Progress Summary

Tanumoy Saha (HTW Berlin)

Dec-Feb

Task

Development of a modular Digital-twin of Pulsars

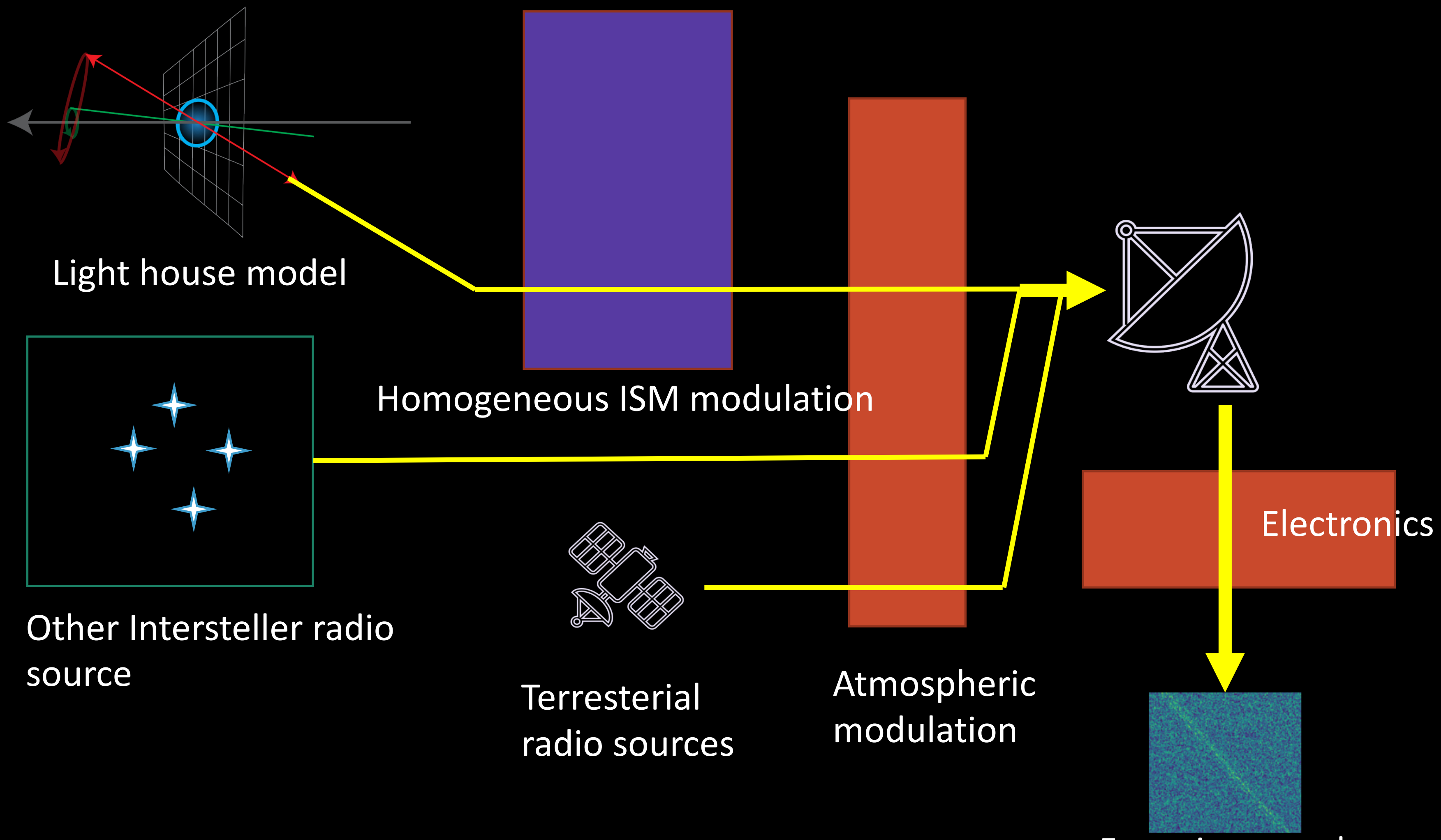
This includes:

1. Development of the basic physical model (animation) of pulsars based on existing theoretical models

2. Modular design of the code/simulation using OOP in python

3. Proper digital auto-documentation of the code

basic physical model (animation) of pulsars



Light house model

Homogeneous ISM modulation

Other Intersteller radio source

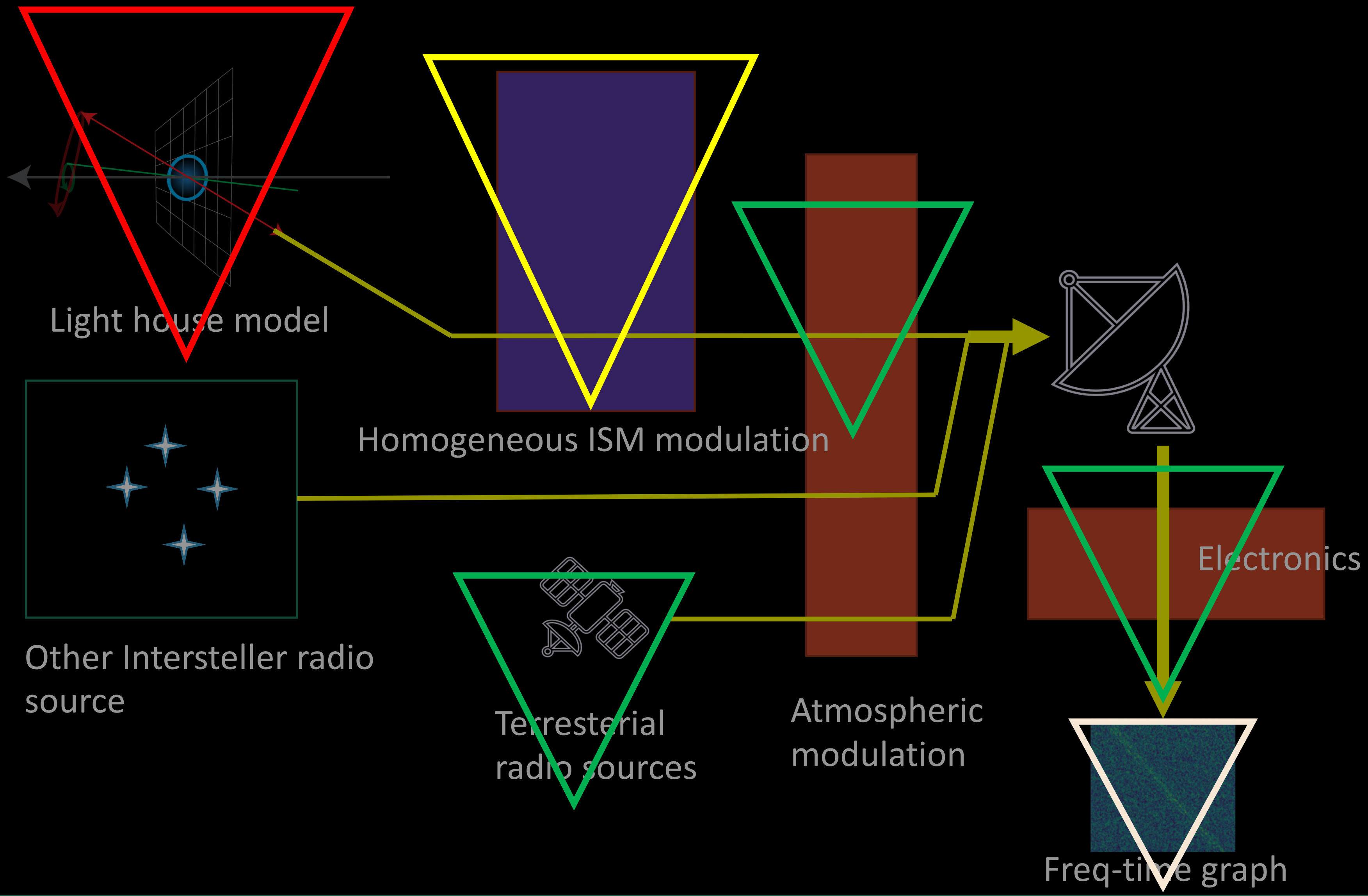
Terresterial radio sources

Atmospheric modulation

Electronics

Freq-time graph

simulation using OOP in python (modular)



Light house model

Homogeneous ISM modulation

Other Intersteller radio source

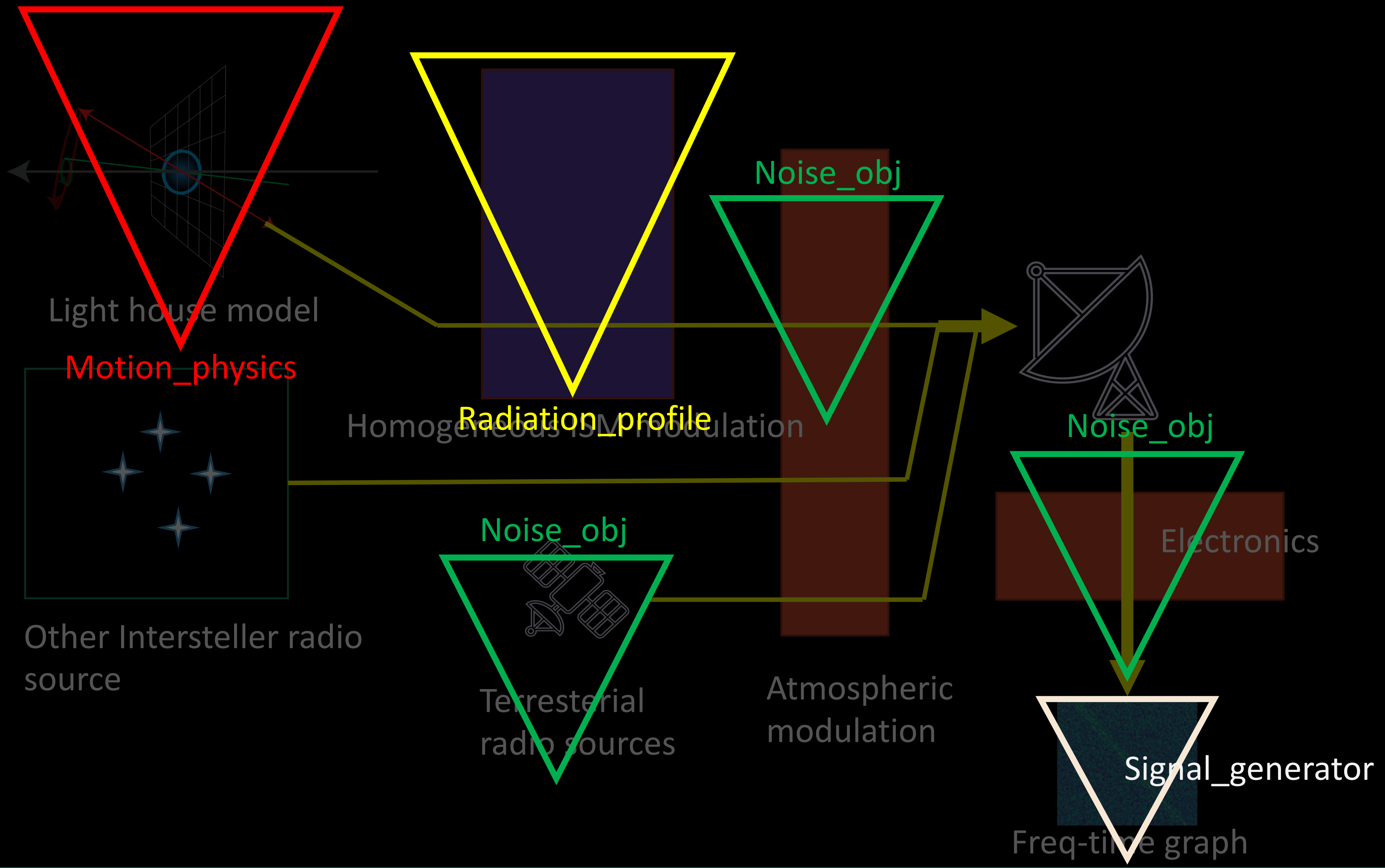
Terresterial radio sources

Atmospheric modulation

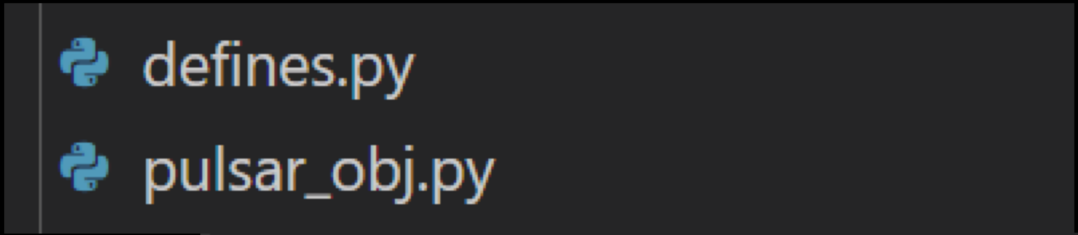
Electronics

Freq-time graph

simulation using OOP in python (modular)

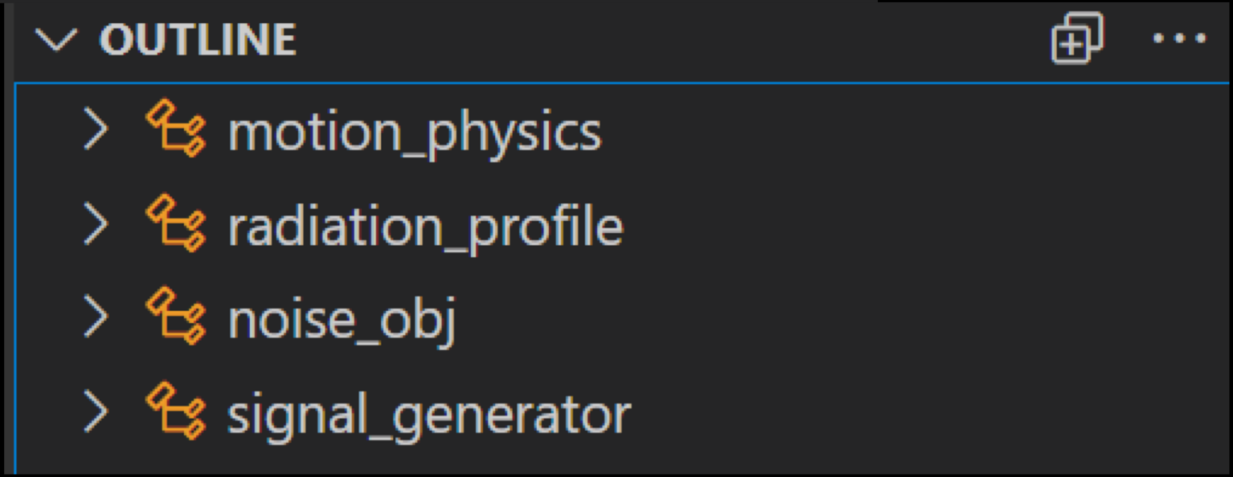


simulation using OOP in python



defines.py

pulsar_obj.py



OUTLINE

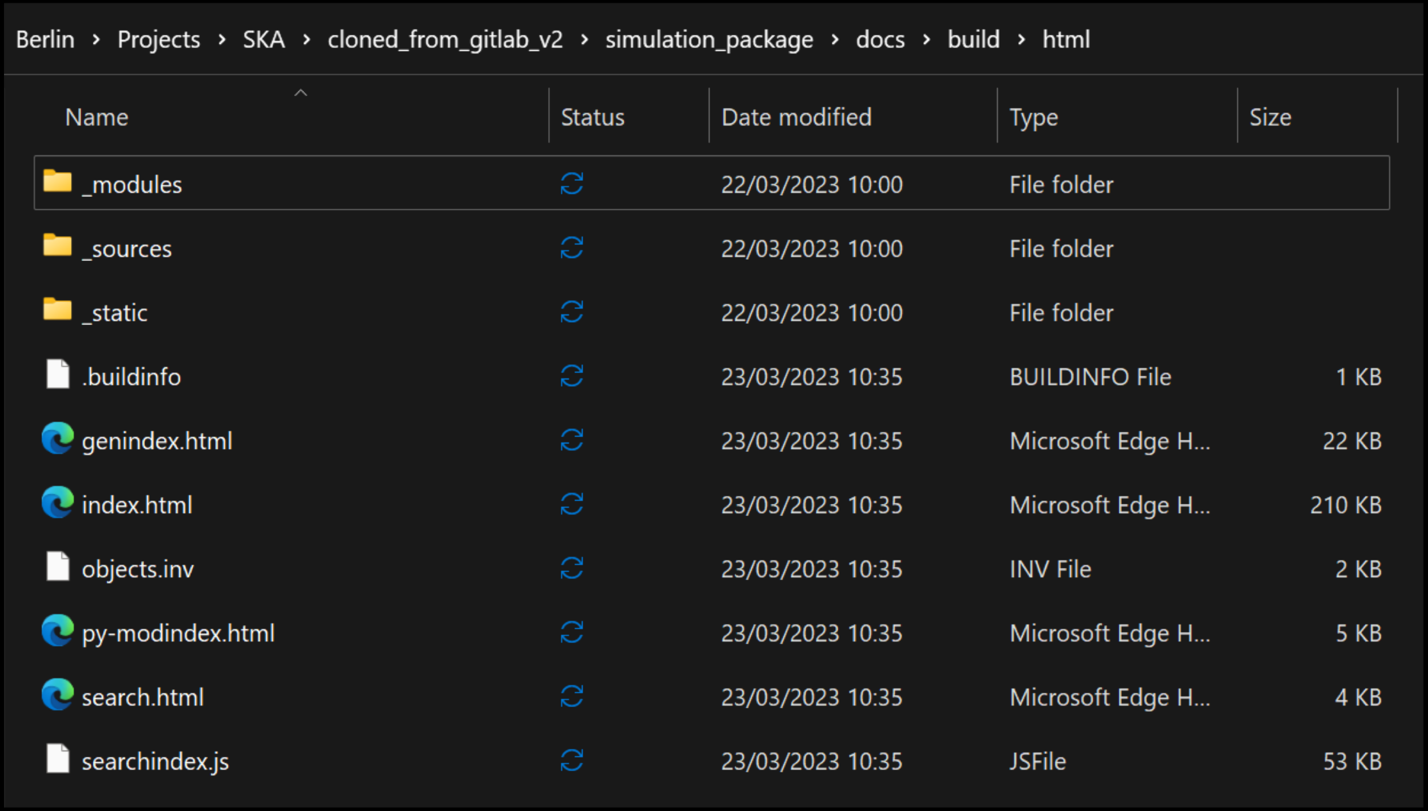
motion_physics

radiation_profile

noise_obj

signal_generator

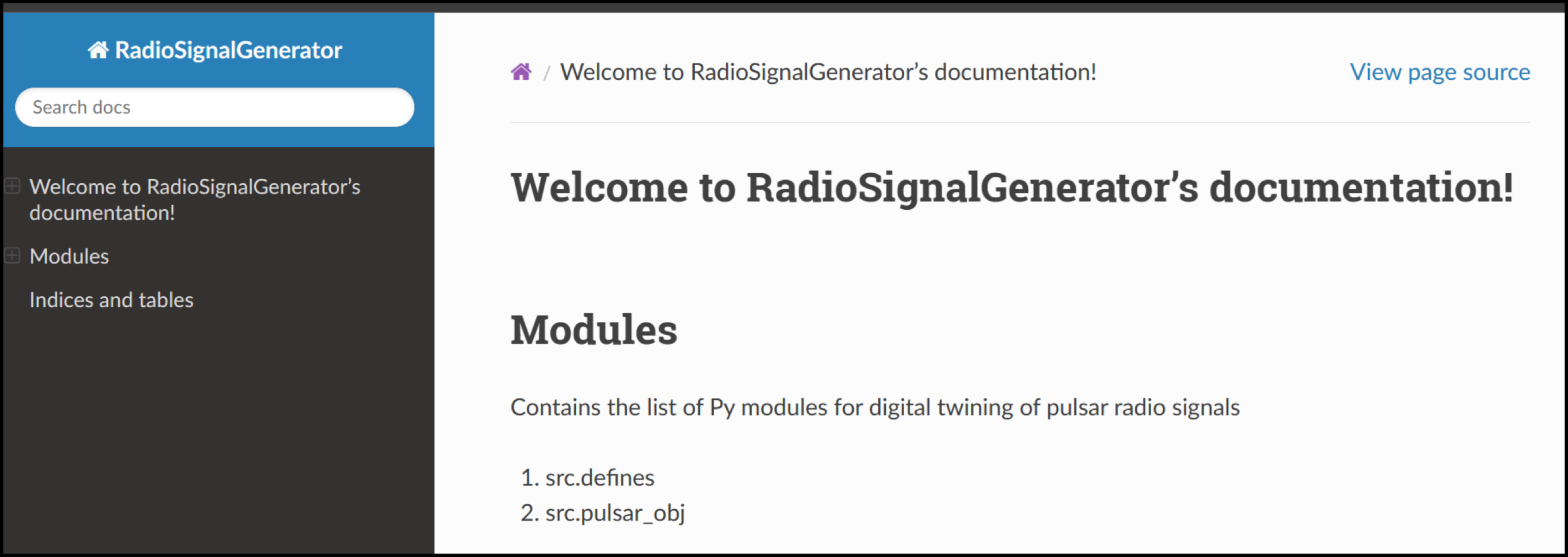
Auto-documentation of the code



Berlin › Projects › SKA › cloned_from_gitlab_v2 › simulation_package › docs › build › html

Name	Status	Date modified	Type	Size
_modules	⟳	22/03/2023 10:00	File folder	
_sources	⟳	22/03/2023 10:00	File folder	
_static	⟳	22/03/2023 10:00	File folder	
.buildinfo	⟳	23/03/2023 10:35	BUILDINFO File	1 KB
genindex.html	⟳	23/03/2023 10:35	Microsoft Edge H...	22 KB
index.html	⟳	23/03/2023 10:35	Microsoft Edge H...	210 KB
objects.inv	⟳	23/03/2023 10:35	INV File	2 KB
py-modindex.html	⟳	23/03/2023 10:35	Microsoft Edge H...	5 KB
search.html	⟳	23/03/2023 10:35	Microsoft Edge H...	4 KB
searchindex.js	⟳	23/03/2023 10:35	JSFile	53 KB

Auto-documentation of the code

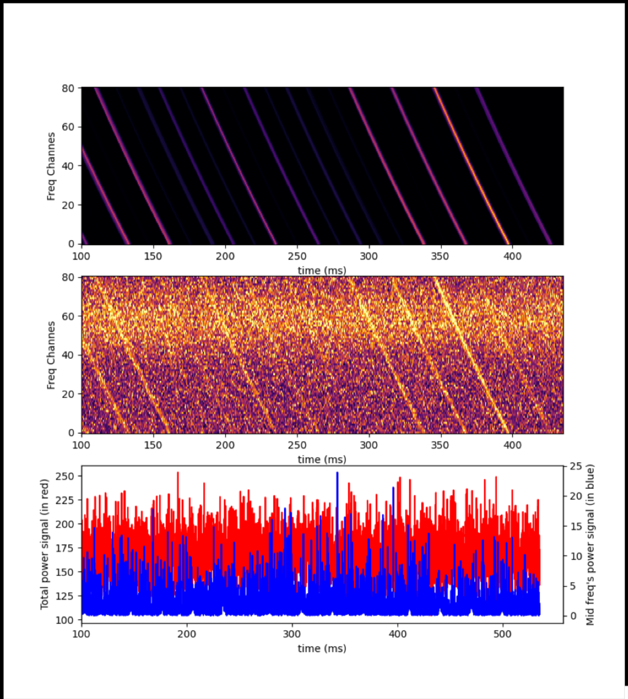

RadioSignalGenerator
Search docs
Welcome to RadioSignalGenerator's documentation!
Modules
Indices and tables
Welcome to RadioSignalGenerator's documentation!
View page source

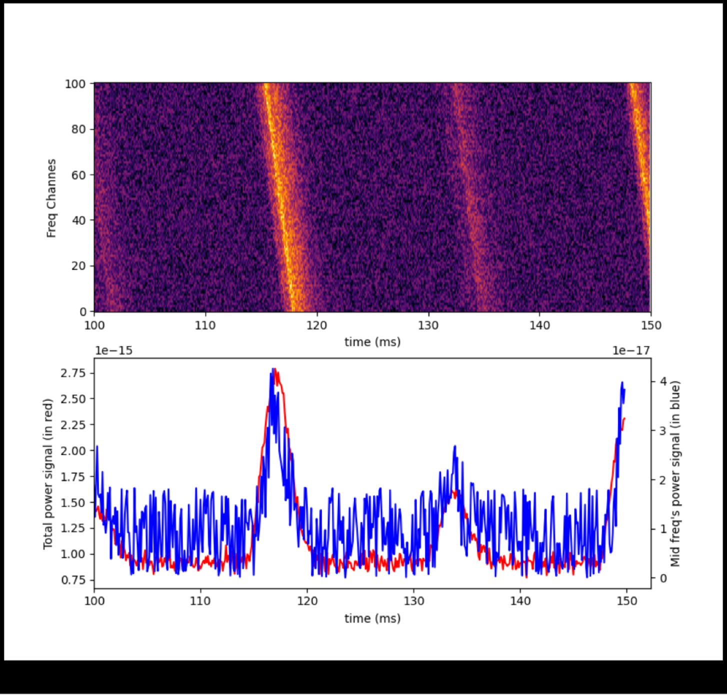
Welcome to RadioSignalGenerator's documentation!

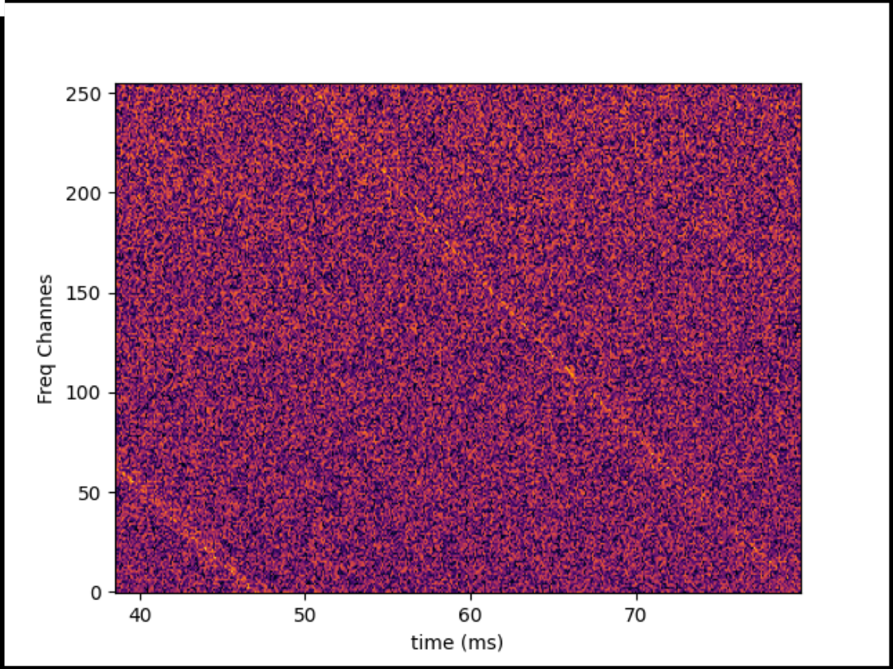
Modules

Contains the list of Py modules for digital twining of pulsar radio signals

1. src.defines
2. src.pulsar_obj


Freq Channes
80
60
40
20
0
100
150
200
250
300
350
400
time (ms)
Freq Channes
80
60
40
20
0
100
150
200
250
300
350
400
time (ms)
Total power signal (in red)
250
225
200
175
150
125
100
100
200
300
400
500
time (ms)
Mid freq's power signal (in blue)
25
20
15
10
5
0


Freq Channes
100
80
60
40
20
0
100
110
120
130
140
150
time (ms)
1e−15
1e−17
Total power signal (in red)
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
Mid freq's power signal (in blue)
4
3
2
1
0
100
110
120
130
140
150
time (ms)


Freq Channes
250
200
150
100
50
0
40
50
60
70
time (ms)

Task progress

Development of a modular Digital-twin of Pulsars

This includes:

1. Development of the basic physical model (animation) of pulsars based on existing theoretical models ✓

2. Modular design of the code/simulation using OOP in python ✓

3. Proper digital documentation of the code ✓

Future tasks

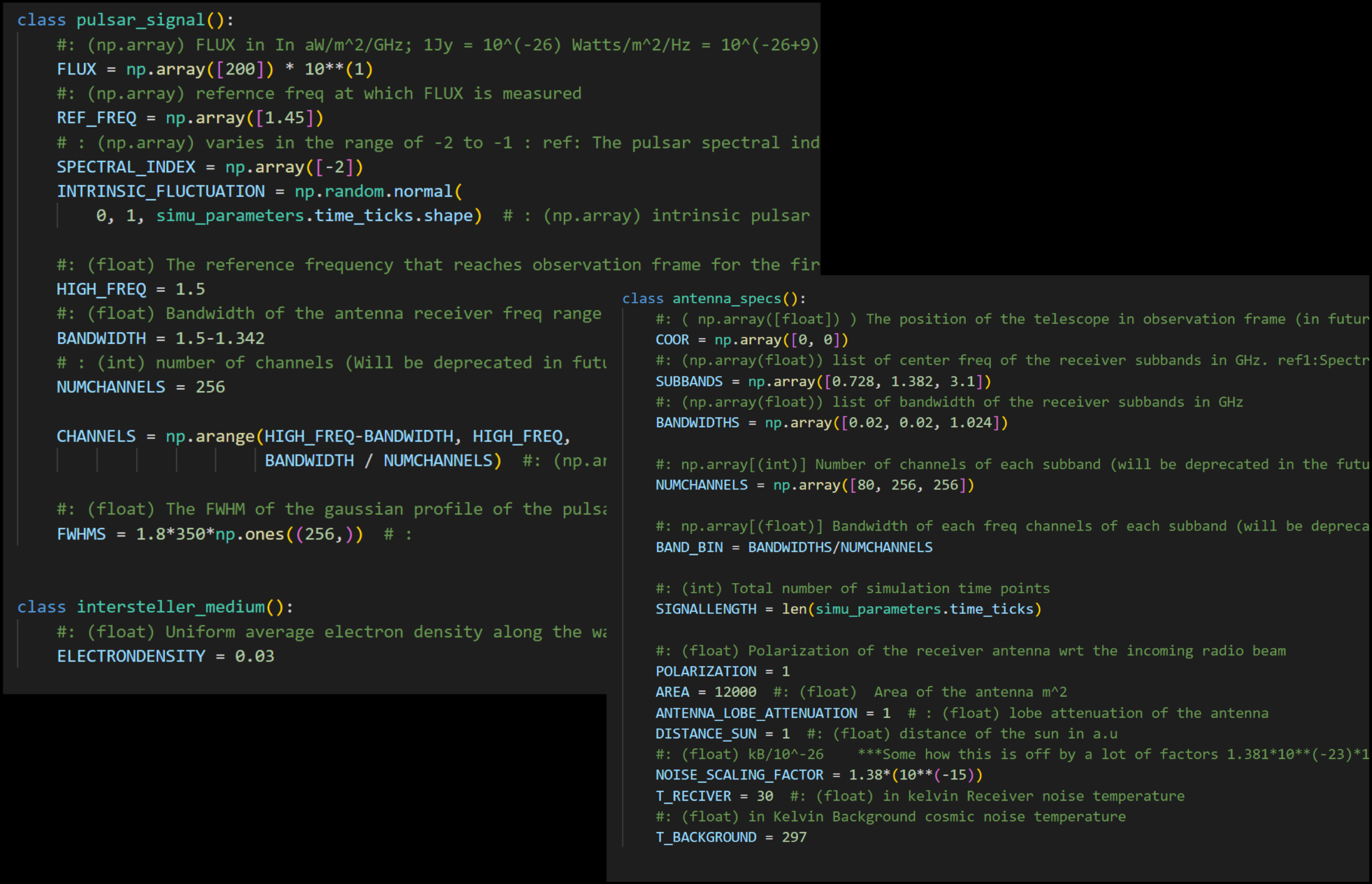
This includes:

1. Realization of model parameters

2. Amplification of pulsar signal using classical image-processing tools

3. Realization of the pulsar model based on more realistic physical models

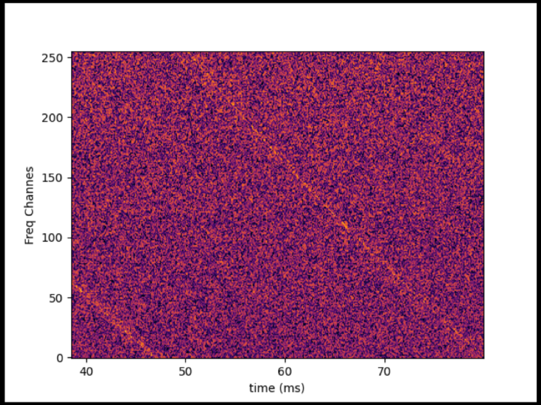
Realization of model parameters

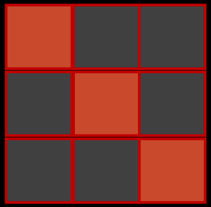


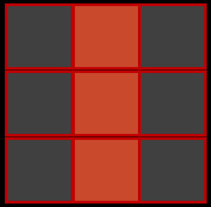
```python
class pulsar_signal():
    #: (np.array) FLUX in In aW/m^2/GHz; 1Jy = 10^(-26) Watts/m^2/Hz = 10^(-26+9)
    FLUX = np.array([200]) * 10**(1)
    #: (np.array) refernce freq at which FLUX is measured
    REF_FREQ = np.array([1.45])
    # : (np.array) varies in the range of -2 to -1 : ref: The pulsar spectral ind
    SPECTRAL_INDEX = np.array([-2])
    INTRINSIC_FLUCTUATION = np.random.normal(
        0, 1, simu_parameters.time_ticks.shape)  # : (np.array) intrinsic pulsar

    #: (float) The reference frequency that reaches observation frame for the fir
    HIGH_FREQ = 1.5
    #: (float) Bandwidth of the antenna receiver freq range
    BANDWIDTH = 1.5-1.342
    # : (int) number of channels (Will be deprecated in futu
    NUMCHANNELS = 256

    CHANNELS = np.arange(HIGH_FREQ-BANDWIDTH, HIGH_FREQ,
                         BANDWIDTH / NUMCHANNELS)  #: (np.ar

    #: (float) The FWHM of the gaussian profile of the pulsa
    FWHMS = 1.8*350*np.ones((256,))  # :


class intersteller_medium():
    #: (float) Uniform average electron density along the wa
    ELECTRONDENSITY = 0.03
```

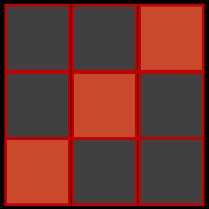
```python
class antenna_specs():
    #: ( np.array([float]) ) The position of the telescope in observation frame (in futur
    COOR = np.array([0, 0])
    #: (np.array(float)) list of center freq of the receiver subbands in GHz. ref1:Spectr
    SUBBANDS = np.array([0.728, 1.382, 3.1])
    #: (np.array(float)) list of bandwidth of the receiver subbands in GHz
    BANDWIDTHS = np.array([0.02, 0.02, 1.024])

    #: np.array[(int)] Number of channels of each subband (will be deprecated in the futu
    NUMCHANNELS = np.array([80, 256, 256])

    #: np.array[(float)] Bandwidth of each freq channels of each subband (will be depreca
    BAND_BIN = BANDWIDTHS/NUMCHANNELS

    #: (int) Total number of simulation time points
    SIGNALLENGTH = len(simu_parameters.time_ticks)

    #: (float) Polarization of the receiver antenna wrt the incoming radio beam
    POLARIZATION = 1
    AREA = 12000  #: (float)  Area of the antenna m^2
    ANTENNA_LOBE_ATTENUATION = 1  # : (float) lobe attenuation of the antenna
    DISTANCE_SUN = 1  #: (float) distance of the sun in a.u
    #: (float) kB/10^-26    ***Some how this is off by a lot of factors 1.381*10**(-23)*1
    NOISE_SCALING_FACTOR = 1.38*(10**(-15))
    T_RECIVER = 30  #: (float) in kelvin Receiver noise temperature
    #: (float) in Kelvin Background cosmic noise temperature
    T_BACKGROUND = 297
```

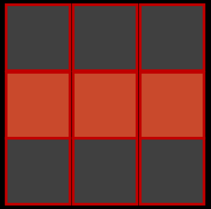
Amplification of pulsar signal using classical image-processing tools

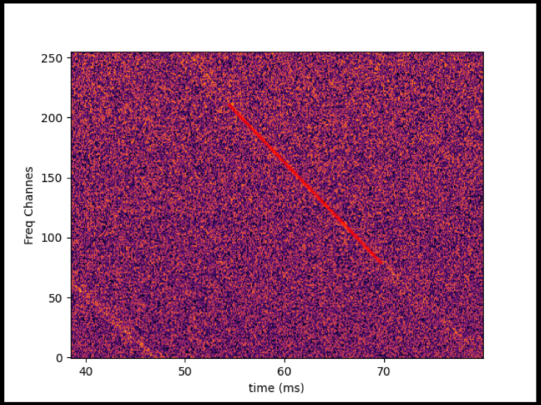




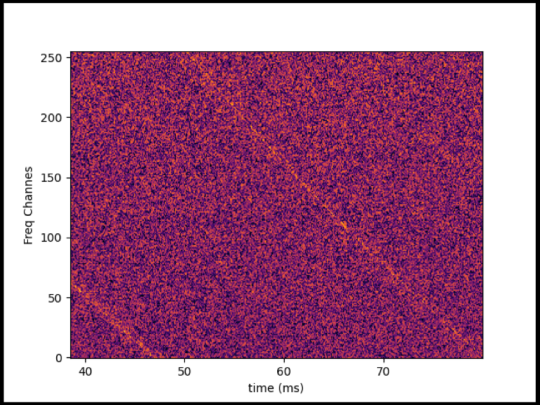




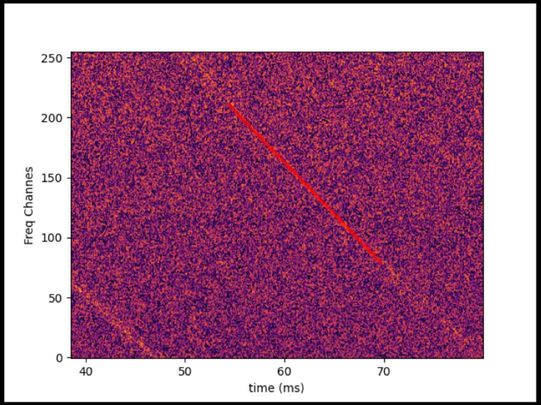




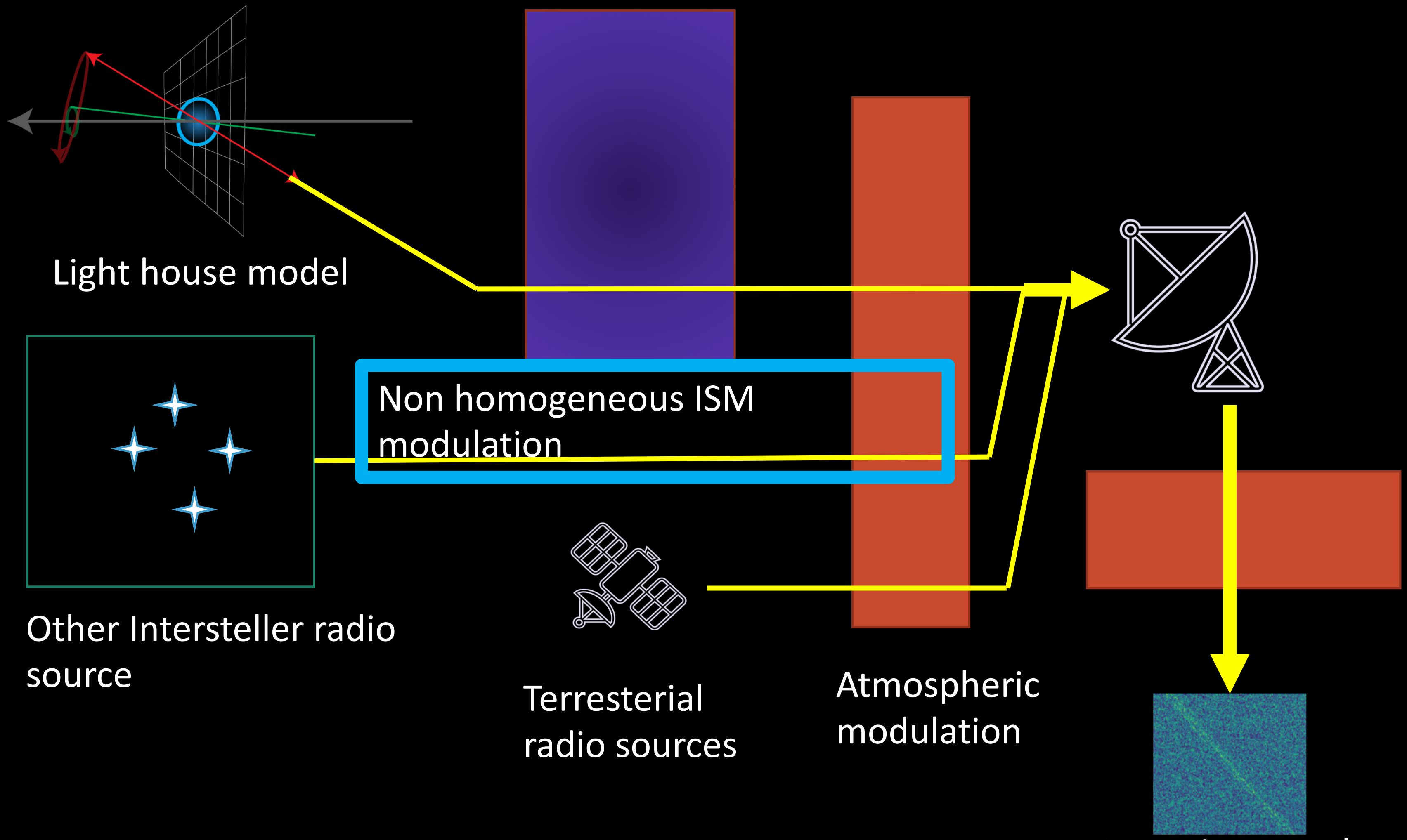
Amplification of pulsar signal using classical image-processing tools



Line/parabola detection models



Realization of the pulsar model based on more realistic physical models



Light house model

Non homogeneous ISM modulation

Other Intersteller radio source

Terresterial radio sources

Atmospheric modulation

Freq-time graph