

---

# **developer.skao.int Documentation**

***Release 6.0.1***

**Marco Bartolini**

**Jul 14, 2023**



## OVERVIEW

<b>1</b>	<b>Design</b>	<b>3</b>
<b>2</b>	<b>MID Theoretical Background</b>	<b>5</b>
<b>3</b>	<b>MID Implementation</b>	<b>13</b>
<b>4</b>	<b>MID REST API</b>	<b>15</b>
<b>5</b>	<b>Further Work</b>	<b>19</b>
<b>6</b>	<b>LOW Theoretical Background</b>	<b>21</b>
<b>7</b>	<b>LOW Implementation</b>	<b>23</b>
<b>8</b>	<b>LOW REST API</b>	<b>25</b>
<b>9</b>	<b>ska_ost_senscalc.mid</b>	<b>27</b>
<b>10</b>	<b>ska_ost_senscalc.subarray</b>	<b>29</b>
<b>11</b>	<b>ska_ost_senscalc.utilities</b>	<b>31</b>
<b>12</b>	<b>ska_ost_senscalc.mid_utilities</b>	<b>33</b>
<b>13</b>	<b>ska_ost_senscalc.low.api</b>	<b>35</b>
<b>14</b>	<b>ska_ost_senscalc.low.validation</b>	<b>37</b>
<b>15</b>	<b>ska_ost_senscalc.low.model</b>	<b>39</b>
<b>16</b>	<b>ska_ost_senscalc.low.service</b>	<b>41</b>
<b>17</b>	<b>ska_ost_senscalc.low.calculator</b>	<b>43</b>
<b>18</b>	<b>ska_ost_senscalc.low.sefd_lookup</b>	<b>45</b>
	<b>HTTP Routing Table</b>	<b>47</b>



The Sensitivity Calculator provides functionality for SKA scientists and engineers to estimate sensitivities and integration times for observations on both SKA LOW and MID.

This document describes the theoretical background, the current Calculator implementation, explaining the decisions behind its design, and links to a space where future work can be planned.

For more details on developing and deploying the code, see the [README](#)



## 1.1 Goals and Considerations

A prime consideration in the design of the calculator is exactly how the user will interface with it. Some users may prefer a simple, accessible interface, e.g. a web page, while others may prefer to be able to download a GUI to their own device. Some may even prefer to access the calculator API directly to use with their own code. Importantly, however, the sensitivity calculator will ultimately be part of some larger observing tool. The calculator is expected to provide the user with a report for attachment to their observing proposal supporting their use of the telescope. After speaking to the developer of the ALMA Observing Support Tool, it became clear that the vast majority of their users would use the web-based tool where possible and nearly always have internet access when wanting to use the tool. Therefore it was decided that the prototype calculator should use a web-based interface to demonstrate functionality, but feature a distinct front- and back-end, allowing the interface to be modified, or for other interfaces to be added (if needed) as the project evolves. Because the calculator is publicly available via the SKA GitLab, anyone who wants to directly interact with the source code can do so.

The scientific model behind the sensitivity calculation will be updated as the life-cycle of the telescope progresses. At the time of writing it is based on the calculation framework of ‘SKA1: Design Baseline Description - SKA-TEL-SKO-0001075’ and ‘SKA1 System Performance Assessment Report - SKA-TEL-SKO-0001089’, with some additional information from the earlier document [Anticipated SKA1 Science Performance](#). Once the telescope is live, we should be able to actively record, for example, the system temperature. This, among other possible developments, will change how the sensitivity calculator functions. In addition, with the current calculator being a prototype, a number of features will certainly be added as time goes on. With all of this considered, it is sensible to maintain a modular design for the calculator back-end, where functionality is separated into independent modules. This means that if, say, the model describing the receiver temperature for SKA1 dishes is changed, the relevant code can easily be modified and the rest of the program should still run without any issues.

One of the desired features of the calculator is to be able to both calculate sensitivity, given an on-source integration time (and the other required parameters) and calculate the integration time required to reach a given sensitivity. The user should also be offered a range of different observing modes, so they can calculate e.g. total continuum noise, line noise, etc. These observing modes are not mutually exclusive, however. A user may be interested in performing a continuum observation with a number of zooms and would therefore want to know the sensitivity they could obtain in each case. Allowing for this while also allowing the user to swap between calculating sensitivity and integration time potentially makes both the front- and back-end design quite complicated. The solution ultimately was to separate out the different observing modes into individual tabs.





## MID THEORETICAL BACKGROUND

### 2.1 Reference Documents

RD1	Anticipated SKA1 Science Performance
RD2	‘SKA1 Performance Assessment Report’ SKA-TEL-SKO-0001089

### 2.2 Applicable Documents

AD1	An improved source-subtracted and destriped 408 MHz all-sky map
-----	---

An overview of the theoretical performance of SKA Mid comprising SKA1 and MeerKAT dishes is given in RD1. A more detailed analysis is given in RD2 for an SKA Mid made up of only SKA1 dishes. We are grateful to Songlin Chen for help in navigating and understanding the documentation.

The Mid Sensitivity Calculator (SC) was originally implemented following the theoretical framework of RD1 but is moving to the more rigorous framework of RD2, though some details remain simplified.

### 2.3 Dish SEFD

The ‘system equivalent flux density’ (SEFD) for a single dish is the flux density of a source that produces a signal equal to the background power of the system:

$$SEFD_{dish} = \frac{2kT_{sys}}{\eta_A A}$$

where:

- $k$  is the Boltzmann constant so that  $kT_{sys}$  measures the power received from background emission and all other sources of unwanted signal within the system, that is  $T_{sys} = T_{spl} + T_{sky} + T_{rcv} + T_{cmb} + \dots$
- $\eta_A$  is the dish efficiency
- $A$  is the geometric dish area.
- The 2 is there because a radio telescope measures only one polarization and it is assumed for this purpose that the other polarization has the same strength.

## 2.4 Array SEFD

SKA Mid is an interferometer that works by combining the signal from multiple dishes. There are 2 types of dishes involved, SKA1 and MeerKAT, with distinct characteristics. It can be shown, by adding up the signals from each baseline, that the array SEFD is given by:

$$SEFD_{array} = \frac{1}{\sqrt{\frac{n_{SKA}(n_{SKA}-1)}{SEFD_{SKA}^2} + \frac{2n_{SKA}n_{MeerKAT}}{SEFD_{SKA}SEFD_{MeerKAT}} + \frac{n_{MeerKAT}(n_{MeerKAT}-1)}{SEFD_{MeerKAT}^2}}}$$

where:

- $n_{SKA}$  is the number of SKA antennas
- $n_{MeerKAT}$  is the number of MeerKAT antennas
- $SEFD_{SKA}$  is the SEFD computed for an individual SKA antenna
- $SEFD_{MeerKAT}$  is the SEFD computed for an individual MeerKAT antenna.
- and the assumption has been made (?) that all baselines are equally efficient.

## 2.5 Array Sensitivity

The ‘sensitivity’ of a radio telescope is an overloaded term. For the purpose of the SC we define the sensitivity as the minimum detectable Stokes I flux ( $1\sigma$ ). This is equal to the noise on the background power, obtained using the radiometer equation  $\sigma = SEFD/\sqrt{2Bt}$ , corrected for atmospheric absorption:

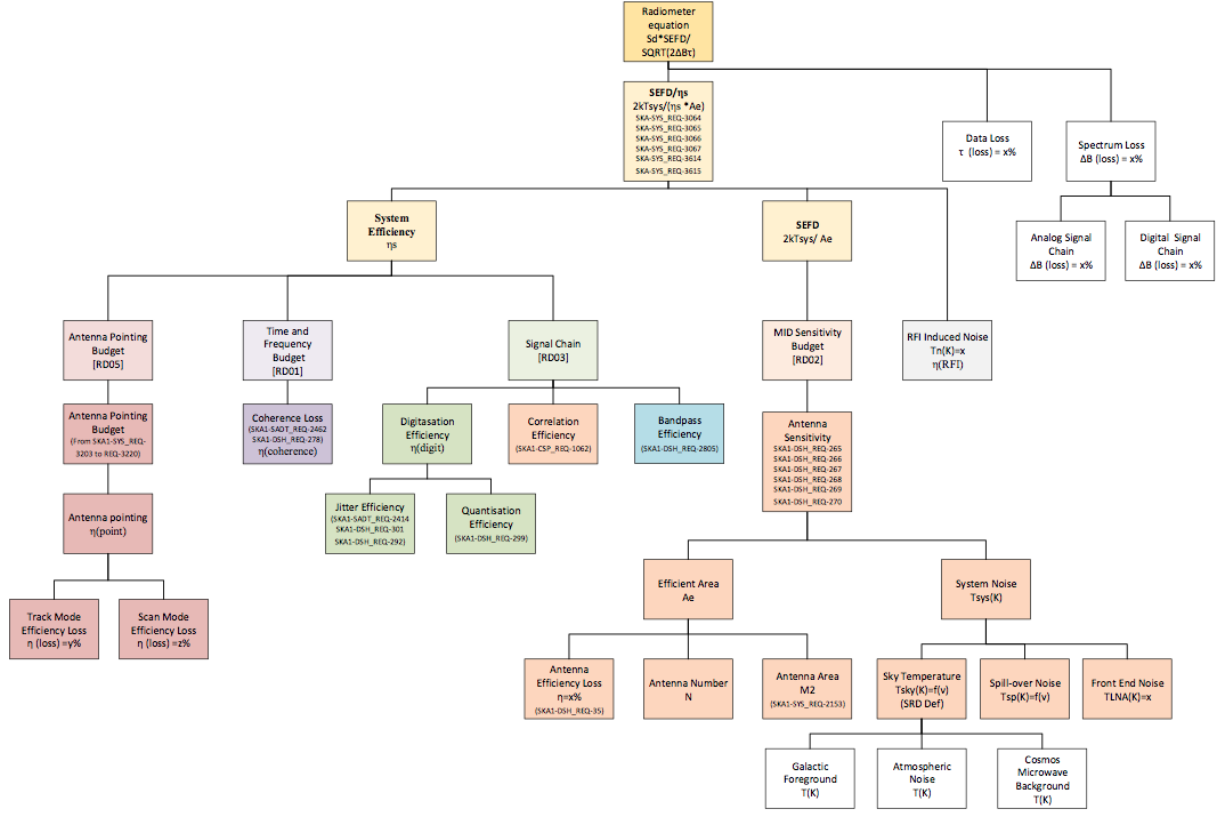
$$\Delta S_{min} \exp(-\tau_{atm}) = \frac{SEFD_{array}}{\eta_s \sqrt{2Bt}} Jy$$

where:

- $\Delta S_{min}$  is the source flux density above the atmosphere
- $\eta_s$  is the efficiency factor of the interferometer
- $B$  is bandwidth
- $t$  is integration time
- $\tau_{atm}$  is the optical depth of the atmosphere towards the target
- the formula applies to the centres of fields-of-view where the dish aperture response is unity.

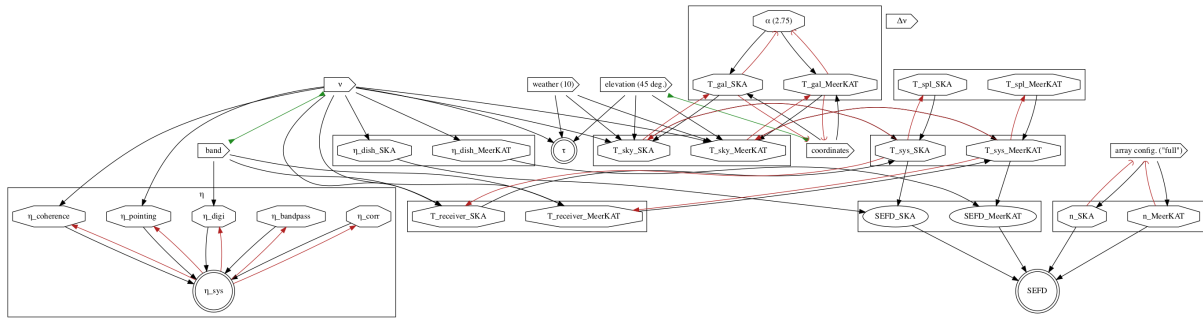
## 2.6 Dependency Tree

The devil is in the detail of calculating  $T_{sys}$  and the efficiency factors  $\eta_A$  and  $\eta_s$ . Fig.1 shows how these values depend on other factors that must be estimated.



**Figure 1** . The dependency tree for factors in the sensitivity calculation (from RD2).

The development of a new API following best practices allowed the refactoring of the sensitivity calculator class to implement the dependencies between variables. The last implementation of the dependency tree in the sensitivity calculator is shown in the following image:



**Figure 2** . The dependency tree for factors in the new implementation of the sensitivity calculator. Boxes with a triangular shape on the right are the fundamental inputs for the calculator. Variables that are in an octagon can be overridden in expert mode (see later). Variables in double circles are the result of the calculation through the dependencies. The black arrows mark dependencies. The green lines sanity and consistency checks. Red arrows indicate variables that are overridden if one of the downstream variables is set. If their arrow tip is circular it means that all the lines pointing to a variable (red arrows) must be activated so that the variable is actually overridden.

The new implementation will be used from API v1 onwards.

## 2.7 System Temperature

The system temperature is given by:

$$T_{sys} = T_{spl} + T_{sky} + T_{rcv}$$

where:

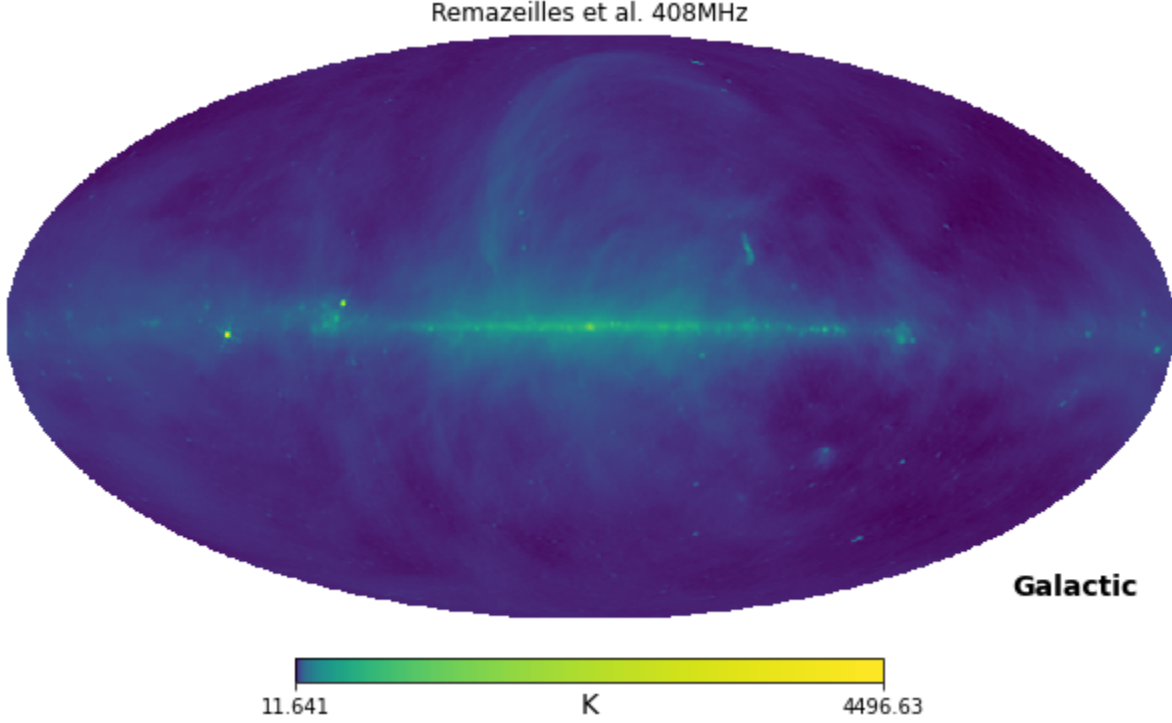
$$T_{sky} = T_{CMB} + T_{gal} + T_{atm}$$

and:

- $T_{spl}$  is the spillover temperature, measuring power from the ground reaching the receiver. Currently this is set to 3K for SKA1 dishes and 4K for MeerKAT.
- $T_{rcv}$  measures noise from the receiver and electronics, depending on band and dish type.
- $T_{sky}$  is the total emission from the sky.
- $T_{CMB}$  is the cosmic microwave background, 2.73K.
- $T_{gal}$  is the Galactic astronomical emission in the target direction.  $T_{gal} = T_{408}(0.408/\nu_{GHz})^{\alpha}$  K, where  $T_{408}$  is the Galactic emission at 408MHz whose estimation is described in [Brightness at 408MHz](#).
- $T_{atm}$  measures the brightness of the atmosphere, which depends on weather, observing frequency and elevation.  $T_{atm}$  and  $\tau_{atm}$  at the zenith are interpolated from lookup tables of results from the CASA atmosphere module, run for a grid of frequencies and weather PWVs.  $T_{atm}$  at the target elevation is estimated by relating it to the physical temperature by  $T_{phys} \sim T_{atm}(1 - \exp(-\tau_{atm}))$ , where  $\tau_{atm}$  varies as  $\sec(z)$ .

## 2.8 Brightness at 408MHz

The brightness of the astronomical background signal at 408MHz is estimated using the all-sky non source-subtracted HEALPix map described by AD1 (Fig.2). The brightness seen by a dish is calculated by multiplying map pixels that lie under the beam by the beam profile. The beam is assumed to be Gaussian, truncated at a radius equal to the FWHM.



**Figure 3** . The all-sky 408Mhz map from AD1, used to calculate  $T_{408}$ .

## 2.9 Efficiencies

### 2.9.1 Aperture

Following RD2, the aperture efficiency  $\eta_A$  is given by:

$$\eta_a = \eta_{dish} \eta_{feed}$$

where:

$$\eta_{dish} = \eta_{block} \eta_{transp} \eta_{surface} \eta_{rad.r}$$

$$\eta_{feed} = \eta_{rad.f} \eta_{ill}$$

and:

- $\eta_{dish}$  accounts for the efficiencies attributable to the dish optics
- $\eta_{block}$  accounts for physical aperture blockage
- $\eta_{transp}$  accounts for losses by transmission through the reflector surface
- $\eta_{surface}$  accounts for all losses due to incoherent propagation through the optics, including panel roughness, systematic deformation and mis-alignment;
- $\eta_{rad.r}$  accounts for the Ohmic dielectric and scattering losses in the reflector system only
- $\eta_{feed}$  accounts for the efficiencies attributable to the feeds
- $\eta_{rad.f}$  accounts for feed mismatches and losses
- $\eta_{ill}$  is the efficiency due to the actual illumination pattern

Currently, the SC follows RD1 and calculates an overall  $\eta_{dish}$  from estimates of  $\eta_{hill}$ ,  $\eta_{surface}$  and  $\eta_{diffraction}$  (?).

## 2.9.2 Array

The system efficiency  $\eta_s$  is the result of multiplying together the following factors:

- **eta\_bandpass** This factor describes the loss of efficiency due to the departure of the bandpass from an ideal, rectangular shape. At present the value is set to 1.0.
- **eta\_coherence** This factor describes the loss of efficiency due to coherence loss on a baseline.

$$\eta_{coherence} = \exp - \frac{\langle \phi_\epsilon^2(t) \rangle}{2} = \exp - 2\pi^2 \nu_0^2 \langle \tau_\epsilon^2(t) \rangle$$

We take the coherence loss at 1s integration time, which is white phase-noise dominated. The total phase delay is due to the sum in quadrature of the phase delay of the clock and signal path on both receptors:

$$\langle \tau_\epsilon^2 \rangle = \langle \tau_{clk,i}^2 \rangle + \langle \tau_{clk,j}^2 \rangle + \langle \tau_{dsh,i}^2 \rangle + \langle \tau_{dsh,j}^2 \rangle$$

The signal path depends on the environment (atmosphere, gusty wind) and the calibration quality, which is quite complicated to estimate in practice. For now we adopt a value of  $\eta_{coherence} = 0.98$  at  $\nu_0 = 15.4GHz$  as coherence loss for the worst case, and scale it to the frequency of observation using the given formula.

- **eta\_correlation** This factor describes the loss of efficiency due to imperfection in the correlation algorithm, e.g. truncation error. Analysis described in “SKA CSP SKA1 MID array Correlator and Central beamformer sub element Signal Processing Matlab Model” (311-000000-007) shows that the CSP correlation efficiency is almost 100% in the case of zero RFI, and better than 98% in the case of strong RFI (defined as <10% RFI in the outside visibility query, what does this mean).

Currently the efficiency value is set to 0.98.

- **eta\_digitisation** This factor describes the loss of efficiency due to quantization during signal digitisation. The process is independent of the telescope and environment, and depends only on the ‘effective number of bits’ (ENOB) of the system, which depends in turn on digitiser quality and clock jitter, and on band flatness.

The values used for each band are as follows:

Band	ENOB	Band Flatness (dB)	$\eta$
Band 1	8	6.5	0.999
Band 2	8	6.5	0.999
Band 3	6	6.5	0.998
Band 4	4	6.5	0.98
Band 5a	3	4 (in any 2.5GHz BW)	0.955
Band 5b	3	4 (in any 2.5GHz BW)	0.955

- **eta\_point** This factor describes the loss of efficiency due to dish pointing errors. Here we currently use an approximate formula:

$$\eta_{point} \sim \frac{1}{1 + 8 \ln 2 \frac{\sigma_\theta^2}{FWHM^2}}$$

where FWHM is the beam full-width at half maximum power for the dish, given by the approximate formula  $FWHM \sim 66\lambda/D$  (degrees), and  $\sigma_\theta$  is the RMS pointing error.

### 2.9.3 Design Independent

This section lists efficiency factors that are independent of the telescope design.

- **eta\_rfi** This factor describes the loss of efficiency due to parts of the spectrum that are lost due to strong RFI noise corrupting the astronomical signal. Currently set to 1.
- **eta\_data\_loss** This describes the loss of observing time due to the need for calibration, time spent moving to source, etc.

It is currently not used in the calculator, so implicitly set to 1.

### 2.9.4 Sensitivity Degradation due to RFI

The effect of RFI is currently removed from the system efficiency budget because of the complexity of the RFI impact. Estimates for the impact of RFI are not solid and work continues to understand them.

- **Strong RFI** Strong RFI which results in saturation in the analogue chain or clipping in digitisation will be flagged. The data loss and spectrum loss are instrument independent.
- **Moderate RFI** Moderate RFI is not flagged but contributes significant input power and might induce extra noise in the digitisation and correlation processes.
- **Weak RFI** Weak RFI, or the high-order intermodulation components of strong and moderate RFI, contribute to the sensitivity in the form of additive system noise.



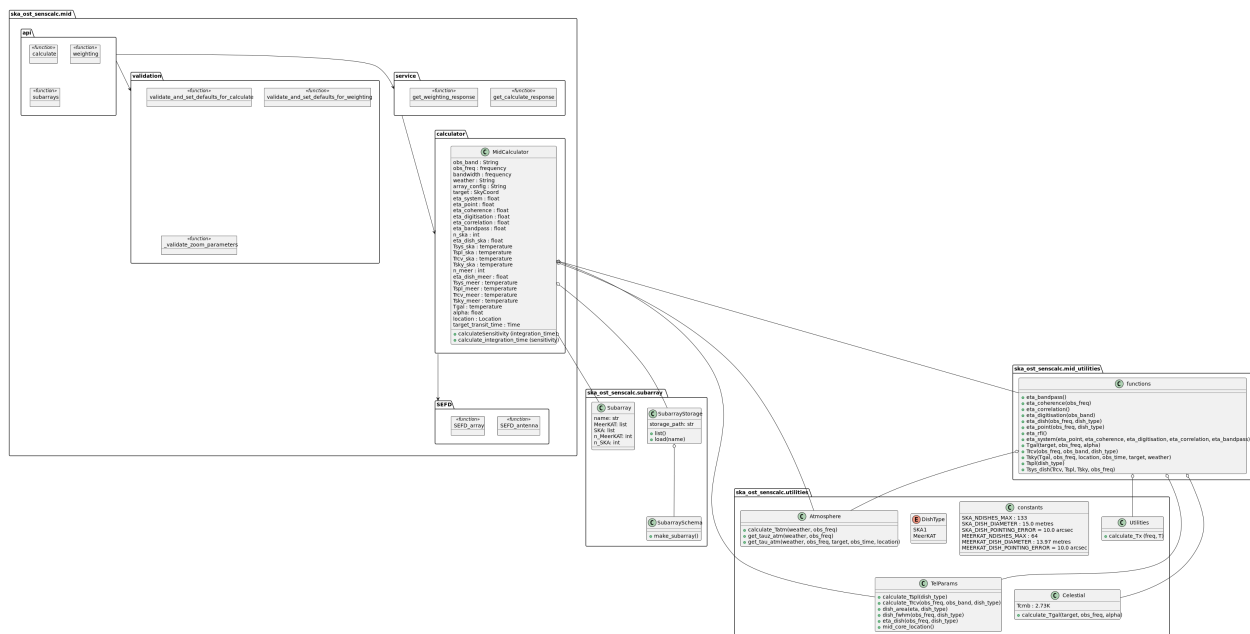


## MID IMPLEMENTATION

---

**Note:** The MID Calculator is in the process of being refactored into a more functional style, similar to LOW. This diagram reflects the current state, but in the end we expect the calculator.py modules to be more functional and rely less on object instantiation and state mutation, and the utilities classes to be structured differently.

---



**Figure 4 .** Class diagram of the Sensitivity Calculator back-end.

The class *MidCalculator* has 2 public methods: *calculate\_sensitivity* to get the array sensitivity in Jy for the given integration time, and *calculate\_integration\_time* to get the integration time required for the given sensitivity.

The *MidCalculator* constructor has a number of required parameters that define the observing configuration, target and weather. The rest default to None, in which case their values will be calculated automatically. The automatic values can be overridden by setting them here.

All parameters, internal variables and results that describe ‘physical’ measures are implemented as astropy Quantities to prevent mixups over units.



## MID REST API

### GET /calculate

Returns the integration time in seconds given a target sensitivity for a continuum observation

#### Query Parameters

- **rx\_band** (*string*) – Receiver band (Required)
- **frequency** (*number*) – Frequency (Hz) (Required)
- **ra\_str** (*string*) – Right ascension in ICRS frame (Required)
- **dec\_str** (*string*) – Declination in ICRS frame (Required)
- **array\_configuration** (*string*) – Array configuration
- **eta\_system** (*number*) – System efficiency. If provided, will be used instead of calculating it from eta\_pointing, eta\_coherence, eta\_digitisation, eta\_bandpass and eta\_correlation
- **eta\_pointing** (*number*) – Pointing efficiency
- **eta\_coherence** (*number*) – Coherence efficiency
- **eta\_digitisation** (*number*) – Digitisation efficiency
- **eta\_correlation** (*number*) – Correlation efficiency
- **eta\_bandpass** (*number*) – Bandpass efficiency
- **t\_sys\_ska** (*number*) – System Temperature of the SKA antennas (K)
- **t\_rx\_ska** (*number*) – Receiver Temperature of the SKA antennas (K)
- **t\_spl\_ska** (*number*) – Spillover Temperature of the SKA antennas (K)
- **t\_sys\_meer** (*number*) – System Temperature of the MeerKAT antennas (K)
- **t\_rx\_meer** (*number*) – Receiver Temperature of the MeerKAT antennas (K)
- **t\_spl\_meer** (*number*) – Spillover Temperature of the MeerKAT antennas (K)
- **t\_sky\_ska** (*number*) – Sky temperature for SKA antennas (K)
- **t\_sky\_meer** (*number*) – Sky temperature for MeerKAT antennas (K)
- **t\_gal\_ska** (*number*) – Galactic temperature for SKA antennas (K)
- **t\_gal\_meer** (*number*) – Galactic temperature for MeerKAT antennas (K)
- **alpha** (*number*) – Spectral index of Galactic emission
- **n\_ska** (*integer*) – Number of SKA antennas (overrides array\_configuration; n\_meer should be provided too)

- **n\_meer** (*integer*) – Number of MeerKAT antennas (overrides array\_configuration; n\_ska should be provided too)
- **eta\_meer** (*number*) – Dish efficiency for MeerKAT antennas
- **eta\_ska** (*number*) – Dish efficiency for SKA antennas
- **pwv** (*number*) – Precipitable water vapour in mm
- **el** (*number*) – Elevation above the horizon in degrees (default 45 deg.)
- **bandwidth** (*number*) – Bandwidth (Hz) (Required)
- **resolution** (*number*) – Frequency Resolution (Hz)
- **integration\_time** (*number*) – Integration time (s). This makes the output to include the computed sensitivity.
- **sensitivity** (*number*) – Sensitivity (Jy). This makes the output to include the integration time.
- **n\_chunks** (*integer*) – Number of chunks. This provides additional outputs dividing the bandwidth in N chunks.
- **zoom\_frequencies** (*array*) – Array of zoom frequencies. This provides additional outputs for line observations (zoomResolutions must be set as well and have the same number of elements).
- **zoom\_resolutions** (*array*) – Array of zoom spectral resolutions. This provides additional outputs for line observations (zoomFreqs must be set as well and have the same number of elements).
- **zoom\_sensitivities** (*array*) – Array of zoom sensitivities. This provides additional outputs for line observations (zoomResolutions and zoomFreqs must be set as well and have the same number of elements).

#### Status Codes

- **200 OK** – Successful response with the results of the computations.
- **400 Bad Request** – Request parameters failed validation, either against the OpenAPI spec or the domain specific validation, eg the spectral window not lying within the allowed range.
- **5XX** – An error occurred making the calculation

#### GET /weighting

Compute the weighting correction factor to apply

##### Query Parameters

- **frequency** (*number*) – Frequency (Hz)
- **zoom\_frequencies** (*array*) – Array of zoom frequencies (Hz)
- **dec\_str** (*string*) – Declination in ICRS frame. Allowed range -90:00:00.00 to 44:00:00.00. (Required)
- **weighting** (*string*) – Weighting strategy used (uniform, robust, or natural). (Required)
- **robustness** (*number*) – Robust parameter. It is only required and considered if the weighting is set to robust.
- **array\_configuration** (*string*) – Array configuration (Required)
- **calculator\_mode** (*string*) – Mode of the Calculator (Required)

- **taper** (*number*) – Gaussian taper (specified as arcseconds in image plane). No taper applied if left empty.
- **chunk\_frequencies\_hz** (*array*) – Chunk frequencies in Hz. Providing chunk frequencies enables the provision of additional weighting outputs..

#### Status Codes

- **200 OK** – Successful response with the correction factor.
- **400 Bad Request** – Request parameters failed validation, either against the OpenAPI spec or the domain specific validation, eg the spectral window not lying within the allowed range.
- **5XX** – An error occurred making the calculation

#### GET /subarrays

Returns the array configurations available

#### Status Codes

- **200 OK** – Successful response with the list of array configurations.



## FURTHER WORK

The calculator, in its current prototype state, is the product of 6 months of work by one developer. It is intended to be a sufficient platform to allow the development of a more sophisticated tool. There are already several suggestions for work that needs to be done in the future. These are grouped below according to their source.

### 5.1 SKA Engineering

It has been pointed out that SKA Engineers and Astronomers define ‘sensitivity’ differently, and there is some question whether the Sensitivity Model used by the Calculator matches that already in use by the Engineers. We need to ensure that we define and use our terms carefully and that the sensitivity models are reconciled.

### 5.2 Ideas from the Prototype

These ideas cropped up during development of the prototype, but were considered too time-consuming or too far off in the future to be added at that stage:

- **Shadowing.** In reality, depending on the pointing direction of the dishes in the array, the dishes are likely to obscure one another, resulting in a loss of effective area. This can be accounted for by determining a shadowing fraction (i.e. what proportion of the total area is shadowed) and reducing the effective area by a proportional amount.
- **Image weighting.** The type of image weighting used will affect the sensitivity one can achieve with the observation. Incorporating the Briggs robust weighting parameter into the calculation will help reflect this.
- **Beam synthesis.** Running some simulations to synthesise beams would be incredibly useful and open up a lot of other options for functionality for the calculator.
- **Weather.** Currently the user is given the options of “Good”, “Average” and “Bad” weather, corresponding to pwv values of 5.8mm, 10.7mm and 19.2mm respectively. While this is important for the calculation of the atmospheric temperature,  $T_{atm}$ , it is impossible for the user to predict what the weather is going to be like when their observation gets scheduled. Instead, it may make more sense to give options for different months/seasons, since then the user would at least get an idea what the weather conditions will likely be over the time their observations could be scheduled.
- **Optional smoothing for zooms.** Down the line it is probably a good idea to add a optional line smoothing option for zooms.
- **More observing modes.** The calculator currently sports two observing modes - continuum and line observations. As it is developed, it would be good to have more observing modes added. The prototype has a tab for pulsar observations (and some comments throughout the code), but there is nothing yet implemented for this mode - it is just a placeholder/suggestion.

- **Report resources.** Adding some report of the resources that will be used for the observation (e.g. compute time) would be a valuable addition to the calculator output.
- **Populate inputs from URL.** A handy feature would be if the calculator would parse the query string from the URL and preload the calculator inputs with those values. When combined with a ‘link generator’ feature which would be fairly straightforward to add, this would allow users to generate links to the calculations they have performed and share them with colleagues. When the colleague clicked the link/pasted it into their address bar, they would be taken to the page and the inputs would be loaded with the same values the first user had used.
- **Other Calculators.** In developing this calculator, it was useful to regularly look at other, similar calculators/tools which exist. These other tools helped inform design and inspire new feature ideas. A list of such calculators follows here, which will hopefully be of use as the calculator is further developed.
  - [ALMA Sensitivity Calculator](#)
  - [ATCA Sensitivity Calculator](#)
  - [e-MERLIN Sensitivity Calculator](#)
  - [VLA Exposure Calculator](#)

## 5.3 The Vision Thing

Would it be worth asking some people to write a (very) short story describing how they imagine they would use the SKA ‘in the ideal world’, especially with reference to the Sensitivity Calculator? Consider different scenarios e.g. standard observing, response to transient triggers, survey planning, whatever you can think of.



## **LOW THEORETICAL BACKGROUND**

TODO - a background on the calculation and the science behind it. See the MID Theoretical Background for inspiration.

### **6.1 Dynamic range warning**

In addition to calculating the theoretical sensitivity, the LOW sensitivity calculator backend warns if it detects a bright ( $>10$  Jy) off-axis source within the full width at half maximum (FWHM) of the LOW station beam calculated at the centre of the specified band. The backend uses a bright source catalogue derived from the GaLactic and Extragalactic All-sky Murchison Widefield Array (GLEAM) survey ([Hurley-Walker et al. \(2017\)](#)).



## LOW IMPLEMENTATION

The LOW Sensitivity Calculator follows the basic structure of an API layer (see LOW API docs page), validation, then calculation, using a look up table. The code is functional in nature, relying on pure functions which do not mutate state. The diagram below aims to give a view of this functional structure.

The Python modules can also be seen below. Mostly the implementation uses module level pure functions, and makes use of classes only for data objects and to encapsulate the look up tables.



## LOW REST API

### GET /subarrays

Returns the array configurations available

#### Status Codes

- **200 OK** – Successful response with the list of array configurations.

### GET /continuum/calculate

Calculate the sensitivity of an SKA LOW observation in continuum for the input parameters.

#### Query Parameters

- **num\_stations** (*number*) – The number of LOW antenna stations
- **duration** (*number*) – The duration of the observation in hours
- **pointing\_centre** (*string*) – Pointing centre in HH:MM:SS[.ss] DD:MM:SS[.ss]
- **freq\_centre** (*number*) – Central frequency in MHz
- **bandwidth\_mhz** (*number*) – Bandwidth for the continuum observation in MHz

#### Status Codes

- **200 OK** – Successful response with the results of the sensitivity calculation.
- **400 Bad Request** – Request parameters failed validation, either against the OpenAPI spec or the domain specific validation, eg the spectral window not lying within the allowed range.

### GET /{spectral\_mode}/weighting

Compute the weighting correction factor to apply for an SKA LOW continuum observation

#### Parameters

- **spectral\_mode** (*string*) – Calculator spectral mode

#### Query Parameters

- **weighting\_mode** (*string*) – Weighting strategy used (Required)
- **robustness** (*number*) – Robust parameter. Only required and considered if the weighting is set to robust.
- **subarray\_configuration** (*string*) – SKA LOW subarray configuration (Required)
- **pointing\_centre** (*string*) – Pointing centre in HH:MM:SS[.ss] DD:MM:SS[.ss] (Required)
- **freq\_centre** (*number*) – Central frequency in MHz (Required)

#### Status Codes

- **200 OK** – Result of an image weighting calculation
- **400 Bad Request** – Request parameters failed validation, either against the OpenAPI spec or the domain specific validation, eg the spectral window not lying within the allowed range.
- **5XX** – An error occurred making the calculation

## GET /zoom/calculate

Calculate the sensitivity of an SKA LOW observation in zoom mode for the input parameters.

### Query Parameters

- **num\_stations** (*number*) – The number of LOW antenna stations
- **duration** (*number*) – The duration of the observation in hours
- **pointing\_centre** (*string*) – Pointing centre in HH:MM:SS[.ss] DD:MM:SS[.ss]
- **freq\_centre** (*number*) – Central frequency in MHz
- **spectral\_resolution\_hz** (*number*) – Spectral resolution/channel width of the zoom window. The channel width of the narrowest zoom window is given by  $(781250 * 32/27)/(4096 * 16) = 14.129$  Hz (rounded to 3 d.p.).
- **total\_bandwidth\_khz** (*number*) – The total bandwidth of the zoom window in kHz. Smallest allowed value is the channel width given above multiplied by 1728 channels (~24.4 kHz). Maximum allowed value is 3125 kHz.

### Status Codes

- **200 OK** – Successful response with the results of the sensitivity calculation.
- **400 Bad Request** – Request parameters failed validation, either against the OpenAPI spec or the domain specific validation, eg the spectral window not lying within the allowed range.

**SKA\_OST\_SENSCALC.MID**





## SKA\_OST\_SENSCALC.SUBARRAY



## SKA\_OST\_SENSCALC.UTILITIES



## **SKA\_OST\_SENSCALC.MID\_UTILITIES**



**SKA\_OST\_SENSCALC.LOW.API**





**SKA\_OST\_SENSCALC.LOW.VALIDATION**



**SKA\_OST\_SENSCALC.LOW.MODEL**



**SKA\_OST\_SENSCALC.LOW.SERVICE**



**SKA\_OST\_SENSCALC.LOW.CALCULATOR**





**SKA\_OST\_SENSCALC.LOW.SEFD\_LOOKUP**



## HTTP ROUTING TABLE

**/calculate**

GET /calculate, 15

**/continuum**

GET /continuum/calculate, 25

**/subarrays**

GET /subarrays, 17

**/weighting**

GET /weighting, 16

**/zoom**

GET /zoom/calculate, 26

**/spectral\_mode**

GET /spectral\_mode/weighting, 25