.2.10 An implementation of Fringe-rate mapping shall be provided.

Note: This is required for maser sources where it is hard/impossible to get a good starting postion. The algorithm implemented in AIPS is sufficient but cumbersome to use. More advanced algorithms exists but may be out of scope.

6 Polarization

The following requirements relate to the handling of data correlated with full polarization (i.e., all 4 polarization products).

6.1 Polatization calibration

6.1.1 It must be possible to determine and correct for antenna polarization leakages including using *resolved* polarization calibrators.

Note: Prototype implementation exists (by Ivan Marti-Vidal) in (mostly) Python.

6.1.2 It must be possible to apply the (co)-parallactic angle correction for X-Y mounts. This is necessary to support calibration of data that includes the Hobart 26m antenna which is part of the Australian Long Baseline Array (LBA).

Note: Implementation (for casacore) exists but has not been verified yet.

6.2 Calibration

6.2.1 It shall be possible to transfer delay calibration information across polarizations to increase coherence time:

6.2.1.1 It shall be possible to use fringe solutions in one polarization (e.g., RR) to assist in fringe detection in another polarization (e.g., RL) via baseline stacking techniques.

6.2.1.2 It shall be possible to use the visibility phases as a function of time on a baseline in one polarization to attempt to extend the coherence time in another polarization.

7 Model accountability and manipulation

Note: The original introduction here scetched an ambition to implement geodetic capabilities into CASA to allow absolute astrometry. This is not the ambition of the current work that is being done. However some of these aspects are relevant for good relative astrometry support as well.

7.1 Delay model propagation

7.1.1 Throughout processing the delay model shall be stored in conjunction with the data, further the delay model shall be kept consistent with the current state of the data.

7.1.2 Delay model versioning must be maintained even after splitting and recombining measurement sets.

7.1.3 Throughout processing delays shall be represented to at least1 femtosecond resolution.

7.1.4 It shall be possible to maintain separation of delay effects (i.e., vacuum propagation, atmospheric terms, etc.) within the delay model.

7.1.5 It must be possible to store the delay model as a polynomial spline. Polynomials with up to 6 terms must be supported. The interval of validity of a specific polynomial must be stored and can range from 1 second to 1 hour. A given polynomial must be identified with a particular source.

7.1.6 The delay model used during correlation shall be imported with the visibility data..

7.1.7 All calibrations containing a delay shall modify the delay model table to ensure consistence with the data, in particular the total delay shall be unchanged.

7.1.8 During any operation where a new visibility database is formed, a delay model table consistent with the state of the new database shall be written

Note: Some of these requirements are not compatible with the way CASA operates and/or fundamental design principles of the MeasurementSet. It is not unlikely that these requirements will need adjustment. These requirements are maintained for now as some of them will be needed to implement calibrations mentioned in the next section.

7.2 Delay model adjustments

Earth Orientation Parameters (EOPs) are used to describe the orientation and spin phase of the earth relative to a standard model of a uniformly spinning orb. The deviations from uniform motion are unpredictable as they are largely driven by transfer of angular momentum between the earth's crust, the oceans, atmosphere and earth core. Typically final best estimates for the EOPs are only available a week or two after observation, which may be after correlation.

7.2.1 It shall be possible to apply delay corrections to the data with improved EOPs[2].

Note: Current thinking is that it would be better to recalculate the model using updated EOPs instead of attempting to do a differential correction based on the difference in EOPs

7.2.2 A mechanism for automatically updating EOP data should be implemented.

Note: VLBI correlators tend to use a different EOP data product than the IERS EOPs that are already present in the CASA data repository.

7.2.3 Source position adjustment should be possible.

7.2.3.1 New source positions shall be specified in J2000 frame coordinates.

7.2.3.2 The applied correction shall produce phases equivalent to those that would come from. correlation with the new source coordinates.

7.2.3.3 The coordinates of the correlation center in the data set shall be updated.

7.2.3.4 The baseline vectors (U,V,Ws) shall be updated to be consistent with the new correlation center.

7.2.3.5 Source position corrections up to 1 arcminute should be supported.

[2]The equivalent functionality is implemented in AIPS task `CLCOR` when using `OPCODE = 'EOPS'`.

Note: While this functionality is in principle implemented in the `fixvis` task, there is a strong suspicion that the underlying model calculations may not be accurate enough.

7.3 Delay model replacement

7.3.1 It shall be possible to derive a delay correction from the difference between the current delay model table and an external delay model.

8 Miscellaneous requirements

8.1 Ionospheric Correction

8.1.1 It shall be possible to derive a dispersive delay calibration table from external ionosphere data. (Equivalent to AIPS task `TECOR`)

8.1.2 External ionosphere data in IONEX format should be downloaded on demand from cddis.gsfc.nasa.gov. (For reference implementation, see AIPS task `TECOR`)

A Completed Requirements

A.1 Amplitude calibration

A.1.1 Autocorrelation corrections

A.1.1.1 A facility to divide cross-correlation values by the geometric mean of the associated and time- coincident autocorrelations is - required.

Note: implemented by accor.

A.1.1.2 A mechanism to plot the time variability of sub-band average autocorrelations would be useful.

Note plotms can do this already.

A.1.2 System temperature

A.1.2.1 System temperature tables imported with visibility data shall be able to be applied to the data and weights.

Note: implemented in gaincal.

A.1.2.2 It should be possible to import system temperature data in the standard "TSM" format (See AIPS task ANTAB)

Note: CASA expects this data to be present in the SYSTEM_TEMPERATURE table of the FITS-IDI file upon import. Tools exist to append this data to the FITS-IDI file. These tools accept both "standard" AIPS ANTAB format and VLBA VLOG format.

A.1.3 Digital corrections

A.1.3.1 It shall be possible to apply the Van Vleck (and its ¿1-bit equivalents) to the visibility data.

Note: implemented in importfitsidi

A.2 Fringe Fitting

A.2.1 General Delay Fitting Requirements

A.2.1.1 It shall be possible to determine singleband delay, multiband delay, and rate solutions on a single baseline.

Note: While this requirement is met, more work may be necessary to make its use practical as HOPS-style baseline-based fringe fitting isn't fully implemented.

A.2.1.2 It shall be possible to determine global antenna-based single-band delay, multiband delay, and rate solutions on an array of baselines or subset thereof.

Note: implemented in fringefit.

A.2.2 Selection

Note: Full power of CASA's data selection interface is availabe in fringeit.

A.3 Modes of operation

A.3.0.1 It must be possible to specify a source model for fringe fitting.

Note: Generic CASA mechanism to divide out a model is available in fringefit.

A.3.0.2 It must be possible to specify a desired solution cadence (equivalent to the SOLINT parameter in the AIPS task FRING).

Note: implemented by the solint parameter in fringefit.

A.3.0.3 It must be possible to request a single fringe solution on each and every scan.

Note: implemented in fringefit though generic CASA data selection.

A.3.0.4 It must be possible to request fringe solutions spanning more than one scan.

Note: implemented in fringefit though generic CASA data selection.

A.3.0.5 It should be possible to combine data by IF and/or polarization prior to determination of the delay paramters.

Note: implemented in fringefit through generic CASA calibration infrastructure. Not sure if/how combining by polarization works.

A.3.0.6 It shall be possible for the user to specify delay and rate windows in which to search.

Note: Implemented in fringefit, but parameters are not per-antenna.

A.3.0.7 It must be possible for the user to specify the minimum SNR of acceptable solutions with a sensible default value.

Note: Implemented in fringefit.

A.4 Polarization

A.4.1 Polarization Calibration

A.4.1.1 It must be possible to correct observed visibility phases for field rotation angle (parallactic angle and elevation angle) effects.

Note: Implemented through parang parameter in various calibration tasks. Support for Nasmyth mounts has been implemented. X-Y mount still missing.

A.4.2 Polarization Plotting

Note: Implemented in plotms.

A.4.3 Polarization Basis Conversion

Note: Implemented in Ivan Mart-Vidal's polconvert package.

A.5 Model accountability and manipulation

A.5.1 Delay model adjustments

A.5.1.1 It shall be possible to correct for antenna position errors.

Note: Implemented in the `gencal` task.

A.5.1.1.1 New antenna position shall be specified as new ITRF frame coordinates for one or more antennas.

A.5.1.1.2 Antenna position corrections up to 10 meters should be supported.

A.5.1.1.3 The antenna position as reported in the antenna table shall be updated.

Question: Are the positions actually updated?

A.5.1.2 Manual delay and rate adjustment shall be supported

Note: This is certainly possible by creating calibration tables using the Python interfaces. Support in the `gencal` task may be desirable.

A.6 Data export

A.6.1 Data

A.6.1.1 It shall be possible to export data in AIPS FITS (UVFITS) format.

Note: Implemented in exportuvfits.

A.6.1.2 It should be possible to export data in a well-defined text format.

Note: CASA tables can be written out into text format using a python interface.

A.6.2 Calibration

A.6.2.1 It should be possible to export calibration information in a well-defined text format.

B Dropped Requirements

B.1 Data Import

B.1.1 Data selection

B.1.1 Selection during import shall be supported. Where appropraite data selection should apply to calibration tables as well as visibility data.

Rationale: Data selection during import is not something that CASA supports in general (with the exception of selecting scans in importasdm). CASA has powerfull tools for data selection at various other stages. For example, the `mstransform` task can be used to create a Measurement Set based on a subset of the data.

B.1.2 Precision and data dimensions

Rationale: CASA meets all these requirements, although subarray support is largely implicit.

B.2 Amplitude calibration

B.2.1 Primary beam correction

B.2.1.1 A correction for the antenna primary beam based on the vector offset between the antenna pointing center and the correlator phase center shall be possible.

B.2.1.2 It should be possible to correct for the primary beam without detailed user input, either through built in models or through automated download.

B.2.1.3 It should be possible for a user to supply a detailed primary beam model tabulated in two dimensions.

B.2.1.4 It should be possible for a user to supply a simple radial polynomial model.

B.2.1.5 Primary beam corrections for phased arrays shall be possible.

- B.2.1.5.1 Input parameters describing a general elliptical Bessel or Gaussian zenith beam shall be supported.
- B.2.1.5.2 This zenith beam shall be appropriately stretched as a function of time to account for fore-shortening of the array as seen by the source.
- B.2.1.5.3 The beam parameters should be scaled in angle with observing wavelength during application.

Note: This is probably best implemented in a pipeline such as rPICARD based on the building blocks already provided by CASA.

B.2.2 Autocorrelation template fitting

Was used in the past to measure antenna gains by comparing autocorrelation spectra on masters. Not clear if this is still used.

B.3 Fringe Fitting

B.3.1 Modes of operation

B.3.1.1

lowpriority

It shall be possible to fit delay as a spline function over a period of time possibly greater than the coherence time.

- B.3.1.1.1 The user shall have control over the spline degrees of freedom.
- B.3.1.1.2 Continuity of the spline across scan boundaries must be selectable.
- B.3.1.1.3 It should be possible to construct a spline solution based on tabulated solutions.

Note: Short coherence time algorithms have been implemented in rPICARD.

B.3.2 Visualization

Considered out of scope of the current effort. Probably should be done as a seprate tool, potentially as an extension to jiveplot.

B.4 Short coherence time algorithms

Note: Short coherence time algorithms have been implemented in rPICARD.

B.5 Data export

B.5.1 Data

B.5.1.1 It must be possible to export data, including visibilities and closure quantities.

B.5.1.2 It shall be possible to export data in OIFITS format.

B.5.1.3 It shall be possible to export data in Mark4 format.