HSIM Manual

v310

1 June, 2021

Contents

Revision History

Version	Date	Comments
1.0	2018-09-27	HSIM v204 release
1.1	—	HSIM v210 release
1.2	2019-03-28	HSIM v211 release
1.2b	2019-04-30	HSIM v212 release
1.3	2019-10-11	Detector systematics
1.3b	2019-10-16	HSIM v300 release
1.4	2019-12-16	HSIM v302 release
1.5	2020-04-21	HSIM v303 release
1.6	2021-06-01	HSIM v310 release

Authors of this document

Miguel Pereira Santaella (CAB)
Laurence Routledge (University of Oxford)

1. Quick Start Guide

1.1 Introduction and Installation

HSIM is a dedicated pipeline for simulating observations with HARMONI on the Extremely Large Telescope. HSIM takes high spectral and spatial resolution input data cubes, encoding physical descriptions of astrophysical sources, and generates mock observed data cubes. The simulations incorporate detailed models of the sky, telescope, instrument, and detectors to produce realistic mock data (Zieleniewski et al., 2015).

HSIM is programmed in Python3 and the source code can be found at https://github.com/HARMONI-ELT/HSIM. It does not require any installation, just clone the GitHub repository

```
$ git clone https://github.com/HARMONI-ELT/HSIM.git
```

HSIM depends on the following Python packages to work:

- astropy 4.1
- numpy 1.19.5
- scipy 1.5.4
- matplotlib 3.3.4
- photutils 0.7

The code has been tested with the indicated package version, although more recent releases of these packages are likely to work as well.

1.2 Running Simulations

1.2.1 Preparing the Input Datacube

Before running HSIM, you will need an input model of your astronomical object stored as a datacube in a FITS file. The recommended sizes for the spatial pixels (spaxels) are 1, 2, or 3 mas depending on the selected spaxel scale for HARMONI (see Table 1.1). If your datacube has a different spaxel scale, HSIM will automatically interpolate/rebin the input datacube to the recommended values.

Similarly, it is recommended to oversample the spectral dimension of the input cube by a factor of 4 with respect to the nominal resolving power (R) of the selected grating (e.g., the spectral sampling for the K-grating R=7100 at 2.2μm should be 0.078 nm). If your input cube has a different spectral sampling, HSIM will interpolate/rebin it.

Table 1.1: Recommended input pixel size

HARMONI Spaxel scale (mas)	Recommended input scale (mas)
4×4	1
10×10	2
20×20	3
30×60	3

The information on the spatial and spectral sampling of the input cube are passed through the FITS file header to HSIM (Table 1.2 shows a summary of all the header keywords used by HSIM). In particular, the `CDELT1` and `CDELT2` header values are used to get the spatial scale, and `CDELT3` to obtain the spectral sampling. The spectral resolution of the input cube is indicated by `SPECRES` in wavelength units similar to `CDELT3`. The units of these values are those indicated by the `CUNIT1`, `CUNIT2`, and `CUNIT3` header values. For the spatial units (`CUNIT1` and `CUNIT2`), the accepted values are any angular units supported by astropy (e.g., `mas`, `arcsec`). For the spectral units, `CUNIT3` and `SPECRES`, HSIM recognizes any length unit supported by astropy (e.g., `angstrom`, `nm`, `micron`). The size of the input cube is obtained from `NAXIS1`, `NAXIS2`, and `NAXIS3`.

The wavelength of each slice of the data cube are also calculated from the FITS header using the following relation:

$$\lambda_i = \texttt{CRVAL3} + \texttt{CDELT3} \times (i - \texttt{CRPIX3}) \tag{1.1}$$

where i is the slice number from 1 to `NAXIS3`. The wavelength range of the input cube and the selected grating should, at least, partially overlap.

Finally, the input data cube should be in surface brightness units indicated by `BUNIT`. The accepted values are any surface spectral flux density units supported by astropy (e.g., `erg/s/cm2/um/arcsec2`, `erg/s/cm2/AA/arcsec2`, `MJy/sr`).

1.2.2 HSIM Gratings

The gratings that can be selected in HSIM are detailed below in Table 1.3.

1.2.3 Detector Systematics

If chosen, HSIM can be run with more detailed detector systematics to simulate some of the features seen on infra-red detectors. When this option is selected, HSIM creates 8 simulated HAWAII 4RGs where each pixel has a unique read noise value derived from analysing the KMOS H2RG detectors. The read noise values have a distribution, with the peak at the expected read noise value for the H4RGs. An example HSIM H4RG is seen in Figure 1.1. This also utilises the HXRG noise generation tool provided by Teledyne (Rauscher, 2015) to add additional features, such as ACN and amplifier bias differences. The parameters of these have been tuned to an expected performance level, however can be adjusted in the config file. Additionally, in the config file there is an option to use a different read noise file, by changing the `rn_file` to point to a different FITS file. This file determines the distribution the read noise will follow, so adding a file with a uniform value across the image will produce a flat distribution. The last

Table 1.2: FITS header keywords used by HSIM

Keyword	Description	Accepted values
	Spatial information	
NAXIS{1,2}	Number of pixels along x-axis and y-axis	
CTYPE1	Type of the spatial x-axis	x, ra, RA---SIN, RA---TAN
CTYPE2	Type of the spatial y-axis	y, dec, DEC--SIN, DEC--TAN
CDELT{1,2}	X and Y pixel size	
CUNIT{1,2}	Units of CDELT{1,2}	any angular unit supported by astropy (e.g., mas, arcsec)
	Spectral information	
NAXIS3	Number of spectral pixels	
CTYPE3	Type of the spectral axis	wavelength
CDELT3	Size of the spectral pixel	
SPECRES	Spectral resolution in wavelength units	
CRPIX3	Reference pixel for the spectral axis (see Equation 1.1)	
CRVAL3	Wavelength of the reference pixel	
CUNIT3	Units of CDELT3, CRVAL3, and SPECRES	any length unit supported by astropy (e.g., angstrom, AA, nm, micron)
	Flux units	
BUNIT	Flux units of the cube	any surface spectral flux density unit supported by astropy (e.g., erg/s/cm2/um/arcsec2, erg/s/cm2/AA/arcsec2, MJy/sr)

Table 1.3: HSIM grating parameters

Grating	λ_{min} (μm)	λ_{max} (μm)	R
Low resolution			
V+R	0.458	0.820	3100
Iz+J	0.811	1.369	3355
H+K	1.450	2.450	3355
Medium resolution			
Iz	0.830	1.050	7104
J	1.046	1.324	7104
H	1.435	1.815	7104
K	1.951	2.469	7104
High resolution			
z-high	0.828	0.902	17385
J-short	1.012	1.102	17385
J-long	1.098	1.189	17385
H-high	1.538	1.678	17385
K-short	2.017	2.201	17385
K-long	2.199	2.400	17385

detector systematics option, `force_new`, determines whether or not to generate a new set of detectors with each simulation. By default, the first time you run detailed detector systematics, it will save the read noise maps generated to a location specified when initialising HSIM, and use these in subsequent runs of the code to speed up the simulation. New shot noise will still be generated each time. If desired however, new detectors can

be made each time, either by setting this parameter to False, or by manually deleting the created `HARMONI_dets.fits` file in the location in which they are saved.

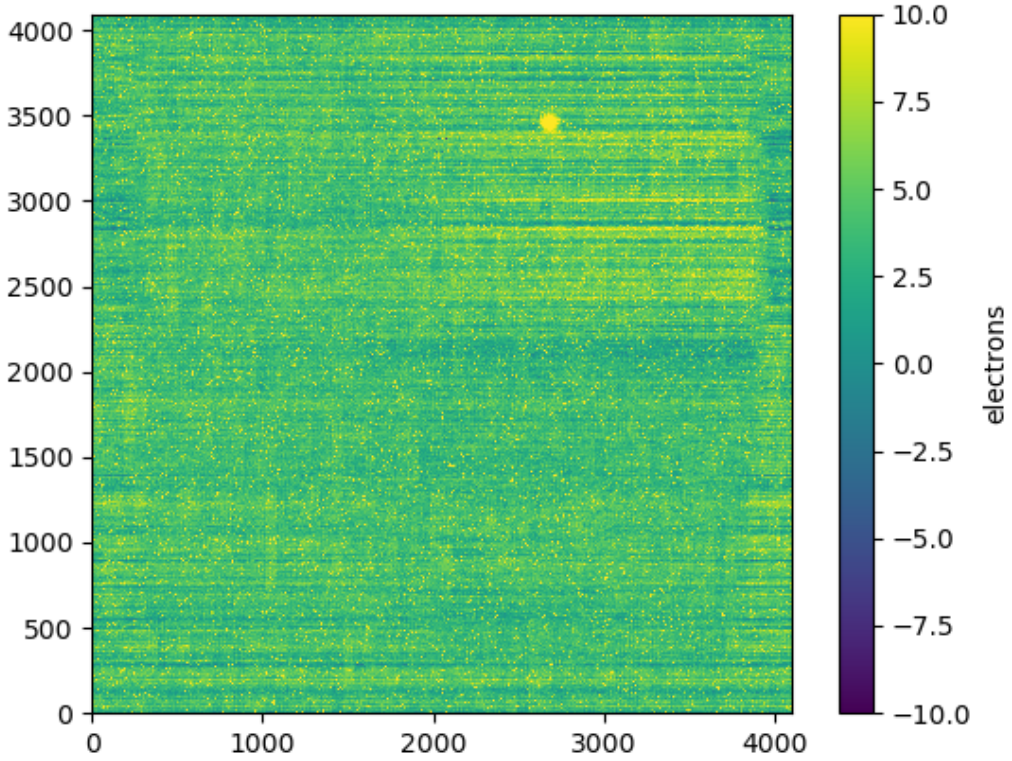

Figure 1.1: HSIM simulated detector.

The simulated data cube is then sliced up and added to these detectors, in a manner similar to the effect of the image slicer. This ensures that features that appear linear on the detectors no longer appear linear in the output data cube. The simulation is then folded back into a standard data cube to maintain the 'cube in, cube out' nature of HSIM. When creating the reduced cube, a separate instance of the detectors is generated and used in the reduction, so the residual features will remain in the reduced cube. A separate cube is returned containing the detectors used in the simulation.

1.2.4 Running HSIM

Once the input data cube has been prepared, the next step is to run HSIM. HSIM comes with a graphical user interface (GUI) which can be used to define the input parameters of the simulations. Alternatively, the same options can be accessed from a configuration file or directly from the command line.

GUI

The GUI provides a relatively easy way to list the available options. To launch HSIM in the GUI mode, type the following command

```
$ python3 hsim3.py
```

Then, a window similar to that shown in Figure 1.2 should appear. In that window, it is possible to select the input cube and the output directory and define all the parameters needed by HSIM to run the simulation (these parameters are explained in Section 1.2.5). Clicking on the `Commence simulation` button will start the simulation. You can see how the simulation advances in the terminal window from where HSIM was called.

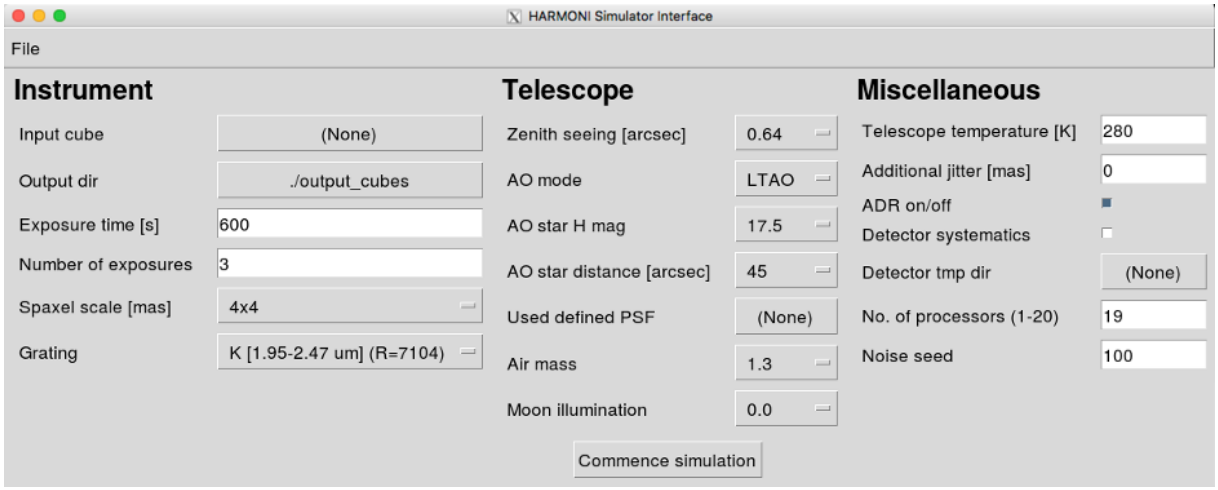

Figure 1.2: HSIM GUI.

Command-line

All the GUI options can be directly accessed from the command line mode using the following command

```
$ python3 hsim3.py -b --arg1 value1 --arg2 value2 ...
```

where `arg*` are the parameters of the simulation (see Section 1.2.5). `-b` forces HSIM to start the simulation not showing the GUI. Otherwise, the parameters are read from the command line (and the configuration file if specified with `-c`, see below) and then the GUI will appear. Adding `-h` will show all the command line options and their default and accepted values.

Configuration file

It is also possible to define the simulation input parameters using a configuration file using

```
$ python3 hsim3.py -b -c config_file
```

Additional input parameters can be added to the command line if required. As in the command-line case, `-b` forces HSIM to start the simulation not showing the GUI.

In the case of conflicting input parameters, preference is given to the GUI, then the command-line, and finally the configuration file.

1.2.5 List of Options

The available HSIM options are listed below. The complete list is also available by running

```
python3 hsim3.py -h
```

These options are preceded by `--` when included in the command-line. When they are used in a configuration file the "`-`" in their names should be replaced by "`_`" (see example below).

`input-cube`: Path of the FITS input datacube.

`output-dir`: Output directory.

`grating`: HARMONI grating. Available values are: `V+R, Iz+J, H+K, Iz, J, H, K, z-high, J-short, J-long, H-high, K-short, K-long`.

`spaxel-scale`: HARMONI spatial scale in mas. Available values are: `4x4, 10x10, 20x20, 30x60, 60x60, 120x60`. The 60x60 and 120x60 scales are only available for the V+R grating.

`exposure-time`: Integration time of each exposure in seconds (e.g., `600`).

`n-exposures`: Number of exposures (e.g., `3`).

`ao-mode`: Adaptive optics mode of the observations. Available values are: `LTAO, SCAO, noAO, Airy, User`. `LTAO` corresponds to the Laser Tomography Adaptive Optics mode, `SCAO` to the Single Conjugate Adaptive Optics mode, `noAO` indicates that no adaptive optics corrections are applied, `Airy` uses the diffraction limit of the ELT, and `User` uses the 2D user-defined PSF indicated by the parameter `user-defined-psf`.

`user-defined-psf`: FITS file containing a 2D user-defined PSF.

`ao-star-hmag`: H magnitude of the LTAO AO star. Available values are: `15.0, 17.5, 19.0`. The default is `17.5`.

`ao-star-distance`: Distance from the HARMONI FoV center to the LTAO AO star in arcsec. Available values are: `30, 45, 60`. The default is `45`.

`zenith-seeing`: Optical (500 nm) zenith atmospheric seeing FWHM in arcsec. Available values are: `0.43, 0.57, 0.64, 0.72, 1.04`.

`air-mass`: Average air mass during the HARMONI observation. Available values are: `1.1, 1.3, 1.5, 2.0`, which correspond to zenith angles of 25°, 40°, 48°, and 60°, respectively.

`moon-illumination`: Fraction of the Moon that is illuminated at the time of the observation. A 30° separation between target and Moon is assumed. Available values are: `0, 0.5, 1.0`. The default is `0.0`.

`detector-systematics`: Indicates if the detector systematics are simulated. It only affects the observed outputs (see Section 1.2.6). Available values are `True, False`.

`detector-tmp-path`: Path of the desired location to save the created detectors if using advanced detector systematics. Use `None` to use the default location (`sim_data/detectors`).

`adr`: When this option is set, the atmospheric differential refraction is first simulated and then corrected in the output cubes in a way similar to what is expected from the pipieline. If it is set to `False`, the atmospheric differential refraction and its correction are not simulated. Available values are `True, False`. The default is `True`.

`telescope-temp`: Temperature of the site/telescope in K. This is used to calculate the telescope and part of the instrument background emission. The default value is `280`.

`extra-jitter`: Additional telescope PSF blur in mas. The default is `0`. It is possible to define independent jitter values for the each axis (e.g., `2x4`).

`noise-seed`: Random number generator seed using during the simulation. It only affects the observed outputs (see Section 1.2.6).

`n-cpus`: Number of processors used by HSIM.

Configuration file

Configuration files are text files containing the parameters that define the simulation. All the parameters listed in Section 1.2.5 can be included in the configuration file, but all the "-" in their names should be replaced by "_". A minimal example configuration file is shown below:

```
[HSIM]
input_cube = datacube.fits
output_dir = output_cubes
grating = J
spaxel_scale = 10x10
exposure_time = 300
n_exposures = 10

ao_mode = LTAO
ao_star_Hmag = 17.5
ao_star_distance = 45

zenith_seeing = 0.72
air_mass = 1.1
```

1.2.6 Output Files

HSIM stores the results of the simulation in the specified output directory (the default output directory is `output_cubes`). The name of the output files begins with the name of the input data cube and it is followed by a suffix indicating the content of the file. The input parameters of the simulation are stored in the header of the FITS files and also in the log file.

1. `.log`. Detailed log of the simulation. The parameters of the simulations are listed in the configuration file format at the beginning of the log. The last line of this file shows if any problem was found during the simulation.
2. `_noiseless_obj_plus_back.fits`. Data cube containing the expected number of electrons per pixel detected by HARMONI due to the target and the background (sky, telescope, and instrument) emissions. Cross-talk is also applied to the noiseless outputs to match the LSF of the observed and reduced outputs cubes.
3. `_noiseless_obj.fits`. Expected number of electrons per pixel solely due to the target.
4. `_noiseless_back.fits`. Expected number of electrons per pixel from the background sources (sky, telescope, and instrument).
5. `_observed_obj_plus_back.fits`. Simulated observed data cube taking into account the photon noise, read noise, dark current, and cross-talk.
6. `_observed_back.fits`. Simulated observed sky background taking into account the photon noise, read noise, dark current, and cross-talk.
7. `_reduced.fits`. Simulated reduced data cube created by subtracting a simulated sky observation to the `_observed_obj_plus_back.fits` data cube.
8. `_reduced_SNR.fits`. Signal to noise ratio of the reduced data cube per pixel.
9. `_all_dets.fits`. If detector systematics are used, a fits file containing all of the detectors used in the simulation, in the detector plane.
10. `_used_dets.fits`. If detector systematics are used, a fits file containing the portion of the detectors used, in the plane of the datacube.
11. `_std.fits`. Noise standard deviation. Includes object and sky noise (Poisson) and dark current and read noise (Gaussian).
12. `_PSF.fits`. Point spread function at the mean wavelength of the input data cube. The PSF has the same dimensions and sampling as the output data cubes.
13. `_PSF_internal.fits`. Oversampled point spread function at the mean wavelength of the input data cube used internally. The PSF is oversampled by a factor of 5 to 20 with respect to the spaxel scale of the output data cubes.
14. `_total_em.pdf`. Plot of the background emission broken up into the individual modeled components. The average HARMONI to telescope+sky background ratio is indicated.
15. `_total_tr.pdf`. Plot of system transmission broken up into the individual modeled components.
16. `_total_tr.txt`. Total transmission stored as a text file.

1.2.7 AO jitter in HSIM

To calculate the AO performance, HSIM adds the following jitter to the PSF depending on the AO mode and observing conditions.

For SCAO, the jitter is always 2 mas. For LTAO, the jitter is estimated from the H magnitude and distance to the AO star according to Table 1.4. The jitter value from this table, j_{star}, is then scaled depending on the seeing and air mass. To do this, the input zenith seeing, α_{zenith}, is transformed into an effective seeing, $\alpha(\theta)$, at the zenith angle θ (i.e. air mass) using

$$\alpha(\theta) = \frac{\alpha_{\text{zenith}}}{\sqrt{\cos\theta}} \tag{1.2}$$

Table 1.4: LTAO jitter in mas

H mag \ Distance	15”	30”	45”
15.0	2	3	4
17.5	3	4	6
19.0	5	7	10

Table 1.5: Seeing scaling jitter factor

Effective Seeing (arcsec)	f_s
0.64	1.0
0.74	1.5
1.04	2.0
1.40	3.0

Table 1.6: Air mass scaling jitter factor

Air mass	f_a
1.1	1.00
1.3	1.02
1.5	1.20
2.0	1.40

A seeing scaling factor, f_s, is derived from the effective seeing interpolating the values in Table 1.5. The air mass scaling factor, f_a, is obtained from Table 1.6. The final LTAO jitter is $f_s \times f_a \times j_{star}$.

Bibliography

Rauscher B., 2015, Publications of the Astronomical Society of the Pacific, 127

Zieleniewski S., Thatte N., Kendrew S., Houghton R. C. W., Swinbank A. M., Tecza M., Clarke F., Fusco T., 2015, MNRAS, 453, 3754