
Hydrogen (HI) Data Emulator Documentation

Release 0.1.0

Joel Akeret

July 28, 2016

1	Contents:	3
1.1	Installation	3
1.2	Usage	3
1.3	hide Package	4
1.4	Contributing	21
1.5	Credits	22
1.6	History	23
2	Feedback	25
	Python Module Index	27

HIDE is a package for simulating of a single dish radio telescope survey. As such, it takes *healpix* maps as inputs and processes them into TOD. The design is flexible and can be customized to different instruments and survey designs.

The **HIDE** package has been developed at ETH Zurich in the [Software Lab of the Cosmology Research Group](#) of the [ETH Institute of Astronomy](#).

The development is coordinated on [GitHub](#) and contributions are welcome. The documentation of **HIDE** is available at [readthedocs.org](#).

Contents:

1.1 Installation

The project is hosted on GitHub. Get a copy by running:

```
$ git clone https://github.com/cosmo-ethz/hide.git
```

Install the package like this:

```
$ cd hide
$ pip install -r requirements.txt
$ python setup.py install --user
```

Alternatively, if you want to develop new features:

```
$ cd hide
$ python setup.py develop --user
```

1.2 Usage

To use Hydrogen (HI) Data Emulator in a project execute the following on the command line:

```
$ hide --strategy-start=2016-03-21-00:00:00 --strategy-end=2016-03-21-23:59:00 --verbose=True hide.co
```

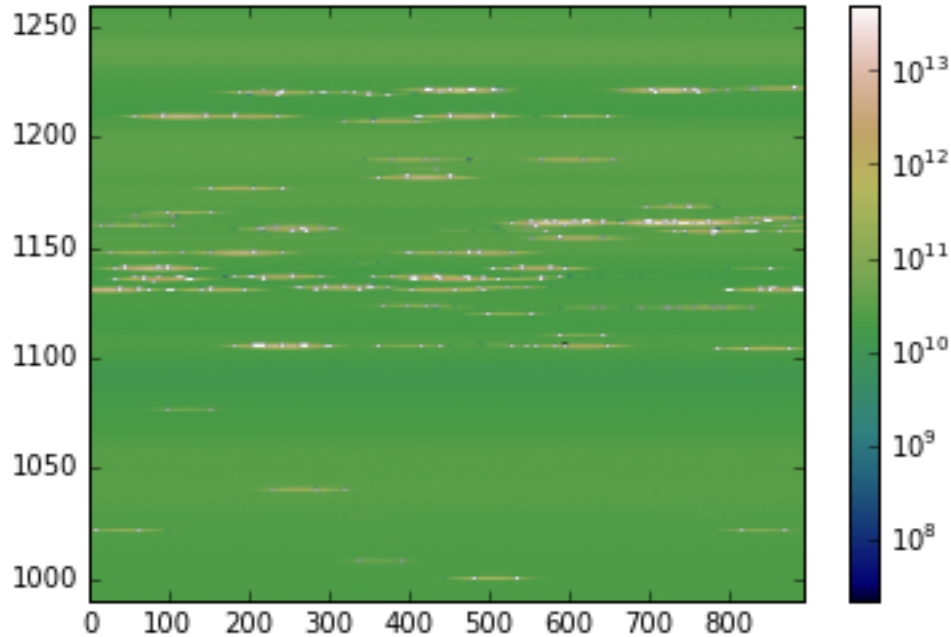
This will simulate one day of time-ordered-data from the Bleien 7m radio telescope.

To visualize 15 minutes of the generated data run this:

```
import matplotlib.pyplot as plt
import matplotlib
import h5py

with h5py.File("./2016/03/21/TEST_MP_PXX_20160321_000000.h5", "r") as fp:
    tod = fp["P/Phase1"].value
    time = fp["TIME"].value

plt.imshow(tod, aspect="auto",
            extent=(time[0], time[-1], 990, 1260),
            cmap="gist_earth", norm=matplotlib.colors.LogNorm())
plt.colorbar()
```



1.3 hide Package

1.3.1 hide Package

1.3.2 Subpackages

astro Package

`gsm` Module

Created on Feb 26, 2015

author: jakeret

`hide.astro.gsm.load_signal(ctx)`

Returns an interpolated global sky model (GSM) map dependent on the frequency.

Parameters `params` – The ctx instance with the parameterization

Returns `signal` The astro signal

`gsm_point_src` Module

Created on Apr 22, 2016

In large parts a copy of `astro_calibration_sources` in seek by cchang.

Models taken from: Baars 1997, Hafez 2008, Benz 2009 All numbers divided by 2 to account for polarization.

Coordinates from wikipedia

author: seehars, jakeret

```
class hide.astro.gsm_point_src.AstroSource(model, ra, dec)
    Bases: tuple
```

```
    dec
        Alias for field number 2
```

```
    model
        Alias for field number 0
```

```
    ra
        Alias for field number 1
```

```
hide.astro.gsm_point_src.add_point_sources(freq, nside, astro_signal, objects=None)
```

```
hide.astro.gsm_point_src.barrs77_power_law(freq, a, b, c)
```

```
hide.astro.gsm_point_src.cas_a_model(freq)
```

```
hide.astro.gsm_point_src.convertunits(s, ae)
```

```
hide.astro.gsm_point_src.cyg_a_model(freq)
```

```
hide.astro.gsm_point_src.load_signal(ctx)
```

Returns an interpolated global sky model (GSM) map dependent on the frequency and adds radio point srcs.

Parameters *params* – The ctx instance with the parameterization

Returns *signal* The astro signal

```
hide.astro.gsm_point_src.sag_a_model(freq)
```

```
hide.astro.gsm_point_src.tau_a_model(freq)
```

```
hide.astro.gsm_point_src.vir_a_model(freq)
```

static Module

Created on Dec 8, 2014

author: jakeret

```
hide.astro.static.load_signal(ctx)
```

Creates a sphere with a static flux as signal for the given ctx (and params)

Parameters *params* – The ctx instance with the parameterization

Returns *signal* A static signal

static_gsm Module

Created on Jan 7, 2015

author: jakeret

```
hide.astro.static_gsm.load_signal(ctx)
```

Returns the same global sky model (GSM) map independent of the frequency. Rescales the map if necessary (if param.beam_nside != 512)

Parameters *params* – The ctx instance with the parameterization

Returns signal The astro signal

uniform Module

Created on Jan 7, 2015

author: jakeret

`hide.astro.uniform.load_signal(ctx)`

Creates a sphere with a uniform flux as signal for the given ctx (and params)

Parameters **params** – The ctx instance with the parameterization

Returns signal A static signal

beam Package

beam Package

`hide.beam.AxisSpec`

Beam definition

alias of axis

class `hide.beam.BeamSpec(ra, dec, pixels)`

Bases: tuple

dec

Alias for field number 1

pixels

Alias for field number 2

ra

Alias for field number 0

class `hide.beam.ResponseSpec(pixel_idx, ra, dec)`

Bases: tuple

dec

Alias for field number 2

pixel_idx

Alias for field number 0

ra

Alias for field number 1

airy Module

Created on Apr 25, 2016

author: jakeret

`hide.beam.airy.airy_wrapper(fwhm)`

`hide.beam.airy.bessel_j1(x)`

`hide.beam.airy.load_beam_profile(beam_spec, frequencies, params)`

Creates a 2d airy beam profile using the given gain template

Parameters `params` – The params instance with the parameterization

Returns profile A list of callable beam profiles

`hide.beam.airy.normalization(fwhm, nside)`

`hide.beam.airy.sigma2fwhm(sigma)`

airy_disk Module

Created on Apr 25, 2016

author: jakeret

`hide.beam.airy_disk.load_beam_profile(beam_spec, frequencies, params)`

Creates a 2d airy disk beam profile for the given params definition

Parameters `params` – The params instance with the parameterization

Returns profile A list of callable beam profiles

gaussian Module

Created on Dec 8, 2014

author: jakeret

`hide.beam.gaussian.gauss_wrapper(sigma, beam_response)`

`hide.beam.gaussian.load_beam_profile(beam_spec, frequencies, params)`

Creates a 2d gaussian beam profile for the given params definition

Parameters `params` – The params instance with the parameterization

Returns profile A list of callable beam profiles

`hide.beam.gaussian.normalization(sigma, nside)`

gaussian_interp Module

Created on Dec 8, 2014

author: jakeret

`hide.beam.gaussian_interp.load_beam_profile(beam_spec, frequencies, params)`

Creates a tophat beam profile for the given params definition

Parameters `params` – The params instance with the parameterization

Returns profile The top hat profile

`hide.beam.gaussian_interp.normalization(sigma, nside)`

`hide.beam.gaussian_interp.spline(func, beam_response)`

top_hat Module

Created on Dec 8, 2014

author: jakeret

`hide.beam.top_hat.load_beam_profile` (*beam_spec, frequencies, params*)

Creates a tophat beam profile for the given params definition

Parameters *params* – The params instance with the parameterization

Returns *profile* The top hat profile

`hide.beam.top_hat.normalization` (*r, nside*)

config Package

bleien Module

Created on Dec 8, 2014

author: jakeret

bleien7m Module

Created on Nov 9, 2015

author: jakeret

common Module

Created on Dec 8, 2014

author: jakeret

earth Package

constant Module

Created on Nov 25, 2015

author: jakeret

`hide.earth.constant.load_signal` (*ctx*)

Returns a constant signal

Parameters *ctx* – context object containing params

Returns *earth_signal* healpy map with the signal

horizon Module

Created on Dec 8, 2014

author: jakeret

`hide.earth.horizon.load_signal(ctx)`

Models effect of the horizon

Parameters `ctx` – context object containing params

Returns `earth_signal` healpy map with the signal

plugins Package

add_background Module

Created on Mar 23, 2016

author: seehars

class `hide.plugins.add_background.Plugin(ctx, **kwargs)`

Bases: `ivy.plugin.base_plugin.BasePlugin`

Adds a time-constant, elevation dependent background to the TOD.

add_point_sources Module

Created on Apr 22, 2016

author: seehars

class `hide.plugins.add_point_sources.Plugin(ctx, **kwargs)`

Bases: `ivy.plugin.base_plugin.BasePlugin`

add_reference Module

Created on Dec 8, 2014

author: jakeret

class `hide.plugins.add_reference.Plugin(ctx, **kwargs)`

Bases: `ivy.plugin.base_plugin.BasePlugin`

add_rfi Module

Created on Dec 8, 2014

author: jakeret

class `hide.plugins.add_rfi.Plugin(ctx, **kwargs)`

Bases: `ivy.plugin.base_plugin.BasePlugin`

Adds RFI to the time ordered data

`hide.plugins.add_rfi.gaussian(height, center_x, center_y, width_x, width_y)`

Returns a gaussian function with the given parameters

add_rfi_phaseswitch Module

Created on Feb 17, 2016

author: seehars

```
class hide.plugins.add_rfi_phaseswitch.Plugin(ctx, **kwargs)
```

```
    Bases: ivy.plugin.base_plugin.BasePlugin
```

Adds RFI to the time ordered data (phase switch).

```
    getTime()
```

```
hide.plugins.add_rfi_phaseswitch.calcRFI(background, amplitude, fraction, deltat, deltaf,  
                                           exponent, enhance, nf, nt)
```

Get time-frequency plane of RFI.

Parameters

- **background** – background level of data per channel
- **amplitude** – maximal amplitude of RFI per channel
- **fraction** – fraction of RFI dominated pixels per channel
- **deltat** – time scale of rfi decay (in units of pixels)
- **deltaf** – frequency scale of rfi decay (in units of pixels)
- **exponent** – exponent of rfi model (either 1 or 2)
- **enhance** – enhancement factor relative to fraction
- **nf** – number of frequency channels
- **nt** – number of time steps

Returns RFI time-frequency plane of RFI

```
hide.plugins.add_rfi_phaseswitch.getDayNightMask(rfiday, time)
```

```
hide.plugins.add_rfi_phaseswitch.getRFI(background, amplitude, fraction, deltat, deltaf, ex-  
                                           ponent, enhance, frequencies, time, rfiday, damp-  
                                           ing)
```

Get time-frequency plane of RFI.

Parameters

- **background** – background level of data per channel
- **amplitude** – maximal amplitude of RFI per channel
- **fraction** – fraction of RFI dominated pixels per channel
- **deltat** – time scale of rfi decay (in units of pixels)
- **deltaf** – frequency scale of rfi decay (in units of pixels)
- **exponent** – exponent of rfi model (either 1 or 2)
- **enhance** – enhancement factor relative to fraction
- **frequencies** – frequencies of tod in MHz
- **time** – time of day in hours of tod
- **rfiday** – tuple of start and end of RFI day
- **damping** – damping factor for RFI fraction during the RFI night

Returns RFI time-frequency plane of RFI

`hide.plugins.add_rfi_phaseswitch.kernel (deltaf, deltat, nf, nt, N, exponent)`
Convolution kernel for FFT convolution

Parameters

- **deltaf** – spread of RFI model in frequency
- **deltat** – spread of RFI model in time
- **nf** – number of frequencies
- **nt** – number of time steps
- **N** – size of kernel relative to deltaf, deltat
- **exponent** – exponent of RFI model (see logmodel)

Returns kernel convolution kernel

`hide.plugins.add_rfi_phaseswitch.logmodel (x, dx, exponent)`

Model for the log of the RFI profile:

- $-\text{abs}(x)/dx$ for exponent 1
- $-(x/dx)^2$ for exponent 2

Parameters

- **x** – grid on which to evaluate the profile
- **dx** – width of exponential
- **exponent** – exponent of (x/dx) , either 1 or 2

Returns logmodel log of RFI profile

apply_gain Module

Created on Feb 27, 2015

author: jakeret

class `hide.plugins.apply_gain.Plugin (ctx, **kwargs)`
Bases: `ivy.plugin.base_plugin.BasePlugin`

Transform the temperature based (Kelvin) TOD into ADU by applying a spectrometer specific gain

astro_signal Module

Created on Dec 8, 2014

author: jakeret

class `hide.plugins.astro_signal.Plugin (ctx, **kwargs)`
Bases: `ivy.plugin.base_plugin.BasePlugin`

background_noise Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.background_noise.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
```

Adds background noise to the time ordered data

```
hide.plugins.background_noise.get_noise(scale, alpha, beta, size)
```

clean_up Module

Created on Sep 14, 2015

author: jakeret

```
class hide.plugins.clean_up.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
```

Cleans up the context to avoid a memory leak

combine_signals Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.combine_signals.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
```

Combines the different signals by convolving the beam profile with the input signals

coord_transform Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.coord_transform.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
```

Applies the coordination transformation to the beam profile by rotating the beam response on the sky sphere according to the defined scanning strategy

```
hide.plugins.coord_transform.plot_beam(beam_spec, coord_ra, coord_dec, rphis, rthetas, ras,
                                         decs)
```

earth_signal Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.earth_signal.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
```


`initialize` Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.initialize.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
    Initialize ...
```

`load_beam_profile` Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.load_beam_profile.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
    Delegates the loading process to the beam profile provider
```

`map_frequency_plugin` Module

Created on Dec 11, 2014

author: jakeret

```
class hide.plugins.map_frequency_plugin.Plugin(ctx)
    Bases: object
    Maps the frequencies to the plugin collection.
    getWorkload()
```

`map_strategy_plugin` Module

Created on Dec 11, 2014

author: jakeret

```
class hide.plugins.map_strategy_plugin.Plugin(ctx)
    Bases: object
    Maps the strategy coordinates to the plugin collection.
    getWorkload()
```

`qu_opt_coord_transform` Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.qu_opt_coord_transform.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
    Applies the coordination transformation to the beam profile by rotating the beam response on the sky sphere
    according to the defined scanning strategy
```

`reduce_frequency_plugin` Module

Created on Mar 18, 2014

author: jakeret

```
class hide.plugins.reduce_frequency_plugin.Plugin(ctx)
    Bases: object

    Combines the time ordered data for all frequencies

    reduce (ctxList)
```

`reduce_signals_plugin` Module

Created on Mar 18, 2014

author: jakeret

```
class hide.plugins.reduce_signals_plugin.Plugin(ctx)
    Bases: object

    Combines all signals to time ordered data

    reduce (ctxList)
```

`scanning_strategy` Module

Created on Dec 8, 2014

author: jakeret

```
class hide.plugins.scanning_strategy.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
```

`write_calibration` Module

Created on Mar 29, 2016

author: seehars

```
class hide.plugins.write_calibration.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin

    Writes the sources of the calibration days to disk

hide.plugins.write_calibration.get_header()
    Get header for calibration day file

hide.plugins.write_calibration.get_path(path, date, name)
    Get path for calibration day
```

Parameters

- **path** – output_path
- **date** – key of calibration day
- **name** – name of calibration file

`hide.plugins.write_calibration.write_day` (*date, entries, path, name, header*)
Write calibration day to file

Parameters

- **date** – key to calibration day
- **entries** – list of sources
- **path** – output_path
- **name** – name of calibration file
- **header** – header information

write_coords Module

Created on Sep 4, 2015

author: jakeret

class `hide.plugins.write_coords.Plugin` (*ctx, **kwargs*)
Bases: `ivy.plugin.base_plugin.BasePlugin`

Writes the time ordered data to the file system

write_rfi Module

Created on Feb 25, 2016

author: seehars

class `hide.plugins.write_rfi.Plugin` (*ctx, **kwargs*)
Bases: `ivy.plugin.base_plugin.BasePlugin`

Writes the RFI contribution to the time ordered phase switch data to the file system. Works only after tod is written to disk.

`hide.plugins.write_rfi.write_data` (*ctx, rfi_key, rfi_name*)

write_tod Module

Created on Dec 8, 2014

author: jakeret

class `hide.plugins.write_tod.Plugin` (*ctx, **kwargs*)
Bases: `ivy.plugin.base_plugin.BasePlugin`

DEPRECATED: Writes the time ordered data to the file system

write_tod_fits Module

Created on Sep 4, 2015

author: jakeret

```
class hide.plugins.write_tod_fits.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
    Writes the time ordered data to the file system in a fits file
```

write_tod_phaseswitch Module

Created on Dec 16, 2015

author: seehars

```
class hide.plugins.write_tod_phaseswitch.Plugin(ctx, **kwargs)
    Bases: ivy.plugin.base_plugin.BasePlugin
    Writes the time ordered phase switch data to the file system
```

```
hide.plugins.write_tod_phaseswitch.add_dataset(grp, name, data)
    Adds a dataset to the group applying moderate compression
```

Parameters

- **grp** – The group
- **name** – Name of the dataset
- **data** – the actual data to be added

```
hide.plugins.write_tod_phaseswitch.get_path(ctx, pol)
    Get path for output :param ctx: instance of ivy context :param pol: identifier for polarization
```

```
hide.plugins.write_tod_phaseswitch.write_data(ctx, tod_key, Pname, P2name=None)
    Write the phase switch data to disk.
```

Parameters

- **ctx** – instance of ivy context
- **tod_key** – key for the data to write
- **Pname** – group name for TOD
- **P2name** – group name for kurtosis data (None at the moment)

spectrometer Package

M9703A Module

Created on Nov 9, 2015

author: jakeret

```
hide.spectrometer.M9703A.convert_frequencies(frequencies)
    Convert frequencies to internal frequencies of M9703A :param frequencies: true frequencies :returns freq: in-
    ternal frequencies

hide.spectrometer.M9703A.get_background(frequencies, el_model)

hide.spectrometer.M9703A.get_gain(frequencies)

hide.spectrometer.M9703A.get_noise_params(frequencies)

hide.spectrometer.M9703A.get_rfi_params(frequencies)
```

```
hide.spectrometer.M9703A.get_schedule()
```

callisto Module

Created on Nov 9, 2015

author: jakeret

```
hide.spectrometer.callisto.apply_background(ctx)
```

```
hide.spectrometer.callisto.apply_gain(ctx)
```

Logarithmizes the data, adds an fix offset and finally applies a model of a standing wave to the TOD :param ctx:

```
hide.spectrometer.callisto.get_sw(frequencies, ft_model, fmin, fmax, nf, slope)
```

strategy Package

strategy Package

```
class hide.strategy.CoordSpec(time, alt, az, ra, dec)
```

Bases: tuple

alt

Alias for field number 1

az

Alias for field number 2

dec

Alias for field number 4

ra

Alias for field number 3

time

Alias for field number 0

center Module

Created on Dec 9, 2014

author: jakeret

```
hide.strategy.center.load_strategy(ctx)
```

Creates a dummy scanning strategy by always centering on RA/DEC 0/0

Parameters **ctx** – The ctx instance with the parameterization

Returns **strategy** A dummy scanning strategy

crosshair Module

Created on Jan 7, 2015

author: jakeret

```
hide.strategy.crosshair.load_strategy(ctx)
```

Creates a crosshair in RA/DEC scanning strategy

Parameters `ctx` – The ctx instance with the parameterization

Returns `strategy` A crosshair scanning strategy

`drift_scan` Module

Created on Jan 15, 2015

author: jakeret

`hide.strategy.drift_scan.load_strategy(ctx)`

Creates a scanning strategy that uses drift mode i.e. the telescope stares at the same position for 24 hours and then changes the altitude by a certain angle

Parameters `ctx` – The ctx instance with the parameterization

Returns `strategy` A list of CoordSpec with the scanning strategy

`full_sky` Module

Created on Jan 15, 2015

author: jakeret

`hide.strategy.full_sky.load_strategy(ctx)`

Creates a scanning strategy that covers the full sky

Parameters `ctx` – The ctx instance with the parameterization

Returns `strategy` A list of CoordSpec with the scanning strategy for the full sky

`scheduler` Module

Created on Mar 24, 2016

author: seehars

`class hide.strategy.scheduler.ScheduleEntry(date, az, el, mode)`

Bases: object

`day()`

`delta(other)`

`is_survey()`

`hide.strategy.scheduler.load_strategy(ctx)`

Creates a scanning strategy from a scheduler file.

Parameters `ctx` – The ctx instance with the path to the scheduler file

Returns `strategy` A list of CoordSpec with the scanning strategy

`hide.strategy.scheduler.parse_schedule(path, strategy_start)`

Parses a scheduler file :param path: the path to the scheduler file :param strategy_start: start date of the strategy

Returns `schedule_entries` list of `ScheduleEntry`

`hide.strategy.scheduler.process_schedule` (*schedule*, *step_size*, *strategy_start*, *strategy_end*,
obs)
Processes a list of schedule entries :param *schedule*: the list of schedule entries :param *step_size*: the step size to use :param *strategy_start*: start date of the strategy :param *strategy_end*: end date of the strategy :param *obs*: telescope position

Returns *strategy*, *calibration_days* a list of *CoordSpec* and a dict for the calibration days

`scheduler_virtual` Module

Created on May 4, 2016

author: jakeret

`hide.strategy.scheduler_virtual.load_strategy` (*ctx*)
Creates a scanning strategy from a scheduler file.

Parameters *ctx* – The ctx instance with the path to the scheduler file

Returns *strategy* A list of *CoordSpec* with the scanning strategy

`hide.strategy.scheduler_virtual.replace_calibrations` (*schedule*, *obs*)

`utils` Package

`utils` Package

Created on Feb 20, 2015

author: jakeret

`hide.utils.arccos` (*x*)

`hide.utils.parse_datetime` (*s*)

`hide.utils.sin_cos` (*x*)

`quaternion` Module

class `hide.utils.quaternion.Rotator` (*q*)

Bases: object

Quaternion based rotator implementation for theta, phi

class `hide.utils.quaternion.VecRotator` (*q*)

Bases: object

Quaternion based rotator implementation for theta, phi

`hide.utils.quaternion.inv` (*q*)

Inverse of quaternion array *q*

`hide.utils.quaternion.mult` (*p*, *q*)

Multiply arrays of quaternions, ndarray objects with 4 columns defined as x y z w see:
http://en.wikipedia.org/wiki/Quaternions#Quaternions_and_the_geometry_of_R3

`hide.utils.quaternion.norm` (*q*)

Normalize quaternion array *q* to unit quaternions

`hide.utils.quaternion.power(q, t)`
raise quaternion to the power of t

`hide.utils.quaternion.rotate_vec(q, v)`
Rotate or array of vectors v by quaternion q

`hide.utils.quaternion.rotate_vec_opt(q, v)`
Rotate or array of vectors v by quaternion q

`hide.utils.quaternion.rotate_vec_slow(q, v)`
Rotate or array of vectors v by quaternion q

`hide.utils.quaternion.slerp(q, r, t)`
spherical linear interpolation between q and r by t

`hide.utils.quaternion.toAxisAngle(q)`

`hide.utils.quaternion.vecquad(x, y, z, w)`
create a quaternion from a euler vector with angle

signal Module

Created on Nov 10, 2015

author: jakeret

`hide.utils.signal.noisegen(beta=0, N=8192)`

Noise will be generated that has spectral densities that vary as powers of inverse frequency, more precisely, the power spectra $P(f)$ is proportional to $1/f^\beta$ for $\beta \geq 0$. When β is 0 the noise is referred to white noise, when it is 2 it is referred to as Brownian noise, and when it is 1 it normally referred to simply as $1/f$ noise which occurs very often in processes found in nature.

The basic method involves creating frequency components which have a magnitude that is generated from a Gaussian white process and scaled by the appropriate power of f . The phase is uniformly distributed on $0, 2\pi$.

from <http://paulbourke.net/fractals/noise/>

Parameters

- **beta** –
- **N** – number of samples (can also be shape of array)

Returns out the sampled noise

sphere Module

Created on Dec 22, 2014

author: jakeret

class `hide.utils.sphere.ArcKDTree(theta, phi)`

Bases: object

Wraps the `scipy.spatial.cKDTree` such that the tree can be used with spherical coords

query `(theta, phi, k=1, eps=0, p=2, distance_upper_bound=inf)`

Query the kd-tree for nearest neighbors using theta, phi :param theta: :param phi: :param k: :param eps: :param p: :param distance_upper_bound:

Returns d, i The distances to the nearest neighbors, the locations of the neighbors in self.data.


```
query_ball_point (theta, phi, r, eps=0)
hide.utils.sphere.altaz_to_ra_dec (date, az, alt, obs=None, ctx=None)
hide.utils.sphere.dec2theta (dec)
hide.utils.sphere.dir2vec (theta, phi)
    converts angle to vector
hide.utils.sphere.get_observer (ctx)
hide.utils.sphere.phi2ra (phi)
hide.utils.sphere.ra2phi (ra)
hide.utils.sphere.radec_to_altaz (date, ra, dec, obs=None, ctx=None)
hide.utils.sphere.rotate_map (Map, rotator, mask=None)
    Map is map in system A rotator is rotator from system B to A mask is a mask in system B returns new map in
    system B
hide.utils.sphere.separation (d1, a1, d2, a2)
    great circle distance http://en.wikipedia.org/wiki/Great-circle\_distance#Computational\_formulas

Parameters

    • d1 – dec 1
    • a1 – ra 1
    • d2 – dec 2

:param a2:ra 2
hide.utils.sphere.theta2dec (theta)
hide.utils.sphere.vec2dir (vec)
    converts vector to angles
```

1.4 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

1.4.1 Types of Contributions

Report Bugs

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Implement Features

Write Documentation

Hydrogen (HI) Data Emulator could always use more documentation, whether as part of the official Hydrogen (HI) Data Emulator docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

1.4.2 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. make sure that the tests pass for all supported Python versions.

1.4.3 Tips

To run a subset of tests:

```
$ py.test test/test_hide.py
```

1.5 Credits

1.5.1 Development Lead

- Joel Akeret <jakeret@phys.ethz.ch>
- Sebastian Seehars <seehars@phys.ethz.ch>
- Chihway Chang <chihway.chang@phys.ethz.ch>

1.5.2 Contributors

None yet. Why not be the first?

1.6 History

1.6.1 0.1.0 (2016-07-20)

- Publication

Feedback

If you have any suggestions or questions about **Hydrogen (HI) Data Emulator** feel free to email me at jak-eret@phys.ethz.ch.

If you encounter any errors or problems with **Hydrogen (HI) Data Emulator**, please let me know!

h

- hide.__init__, 4
- hide.astro.gsm, 4
- hide.astro.gsm_point_src, 4
- hide.astro.static, 5
- hide.astro.static_gsm, 5
- hide.astro.uniform, 6
- hide.beam, 6
- hide.beam.airy, 6
- hide.beam.airy_disk, 7
- hide.beam.gaussian, 7
- hide.beam.gaussian_interp, 7
- hide.beam.top_hat, 8
- hide.config.bleien, 8
- hide.config.bleien7m, 8
- hide.config.common, 8
- hide.earth.constant, 8
- hide.earth.horizon, 9
- hide.plugins.add_background, 9
- hide.plugins.add_point_sources, 9
- hide.plugins.add_reference, 9
- hide.plugins.add_rfi, 9
- hide.plugins.add_rfi_phaseswitch, 10
- hide.plugins.apply_gain, 11
- hide.plugins.astro_signal, 11
- hide.plugins.background_noise, 12
- hide.plugins.clean_up, 12
- hide.plugins.combine_signals, 12
- hide.plugins.coord_transform, 12
- hide.plugins.earth_signal, 12
- hide.plugins.initialize, 13
- hide.plugins.load_beam_profile, 13
- hide.plugins.map_frequency_plugin, 13
- hide.plugins.map_strategy_plugin, 13
- hide.plugins.qu_opt_coord_transform, 13
- hide.plugins.reduce_frequency_plugin, 14
- hide.plugins.reduce_signals_plugin, 14
- hide.plugins.scanning_strategy, 14
- hide.plugins.write_calibration, 14
- hide.plugins.write_coords, 15
- hide.plugins.write_rfi, 15
- hide.plugins.write_tod, 15
- hide.plugins.write_tod_fits, 15
- hide.plugins.write_tod_phaseswitch, 16
- hide.spectrometer.callisto, 17
- hide.spectrometer.M9703A, 16
- hide.strategy, 17
- hide.strategy.center, 17
- hide.strategy.crosshair, 17
- hide.strategy.drift_scan, 18
- hide.strategy.full_sky, 18
- hide.strategy.scheduler, 18
- hide.strategy.scheduler_virtual, 19
- hide.utils, 19
- hide.utils.quaternion, 19
- hide.utils.signal, 20
- hide.utils.sphere, 20

A

add_dataset() (in module
hide.plugins.write_tod_phaseswitch), 16

add_point_sources() (in module
hide.astro.gsm_point_src), 5

airy_wrapper() (in module hide.beam.airy), 6

alt (hide.strategy.CoordSpec attribute), 17

altaz_to_ra_dec() (in module hide.utils.sphere), 21

apply_background() (in module
hide.spectrometer.callisto), 17

apply_gain() (in module hide.spectrometer.callisto), 17

arccos() (in module hide.utils), 19

ArcKDTTree (class in hide.utils.sphere), 20

AstroSource (class in hide.astro.gsm_point_src), 5

AxisSpec (in module hide.beam), 6

az (hide.strategy.CoordSpec attribute), 17

B

barrs77_power_law() (in module
hide.astro.gsm_point_src), 5

BeamSpec (class in hide.beam), 6

bessel_j1() (in module hide.beam.airy), 6

C

calcRFI() (in module hide.plugins.add_rfi_phaseswitch), 10

cas_a_model() (in module hide.astro.gsm_point_src), 5

convert_frequencies() (in module
hide.spectrometer.M9703A), 16

convertunits() (in module hide.astro.gsm_point_src), 5

CoordSpec (class in hide.strategy), 17

cyg_a_model() (in module hide.astro.gsm_point_src), 5

D

day() (hide.strategy.scheduler.ScheduleEntry method), 18

dec (hide.astro.gsm_point_src.AstroSource attribute), 5

dec (hide.beam.BeamSpec attribute), 6

dec (hide.beam.ResponseSpec attribute), 6

dec (hide.strategy.CoordSpec attribute), 17

dec2theta() (in module hide.utils.sphere), 21

delta() (hide.strategy.scheduler.ScheduleEntry method), 18

dir2vec() (in module hide.utils.sphere), 21

G

gauss_wrapper() (in module hide.beam.gaussian), 7

gaussian() (in module hide.plugins.add_rfi), 9

get_background() (in module
hide.spectrometer.M9703A), 16

get_gain() (in module hide.spectrometer.M9703A), 16

get_header() (in module hide.plugins.write_calibration), 14

get_noise() (in module hide.plugins.background_noise), 12

get_noise_params() (in module
hide.spectrometer.M9703A), 16

get_observer() (in module hide.utils.sphere), 21

get_path() (in module hide.plugins.write_calibration), 14

get_path() (in module
hide.plugins.write_tod_phaseswitch), 16

get_rfi_params() (in module hide.spectrometer.M9703A), 16

get_schedule() (in module hide.spectrometer.M9703A), 16

get_sw() (in module hide.spectrometer.callisto), 17

getDayNightMask() (in module
hide.plugins.add_rfi_phaseswitch), 10

getRFI() (in module hide.plugins.add_rfi_phaseswitch), 10

getTime() (hide.plugins.add_rfi_phaseswitch.Plugin method), 10

getWorkload() (hide.plugins.map_frequency_plugin.Plugin method), 13

getWorkload() (hide.plugins.map_strategy_plugin.Plugin method), 13

H

hide.__init__ (module), 4

hide.astro.gsm (module), 4

hide.astro.gsm_point_src (module), 4

hide.astro.static (module), 5
hide.astro.static_gsm (module), 5
hide.astro.uniform (module), 6
hide.beam (module), 6
hide.beam.airy (module), 6
hide.beam.airy_disk (module), 7
hide.beam.gaussian (module), 7
hide.beam.gaussian_interp (module), 7
hide.beam.top_hat (module), 8
hide.config.bleien (module), 8
hide.config.bleien7m (module), 8
hide.config.common (module), 8
hide.earth.constant (module), 8
hide.earth.horizon (module), 9
hide.plugins.add_background (module), 9
hide.plugins.add_point_sources (module), 9
hide.plugins.add_reference (module), 9
hide.plugins.add_rfi (module), 9
hide.plugins.add_rfi_phaseswitch (module), 10
hide.plugins.apply_gain (module), 11
hide.plugins.astro_signal (module), 11
hide.plugins.background_noise (module), 12
hide.plugins.clean_up (module), 12
hide.plugins.combine_signals (module), 12
hide.plugins.coord_transform (module), 12
hide.plugins.earth_signal (module), 12
hide.plugins.initialize (module), 13
hide.plugins.load_beam_profile (module), 13
hide.plugins.map_frequency_plugin (module), 13
hide.plugins.map_strategy_plugin (module), 13
hide.plugins.qu_opt_coord_transform (module), 13
hide.plugins.reduce_frequency_plugin (module), 14
hide.plugins.reduce_signals_plugin (module), 14
hide.plugins.scanning_strategy (module), 14
hide.plugins.write_calibration (module), 14
hide.plugins.write_coords (module), 15
hide.plugins.write_rfi (module), 15
hide.plugins.write_tod (module), 15
hide.plugins.write_tod_fits (module), 15
hide.plugins.write_tod_phaseswitch (module), 16
hide.spectrometer.callisto (module), 17
hide.spectrometer.M9703A (module), 16
hide.strategy (module), 17
hide.strategy.center (module), 17
hide.strategy.crosshair (module), 17
hide.strategy.drift_scan (module), 18
hide.strategy.full_sky (module), 18
hide.strategy.scheduler (module), 18
hide.strategy.scheduler_virtual (module), 19
hide.utils (module), 19
hide.utils.quaternion (module), 19
hide.utils.signal (module), 20
hide.utils.sphere (module), 20

I

inv() (in module hide.utils.quaternion), 19
is_survey() (hide.strategy.scheduler.ScheduleEntry
method), 18

K

kernel() (in module hide.plugins.add_rfi_phaseswitch),
11

L

load_beam_profile() (in module hide.beam.airy), 6
load_beam_profile() (in module hide.beam.airy_disk), 7
load_beam_profile() (in module hide.beam.gaussian), 7
load_beam_profile() (in module
hide.beam.gaussian_interp), 7
load_beam_profile() (in module hide.beam.top_hat), 8
load_signal() (in module hide.astro.gsm), 4
load_signal() (in module hide.astro.gsm_point_src), 5
load_signal() (in module hide.astro.static), 5
load_signal() (in module hide.astro.static_gsm), 5
load_signal() (in module hide.astro.uniform), 6
load_signal() (in module hide.earth.constant), 8
load_signal() (in module hide.earth.horizon), 9
load_strategy() (in module hide.strategy.center), 17
load_strategy() (in module hide.strategy.crosshair), 17
load_strategy() (in module hide.strategy.drift_scan), 18
load_strategy() (in module hide.strategy.full_sky), 18
load_strategy() (in module hide.strategy.scheduler), 18
load_strategy() (in module
hide.strategy.scheduler_virtual), 19
logmodel() (in module
hide.plugins.add_rfi_phaseswitch), 11

M

model (hide.astro.gsm_point_src.AstroSource attribute),
5
mult() (in module hide.utils.quaternion), 19

N

noisegen() (in module hide.utils.signal), 20
norm() (in module hide.utils.quaternion), 19
normalization() (in module hide.beam.airy), 7
normalization() (in module hide.beam.gaussian), 7
normalization() (in module hide.beam.gaussian_interp), 7
normalization() (in module hide.beam.top_hat), 8

P

parse_datetime() (in module hide.utils), 19
parse_schedule() (in module hide.strategy.scheduler), 18
phi2ra() (in module hide.utils.sphere), 21
pixel_idxs (hide.beam.ResponseSpec attribute), 6
pixels (hide.beam.BeamSpec attribute), 6

plot_beam() (in module hide.plugins.coord_transform), 12

Plugin (class in hide.plugins.add_background), 9

Plugin (class in hide.plugins.add_point_sources), 9

Plugin (class in hide.plugins.add_reference), 9

Plugin (class in hide.plugins.add_rfi), 9

Plugin (class in hide.plugins.add_rfi_phaseswitch), 10

Plugin (class in hide.plugins.apply_gain), 11

Plugin (class in hide.plugins.astro_signal), 11

Plugin (class in hide.plugins.background_noise), 12

Plugin (class in hide.plugins.clean_up), 12

Plugin (class in hide.plugins.combine_signals), 12

Plugin (class in hide.plugins.coord_transform), 12

Plugin (class in hide.plugins.earth_signal), 12

Plugin (class in hide.plugins.initialize), 13

Plugin (class in hide.plugins.load_beam_profile), 13

Plugin (class in hide.plugins.map_frequency_plugin), 13

Plugin (class in hide.plugins.map_strategy_plugin), 13

Plugin (class in hide.plugins.qu_opt_coord_transform), 13

Plugin (class in hide.plugins.reduce_frequency_plugin), 14

Plugin (class in hide.plugins.reduce_signals_plugin), 14

Plugin (class in hide.plugins.scanning_strategy), 14

Plugin (class in hide.plugins.write_calibration), 14

Plugin (class in hide.plugins.write_coords), 15

Plugin (class in hide.plugins.write_rfi), 15

Plugin (class in hide.plugins.write_tod), 15

Plugin (class in hide.plugins.write_tod_fits), 15

Plugin (class in hide.plugins.write_tod_phaseswitch), 16

power() (in module hide.utils.quaternion), 19

process_schedule() (in module hide.strategy.scheduler), 18

Q

query() (hide.utils.sphere.ArcKDTree method), 20

query_ball_point() (hide.utils.sphere.ArcKDTree method), 20

R

ra (hide.astro.gsm_point_src.AstroSource attribute), 5

ra (hide.beam.BeamSpec attribute), 6

ra (hide.beam.ResponseSpec attribute), 6

ra (hide.strategy.CoordSpec attribute), 17

ra2phi() (in module hide.utils.sphere), 21

radec_to_altaz() (in module hide.utils.sphere), 21

reduce() (hide.plugins.reduce_frequency_plugin.Plugin method), 14

reduce() (hide.plugins.reduce_signals_plugin.Plugin method), 14

replace_calibrations() (in module hide.strategy.scheduler_virtual), 19

ResponseSpec (class in hide.beam), 6

rotate_map() (in module hide.utils.sphere), 21

rotate_vec() (in module hide.utils.quaternion), 20

rotate_vec_opt() (in module hide.utils.quaternion), 20

rotate_vec_slow() (in module hide.utils.quaternion), 20

Rotator (class in hide.utils.quaternion), 19

S

sag_a_model() (in module hide.astro.gsm_point_src), 5

ScheduleEntry (class in hide.strategy.scheduler), 18

separation() (in module hide.utils.sphere), 21

sigma2fwhm() (in module hide.beam.airy), 7

sin_cos() (in module hide.utils), 19

slerp() (in module hide.utils.quaternion), 20

spline() (in module hide.beam.gaussian_interp), 7

T

tau_a_model() (in module hide.astro.gsm_point_src), 5

theta2dec() (in module hide.utils.sphere), 21

time (hide.strategy.CoordSpec attribute), 17

toAxisAngle() (in module hide.utils.quaternion), 20

V

vec2dir() (in module hide.utils.sphere), 21

vecquad() (in module hide.utils.quaternion), 20

VecRotator (class in hide.utils.quaternion), 19

vir_a_model() (in module hide.astro.gsm_point_src), 5

W

write_data() (in module hide.plugins.write_rfi), 15

write_data() (in module hide.plugins.write_tod_phaseswitch), 16

write_day() (in module hide.plugins.write_calibration), 14