developer.skatelescope.org Documentation

Release 0.1.0-beta

Marco Bartolini

Mar 21, 2024

HOME

This package collects RASCIL scripts for various SKA-MID simulations: continuum imaging, continuum imaging with direction-dependent effects, high data rate observations, observation sizing, and radio frequency interference.

The package uses RASCIL for workflow functions, ska-sdp-func-python for processing functions and ska-sdp-datamodels for data models.

CHAPTER

ONE

CONTINUUM IMAGING

These scripts simulate MID continuum imaging observations of multiple point sources.

- The sky model is constructed from Oxford S3-SEX catalog. These are unpolarised point sources.
- The observation is by MID or MEERKAT+ over a range of hour angles.
- The visibility is calculated by Direct Fourier transform after application of the gaintable for each source.
- Dask is used to distribute the processing over a number of workers.
- Processing can be divided into chunks of time (default 1800s).

The core simulation functions reside in RASCIL. The RASCIL driver script for simulation in this repository is direction_dependent/src/mid_simulation.py. This driver scripts can be run directly using the command line arguments listed below. Canonical bash scripts have been provided in continuum_imaging/scripts. The scripts will need to be altered for location of the various files needed. We recommend use of the bash scripts at first.

These simulations typically requiring a cluster to run. In Cambridge They work well on the P3 cluster using 16 nodes of 128GB each.

There are types of output: MeasurementSets and Images. Each are calculated for actual, nominal, and difference. The time consumed in the calculation of the MeasurementSets is small compared to the time in writing the MeasurementSet and the time to make the images. Images may be calculated using the dist_imager.

The fits images can be viewed using the casaviewer or carta. The MeasurementSets can be viewed using casaviewer.

If you are running on your own machine, make sure you have the below environment variables set up:

> SSMROOT : location where the ska-sim-mid repository is SSMRESOURCES : location of the resources e.g. beam models SSMRESULTS: location of results folder

1.1 Effects simulated

1.1.1 Nominal

- A set of point sources is simulated and the relevant nominal voltage pattern for each end of interferometer is applied before Fourier transform. The nominal pattern is constructed from a tapered symmetric illumination pattern, with the diameter of the SKA dishes.
- stokesIQUV, band B2

1.1.2 Heterogenous

- A set of point sources is simulated and the relevant voltage pattern for each end of interferometer is applied before Fourier transform.
- Simulates observations with all SKA and MEERKAT dishes (on state) and all SKA dishes (off state).
- stokesIQUV, band B2

1.1.3 Heterogenous_meerkat+

- A set of point sources is simulated and the relevant voltage pattern for each end of interferometer is applied before Fourier transform.
- Simulates observations with MEERKAT+ configuration, all SKA and MEERKAT dishes (on state) and all SKA dishes (off state).
- stokesIQUV, band B2

1.1.4 Ionosphere

- A set of point sources is simulated and the phases calculated using a thin screen model for the ionosphere. The screen has units of meters of Total Electron Content. The phase is evaluated at places in the screen where the line of sight from source to a dish pierces the screen.
- Simulates observations with ionospheric screens on and off.
- This requires the screens to be calculated first or downloaded: see ../screens/README.md
- stokesI, band B1LOW

1.1.5 Polarisation

- Simulates observations with SKA and EMSS calculated primary beam models with cross pol (on) and no cross pol set to zero (off)
- Polarisation stokesIQUV, linear, band B2

1.1.6 Surface sag

- Simulates observations with sagging dish (on) and nominal dish (off).
- Models of the voltage pattern are available at +15, +45, +90 deg elevation.
- We interpolate between those to 5 degrees.
- Stokes I, band B2

For more details see: https://confluence.skatelescope.org/display/SE/Dish+deformation+simulations

1.1.7 Troposphere

- Simulates observations with tropospheric screens on and off.
- This requires the screens to be calculated first or downloaded: see ../screens/README.md
- stokesI, bands B2 and B5
- A set of point sources is simulated and the phases calculated using a thin screen model for the atmosphere. The phase is evaluated at places in the screen where the line of sight from source to dish pierces the screen. The screen has units of meters of delay.

1.1.8 Wind pointing

- Simulates observations with wind buffeting of dishes (on) and without (off)
- Stokes I, bands B2 and B5

For more details see: https://confluence.skatelescope.org/display/SE/MID+pointing+error+simulations

Simulate SKA-MID direction dependent errors

```
usage: mid_simulation.py [-h] [--rmax RMAX] [--image_sampling IMAGE_SAMPLING]
                         [--configuration CONFIGURATION]
                         [--antennas [ANTENNAS ...]] [--source SOURCE]
                         [--ra RA] [--declination DECLINATION] [--band BAND]
                         [--nchan NCHAN] [--channel_width CHANNEL_WIDTH]
                         [--integration_time INTEGRATION_TIME]
                         [--time_range TIME_RANGE TIME_RANGE]
                         [--time_step TIME_STEP] [--make_images MAKE_IMAGES]
                         [--only_actual ONLY_ACTUAL] [--apply_pb APPLY_PB]
                         [--pbtype PBTYPE] [--seed SEED]
                         [--flux_limit FLUX_LIMIT] [--npixel NPIXEL]
                         [--cellsize CELLSIZE] [--weighting WEIGHTING]
                         [--robustness ROBUSTNESS] [--results RESULTS]
                         [--elevation_sampling ELEVATION_SAMPLING] [--r0 R0]
                         [--height HEIGHT] [--screen SCREEN]
                         [--isoplanatic ISOPLANATIC] [--time_chunk TIME_CHUNK]
                         [--channel_start CHANNEL_START] [--mode MODE]
                         [--duration DURATION]
                         [--wind_conditions WIND_CONDITIONS]
                         [--global_pe GLOBAL_PE GLOBAL_PE]
                         [--static_pe STATIC_PE STATIC_PE]
                         [--dynamic_pe DYNAMIC_PE]
                         [--pointing_directory POINTING_DIRECTORY]
                         [--vp_directory VP_DIRECTORY]
                         [--vp_support VP_SUPPORT] [--use_dask USE_DASK]
                         [--write_gt WRITE_GT] [--write_pt WRITE_PT]
                         [--imaging_dft_kernel IMAGING_DFT_KERNEL]
                         [--add_noise ADD_NOISE] [--ra_s RA_S] [--dec_s DEC_S]
                         [--vp_real_file VP_REAL_FILE]
                         [--vp_imag_file VP_IMAG_FILE] [--obs_time OBS_TIME]
```

Named Arguments

--rmax	Maximum distance of dish from centre (m) Default: 200000.0
--image_sampling	Number of points per synthesized beam Default: 3.0
--configuration	MID Configuration: MID | MEERKAT+ Default: "MID"
--antennas	Antenna names to include (default is all)
--source	Type of source: s3sky or point or double Default: "s3sky"
--ra	Right ascension of phase centre (degrees) Default: 0.0
--declination	Declination of phase centre (degrees) Default: -40.0
--band	Band B1LOW | B1 | B2 | B5 | Ku Default: "B2"
--nchan	Number of frequency channels Default: 1
--channel_width	Channel bandwidth (Hz) (default is to calculate from band frequency)
--integration_time	Integration time (s) Default: 180
--time_range	Time range in hour angle Default: [-4.0, 4.0]
--time_step	Time step (unit: second of time)
--make_images	Make the images? Default: "True"
--only_actual	Only make the actual data? Default: "False"
--apply_pb	Apply the primary beam in modes troposphere and ionosphere? Default: "True"
--pbtype	Primary beam model: MID_B1 MID_B1LOW MID_B2 MID_Ku MEERKAT_B1 MEERKAT_B2 MEERKAT_Ku Default: "MID_B2"
--seed	Random number seed Default: 18051955

--flux_limit	Lower flux limit (Jy)
	Default: 0.01
--npixel	Number of pixels in the resulting image
	Default: 512
--cellsize	Cellsize in radians (will calculate if set to None)
--weighting	Type of weighting (natural, uniform or robust)
	Default: “uniform”
--robustness	Robustness for robust weighting
	Default: 0.0
--results	Directory for results
	Default: “./”
--elevation_sampling	Sampling in elevation for surface (deg)
	Default: 1.0
--r0	R0 (meters)
	Default: 5000.0
--height	Height of layer (meters)
	Default: 300000.0
--screen	Location of atmospheric phase screen data
	Default: “/home/docs/checkouts/readthedocs.org/user_builds/ska-telescope-ska-sim-mid/envs/latest/lib/python3.10/site-packages/rascil/processing_components/../../data/models/test_mpc_screen.fits”
--isoplanatic	Are the phase screens to be treated as isoplanatic?
	Default: “False”
--time_chunk	Time for a chunk (s)
	Default: 3600.0
--channel_start	First channel of interest if interested in some frequency range. Use with –nchan
--mode	Mode of simulation: wind_pointing|random_pointing|polarisation|ionosphere|troposphere|heterogeneous
	Default: “none”
--duration	Type of duration: long or medium or short. Set to custom when using options –integration_time and –time_range (or–time_step)
	Default: “long”
--wind_conditions	SKA definition of wind conditions: precision, standard or degraded
	Default: “precision”
--global_pe	Global pointing error (absolute values)
	Default: [0.0, 0.0]
--static_pe	Multipliers for static errors
	Default: [0.0, 0.0]

--dynamic_pe	Multiplier for dynamic errors Default: 1.0
--pointing_directory	Location of wind PSD pointing files Default: "/home/docs/checkouts/readthedocs.org/user_builds/ska-telescope-ska-sim-mid/envs/latest/lib/python3.10/site-packages/rascil/processing_components/../../data/models"
--vp_directory	Location of voltage pattern files
--vp_support	Number of pixels in voltage pattern images
--use_dask	Use dask processing? Default: "True"
--write_gt	Write gaintable files as HDF Default: "False"
--write_pt	Write pointing table files as HDF Default: "False"
--imaging_dft_kernel	DFT kernel: cpu_looped | cpu_numba | gpu_raw
--add_noise	Add noises to visibilities Default: "False"
--ra_s	Right ascension of phase centre (degrees) Default: -999.99
--dec_s	Declination of phase centre (degrees) Default: 999.99
--vp_real_file	Voltage pattern real part file name
--vp_imag_file	Voltage pattern imagine part file name
--obs_time	Observation start time

CHAPTER

TWO

DIRECTION-DEPENDENT EFFECTS

These scripts simulate MID observations of multiple point sources and calculate the effect of direction dependent gain errors.

- The sky model is constructed from Oxford S3-SEX catalog. These are unpolarised point sources.
- The observation is by MID or MEERKAT+ over a range of hour angles.
- The visibility is calculated by Direct Fourier transform after application of the gaintable for each source.
- Dask is used to distribute the processing over a number of workers.
- Processing can be divided into chunks of time (default 1800s). This allows

See the following presentation for an overview of the results:

> Simulating MID direction dependent gain effects SPO-1057

The core simulation functions reside in RASCIL. The RASCIL driver script in this repository is direction_dependent/src/mid_simulation.py. This can be run directly using the command line arguments listed below, or some typical bash scripts have been provided in direction_dependent_src/scripts. The bash scripts allow for looping over duration and declination. The scripts will need to be altered for location of the various files needed. We recommend use of the bash scripts at first.

Note that the simulation has two steps: first the visibilities are calculated and written to HDF files, using mid_simulation.py and then all the HDF files are combined into one MeasurementSet using convert_to_ms.py. In this conversion step, a number of diagnostic plots are written.

These simulations typically requiring a cluster to run. In Cambridge They work well on the P3 login node, which has 512GB and 64 threads/32 cores, using a large number of threads. For this approach –use_slurm should be False.

There are types of output: MeasurementSets and Images. Each are calculated for actual, nominal, and difference. The time consumed in the calculation of the MeasurementSets is small compared to the time in writing the MeasurementSet and the time to make the images. Images may be calculated using the dist_imager.

The fits images can be viewed using the casaviewer or carta. The MeasurementSets can be viewed using casaviewer.

If you are running on your own machine, make sure you have the below environment variables set up:

> SSMROOT : location where the ska-sim-mid repository is SSMRESOURCES : location of the resources e.g. beam models SSMRESULTS: location of results folder

The default or these variables are set as common directories on P3.

2.1 Effects simulated

2.1.1 Nominal

- A set of point sources is simulated and the relevant nominal voltage pattern for each end of interferometer is applied before Fourier transform. The nominal pattern is constructed from a tapered symmetric illumination pattern, with the diameter of the SKA dishes.
- stokesIQUV, band B2

2.1.2 Heterogenous

- A set of point sources is simulated and the relevant voltage pattern for each end of interferometer is applied before Fourier transform.
- Simulates observations with all SKA and MEERKAT dishes (on state) and all SKA dishes (off state).
- stokesIQUV, band B2

2.1.3 Heterogenous_meerkat+

- A set of point sources is simulated and the relevant voltage pattern for each end of interferometer is applied before Fourier transform.
- Simulates observations with MEERKAT+ configuration, all SKA and MEERKAT dishes (on state) and all SKA dishes (off state).
- stokesIQUV, band B2

2.1.4 Ionosphere

- A set of point sources is simulated and the phases calculated using a thin screen model for the ionosphere. The screen has units of meters of Total Electron Content. The phase is evaluated at places in the screen where the line of sight from source to a dish pierces the screen.
- Simulates observations with ionospheric screens on and off.
- This requires the screens to be calculated first or downloaded: see ../screens/README.md
- stokesI, band B1LOW

2.1.5 Polarisation

- Simulates observations with SKA and EMSS calculated primary beam models with cross pol (on) and no cross pol set to zero (off)
- Polarisation stokesIQUV, linear, band B2

2.1.6 Surface sag

- Simulates observations with sagging dish (on) and nominal dish (off).
- Models of the voltage pattern are available at +15, +45, +90 deg elevation.
- We interpolate between those to 5 degrees.
- Stokes I, band B2

For more details see: https://confluence.skatelescope.org/display/SE/Dish+deformation+simulations

2.1.7 Troposphere

- Simulates observations with tropospheric screens on and off.
- This requires the screens to be calculated first or downloaded: see ../screens/README.md
- stokesI, bands B2 and B5
- A set of point sources is simulated and the phases calculated using a thin screen model for the atmosphere. The phase is evaluated at places in the screen where the line of sight from source to dish pierces the screen. The screen has units of meters of delay.

2.1.8 Wind pointing

- Simulates observations with wind buffeting of dishes (on) and without (off)
- Stokes I, bands B2 and B5

For more details see: https://confluence.skatelescope.org/display/SE/MID+pointing+error+simulations

CHAPTER

THREE

POINTING-OFFSET SIMULATION

These scripts simulate MID pointing offset observations for 5-point (or more) scans and prepare the simulation data for Pointing-Offset Pipeline.

- The users could define the radio source and observational time.
- The observation is for MID-AA0.5 telescope layout over a range of hour angles.
- Dask is used to distribute the processing over several workers.
- Processing can be divided into chunks of time (default 1800s).

The core simulation functions reside in RASCIL. The RASCIL driver script in this repository is /src/mid_simulation.py. This can be run directly using the command line arguments listed below, or bash scripts, which have been provided in the pointing_offset directory. The bash scripts allow for looping over observational duration and source coordinate (RA and DEC). Before using the scripts, the users could modify observatory location, point source coordinate, integration time, time chunk, and so on.

Note that the simulation has four steps:

1. Calculate 5-point pointing directions (RA/DEC) according to the given offset values, source RA, source DEC, and observation time.
2. Generate Gaussian Voltage Pattern FITS files.
3. Simulate pointing offset data (MeasurementSet format) using mid_simulation.py and generate relevant HDF files.
4. Add SOURCE OFFSET and relevant tables in the MS files.

The simulation can be run in any PC servers. According to the number of channels, the amount of memory is quite important for the simulation. According to the current simulation results, the size of voltage pattern files (real or imagined part) of 8, 256, 1024, and 16384 channels is about 268 MB, 859 MB, 3.43 GB, and 549 GB, respectively.

MeasurementSets (MS), FITS, and HDF5 files save the results. Currently, MS files are generated to store actual and nominal data and their differences. HDF5 files include primary beam components and actual, nominal, and source gaintable files. In the simulation, nominal files mean the expected values and actual data means the expected + pointing offset values. The Voltage Pattern (VP) files are stored in FITS format.

The FITS file can be viewed using the casaviewer or carta <https://cartavis.org/> `_. The MS file can be viewed using the `casaviewer as well.

If the user wants to run on a local machine, make sure to setup the below environment variables:

> SSMROOT: location where the ska-sim-mid repository is SSMRESOURCES: location of the resources, e.g. beam models SSMRESULTS: location of results folder

The default for these variables is set as common directories on the HPC.

3.1 Script Descriptions

3.1.1 get_radec_time.py

- Calculate RA and DEC according to the 5-point observation requirements. AstroPy is used in the position calculation.
- Output data files are used in the simulation.

3.1.2 gen_band2_gaussian_vp.py

- A point source is simulated, and the relevant voltage pattern is applied before the Fourier transform.
- Simulates observations with SKA1-MID AA0.5.
- stokesIQUV, Band B2

3.1.3 make_B2_AA05_5points.sh

- A shell script to run the simulation.

For more details, see: https://confluence.skatelescope.org/display/SE/Pointing+Offset+Simulation+Using+SKA-SIM-MID

CHAPTER

FOUR

PSI SIMULATIONS

These are simulated visibility data sets in Measurement Set format, as needed to test the operation of the CBF-SDP interface and the real-time signal displays.

These simulations generate fully time-sampled, high number of spectral channel observations using the simulated Prototype System Integration (PSI) test arrays: a small number of antennas or stations (between 4 and 8), with a large number of frequency channels (of order 10000).

The simulation currently does:

- 10 minutes of 0.28s integrations
- six antennas chosen to be close to the centre but to give baselines upto 200m
- 50% fractional bandwidth: centre frequency 1.369GHz +/- 0.5 * 1.369GHz

The simulation requires a large amount of memory, primality because of inefficiencies in the RASCIL visibility format. We expect to reduce these over time. The best Dask setup on Alaska P3 is to use one worker and a large number of threads (up to 64).

Simulate time and frequency dense PSI observations

```
usage: mid_write_psi.py [-h] [--rmax RMAX] [--configuration CONFIGURATION]
                        [--antennas [ANTENNAS ...]] [--badantenna BADANTENNA]
                        [--howbad HOWBAD] [--ra RA]
                        [--declination DECLINATION] [--band BAND]
                        [--nchan NCHAN] [--nspw NSPW]
                        [--channel_width CHANNEL_WIDTH]
                        [--integration_time INTEGRATION_TIME]
                        [--time_range TIME_RANGE TIME_RANGE]
                        [--image_pol IMAGE_POL] [--vis_pol VIS_POL]
                        [--duration DURATION] [--results RESULTS]
                        [--msname MSNAME] [--split SPLIT]
                        [--nthreads NTHREADS] [--processes PROCESSES]
                        [--memory MEMORY] [--nworkers NWORKERS]
                        [--use_dask USE_DASK] [--interface INTERFACE]
```

4.1 Named Arguments

--rmax	Maximum distance of dish from centre (m)
	Default: 200.0
--configuration	MID Configuration: MID | MEERKAT+
	Default: “MID”
--antennas	antenna names to include (default is all)
--badantenna	Bad antenna
	Default: -1
--howbad	How many meters to move bad antenna
	Default: 1
--ra	Right ascension of phase centre (degrees)
	Default: 0.0
--declination	Declination of phase centre (degrees)
	Default: -55.0
--band	Band B1LOW | B1 | B2 | Ku
	Default: “B2”
--nchan	Number of frequency channels
	Default: 1024
--nspw	Number of spectral windows
	Default: 8
--channel_width	Channel bandwidth (Hz) (default is to calculate from frequency
--integration_time	Integration time (s)
	Default: 0.14
--time_range	Time range in hour angle
	Default: [0, 0.16667]
--image_pol	RASCIL polarisation frame for image: stokesI | stokes IQ | stokesIQUV
	Default: “stokesIQUV”
--vis_pol	RASCIL polarisation frame for visibility: linear | linearnp
	Default: “linear”
--duration	Type of duration: long or medium or short
	Default: “long”
--results	Directory for results
	Default: “./”
--msname	Root name of MS (i.e. without .ms)
	Default: “sim_mid_psi”

--split	In concat, type of tree e.g. 2->binary Default: 2
--nthreads	Number of threads per worker Default: 1
--processes	Number of processes per worker Default: 1
--memory	Memory per worker (GB)
--nworkers	Number of workers Default: 4
--use_dask	Use dask processing? Default: "True"
--interface	Network interface Default: "ib0"

Todo:

- Insert todo's here

CHAPTER

FIVE

SIZING OBSERVATIONS

This script estimates the size of CASA MeasurementSets for various MID observations. it does this by constructing a scaled down Visibility and writing it to a MeasurementSet.

The maximum size of the array (maximum distance from the array centre) can be set. This determines the angular resolution. See parameter rmax.

The field of view to be imaged is set to a multiple of the primary beam. See parameter guardband.

The time and frequency sampling is adjusted to match the SKA specification of 2% decorrelation at the half-power point of the primary beam. See parameter dela.

Typical output:

```
mid_size: Starting MID simulation sizing

{'band': 'B2',
 'context': 's3sky',
 'declination': -45.0,
 'dela': 0.02,
 'fractional_bandwidth': 0.1,
 'guardband': 2.0,
 'integration_time': 80.0,
 'output_msname': 'mid_simulation.ms',
 'ra': 15.0,
 'rmax': 1000.0,
 'time_range': [-4.0, 4.0],
 'verbose': 'False'}
mid_size: Using only dishes within 1000.0m of the array centre requires 12 channels of 1.
↪13e+07Hz and integration time 683.8s, the MS size is 0.338GB
mid_size: The time and frequency sampling is to provide no more than 2% smearing at the␣
↪half power point of the primary beam.
mid_size: Image has shape 768 by 768, 4 polarisations, and size 0.018GB
mid_size: W processing requires 49 w planes of step 238.8 wavelengths and maximum␣
↪support 0 pixels
mid_size: Allowing fractional bandwidth 0.100  to fill in the uv sampling.
nmid_size: Starting MID simulation sizing

{'band': 'B2',
 'context': 's3sky',
 'declination': -45.0,
 'dela': 0.02,
 'fractional_bandwidth': 0.1,
```

(continues on next page)

(continued from previous page)

```
 'guardband': 2.0,
 'integration_time': 80.0,
 'output_msname': 'mid_simulation.ms',
 'ra': 15.0,
 'rmax': 10000.0,
 'time_range': [-4.0, 4.0],
 'verbose': 'False'}
mid_size: Using only dishes within 10000.0m of the array centre requires 107 channels of␣
↪1.27e+06Hz and integration time 80.4s, the MS size is 50.382GB
mid_size: The time and frequency sampling is to provide no more than 2% smearing at the␣
↪half power point of the primary beam.
mid_size: Image has shape 8192 by 8192, 4 polarisations, and size 2.000GB
mid_size: W processing requires 428 w planes of step 238.8 wavelengths and maximum␣
↪support 12 pixels
mid_size: Allowing fractional bandwidth 0.100  to fill in the uv sampling.
nmid_size: Starting MID simulation sizing

{'band': 'B2',
 'context': 's3sky',
 'declination': -45.0,
 'dela': 0.02,
 'fractional_bandwidth': 0.1,
 'guardband': 2.0,
 'integration_time': 80.0,
 'output_msname': 'mid_simulation.ms',
 'ra': 15.0,
 'rmax': 200000.0,
 'time_range': [-4.0, 4.0],
 'verbose': 'False'}
mid_size: Using only dishes within 200000.0m of the array centre requires 1161 channels␣
↪of 1.17e+05Hz and integration time 7.4s, the MS size is 8512.629GB
mid_size: The time and frequency sampling is to provide no more than 2% smearing at the␣
↪half power point of the primary beam.
mid_size: Image has shape 98304 by 98304, 4 polarisations, and size 288.000GB
mid_size: W processing requires 4065 w planes of step 238.8 wavelengths and maximum␣
↪support 132 pixels
mid_size: Allowing fractional bandwidth 0.100  to fill in the uv sampling.
```

CHAPTER

SIX

RFI SIMULATIONS

These simulations are designed to provide simulated data containing received signal from input Radio Frequency Interference (RFI) sources. The aim of providing these simulations is to enable testing of RFI-mitigation techniques and to understand the effects of RFI signals at the SKA Mid site.

The **ska-sim-mid/rfi** directory contains a python script, a bash script and a data directory. For the main simulation providing output images and measurement sets, the bash script rfi_sim.sh should be used. This has three main parts:

- First, it runs the mid_rfi_simulation.py script, which takes an HDF5 file as input, and produces a measurement set with RFI signal added to the appropriate frequency channels and time samples.
- Second, the RASCIL Imager app is called using the measurement set as input and FITS images are generated.
- Finally, the RASCIL vis_ms visualization app is called to generate various plots from the measurement set.

6.1 Input data

The input data need to be supplied in the form of an HDF5 file, with the following information and array structure:

- Source ID, string, dimensions: (nsources)
- Source type, string, dimensions: (nsources)
- Time samples, timestamp, dimensions: (ntimes)
- Frequency channels, FP64, dimensions: (nfreqs), units: [Hz]
- SKA station ID, string, dimensions: (nstations)
- Apparent source coordinates in antenna rest frame, FP64, dimensions: (nsources, ntimes, nants, 3) These are [azimuth, elevation, distance], units: [degree, degree, m]
- Transmitter power as received by an isotropic antenna, FP64, dimensions: (nsources, ntimes, nants, nfreqs) This does not include the antenna beam pattern which will be applied in the visibility simulation pipeline. units: [W/(Hz m^2)]

Note: at the moment the code is set up to handle Unix time for the provided time samples. This will be updated once the input RFI data of aeronautical sources are generated using MJD [s].

6.2 Input data

The most up-to-date RFI source data is located in **ska-sim-mid/rfi/data/aeronautical_v2_SKA_Mid_coord/aeronautical_sim_datacub**
This file contains data for 17 aircraft (RFI sources) for 180 time samples covering one hour of observations, for the full AA1 Mid array (197 antennas) and a single frequency channel (1090 MHz).

ska-sim-mid/rfi/data/aeronautical_v2_SKA_Mid_coord/SKA1MID.csv contains the Mid antenna coordinates used for generating the above data file. The mid RFI simulation code is also updated to generate the telescope configuration using this file. Alternatively, one can use the default RASCIL Mid configuration for the simulations.

6.3 Beam gain

The beam gain is applied as part of the RASCIL RFI simulations script simulate_rfi_block_prop, specifically the apply_beam_gain_for_mid function. A voltage pattern for the Mid antennas is generated, together with pointing tables (latter is one for each RFI source). These are used to calculate the gain tables, which are applied to the BlockVisibilities containing the RFI signal, at the end of the simulation.

At the moment we do not have information about the far outside lobes of the Mid dishes, we can only approximate the beam there, which distorts the results further from the beam centre.

We recommend that when using the data file listed above, you do not apply the beam gain directly, or you set the phase centre to be low on the sky, close to the horizon. The aircraft in the data are closer to the horizon, no more than 20 degrees above it. Since the beam we have is small, if the pointing (phase centre) is far from the horizon, all of the RFI signal is missed and removed from the results when the beam gain is applied. If the pointing is set to be closer to the horizon, then some of the aircraft signal will appear in the results (attenuated by the beam as expected).

6.4 CLI

Simulate RFI data with RASCIL

```
usage: mid_rfi_simulation.py [-h] [--input_file INPUT_FILE]
                             [--output_dir OUTPUT_DIR] [--msout MSOUT]
                             [--antenna_file ANTENNA_FILE] [--seed SEED]
                             [--noise NOISE] [--ra RA] [--dec DEC]
                             [--nchannels NCHANNELS]
                             [--frequency_range FREQUENCY_RANGE FREQUENCY_RANGE]
                             [--apply_primary_beam APPLY_PRIMARY_BEAM]
                             [--return_average RETURN_AVERAGE]
                             [--time_average TIME_AVERAGE]
                             [--channel_average CHANNEL_AVERAGE]
```

6.4.1 Named Arguments

--input_file	Full path to the HDF5 file, which contains necessary RFI information for each RFI source.
--output_dir	Output directory for storing files
--msout	Name for MeasurementSet. If empty string, no MeasurementSet is written
--antenna_file	Path to antenna file. If left empty, MID configuration from RASCIL will be used.
--seed	Random number seed
--noise	Add random noise to the visibility samples?
--ra	Right Ascension (degrees)
--dec	Declination (degrees)
--nchannels	How many channels to create within given –frequency_range
--frequency_range	Frequency range (Hz)
--apply_primary_beam	Apply primary beam to RFI sources?
--return_average	Return visibility with time and frequency averaged data?
--time_average	Number of integrations in a chunk to average
--channel_average	Number of channels in a chunk to average

CHAPTER

SEVEN

INSTRUCTIONS

To install these scripts, use git:

```
git clone https://gitlab.com/ska-telescope/sim/ska-sim-mid.git
```

Or download the source code from https://gitlab.com/ska-telescope/sim/ska-sim-mid

None of these simulations need to be installed into the python path. Instead please set the below environment variables:

```
export SSMROOT=/path/to/code
export SSMRESULTS=/path/to/results
```

In direction dependent simulations, one will also need the additional data resources (e.g. beam models) which requires setting the below environment variable:

```
export SSMRESOURCES=/path/to/resources
```

RASCIL must be installed - the options are via pip, git clone, or docker. The simplest approach is to use pip. In the ska-sim-mid directory, do:

```
pip3 install -r requirements.txt
```

A script can be run as:

```
export SMSROOT=`pwd`
cd sizing/scripts
sh make_sizes.sh
```

The resource files for the SKA direction-dependent MID simulations are 80GB in size and are kept on the Google Cloud Platform. The python command line tool gsutil allows for interacting with the Google Cloud Platform:

```
https://cloud.google.com/storage/docs/gsutil
```

After installing gsutil, you may download the resources as follows:

```
cd resources
gsutil -m rsync -r gs://ska1-simulation-data/resources/mid/beam_models/ beam_models
gsutil -m rsync -r gs://ska1-simulation-data/resources/shared/screens screens
```