ESO "Advanced Study for Upgrades of the Atacama Large Millimeter/submillimeter Array (ALMA)"
(CFP/ESO/16/11115/OSZ)

High-cadence Imaging of the Sun

Final study report review data package - DS4

Technical documentation of the developed software packages

REVISED VERSION
July 31, 2023

(Original version: January 30, 2023)

Prepared by	Sven Wedemeyer Mikołaj Szydlarski	University of Oslo, Norway (UiO)
	M. Carmen Toribio Tobia D. Carozzi	Nordic ARC Node, Onsala Space Observatory (OSO), Sweden
	Contributions to the version presented at the mid-project review	
	Daniel Jakobsson Juan Camilo Guevara Gómez Henrik Eklund Vasco M. J. Henriques Shahin Jafarzadeh	University of Oslo, Norway (UiO)
Validated by		
Approved by		

Acknowledgements This work is supported by the SolarALMA project, which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 682462, until September 2021), and by the Research Council of Norway through its Centres of Excellence scheme, project number 262622. The study team acknowledges support from the Nordic ALMA Regional Centre (ARC) node based at Onsala Space Observatory. The Nordic ARC node is funded through Swedish Research Council grant No 2017-00648.

Contents

1. Overview

1.1 Context

The main software packages described in this document are developed with the aim to optimise the processing of data obtained by observing the Sun with the Atacama Large Millimeter/submillimeter Array [1] (ALMA). For that purpose, simulation tools are developed and employed that enable a quantitative assessment of the quality of processed images and thus the optimisation of the imaging procedure. The latter is essential in order to take full advantage of ALMA's capabilities for solar observing, which includes unprecedented spatial and temporal resolution in the millimeter wavelength range. The Sun is a particularly challenging target for the following main reasons:

- The angular extent of the Sun on the sky is much larger than ALMA's primary beam. The primary beam of ALMA thus gets filled with a complex emission pattern covering a large range of spatial scales, which complicates the calibration and sub-sequent imaging.

- The small-scale structure of the Sun evolves on extremely short time scales on the order of seconds and below. Fully exploiting ALMA's possibilities therefore requires imaging at very high cadence.

- ALMA has entered a domain in terms of spatial and temporal scales that bears great scientific potential for observing the Sun but also causes new technical challenges, which need to be overcome. Daytime observing of the Sun with ALMA at high cadence (currently 1 s) results in sparse sampling of the u-v space and also becomes more susceptible to variations in "seeing" conditions induced by the Earth's atmosphere. Such effects are less important when observing at lower angular resolution, as typical in solar radio astronomy before, but are essential when dealing with the highly dynamic and intermittent nature of the solar atmosphere at small spatial and temporal scales as with ALMA. Currently, solar ALMA observations are carried

[1] See the ALMA Technical Handbook for details of the solar observing mode:
`https://almascience.org/documents-and-tools/latest/alma-technical-handbook`

out for continuum only as this is favourable in view of the sparse instantaneous u-v sampling in terms of a better signal-to-noise ratio.

A direct result of these differences and challenges is that solar observing produces data that are to this date not processed with the ALMA Science Pipeline [2]. At least until Cycle 8, the solar observing mode is qualified by the ALMA Observatory as a non-standard observing mode and thus requires special attention. PIs of solar observing projects are provided with calibrated data and a `scriptForPI.py` that produces one reference image. This image, however, is derived from the visibilities across the whole measurement set and thus the whole observing period, resulting in a time-average map with very limited scientific value. The PIs have therefore to take own initiative to process the data to derive image time series appropriate for their proposed and approved scientific goal. As this is challenging for many PIs who are new to millimeter astronomy and interferometry and also as solar observing with ALMA in itself is still novel in many aspects, the solar/stellar ALMA group at the Rosseland Centre for Solar Physics at the University of Oslo, Norway, has developed the Solar ALMA Pipeline (SoAP) in collaboration with the international solar development group and supported by a grant by the European Research Council[3]. Data processed with SoAP are shared with the solar community in form of the Solar ALMA Science Archive (SALSA[4]). The software tools here described are essential for optimising SoAP and thus for ensuring the best possible quality of scientific data derived from ALMA observations of the Sun.

[2] See the ALMA Cycle 10 Technical Handbook: `https://almascience.org/documents-and-tools/cycle10/alma-technical-handbook`

[3] Project SolarALMA, grant 682462, 2016-2021.

[4] `http://sdc.uio.no/salsa/`, see also Henriques et al. (2022).

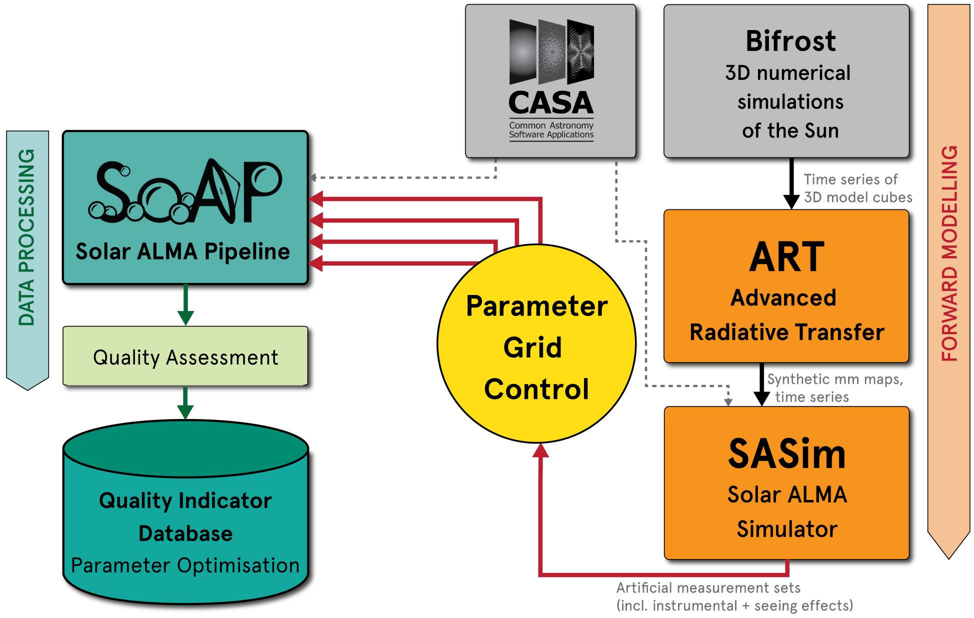


Figure 1.1: Software packages and data used for the study.

1.2 Software packages

- The **Solar ALMA Simulator (SASIM)** is the main software package developed for this study. SASIM takes time series of synthetic brightness temperature maps as input and applies instrumental and weather (or "seeing") effects. The output is a simulated measurement set. See Sect. 2.

- The **Solar ALMA Pipeline (SOAP)** is the main tool used for the reconstruction of time series of brightness temperature maps. SOAP accepts as input measurement sets as downloaded from the ALMA Science Archive but also simulated measurement sets produced with SASIM. See Sect. 3.

- The stellar atmosphere simulation code **BIFROST** (Gudiksen et al. 2011) is used for the production of time sequences of 3D solar model atmospheres. See Sect. 5.1.

- The **Advanced Radiative Transfer (ART) code** (De La Cruz Rodríguez et al. 2021) calculates synthetic brightness temperature maps that are used as input for SASIM. See Sect. 5.2.

- The **Parameter Grid Control**[5] is a simple python script that takes the simulated measurement sets produced with SASIM and starts SOAP runs for a user-defined grid of CLEAN parameters.

- The **Imaging Quality Assessment** consists of a suite of python files that read SOAP output files (i.e., brightness temperature maps), constructs reference models from the ART brightness temperature maps and then calculate Imaging Quality Indicators and save them in a collected database file. Additional IDL scripts have been used for the analysis and production of figures (as seen in the *High-cadence Report*, Wedemeyer et al. 2023). Sect. 4.

[5]Due to its simplicity, no separate description is provided.

2. Solar ALMA Simulator (SASIM)

2.1 Overall design

The Solar ALMA Simulator utilises the CASA[1] task ***simobserve()***[2] for the simulation of measurements sets that correspond to solar observations with ALMA. The resulting data are based on state-of-the-art time-dependent 3D simulations that provide a realistic test case for observing the Sun with ALMA. Please note that SASIM uses CASA in the form of a package called from within an own python (3.7) environment[3].

SASIM involves the following steps:

Stage	Description
0	Preparation of input files
1	Construction of (uncorrupted) measurement set
2	Corruption of measurement set
3	Automatic SOAP processing
4	Quick Visualisation and Analysis
5	Construction of reference maps

[1]The Common Astronomy Software Applications (CASA) package is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and it is available at `https://casa.nrao.edu/`

[2]`https://casadocs.readthedocs.io/en/stable/api/tt/casatasks.simulation.simobserve.html#casatasks.simulation.simobserve`

[3]The ***settrop()*** function of the simulator tool as available in CASA during the course of this study was not working properly with the short time steps needed here. A patched version was provided by D. Petry (ESO) and used for corrupting measurement sets for this study. The new version will eventually be part of official CASA releases so that SASIM can be used in connection with these future releases.

Name	Type	Description
`sasim_prep_fits_input.pro`	routine	Reads in ART FITS files with brightness temperature maps and outputs FITS files with 4D cubes that can be used with SASIM
`sasim_make_fits_header.pro`	function	Creates standard FITS headers.
`sasim_unit_conversion.pro`	function	Converts units of data, e.g. from brightness temperature in K to flux in Jy/pixel.

Table 2.1: IDL routines used for the conversion from ART brightness temperature maps to SASIM input files.

2.2 Stage 0 - Input preparation

Examples for the original maps produced with ART are shown in the top row of Fig. 2.1. The ART maps, which are available in HDF5 format, are converted to SASIM input files with the IDL[4] routine `sasim_prep_fits_input.pro`. This step is necessary in order to comply with the input format expected by relevant CASA routines such as ***simobserve()***. The latter expects 4D cubes with the dimensions $[n_x, n_y, n_{\text{stokes}}, n_\nu]$ being the two horizontal spatial dimensions, the Stokes dimension, and the frequency dimension. Each resulting FITS file contains one 4D cube for one spectral window (sub-band) with n_ν spectral channels[5]. In the current version of SASIM, only Stokes I is considered (n_{stokes}) but an extension to all four Stokes components will be considered for future releases.

The brightness temperature in units of Kelvin as read from the ART FITS files is converted to spectral flux density in units of Jansky/pixel. The pixel size is set by the grid cell size in the original 3D model, which is propagated as pixel size to the ART maps. However, it is generally possible to interpolate the input models to custom grid cell/pixel sizes.

The horizontal extent of the 3D computational domain of the default solar atmosphere model (Carlsson et al. 2016), i.e. the 3D BIFROST model seen from the top and thus the extent of the corresponding ART maps, is 33.3” × 33.3”. This size is just enough to contain the Band 6 primary beam (with 25.3” diameter - Full Width Half Power, FWHP) but too small for the Band 3 primary beam (with 58.3” FWHP). In order to ensure enough margin and to account for the outer range of the primary beam, the ART maps are periodically repeated. This is possible due to the periodic boundary conditions of the solar 3D model. For Band 3, 5×5 tiles are chosen while 3×3 tiles are sufficient for Band 6. Examples for the resulting sky models are shown in the middle row of Fig. 2.1. Please note that this procedure is to be adjusted to the primary beam size of the selected receiver band in comparison to the extent of the numerical model. The size of appropriate models is set by the requirements and limitations of computationally expensive 3D simulations (see Sect. 5.2) but, in principle, models of any extent can be used as input for SASIM.

[4]Interactive Data Language: `https://www.l3harrisgeospatial.com/Software-Technology/IDL`

[5]Please note that the number of spectral channels can be freely chosen, e.g. = $n_\nu = 128$ to include all spectral channels per band as currently offered by ALMA for solar observing.

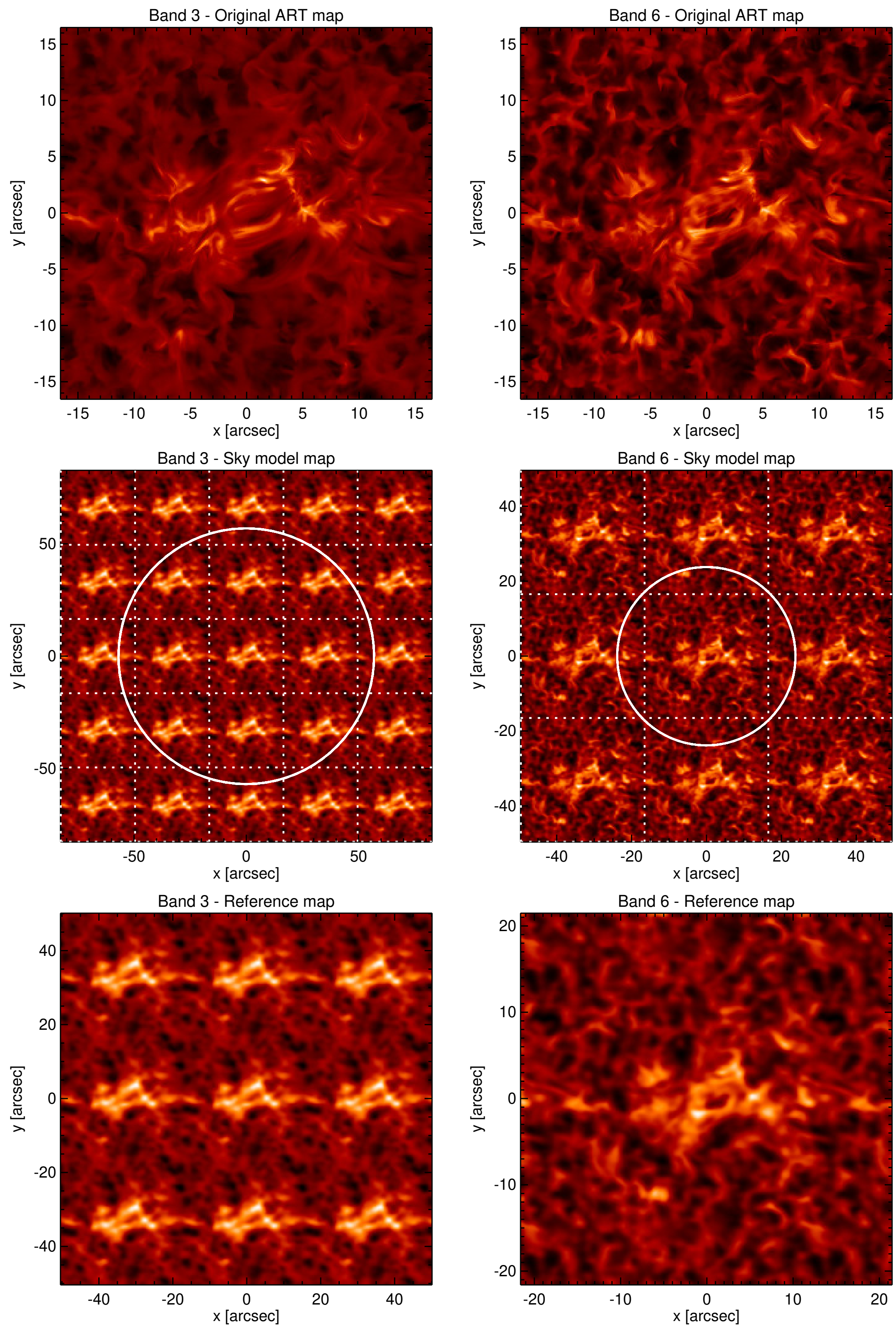


Figure 2.1: *Top:* Original (frequency-averaged) ART maps for Band 3 (left column) and Band 6 (right column). *Middle:* Sky model used in SASIM as constructed from periodically repeating the maps in the top row. *Bottom:* Reference model as derived from the sky model after being convolved with a synthesised beam (see text for details) and clipped to a size sufficient for the final CLEANed maps. The tiles and the primary beams of the 12 m antenna (FWHM) are marked with dotted and solid white lines, respectively, in the middle row. Please note the different axes for the individual panels.

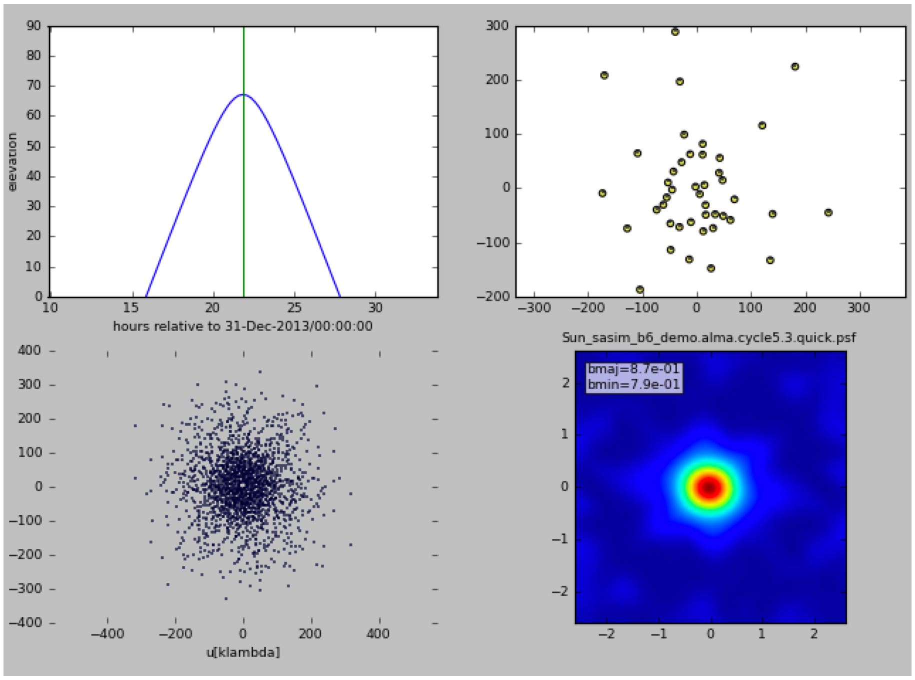


Figure 2.2: Graphics output of ***simobserve()*** during the execution of SASIM for an uncorrupted measurement set for Band 3. *Top left:* Elevation on the sky. *Top right:* Array configuration (axes in units of m). The antenna diameters are plotted to scale. *Bottom left:* u-v space. *Bottom right:* Synthesised dirty beam.

In summary, the following files are produced in this stage:

- **Science target maps:** Four SASIM input FITS files (one for each spectral window as used for the receiver setup of ALMA's solar observing mode) are produced for each simulation time step for a given receiver band. This results in 240 files for each minute of simulation time at 1 s cadence and a total data volume of several 100 GB depending on the receiver band and thus the necessary number of periodic tile repetitions.
- **Initial reference model maps:** One frequency-averaged (i.e. over all 4×128 spectral channels) map for each time step at the original spatial resolution of the numerical model. See Sect. 2.6 for the reference maps that can be constructed based on these initial maps.
- **Calibrator map:** One artificial calibrator map that has the same dimensions and pixel size as the science target maps but only contains a point source in the centre. This is simulated by setting the central pixel to a flux as prescribed in the SASIM parameter file in units of Jansky and all others to zero.

2.3 Stage 1 — Construction of uncorrupted measurement set

In this first stage, SASIM constructs an *uncorrupted* measurement set (MS), meaning that no atmospheric/instrumental corruptions are incorporated. The MS is build up time step by time step.

For each step, a loop over all spectral windows (typically 4) is carried out during which the correct SASIM input FITS file is loaded. Please note that the angular resolution of the input data is set by the pixel grid of the ART maps as described in Sect. 2.2. See also Fig. 2.1 for an example with an angular resolution of 0.066". SASIM supports a combination of two separate arrays such as the 12m-Array and the ACA or alternatively a combined array as defined in an user-provided array configuration file. ***simobserve()*** is then called individually for each spectral window and each array component (one or two). For each of these subsets, interferometric visibilities are and calculated for the current skymodel as defined by the input files. The resulting individual MSs for the current time steps are then inserted into the main MS by using the CASA task ***concat***[6] so that the final MS contains the visibilities for all spectral channels and all time steps. In an additional step, SASIM ensures that all parts of the MS have the correct time stamps, the same scan ID, and the same observation ID. For this purpose, SASIM data of the final MS is edited with the help of the table tool[4] (`tb`) and `msmetadata`. The resulting measurement set then represents an observation at ideal (and thus unrealistic) conditions corresponding to a precipitable water vapour (PWV) level of 0 mm. See Sect. 2.7 for parameters that a user can define to change the properties of this artificial observing process. It should be noted that neither the brightness temperature maps for the spectral channels nor the reference model maps are multiplied by the primary beam prior to being used as input for SASIM. Application of the primary beam occurs as part of ***simobserve()*** when it is called during the execution of stage 1 of SASIM. Additional output (via ***plotms***) for the simulation of the 12m Array and the ACA can be displayed, too. See Figure 2.2 for an example.

In addition, the procedure is repeated with the artificial calibrator map. The same map is used for each time step, simulating an observation of the (static) calibrator for the same duration as for the artificial science target. The result is an artificial MS for the calibrator that has the same dimensions and setup as the MS for the science target. Please note that the resulting simulated MSs correspond to observational MSs *after calibration*. Consequently, no calibration is applied to the simulated MSs produced with SASIM.

2.4 Stage 2 — Measurement set corruption

2.4.1 Atmospheric model

The model used for estimating Earth's atmospheric transmission at mm wavelengths at the ALMA site is called Atmospheric Transmission at Microwaves (ATM) and was developed by Juan R. Pardo et. al. This model is based on spectroscopic absorption parameters complemented with two continuum-terms. One term accounts for collision-induced absorption from dry atmosphere transition dipoles in symmetric molecules and the other for continuum absorption due to water opacity. Coefficients for these additional terms have initially been determined by Fourier Transform Spectroscopy (FTS) at Mauna Kea and updated as new data becomes available (Pardo et al. 2001).

Approximations and assumptions

The Earth's atmosphere is concentrated to $\sim 99.999\%$ of the mass below an altitude of 90 km. Above 90 km considerations of Zeeman splitting of O_2 only needs to be considered (Pardo et al. 1995,

[6] `https://casadocs.readthedocs.io/en/stable/api/tt/casatasks.manipulation.concat.html#casatasks.manipulation.concat`

1998). The pressure and temperature in this range are typically between $\sim 1020 - 0.0015$ mb and $\sim 220 - 320$ K, respectively. At these typical pressures and temperatures, populations of different energy levels are collision-based for a small volume and thus set by the local temperature, allowing for the simplifying assumption of Local Thermodynamic Equilibrium (LTE). With increasing altitude, the collision rate is lowered until, at ~ 80 km, photon pumping dominates and the LTE approximation is no longer valid. This is, however, not a problem since the contribution to the atmospheric opacity in the millimetre/submillimetre range is negligible at those heights. The final altitudes that need to be considered when calculating the atmospheric opacity in the millimetre/submillimetre range for ALMA are therefore ~ 5 km to ~ 80 km (Pardo et al. 2001).

2.4.2 Corruption with the simulator tool

The SIMULATOR TOOL[7] (SM) is a built-in CASA utility, which allows for advanced corrupting and distortion of a MS by considering degrading effects due to atmospheric disturbance and other types of noise such as pointing errors or polarization leakage etc.. The SIMULATOR TOOL can take an existing MS or generate one when called from within ***simobserve()***. The implementation of the phase corruption is done with calibration tables (a Jones matrix) in the measurement equation formalism. Please refer to the following guide for more information: `https://safe.nrao.edu/wiki/pub/ALMA/SimulatorCookbook/corruptguide.pdf`. Examples of measurement sets that have been corrupted with SASIM are illustrated with XX and YY phases as function of time for selected antenna baselines in Fig. 2.3 for Band 3 and in Fig. 2.4 for Band 6, respectively.

Simulating the degrading effects of the Earth's atmosphere with *settrop*

In SASIM, the MS for the science target and subsequently the MS for the calibrator (see Sect. 2.3) are corrupted with the CASA function ***settrop*** for the exact same set of parameters (see Table 2.2). The output of this stage consists of new but corrupted MSs for the science target and the calibrator. The user can provide a name set with the input parameter `corruptset` in order not to overwrite the uncorrupted MS.

For modelling the impact of the Earth's atmosphere on an observation with ALMA, the atmosphere can be described as a combination of a dry and a wet component. While the dry component is usually well mixed, it is not the case for the wet component, i.e. the component containing water vapour. The poor mixing of the wet atmosphere component results in pockets with different PWV levels in which the phase noise dominates over the thermal noise (Nikolic et al. 2013). The CASA function ***settrop*** simulates these water vapour differences as a fluctuating phase screen that moves (driven by wind) over the array and thus decorrelates the signal. In real observations, the resulting phase differences typically increase as a function of baseline although this effect can be substantially corrected for during the data calibration stage.

For simulating the above described atmospheric effects, the *"screen"* mode of ***settrop*** is used. The phase screen is in practice a fluctuating PWV screen that is generated with fractional Brownian motion (fBM). The corresponding water vapour differences in the atmosphere are here controlled by setting the root mean square fluctuation as prescribed by the parameter `deltapwv`. In addition, the parameter `beta` sets the exponent of fractional Brownian motion (see below for more details),

[7] `https://casadocs.readthedocs.io/en/stable/api/tt/casatools.simulator.html#casatools.simulator`

while the `windspeed` parameter sets the speed with which the phase screen moves over the array with time. Please see Table 2.2 for typical default values. There is also the option to store the corruption data in a calibration table. The name of this table is set by the parameter `table` (if not specified, the data will only be applied to the MS and not saved). How the phase screen affects an observation in detail as a function of frequency is later calculated with the ATM library and the troposphere model within the library. More information regarding this step can be found at `https://casaguides.nrao.edu/index.php/Corrupt`.

The screen with fBM that is used here to corrupt a measurement set can be understood as a statistical realisation of the turbulent field $q(r)$. This turbulent field should therefore obey the intrinsic structure function $D_q(r)$ that originates in Kolmogorov's turbulence theory as defined as

$$< [q(r') - q(r'+r)]^2 >= D_q(r) = 6.88(\frac{r}{r_0})^{\xi}. \tag{2.1}$$

The factor 6.88 is due to convention to let r_0 be equal to the Fried parameter. This structure function is then related to the ***settrop*** parameter `beta` as

$$\beta = 1 + 2H \tag{2.2}$$

where H is the Hurst parameter. The Hurst parameter determines the *"smoothness"* of the generated field. The exponent ξ in Eq. 2.1 is related to the type of screen that one wants to simulate. This could either be a 3D or 2D turbulent screen. For longer baselines, the screen behaves typically more like a 2D screen and a value of $\xi = \frac{2}{3}$ is used (Nikolic et al. 2007). The fBM follows the structure function defined in Eq. 2.1 by letting `beta` be equal to either $(5-2D)$ or $(7-2D)$ in the generation of the screen[8]. When predicting the atmospheric corruption with the ATM model the fBM will be converted to a complex screen. This complex fBM screen can then be scaled to have desired properties regarding the PWV rms by settting the ***settrop*** parameter `deltaPWV`[9]. This scaling is used to calculate the angle and amplitude on the complex fBM screen where the rms is calculated as

$$rms = \sqrt{< (fBM - < fBM >) * (fBM - < fBM >) >} \tag{2.3}$$

[8]This can be found in the source code at `https://open-bitbucket.nrao.edu/projects/CASA/repos/casa6/browse/casatools/src/code/synthesis/MeasurementComponents/CalCorruptor.cc`.

[9]This information is deduced from the source code found at `https://open-bitbucket.nrao.edu/projects/CASA/repos/casa6/browse/casatools/src/code/synthesis/MeasurementComponents/StandardVisCal.cc`. The implementation of the scaling can be found in row 1234-1247 in the source code linked in the footnote above.

Parameter	Description	Default
`table`	Store corruption data as a table	' '
`pwv`	Total precipitable water vapour in mm	3.0
`deltapwv`	RMS PWV fluctuations *as a fraction of PWV parameter*	0.15
`beta`	Exponent of fractional brownian motion	1.1
`windspeed`	wind speed for screen type corruption (m/s)	7.0

Table 2.2: Controllable parameters for the CASA function ***settrop***.

where fBM is the 2D fractional Brownian Motion screen and $< fBM >$ the mean value. The amplitude and angle are then calculated and scaled in the following way:

$$amp = fBM * \frac{rms}{scale} \tag{2.4}$$

and

$$angle = fBM * \frac{rms}{scale} * \pi. \tag{2.5}$$

More information regarding atmospheric simulations are found in the ALMA memos pointed to in the CASA corruption guide: `https://casaguides.nrao.edu/index.php/Corrupt`.

Please note that the current implementation of ***settrop()*** in CASA corrupts the XX and YY phases in the exact same way and produces thus identical corrupted XX and YY data. As that is unrealistic with XX and YY in real measurement sets being in general different, SASIM allows for adding random noise that results in different XX and YY data. For that purpose, the simulator tool method ***setnoise()*** with `mode=simplenoise` is used. Already a small value for simplenoise of (a few) 0.1 Jy is sufficient to produce a more realistic difference between the XX and YY data.

Phase variations

The above described turbulent motions in the Earth's atmosphere add phase variations to the visibilities of an astronomical target as measured with ALMA. The resulting phase variations over time on selected baselines are illustrated for Band 3 and Band 6 in Figs. 2.3-2.4, respectively. Each figure includes two columns each for a different set of the corruption parameters. Comparing these results clearly shows the different level of phase noise added with ***settrop*** and ***setnoise*** with the phase noise increasing with increasing PWV value. The shown phase data can be retrieved with SASIM for a simulated (and also observational) MS by extracting the baselines and the phases from the MS table columns `UVW` and `CORRECTED_DATA`, respectively, for all simulated spectral windows/channels and baselines (see Sect. 2.5). Please note that ***settrop()*** produces identical XX and YY phases but adding further noise components with ***setnoise*** results in deviations between XX and YY (see Fig. 2.3). We note that real measurement sets indeed show similar differences (see, e.g., Fig. 2.5 in the *Imaging Report* (Wedemeyer et al. 2023)).

As indicated above, the resulting variations of the phase (Φ), which depend on spatial scale due to the nature of the turbulent motions (cf. Eq. 2.1), can be described with the *Spatial phase Structure Function* (SSF) D_Φ. According to Ishizaki & Sakamoto (2005), D_Φ is defined as the mean square phase as a function of baseline length ρ,

$$D_\Phi(\rho) = \langle(\Phi(\rho_0) - \Phi(\rho_0 + \rho))^2\rangle \simeq \sigma_\Phi^2(\rho), \tag{2.6}$$

which can be approximated by the root-mean-square (rms) of the phase fluctuation $\sigma_\Phi(\rho)$ measured as function of baseline length ρ. The SSF can be calculated from the phases data that SASIM extracts for a MS (see Sect. 2.5). For each extracted spectral window of central frequency ν and baseline of length ρ, the rms phase variation $\sigma'_\Phi(\rho, \nu)$ is then derived as the standard deviation of the phases over the simulated time span. As the resulting values are very similar for the spectral windows, here the rms phase variation $\sigma_\Phi(\rho)$ is calculated as the average across the spectral windows. The resulting values for $\sigma_\Phi(\rho)$ are shown in Fig. 2.5 for Band 3 and in Fig. 2.6 for Band 6 for the same corruption

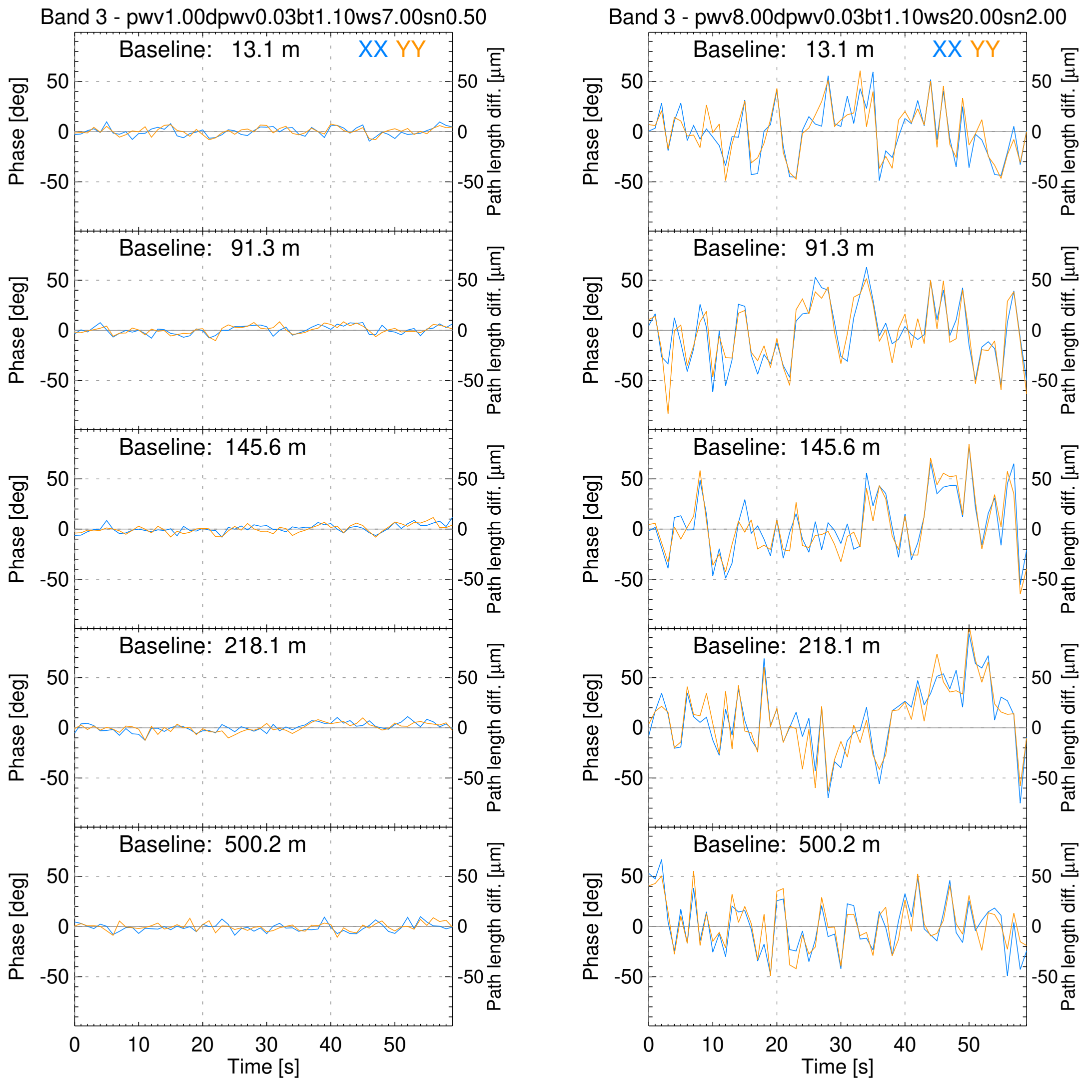


Figure 2.3: Phases in the corrupted measurement set for the artificial calibrator target for Band 3. The two columns show two different cases with the SASIM parameters specified at the top. The rows show the phases as function of time for selected baselines with XX and YY phases plotted as blue and orange lines, respectively. Please note that the phase values for the corresponding uncorrupted measurement set are essentially zero. The parameters used for the MS corruption are shown at the top of the figure: `pwv=1.0, deltapwv=0.03, beta=1.1, windspeed=7.0, simplenoise=0.50`.

parameter combinations as used for Figs. 2.3-2.4 but plotted on linear and log-log scales. Comparing the SSFs clearly shows the different level of phase noise added with ***settrop*** and ***simnoise*** with the phase noise increasing with increasing PWV value. Also the spread of the values increases with the PWV value, whereas the expected increase with baseline is not very prominent in these examples for the compact array configuration 3. We note that in the absence of a clear slope, the simulated SSFs

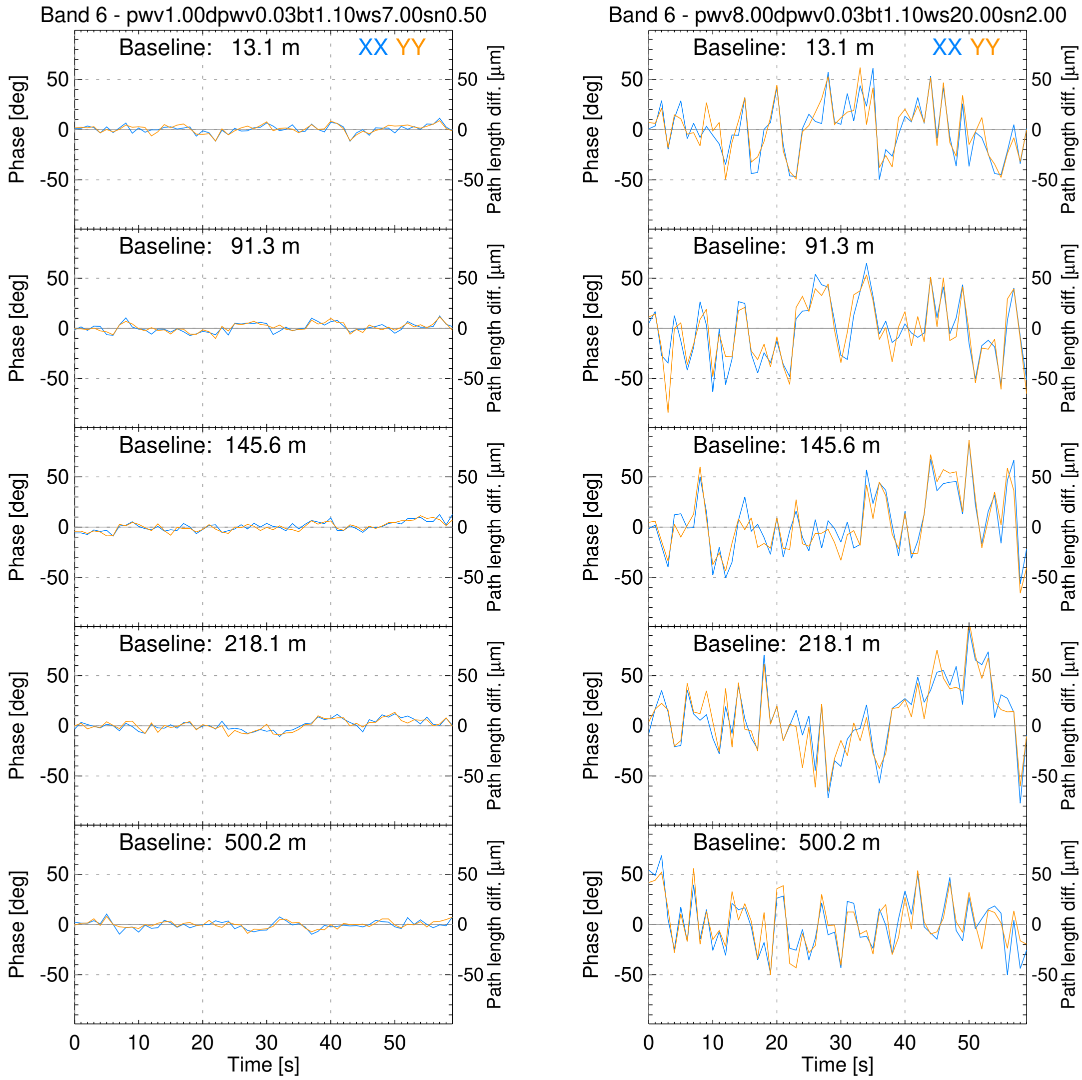


Figure 2.4: Phases in the corrupted measurement set for the artificial calibrator target for Band 6. The two columns show two different cases with the SASIM parameters specified at the top. The rows show the phases as function of time for selected baselines with XX and YY phases plotted as blue and orange lines, respectively. Please note that the phase values for the corresponding uncorrupted measurement set are essentially zero. The parameters used for the MS corruption are shown at the top of the figure: `pwv=1.0, deltapwv=0.03, beta=1.1, windspeed=7.0, simplenoise=0.50.`

can be described well by the median value and the standard deviation of the phase rms values.

For observational (uncalibrated) phase rms data, often a power law with an exponent α to $D_\Phi(\rho)$ (Ishizaki & Sakamoto 2005),

$$\sigma_\Phi(\rho) = \sigma_\Phi(\rho_0)\left(\frac{\rho}{\rho_0}\right)^\alpha, \tag{2.7}$$

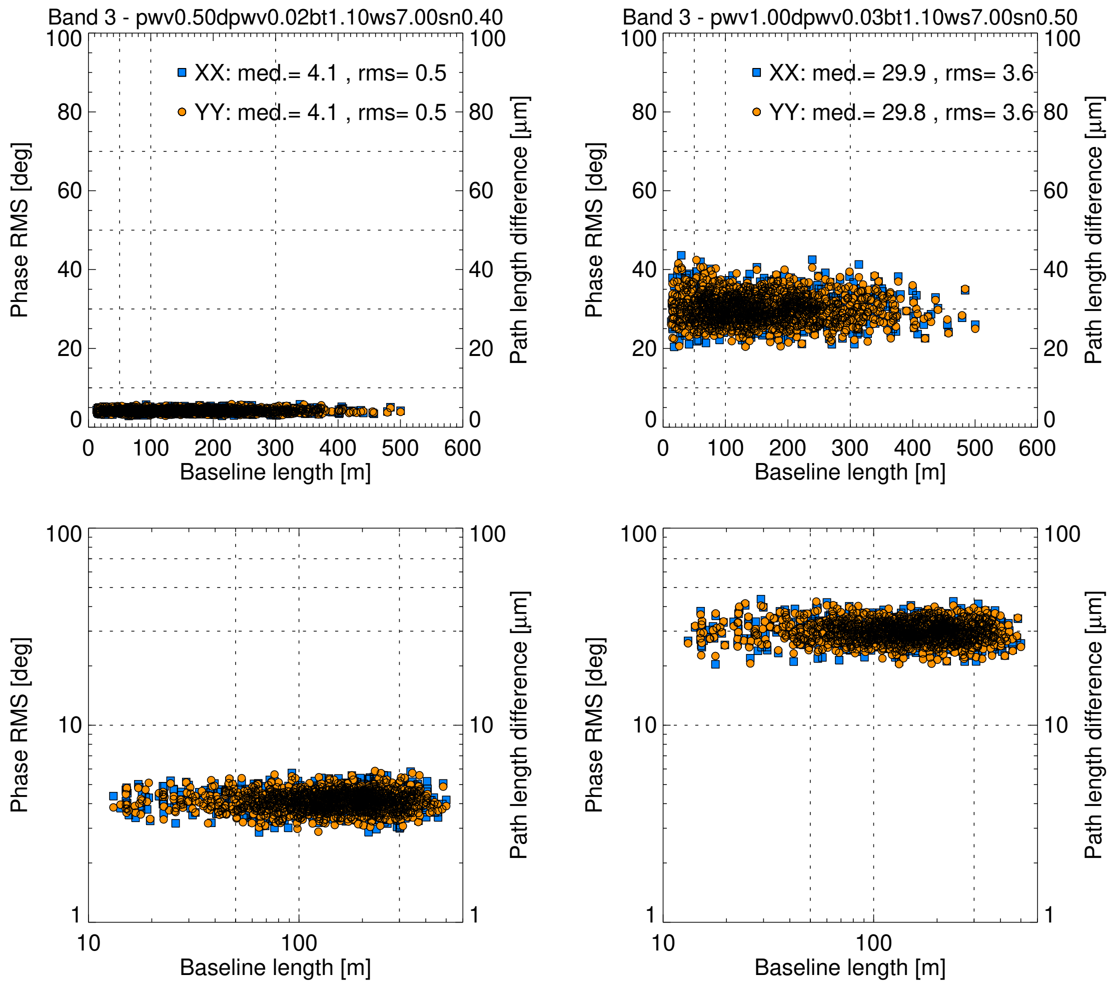


Figure 2.5: Spatial Structure Function for Band 3 (root-mean-square (rms) of the temporal phase variations as function of baseline length) for two corrupted measurement sets in the left and right column. The SASIM parameters are specified at the top. The rows show the same data but on a linear scale (top) and a log-log scale (bottom). The data was calculated by applying the same corruption to the artificial calibrator as for the artificial science target. The XX and YY phases rms values are represented by blue squares (XX) and orange circles (YY), respectively. Please note that the phase rms values for the corresponding uncorrupted measurement set are essentially zero. The parameters used for the MS corruption are shown at the top of the figure: `pwv=0.5, deltapwv=0.02, beta=1.1, windspeed=7.0, simplenoise=0.40`.

is fitted whereas the examples discussed here are flat and do not show trends that could be clearly described with an exponential fit over the relatively short baselines. The main reasons for that are that (i) such slopes become typically notable only for baselines longer than for the compact configuration used for this report and (ii) that the simulated MSs and thus their SSFs correspond to calibrated data for which the dependence on baseline length should have been (substantially) removed. Please see Fig. 2.6 in the *Imaging Report* (Wedemeyer et al. 2023) for the SSF for a real measurement set, exhibiting a slope before calibration. Please note that the spatial phase structure function would be

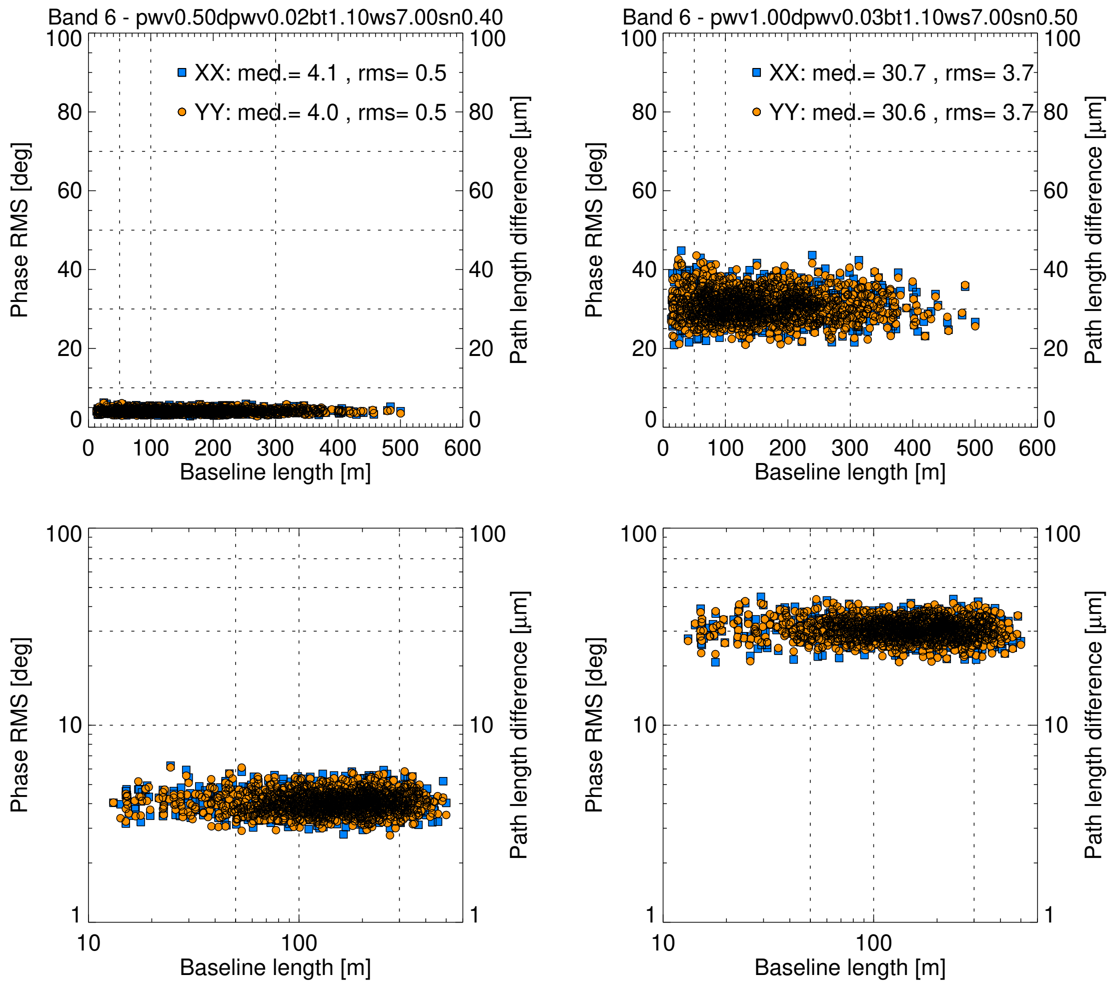


Figure 2.6: Spatial Structure Function for Band 6 (root-mean-square (rms) of the temporal phase variations as function of baseline length) for two corrupted measurement sets in the left and right column. The SASIM parameters are specified at the top. The rows show the same data but on a linear scale (top) and a log-log scale (bottom). The data was calculated by applying the same corruption to the artificial calibrator as for the artificial science target. The XX and YY phases rms values are represented by blue squares (XX) and orange circles (YY), respectively. Please note that the phase rms values for the corresponding uncorrupted measurement set are essentially zero. The parameters used for the MS corruption are shown at the top of the figure: `pwv=0.5, deltapwv=0.02, beta=1.1, windspeed=7.0, simplenoise=0.40.`

calculated as $D_\Phi(\rho) \simeq \sigma^2_\Phi(\rho)$ and thus does not contain information that is not already visible in the provided rms phase plots.

For convenience, the rms phases in Figs. 2.5 and 2.6 are also converted to (frequency-independent) path length differences (see right axis in each panel). The conversion from rms phase ϕ in degree to path length difference is done according to Eq. 5 in Maud et al. (2017),

$$\Phi = \frac{\phi'}{2\pi} \times \frac{c}{\nu_{\rm obs}}, \tag{2.8}$$

where $\phi' = \frac{\pi}{180^o}\phi$ is the rms phase in radians, c is the speed of light, and ν is the observation frequency in Hz. The resulting path length differences are expressed in units of μm (micrometres, microns). The values in the shown examples range from a few microns to a few tens of microns.

Time step

Please note that the version of ***settrop()*** as included in the public CASA releases available at the time of developing SASIM was not capable of handling time series with steps smaller than 10 s. Using such a version of ***settrop()*** only allows the corruption of a MS at 10 s cadence even if the time information in the MS correctly indicates a shorter cadence. D. Petry (ESO) kindly provided an updated version of ***settrop()*** that allows to set shorter time steps via the new additional parameter `simint`. The updated version was introduced in CASA 6.5 [10]

2.5 Stage 3-4 — Test and visualisation

The following steps are optional and can be selected and controlled with the parameters in the SASIM parameter file (see Sect. 2.7). Each step can either be called automatically after the (corrupted) measurement set has been generated or individually and separately.

Automatic SOAP execution (Stage 3)

Once a (un)corrupted MS is produced, SOAP can be started automatically. It will then perform imaging with the parameters specified in SASIM parameter file. While this step is an important test for the simulated MS, it can also be used as a convenient way to start an imaging process with user-specified parameters. Please refer to Sect. 3 for more details regarding SOAP.

Measurement Set – Visualisation and Information

The (interactive) CASA task ***plotms*** allows for a quick and interactive analysis of the MS and can be started automatically after the uncorrupted and/or the corrupted MS has been generated. Please refer to the CASA documentation for more information about ***plotms***. Additionally, ephemeris information, the array configuration, the *uv*-coverage, and the synthesized (dirty) beam can be displayed in the same as way done by the CASA task ***simobserve()*** (see Fig. 2.2).

Phase data extraction

In addition, SASIM can extract phase data that can then be visualised with additional python/IDL routines. For that purpose, the data from the MS table columns `UVW` and `CORRECTED_DATA` for all simulated spectral windows and baselines is retrieved and processed with the help of CASA tools such as `tbtool` and `msmetadata`. The following data products are then stored:

[10] `https://casadocs.readthedocs.io/en/v6.5.2/notebooks/introduction.html#Release-Notes`.

- XX and YY phases as function of time for all baselines. The extracted data is saved in files with names ending on `phase_vs_baseline.dat`.
- XX and YY phase RMS for the time series for each baseline, thus resulting in two values per baseline. The extracted data is saved in files with names ending on `rmsphase.dat` and is used to produce plots of the Spatial Structure Function (SSF).

Imaging

SASIM can (optionally and automatically or separately) start an imaging run with SOAP (see Sect. 3) in order to quickly produce a time series of dirty maps or even full imaging products. The SOAP parameters can be set in the SASIM parameter file. Successful execution of SOAP is a crucial test for any produced measurement set.

Visualisation

Optionally, a movie of the produced time series of images can be produced once the imaging stage is completed. The movie can also be generated for any existing folder with FITS files produced with SOAP.

2.6 Stage 5 — Reference maps for quantitative analysis

For a quantitative evaluation of how well the imaging process performs, a meaningful reference is needed. Such reference maps can be generated based on the initial reference maps that are produced in the preparation stage 0 (see Sect. 2.2) and as shown in Fig. 2.1. It should be noted that the initial maps would represent observations at the same spatial resolution as the initial model and with perfect *uv*-coverage, i.e. not with an interferometric array such as ALMA. Hence, the initial maps are convolved with a synthesised beam as it is produced during the imaging (stage 3) with SOAP and interpolated to the same pixel grid as for the image produced with SOAP. For that purpose, the parameter `robust` is set to a value of -2 as it produces the narrowest beam (cf. Sect. 3). This way, the convolved reference maps can be directly compared to the maps produced with SOAP based on any uncorrupted or corrupted MS. These reference maps still represent a highly idealised case that assumes perfect *uv*-coverage and no atmospheric or instrumental degradation. This approach is chosen as it sets the limit for the degree to which any detail of the sky image can be recovered under perfect conditions with a perfect instrument of the assumed overall (synthesised) aperture. Any detail on spatial scales that cannot be recovered with the given aperture are thus effectively removed as they can never be recovered during the imaging stage and would thus represent an unfair reference.

2.7 Input parameter

All input parameters to SASIM are read from a single configuration file in *YAML*[11] format. An example is shown in Fig. 2.7. The SASIM configuration file has separate sections for each major

[11] *YAML* ("Yet Another Markup Language" or "YAML Ain't Markup Language") is a Unicode-based human-readable data-serialization language. Its syntax is independent of a specific programming language, and it is designed to work well with modern programming languages like Python.

task of the pipeline, e.g., `soap` for SOAP or `sim` to control SIMULATOR TOOL. A default value is used for variables that are not specified by the user or set to `none` or `null`.

Names and directories

Among the basic parameters are the `project` name that will be the base of names of output files and the name extension `corruptset` added for corrupted measurement sets, the output directory (`path_output`), and paths and names for the needed files such as the pointing file (`ptgfile`) and array configuration files (`config12m`, `configaca`).

Sky model and instrumental set-up

SASIM can handle all ALMA receiver bands. The primary selection is done via the parameter `band`.

Please make sure that the correct input maps for the wanted band are chosen by updating the input directory via the parameter `artmodel`. Simulations can be done for receiver bands that have not yet been commissioned for solar observing but the frequencies are not predefined in those cases. The user will have to choose the wanted frequencies (via the ART input files).
The spectral set-up of a receiver band is specified via the ART input files. The default for currently commissioned bands is 4 sub-bands with 128 spectral channels each. The current version of SASIM handles the input data per sub-band.

SASIM allows for simulating different antenna configurations. The wanted configurations for the 12-m Array and ACA can be set by editing the following parameters:

Parameter	Function	Default
`config12m`	Configuration file for the ALMA 12m. Array	’alma.cycle5.3.cfg’
`configaca`	Configuration file for the ACA.	’aca.cycle5.cfg’
`configtp`	Configuration file for the TP antennas.	’none’

The shown configurations are used as default as they are commonly used for solar observations. The parameters then point at the configuration files (in this example alma.cycle5.3.cfg and aca.cycle5.cfg). Alternative configuration files can be retrieved here for Cycle 1-8:
`https://almascience.org/tools/casa-simulator`
SASIM thus allows also for the simulation of array configurations that are not yet commissioned for solar observing. For the results described in the *Imaging Report* (Wedemeyer et al. 2023), all antennas were set to a diameter of 12 m and provided in a combined antenna configuration file as specified with the parameter `config12m`. The parameter `configaca` is set to `none` in this case. Please note that the TP option is not yet available in the current version.

Measurement set

Parameter	Function	Default
`scannr_sci`	Scan number of science scan in final MS.	1
`scannr_cal`	Scan number of calibrator scan in final MS.	2
`obsid`	Observation ID in final MS.	0

```yaml
sim:
  # --- Output file names ---
  project: SASim_Sun_b3_12m_4ch_60s_dt100ms_cal
  corruptset: pwv0.50dpwv0.30bt1.10ws10.00sn0.40
  # --- Program control ---
  simulate: False
  reconcatenate: False
  corrupt: True
  graphics: False
  # --- sky (input) model ---
  artmodel: art_bifrost_en024048_b03
  mapsize: auto
  simtime: [0.0,59.9] # --- sim. time range to consider
  timestep: 0.1 # in seconds
  date:       "2020/01/01"
  timestart: "16:00:00" # Start of observation in UT [HH:MM:SS]
  timenoon:  "16:36:00" # Time of local noon in UT [HH:MM:SS]
  direction: "18h46m46.2 -23d22m48.5"
  ptgfile:     # Created automatically if left empty
  setpeakflux: '7.0Jy/pixel'
  # --- Instrumental setup to be simulated ---
  band: 3
  config12m: alma12mACA12m.cycle5.3.cfg
  configaca: none
 # --- Paths, data, scripts ---
  path_output: output # --- automatically changes to subfolder named [project]
  #
  # --- Advanced options (change only if you know what you are doing) ---
  scannr_sci: 1 # --- Scan number of science data (target) in final MS
  scannr_cal: 2 # --- Scan number of calibrator data in final MS
  obsid: 0 # --- Observation ID in final MS
  sepcal: True # --- produce separate MS files for science and calibration data
  scidata: True  # --- produce (and corrupt) science data
  caldata: False # --- produce (and corrupt) calibration data
  #
  # --- Data corruption ---
  noise_model:
    autocorrweight : 0.0
    table : cal_table_b3dt100ms_cal
    setnoise: # --- casa::setnoise() parameters
      mode : simplenoise
      noiselevel :  0.4Jy
    settrop: # casa::settrop() parameters
      mode:       screen
      pwv:        0.50
      deltapwv:   0.30
      beta:       1.10
      windspeed: 10.00
```

Figure 2.7: Example of user-specified SASIM input parameters. See text for explanation.

```
soap: # SoAP specific input parameters
  msc_folder: soap.input.ms
  output_dir: /tmp/
  lvl: 2
  max_window_size: 180
  min_window_size: 4
  twsize: null
  tclean_cfg: # parameters for tclean
    niter: 1000
    robust: 0.5
    gain: 0.025
    cycleniter: -1
```

Figure 2.7: (Continued) Example of user-specified SASIM input parameters. See text for explanation.

Terrestrial atmosphere (corruption)

Parameter	Function	Default
`pwv`	Total precipitable water vapour in mm	3.0
`deltapwv`	RMS PWV fluctuations *as a fraction of PWV parameter*	0.15
`beta`	Exponent of fractional brownian motion	1.1
`windspeed`	wind speed for screen type corruption (m/s)	7.0
`noiselevel`	Additional noise via ***setnoise***.	0.1Jy

2.8 SASIM execution

SASIM is a standalone python application which uses PiP wheels for CASATOOLS and CASATASKS installed as Python3 modules from the public PyPI server. The most convenient way to use SASIM is through a Python virtual environment which can be easily set up with the following lines executed from the SASIM directory:

```
python3 -m venv casa-sasim
source casa-sasim/bin/activate
pip install --upgrade pip
pip install -r install/requirements.txt
pip install -r install/requirements_external.txt
pip install --index-url https://casa-pip.nrao.edu/repository/pypi-casa-release/
    simple casatools
pip install --index-url https://casa-pip.nrao.edu/repository/pypi-casa-release/
    simple casatasks
python3.7 -m casatools --update-user-data
```

SASIM has been built with the convenient Command-Line Interface (CLI), which allows execution with custom config files. This allows the execution of multiple versions of a project on the same machine without collisions.

Here is a CLI help message:

```
Usage: run.py [OPTIONS]

Options:
  --task [vis|sim|ssf|ssf_obs|soap]
                              Name of the task to execute [required]
  --config_file TEXT          YAML input config file.
  --help                      Show this message and exit.
```

2.9 Limitations of the current version

The version of SASIM as described here was developed with the aim to provide a first recommendation for optimal imaging of solar ALMA observations (see the *High-cadence Report*, Wedemeyer et al. 2023). For that purpose, certain parameters and aspects were fixed but can be implemented in future versions of SASIM.

- **Interferometric observations only.** At this stage, no combination with Total Power (TP) maps is performed within SASIM but is foreseen for future versions. For the imaging study, the TP offset was derived from the reference model maps in a step subsequent to SASIM.

- **Single-pointing time series.** SASIM is used so far for high-cadence single-pointing ("sit-and-stare") observational set-ups. This can, in principle, be modified in the pointing file, which is specified with the input parameter `ptgfile`. The extension towards mosaics can thus be offered in future versions after only minor modifications.

- **Direction / time of observation:** As the Sun moves over the sky in the course of the day, the shape of the synthesised beam will change. It will be the least elongated and thus the closest to a circular shape when the Sun stands highest on the sky at local noon. This effect is small during a typical solar scan of 10min duration but can be notably different for observations during different times of the day. Here, we choose a case of only a slightly elongated beam with a ratio of 1.4-1.5 between the major and minor beam axis. This accounts for non-circular beams while still being representative of the majority of solar ALMA observations.

- **Thermal noise** is not implemented in the current version of SASIM. The inclusion of thermal noise could affect the primary beam correction and thus the resulting CLEANed maps, which will lead to a re-evaluation of the recommended imaging strategies.

- **Heterogeneous baselines** (i.e. baselines between antennas with different diameter as in the 12m Array and ACA) are not included as this option is not supported in the simulator tool package. Please note that heterogeneous baselines are not commonly used for observations of targets other than the Sun and are therefore not a development priority for the simulator tool. For this study, the pragmatic solution was to set the diameter of all ACA antennas to 12 m. We argue that the inclusion of the 12m-ACA baselines, which is enabled by this change, is more important for the synthesised observations than the unrealistic diameters of the simulated ACA antennas. However, we strongly recommend that the developers of the SIMULATOR TOOL implement the treatment of heterogeneous baselines in future versions.

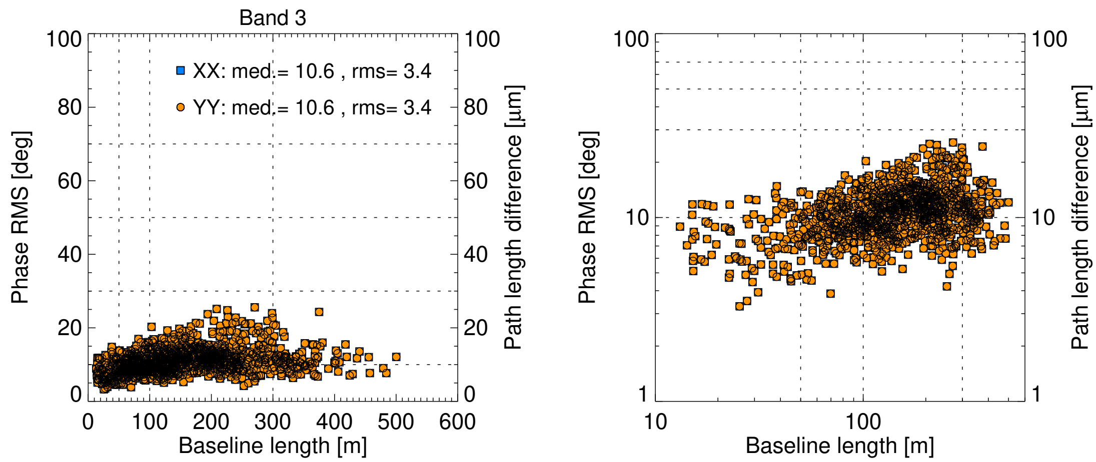


Figure 2.8: Spatial Structure Function for the calibrator MS (i.e. an artifical observations of a point source) after the gain table by Y. Asaki was applied (left: linear scale, right: log-log).

2.10 Future upgrades — Improved phase corruption model

The calibration of real bandpass and phase calibrator scans includes self-calibration, which effectively reduces if not completely removes any baseline dependence and thus any slope in the corresponding SSF. The SSFs for measurement sets that are corrupted with the CASA function ***settrop()*** models phase corruption do not exhibit any notable slope (see Figs. 2.5-2.6) and thus seems to correspond to an approach that includes self-calibration. However, self-calibration is not applied by default on science target scans. Under realistic conditions, science data would thus still show a dependence on baseline length. The phase corruption modelling so far implemented in SASIM based on ***settrop()*** is therefore not ideal for corrupting simulated MSs for science targets. Consequently, it is advisable to explore alternative modelling approaches that could be implemented in the future if found to produce more realistic results.

As a result of the study review process, Y. Asaki kindly provided a gain table that can be applied to a simulated uncorrupted MS instead of using the phase corruption mechanism based on ***settrop***. The gain table was calculated for a time-dependent moving atmospheric phase screen for a calibrator source for Band 3 and a wind speed of 6 m/s . The algorithm devised by Asaki is based on the statistical model for atmospheric phase fluctuation by Dravskikh & Finkelstein (1979), which also uses the concept of a spatial structure function $D\Phi$. Asaki assumes Kolmogorov turbulence with slopes of 5/3 for lengths scales around 1 km and a slope of 2/3 for scales of 6–10 km, respectively.

The simulated calibrator MS, which is here a simple point source, was corrupted with the gain table provided by Asaki. The resulting SSF, which is shown in Fig. 2.8, has notable slope. The median and standard deviation of the SSF are 10.6 and 3.4 degrees, respectively. We note that this slope would be removed if self-calibration would be applied. The same gain table was also successfully applied to the science target MS covering a duration of 60 s with a cadence of 1 s. The resulting corrupted MS was then processed with SOAP with default parameters (`niter=1000, robust=0.5, gain=0.025`). The image for the first time-step is shown in Fig. 2.9 in comparison to the corresponding (uncorrupted) reference map. Already at first glance, the map for the corrupted case appears blurred compared to the uncorrupted reference case, clearly demonstrating that the

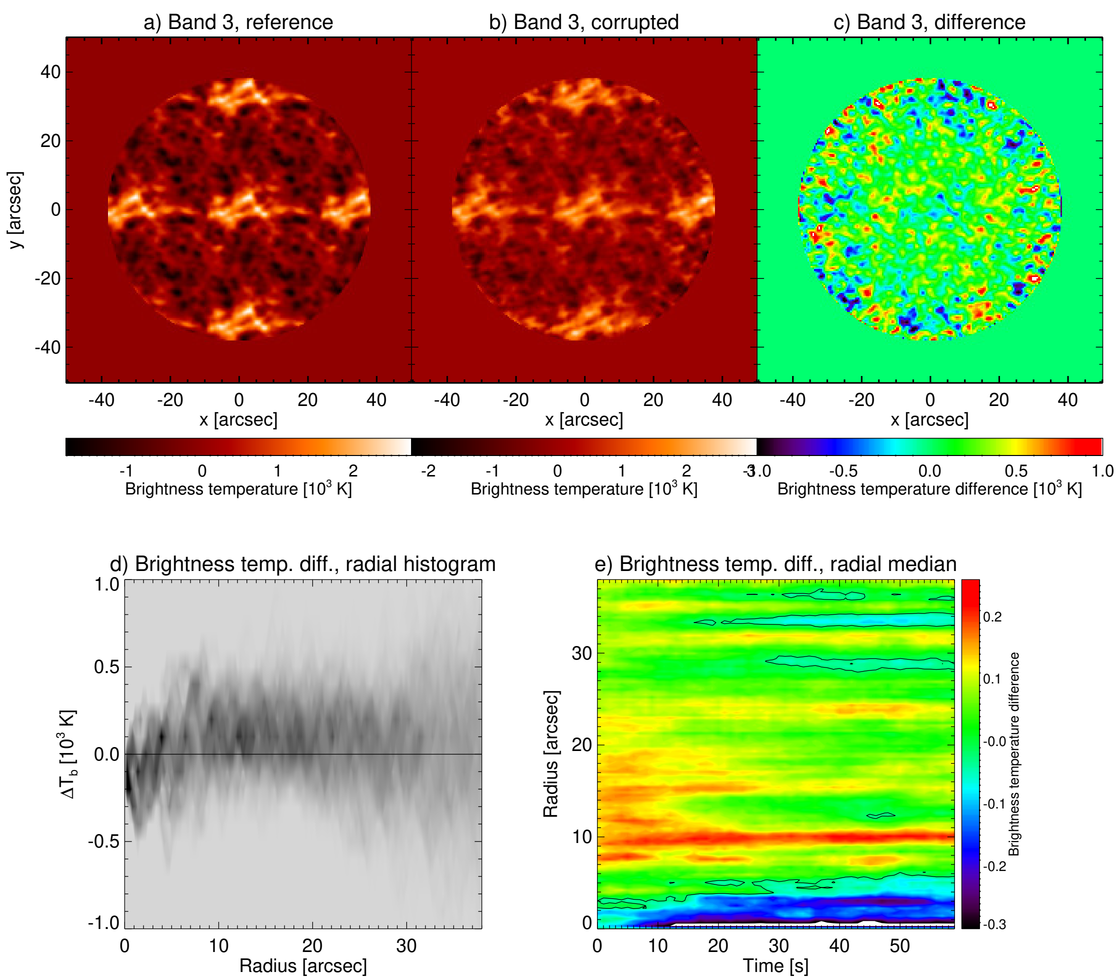


Figure 2.9: Measurement corruption with the gain table provided by Asaki for the first time step of the science target scan for Band 3. a) Reference map (uncorrupted), b) imaging result for the corrupted MS, and c) resulting brightness temperature differences as a map and d) as histogram as function of radius. In addition, the median brightness temperature difference is shown as function radius and time for the whole scan in panel e. The solid black line marks a zero difference.

corruption with the gain table was successful. In particular, the pixels with the highest and lowest brightness temperatures are impacted most notably. The dark regions in the reference map, which are an integral part of chromospheric dynamics produced by propagating shock waves, have much less contrast and are harder to identify in the corrupted case, similar to what is seen in real ALMA observations. The brightness temperature deviation of the imaging result from the uncorrupted reference map is shown in panel c and d as map and histogram as function of radial distance from the centre, respectively. Clearly, the deviations increase as a function of radial distance from the

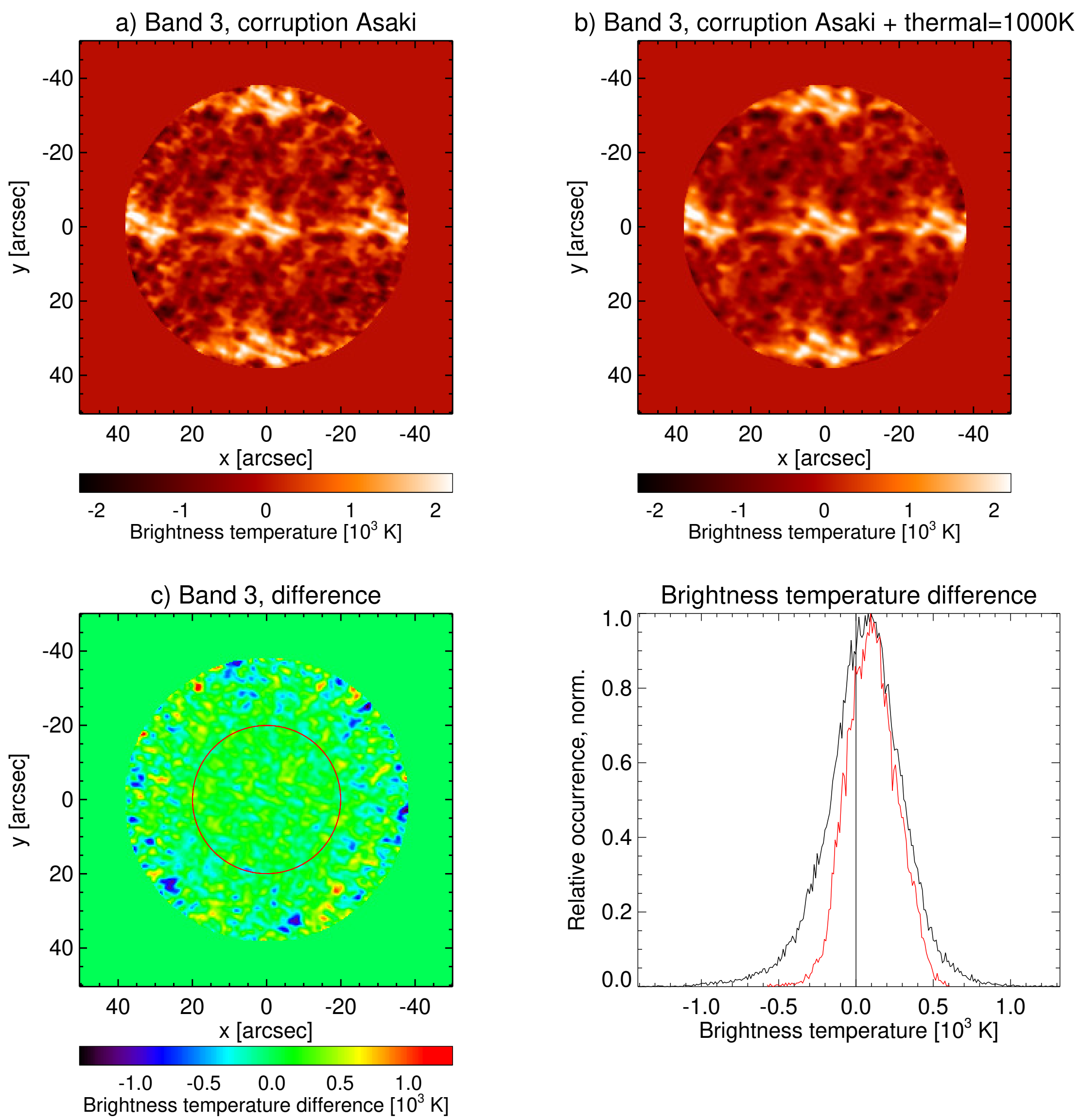


Figure 2.10: Experiments for phase corruption using the gain table by Asaki and an additional thermal receiver noise of $T_{\rm rx} = 1000$ K: a) First time step of the sequence corrupted only with the Asaki gain table and b) the corresponding map with additional receiver noise. Both maps were produced with SoAP (`niter=10000`, `robust=0.5`, `gain=0.2`). The differences between both maps are shown in panel c as a map and as a histogram in panel d. The black histogram includes all pixels within the mask whereas the red histogram is calculated for the inner region with a radius of 20" (see the red circle in panel c).

centre (axis) as expected. The deviations also change in time from frame to frame as intended, illustrating the moving phase screen used for the production of the gain table indeed produces a temporal variation alike to real observing conditions. The temporal variation is shown by means of the median brightness temperature deviation from the reference case as function of time and radial distance from the axis in Fig. 2.9e.

In addition, a thermal receiver noise component was investigated. For this purpose, the CASA ***setnoise()*** was called with a receiver temperature of $T_{rx} = 1000\,K$. The first time step of the resulting time series of brightness temperature maps (produced with SOAP) is compared to the above described data without receiver noise. At first glance, the map with additional receiver noise appears slightly more blurred but the brightness temperature differences remain relatively small in the innermost region. However, as expected, these deviations increase as function of radius and can amount to differences of several 100 K in the outer parts of the map. This experiment demonstrates that such a receiver noise component could be implemented in a future phase corruption model as part of SASIM.

In conclusion, we recommend to study the algorithm by Asaki in more detail and consider its implementation as part of SASIM in the future.

2.11 Experiments with self-calibration

The following experiments are based on the measurement set described in Sect. 2.10 and thus include phase corruption with the gain table provided by Y. Asaki and additional receiver noise of $T_{rx} = 1000\,K$. The corrupted measurement set was used as input for SOAP first for a run without prior self-calibration and then for (independent) runs with self-calibration with different values of the `max_window_size` and `min_window_size` parameters. The same CLEANing parameters were used for all runs (`niter=10000`, `gain=0.2`, `robust=0.5`). The results are shown in Fig. 2.11 for a selected time step. The maps do not exhibit pronounced differences, implying that the choice of the self-calibration parameters is not critical in this case. Directly comparing the deviations with respect to the run without self-calibration (right column in Fig. 2.11) reveals that the deviations increase radially from the axis to the outer parts of the maps.

The deviations are shown as a function of time for the whole simulated time sequence with a duration of 60 s in Fig. 2.12. Again, the different choices for the `max_window_size` and `min_window_size` parameters produce similar results, which are also close to the case without prior self-calibration. Overall, however, the application of self-calibration reduces noise on shorter time scales. Apart from a systematic offset that is likely due to the too idealised nature of the reference model (see Sect. 2.6), the brightness temperatures for the selected pixel follow the same trend as the reference model. We note that (i) the application of self-calibration produces results with less noise and (ii) that the tested phase corruption approach seems to produce useful test cases that should be explored more systematically in the future.

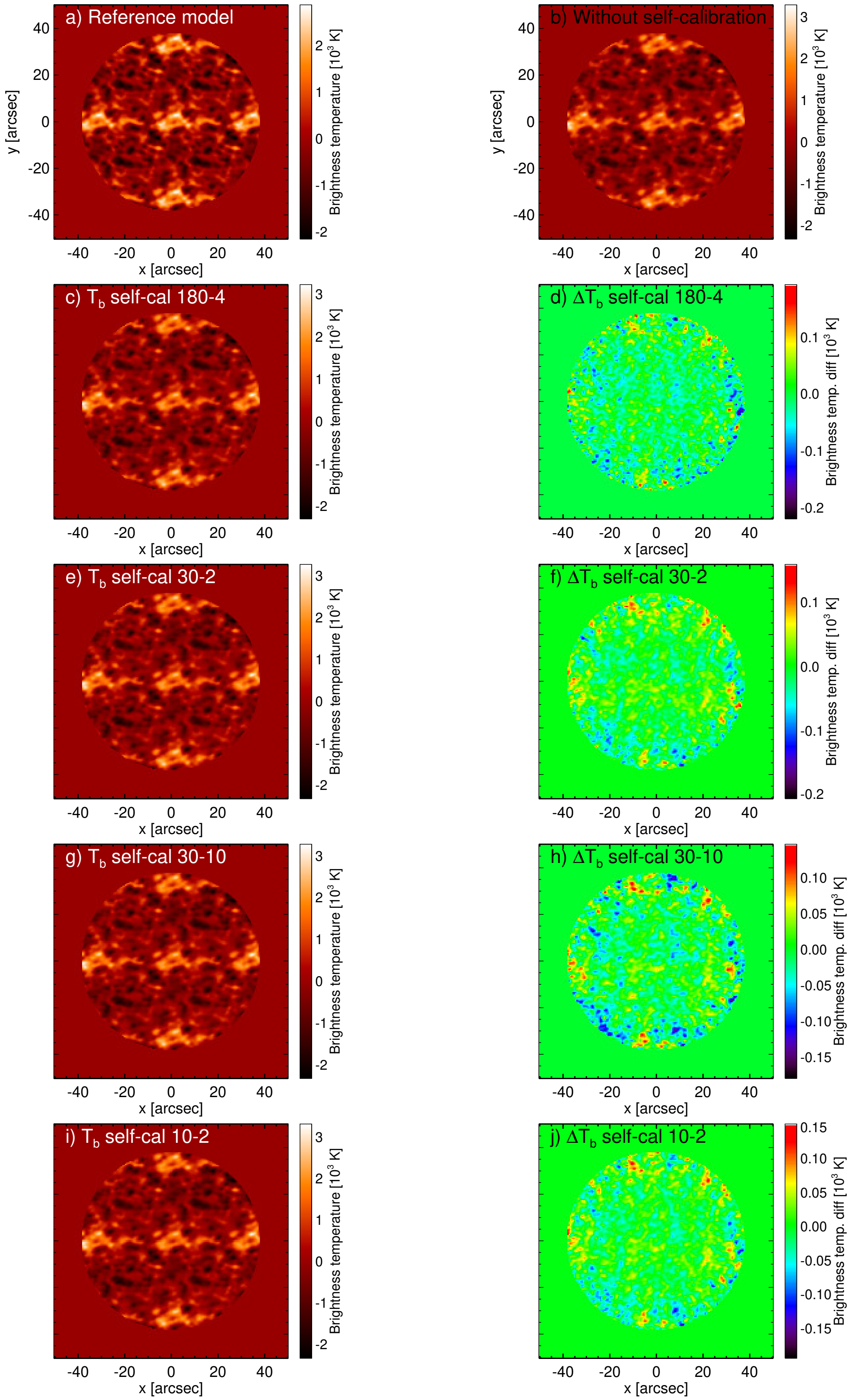


Figure 2.11: Brightness temperature maps for time step 30 for the measurement set corrupted with the gain table by Asaki plus receiver noise ($T_{\rm rx} = 1000$ K) after self-calibration (see `max_window_size` and `min_window_size` parameters in seconds in panels c-j) in comparison to the convolved reference model (panel a) and the map without self-calibration (panel b). The difference of each self-calibrated map and panel b is shown in panels d,f,h,j.

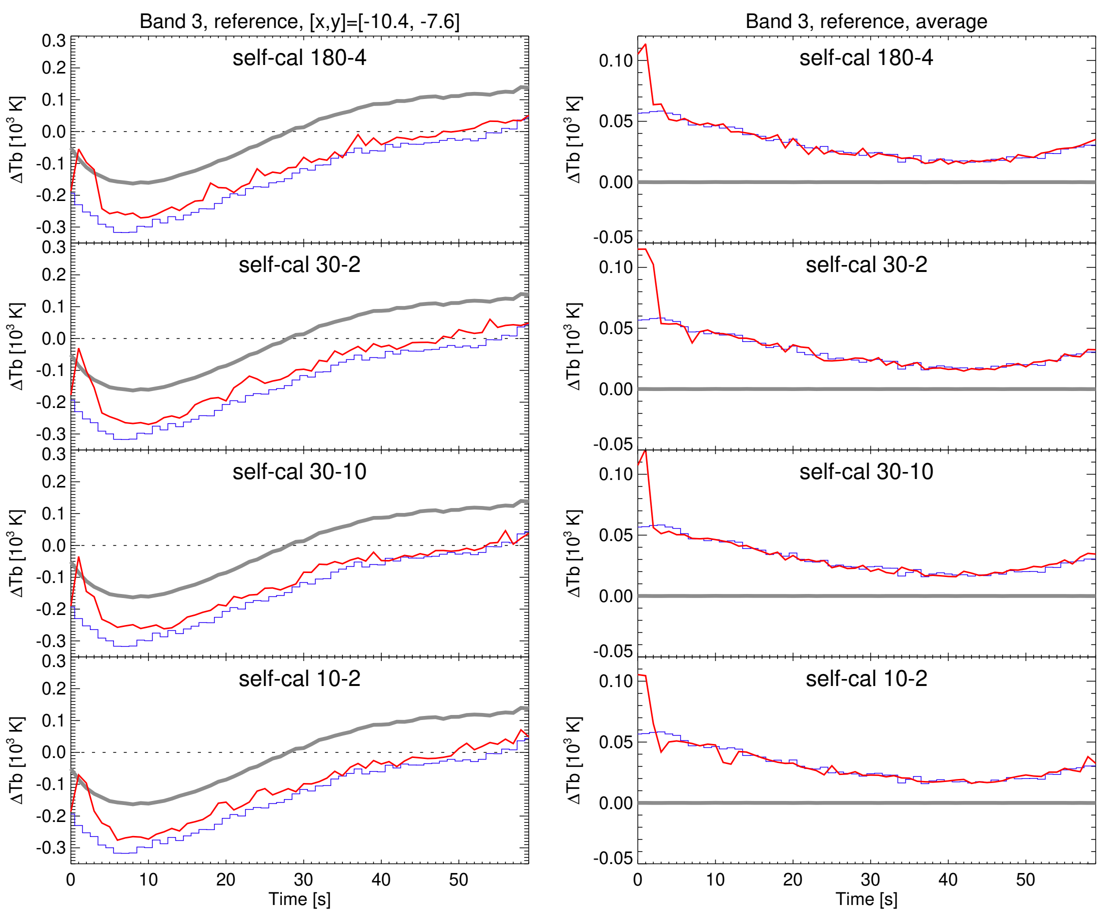


Figure 2.12: Brightness temperature as function time for Band 3 for the measurement set corrupted with the gain table by Asaki plus receiver noise ($T_{\rm rx} = 1000\,$K). Left: Selected pixel with coordinates given above the topmost panel. Right: Horizontal average (for all pixels within the mask). The rows show the results (red lines) for different choices of the self-calibration parameters `max_window_size` and `min_window_size` in seconds (see label in each panel). For comparison, the results without self-calibration (blue) and the convolved reference model (grey) are shown. The same CLEANing parameters were used for all runs (`niter=10000`, `gain=0.2`, `robust=0.5`).

3. Solar ALMA Pipeline

The Solar ALMA Pipeline (SoAP) is a wrapper around the CASA imaging task ***tclean***[1] It has been designed to automatise the deconvolution of measurement sets from time-dependent observations of the Sun with ALMA. Please note that CASA 5.7 was used for the calculations reported in the *Imaging Report* (Wedemeyer et al. 2023) and that SoAP was updated to be compatible with CASA 6.1 in October 2020.

In its current implementation, SoAP has been integrated with SASIM and can be executed from the command line. All SoAP parameters can be adjusted through a config file which is an optional input to SoAP. See Fig 1 for example SoAP specific content of config file.

SoAP is designed so that most of the necessary parameters for ***tclean*** are automatically deduced from the input measurement set (e.g., image cell size, the field of view size etc.). Only a handful number of parameters are left for the user to set up. Namely, parameters which were used for the exploration of the parameter grid described in the *Imaging Report* (Wedemeyer et al. 2023).

The main strength of SoAP is its ability to detect the different timestamps in the input measurement set (here, the artificial MSs produced with SASIM) and produce time series of brightness temperature maps. SoAP also takes care of combining the interferometric data with Total Power (TP) data if available.

3.1 *tclean* task parameters

By default SoAP is using ***tclean*** with Cornwell-Holdaway Multi-Scale deconvolver (Cornwell 2008) as it is designed for images with complicated spatial structure. Scales sizes for the algorithm (in units of the number of pixels) are automatically tuned to be a multiplication of the clean beam size.

For convolutional resampling SoAP always uses a 'mosaic' gridder to correctly construct the primary

[1] `https://casadocs.readthedocs.io/en/stable/api/tt/casatasks.imaging.tclean.html#casatasks.imaging.tclean`

```yaml
soap:
  msc_folder: corr.atm.ms # input ms
  output_dir: /tmp/
  lvl: 2                  # lvl 2, just imaging, lvl 3 img + selfcalibration
   ↪
  specmode: 'fba'
  verbose: True
  max_window_size: 180    # Max intergration time [s] furing selfcalibration
  min_window_size: 4      # Min intergration time [s]
  stokes: I
  refant: A042            # Reference antena (needed for selfcal)
  scans: null             # limit reduction to specific scans
  spws: null              # limit reduction to specific spws
  checkonly: False        # estimate lenght of time series and calculate
   ↪  optimal resolution
  cellsize: null          # overwrite optimal cellsize with own choice
  twsize: null            # window size [s] for box-car like time average
  tsmin: null             # reduce data only for t >= tsmin
  tsmax: null             # reduce data only for t <= tsmax
  uvtaper: null           # use uvtaper
  tclean_cfg:             # setup tclean (selected) parameters
    niter: 1000
    robust: 0.5
    cycleniter: 1000
    gain: 0.025
```

Listing 1: Example of SOAP input parameters in YAML format.

beam from a combination of 12 and 7 m antennae.

The cell size can be set by the user but by default SOAP relay on CASA function ***image.advice()***. The advised cell size is the maximum corresponding to $1/(2 * BL_max)$ where a higher value means undersampling. On the other hand, a bit of over-sampling is recommended. Factor 0.45 chosen as default in SOAP is a 'best fit' to the values found by the QA2 report generator (which knows the properties of the synthesised beam).

For a given cell size, an optimal image size is automatically calculated. The size is rounded up to the integer number that can be written in the form $I = 2^r * 3^s * 5^t$ (with r,s, and t being positive integers). This enables the optimum performance of FFT based on the small prime-numbers radices.

Data weighting during imaging is set to 'Briggs' weighting which provides a compromise between natural and uniform weighting. SOAP also automatically sets a mask at the 0.3 primary beam gain level.

Listing 2 provides a full list of arguments used in ***tclean*** call along with the description and default values.

3.2 SoAP operating modes

SoAP can be configured to produce images in 3 different modes, called levels.

- Level 1: SoAP outputs a time series of primary beam corrected "dirty" images.
- Level 2: The ***tclean*** task is executed with the parameters set in the parameter file (see the beginning of this section) for all unique time stamps (i.e. a small portion of the calibrated MS), resulting in a time series of images.
- Level 3: Self-calibration is applied to a copy of the input MSs and then level 2 imaging is performed on the resulting MS. In the initial step of the self-calibration, the time-averaged (over the length of one scan) level 2 image is provided as an input model for the source. Self-calibration corrections are only applied to the phases.

3.2.1 Sliding time windows

By default, SoAP is cleaning individual images from a very small portion of the measurement set identified by the same unique time stamp t_i, i.e., `timerange` parameter in ***tclean*** is set to `timerange="`$t_i - \delta_t \sim t_i + \delta_t$`"`, where δ_t is equal to half of the cadence of a given data set. However, a user can increase the value of default δ_t via parameter `tw_size`. With `tw_size` $> \delta_t$, individual images will be cleaned from data selected by the time range: $t_i - \text{tw\_size}/2 \sim t_i + \text{tw\_size}/2$. This leads to a reduction in noise and thus, smaller time variations and generally improved image quality.

3.2.2 Self-calibration

SoAP uses aggressive self-calibration tactics for images produced in level 3. It first splits provided scan into chunks of size `max_window_size` (with default value equal to 180s), then for each chunk, SoAP calls ***tclean*** with `savemodel='modelcolumn'` to save the cleaned "model" of the science target with the measurement set. With a model in place for the entire scan, SoAP calls ***gaincal*** with `calmode='p'` and `solint='int` to apply phase corrections only with solution interval per integration. After applying the solution with `applycal`, corrected data is split into a new measurement set which is an input to the new round of self-calibration step, but with scan chopped into even shorted chunks of size `max_window_size / 2`. Rounds of self-calibration are stopped when the chunk size is smaller than min_window_size.

The final corrected measurement set is an input to the final SoAP call which uses it to produce a sequence of images in Level 2.

3.3 SoAP brightness temperature conversion

FITS images produced with SoAP have Jansky/beam as a unit, so that, it is necessary to find and apply a conversion factor if units are wanted in Kelvins. In order to derive the conversion factor the beam size and observed frequency have to be taken from the header in the data file. SoAP using python, RADIO_BEAM and ASTROPY packages for finding correct conversion factor. This transformation method uses the Rayleigh-Jeans approximation of the Planck function to derive the flux:

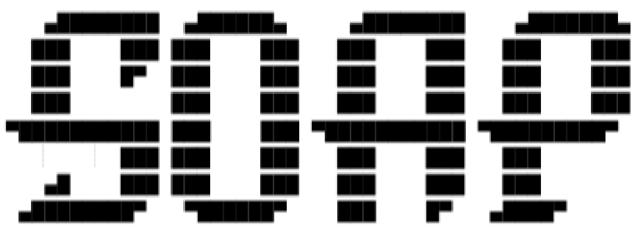
$$T_b = \frac{c^2}{2k_B}\frac{1}{\Omega_A \nu^2}S$$

where T_b is the brightness temperature in K, k_B is the Boltzmann constant [2], c is the light speed in vacuum, ν is the observation frequency in Hz, Ω_A is the beam size in steradians calculated as $\Omega_A = \pi\theta_{maj}\theta_{min}/4\ln 2$ where θ_{maj} and θ_{min} are the BMAJ and BMIN extracted from the the header of FITS image, and finally S is the flux in $\mathrm{Wm^2Hz^{-1}}$.

3.3.1 SoAP runtime informations

SoAP prints lots of information to the standard output and to the *.log file. The user is informed about all automatic adjustments; thus, *.log files should be considered to be a detailed report from the cleaning report. Below an example of SoAP runtime printouts:

```
[SOAP] File $Id: soap_clean_v014 (583ea70) Tue May 12 18:17:30 2020 by Mikolaj $
```



```
                                        ver. 14.0

[SOAP] Started with stamp [ 20230126-220551 ]
[SOAP] Your observation is in Band 3 [ freq = 1.000000e+11 Hz ]
[SOAP] Found field name [ SASim_Sun_b3_12m_4ch_60s.alma12mACA12m.cycle5.3_0 ]
[SOAP] Found [ 1 ] scans for field name [ SASim_Sun_b3_12m_4ch_60s.alma12mACA12m.
    cycle5.3_0 ]
[SOAP] Your scan list: [1]
[SOAP] Estimating the image properties ... (might take some time)
[SOAP] Cellsize estimation from im.advise() [ 0.280 arcsec ]
[SOAP] Your observation is single field
[SOAP] image size : [ 360 x 360 ] with pixel size [ 0.28 arcsec ]
[SOAP] ... [ 4 ] spws for scan number [ 1 ] with [ 60 ] time stamps with cadence [
     1.000000 s ]
[SOAP] ... ..... .... ... scan number [ 1 ] with lenght [ 2020/01/01/16:00:00 ->
    2020/01/01/16:00:59 ]
[SOAP] There is [ nimages = 60 ] to clean to cover obs time [ ~ 1.000000 min ]
[SOAP] Created root output_dir [ /tmp/ ]
[SOAP] Created directory for raw images [ /tmp/SoAP_raw_images.b3.sip.fba.level3.
    Sv014.Cv6.x.mod.20230126-220551 ]
[SOAP] Created directory for fits images [ /tmp/SoAP_fits_images.b3.sip.fba.level3.
    Sv014.Cv6.x.mod.20230126-220551 ]
[SOAP] Created directory for selfcal images [ /tmp/SoAP_SelfCal_ms_20230126-220551
     ]
```

[2] $k_B = 1.3806 \times 10^{-23}\,\mathrm{J\,K^{-1}} = 1.3806 \times 10^{-16}\,\mathrm{cm^2\,g\,s^{-2}\,K^{-1}}$

```
[SOAP] ... processing scan [ 1 ]
[SOAP] ... self-calibrating [ scan = 1 ]
[SOAP] ... self-calibrating: create init model for scan [ 1 ]
[SOAP] ... ... ... self-calibrating : cleaning image [ /tmp/
    SoAP_SelfCal_ms_20230126-220551/selfcal_image_1_0_0 ]
[SOAP] ... ... ... self-calibrating :          [
    2020/01/01/16:00:00~2020/01/01/16:00:59 ]
[SOAP] ... ... ... self-calibrating :          [ dt = 59.000000 s ]
```

```
casatasks.tclean(
    vis          = vis,          # input measurment set
    imagename    = imagename,    # output image name
    field        = field,        # the field to be imaged (usually '0')
    spw          = spw,          # the spectral window(s) to be used (by
     ↪ default all avaible)
    cell         = cell,         # the cell size in angular units
     ↪ (automatically deduced via imager.advise() function)
    stokes       = stokes,       # the Stokes parameters to be imaged
    timerange    = timerange,    # the timerange to be imaged
    imsize       = imsize,       # the size of the output image, deduced via
     ↪ imager.advise() with a tweak to optimal size for FFT
    outframe     = 'LSRK',       # the reference frame of the output image
    deconvolver = deconvolver, # the method used for deconvolution, by
     ↪ default 'mtmfs'
    scales       = [0,6,18],     # the scales to be used for multi-scale clean
    weighting    = 'briggs',     # the weighting scheme to be applied
    robust       = robust,       # the robust parameter for Briggs weighting
    niter        = niter,        # the maximum number of iterations to perform
    cycleniter   = cycleniter,   # the number of iterations after which the
     ↪ clean components will be cycled
    specmode     = tclean_mode, # the spectral, mode to use. By default:
     ↪ 'fba' i.e,. full band average to single frequency
    gridder      = gridder,      # the gridder to be used, default: 'mosaicft'
    usemask      = 'pb',         # the type of mask to be used. Always use
     ↪ primary beam mask
    conjbeams    = True,         # whether to use conjugate beams for
     ↪ non-circular telescopes
    pbmask       = pbmask,       # the primary beam mask to be used, default
     ↪ value 0.3
    pblimit      = pblimit,      # the primary beam correction limit, default
     ↪ value 0.3
    uvtaper      = uvtaper,      # the uv taper to be applied, default `None`
    pbcor        = True,         # whether to correct for the primary beam
    gain         = gain,         # the gain to be used in the cleaning process
     ↪
    threshold    = threshold,    # the cleaning threshold, default: "1.0Jy"
    phasecenter = 0,             # the phase center of the image
    restart      = False,        # whether to restart the cleaning from the
     ↪ previous solution
    savemodel    = savemodel,    # whether to save the model // save only
     ↪ while self-calibration
    parallel     = False         # whether to run in parallel
    )
```

Listing 2: ***tclean*** call from within SOAP with arguments descriptions and default values.

4. Imaging Quality Assessment

In order to decide which imaging parameters produce the best results for a given setup under a certain condition, quantitative imaging quality indicators are needed. A detailed description is given in the *Imaging Report* (Wedemeyer et al. 2023) as the quality indicators are central to the conclusions of the report. Here the most important features regarding the developed software are summarised.

The software for the Imaging Quality Assessment consists of python scripts that do the following:

1. Browse for directories with valid SoAP-produced series of FITS files starting from a user-specified base directory. Alternatively, path can be set specifically by the user.

2. Load the FITS files for one SoAP set at a time.

    - Check if the time series for this set is complete.
    - Retrieve meta data from the FITS header:

        – Major and minor axis of synthesised beam.
        – Pixel sizes and grid extent.
        – Band/frequency.

3. Load the corresponding time series of reference maps for the detected receiver band (see Sect. 2.6).

4. Calculate the brightness temperature differences between each map in the SoAP output and the corresponding reference model map. This results in a time series of ΔT_b maps. Calculate the radial average and save to a file.

5. Calculate the spatial power spectrum for each SoAP output and reference model map. Calculate the power ratio (SoAP map / reference map) for each time step. Save output to a file.

Abbreviation	Description
TDR	Brightness temperature difference with respect to the corresponding reference model (in radial bins or weighted average, across whole time series)
TDRA	Average of TDR
TDRV	Standard deviation of TDR
TDR+	TDRA and TDRV combined, weighted average across all spatial bins
SPR	Spatial power ratio with respect to the corresponding reference model (in spatial bins or weighted average, across whole time series))
SPRA	Average of SPR over a bin
SPRV	Standard deviation of SPR
SPR+	SPRA and SPRV ombined, weighted average across all spatial bins
UQI	Unified Quality Indicator combining TDRA, TDRV, SPRA, SPRV

Table 4.1: Quality indicators used in this study. Please note that the TDR and SPR indicators are always calculated within spatial bins but can then evaluated as corresponding weighted averages. An additional b indicates that the quantity refers to individual bins (e.g., TDRAb) but the weighted average otherwise. See Sects. 2.7.4-2.7.6 in the *Imaging Report* (Wedemeyer et al. 2023) for further explanation.

6. After all files have been processed for all directories: Collect all files with radial brightness temperature difference profiles and spatial power spectrum ratios and calculate the following Imaging Quality Indicators for the current data set: TDRA, TDRV, SPRA, SPRV (see Table 4.1).

7. Save the results in global tables for each receiver band.

After the set of quality indicators is produced, analysis is performed (mostly in IDL). The tables are loaded, which makes the quality indicators available as function of receiver band, corruption parameters, and imaging parameters.

Please refer to Sect. 2.6 in the *Imaging Report* (Wedemeyer et al. 2023) for a detailed description of how the indicators are calculated for the cases considered in this study and to Sect. 3 in the *Imaging Report* (Wedemeyer et al. 2023) for details regarding the analysis of the quality indicators, the construction of the global indicators (TDR+, SPR+ and GQI, see Table 4.1) and for the determination of the best parameter combinations.

5. Additional software packages

Here very brief descriptions of software packages are provided that are essential for the study but have been developed before and/or separately from the study.

5.1 Radiation magnetohydrodynamics (RMHD) code Bifrost

The model of the Sun used in this report is produced with the state-of-the-art 3D radiation magnetohydrodynamics (RMHD) code Bifrost (Gudiksen et al. 2011; Carlsson et al. 2016). The simulations are self-consistent in the sense that the dynamics in the resulting time-dependent 3D data cubes are a natural outcome of the solution of the RMHD equations together with a realistic equation of state and opacities. A particular strength of the Bifrost code, which is essential for the study of emergent radiation at (sub)millimeter wavelengths, is the detailed time-dependent treatment of hydrogen ionisation that provides realistic electron densities. The solar model used for this study is thus as realistic as currently possible.

The computational domain of a resulting 3D simulation usually includes the top of the convection zone, the photosphere, the chromosphere and the lower parts of the corona, and thus all layers that are relevant for the formation of radiation at (sub)millimeter wavelengths.

The data from Bifrost is stored in HDF5 format. Please refer to (Gudiksen et al. 2011) for more details.

5.2 Advanced Radiative Transfer (ART) code

The initial version of the Advanced Radiative Transfer (ART) code has been developed by Co-I J. de la Cruz Rodriguez in collaboration with Oslo's Solar ALMA project and optimised for computational performance supported by a PRACE Preparatory Access grant (De La Cruz Rodríguez et al. 2021). The code reads snapshots from 3D simulations of the solar atmosphere and solves the equation of radiative transfer along many rays through the model atmosphere in much detail, considering all relevant opacity sources. The code then outputs continuum intensity maps at a prescribed wavelength but can also calculate spectral line profiles for all spatial positions in the model snapshot. The original ART output is stored in HDF5 format.

Bibliography

Carlsson, M., Hansteen, V. H., Gudiksen, B. V., Leenaarts, J., & De Pontieu, B. 2016, A&A, 585, A4

Cornwell, T. J. 2008, IEEE Journal of Selected Topics in Signal Processing, 2, 793

De La Cruz Rodríguez, J., Szydlarski, M., & Wedemeyer, S. 2021, ART: Advanced (and fast!) Radiative Transfer code for Solar Physics., Zenodo

Dravskikh, A. F. & Finkelstein, A. M. 1979, Ap&SS, 60, 251

Gudiksen, B. V., Carlsson, M., Hansteen, V. H., et al. 2011, A&A, 531, A154+

Henriques, V. M. J., Jafarzadeh, S., Guevara Gómez, J. C., et al. 2022, A&A, 659, A31

Ishizaki, H. & Sakamoto, S. 2005, ALMA Memo

Maud, L. T., Tilanus, R. P. J., van Kempen, T. A., et al. 2017, A&A, 605, A121

Nikolic, B., Bolton, R. C., Graves, S. F., Hills, R. E., & Richer, J. S. 2013, Astronomy and Astrophysics

Nikolic, B., Hills, R. E., & Richer, J. S. 2007, ALMA Memo

Pardo, J. R., Cernicharo, J., & Serabyn, E. 2001, IEEE Transactions on Antennas and Propagation

Pardo, J. R., Gérin, M., Prigent, C., et al. 1998, Journal of Quantitative Spectroscopy and Radiative Transfer

Pardo, J. R., Pagani, L., Gerin, M., & Prigent, C. 1995, Journal of Quantitative Spectroscopy and Radiative Transfer

Please also refer to the following two reports that are results of this study:

[Imaging Report]
Wedemeyer, S., Szydlarski, M., Carozzi, T., Toribio, M. C. et al. 2023, ESO "Advanced Study for Upgrades of the Atacama Large Millimeter/submillimeter Array (ALMA)" (CFP/ESO/16/11115/OSZ), *High-cadence Imaging of the Sun*, final study report review data package - DS4:
Recommendations for optimal post-processing of solar ALMA data

[High-Cadence Report]
Wedemeyer, S., Szydlarski, M., Carozzi, T., Toribio, M. C. et al. 2023, ESO "Advanced Study for Upgrades of the Atacama Large Millimeter/submillimeter Array (ALMA)" (CFP/ESO/16/11115/OSZ), *High-cadence Imaging of the Sun*, final study report review data package - DS4:
The potential of a high-cadence imaging mode for ALMA observations of the Sun