developer.skatelescope.org Documentation

Release 0.1.0-beta

Marco Bartolini

Nov 10, 2021

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.

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

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

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).

Todo:

- Insert todo's here

CHAPTER

FOUR

SIZING OBSERVATIONS

This script estimates the size of CASA MeasurementSets for various MID observations. it does this by constructing a scaled down BlockVisibility 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

FIVE

RFI SIMULATIONS

These simulations are designed to provide simulated data containing received signal from input RFI sources. The aim of providing these simulations is to enable testing of RFI-mitigation techniques and specifically to understand the level of low-level RFI (radio frequency interference) likely to be present for SKA Mid observations and the limitations of the standard mitigation software.

The **ska-sim-mid/rfi** directory contains a python script, and a bash script. 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.
- Finally, the RASCIL vis_ms visualization app is called to generate various plots from the measurement set.

5.1 Input data

The input data needs 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 we have updated RFI input of aeronautical source. The agreed format will be in MJD [s].

An example input HDF5 file can be found in **ska-sim-mid/tests/rfi/test_data**.

5.2 Beam gain

Beam gains are not applied at the time of writing. These will need further development. The beam gains will be obtained from gain tables created within RASCIL.

5.3 CLI

Simulate RFI data with RASCIL

```
usage: mid_rfi_simulation.py [-h] [--input_file INPUT_FILE]
                             [--output_dir OUTPUT_DIR] [--msout MSOUT]
                             [--seed SEED] [--noise NOISE] [--ra RA]
                             [--declination DECLINATION]
                             [--nchannels_per_chunk NCHANNELS_PER_CHUNK]
                             [--frequency_range FREQUENCY_RANGE FREQUENCY_RANGE]
                             [--apply_primary_beam APPLY_PRIMARY_BEAM]
                             [--return_average RETURN_AVERAGE]
                             [--time_average TIME_AVERAGE]
                             [--channel_average CHANNEL_AVERAGE]
                             [--use_dask USE_DASK]
```

5.3.1 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
--seed	Random number seed
--noise	Add random noise to the visibility samples?
--ra	Right Ascension (degrees)
--declination	Declination (degrees)
--nchannels_per_chunk	Number of channels in a chunk
--frequency_range	Frequency range (Hz)
--apply_primary_beam	Apply primary beam to RFI sources?
--return_average	Return block 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
--use_dask	Use dask to distribute processing?

CHAPTER

SIX

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
```