ML-based Pipeline for Pulsar Analysis (ML–PPA)

Andrei Kazantsev
akazantsev@mpifr-bonn.mpg.de

Tim Oelkers
Tim.Oelkers@student.htw-berlin.de

Yurii Pidopryhora
yurii@astro.uni-bonn.de

Tanumoy Saha
tanumoy.saha@htw-berlin.de

Marcel Trattner
marcel.trattner@HTW-Berlin.de

Hermann Heßling
hessling@HTW-Berlin.de

November 6, 2025

Abstract

ML–PPA (Machine Learning-based Pipeline for Pulsar Analysis) is a new software framework that addresses the challenge of identifying pulsar signals from large-volume data streams during the data acquisition phase. A main task is to identify interference signals coming from different sources. For verification, ML-PPA simulates the propagation of pulsar signals from the source to radio astronomical antennas. Another focus is the development of an analysis pipeline that can be efficiently used for massively parallel computing.

ML-PPA follows the layered software design of the project PUNCH4NFDI [1]: below the user interface layer, "algorithmic tools" are provided that can be combined to build an analysis pipeline. At the bottom level are resources for creating a "pipeline container" that can be used for processing data or generating synthetic data (digital twins, as outlined in the project interTwin [2]).

The framework aims to empower astronomers in their pursuit of uncovering fascinating astronomical signals and enhancing their ability to analyze and interpret large-scale astronomical data.

1 Introduction

In traditional experiments, all data is collected at a nearby data center where an initial analysis is performed, and the resulting data products are then distributed to different data centers for detailed study by scientists, see the large blue arrows in Fig. 1.

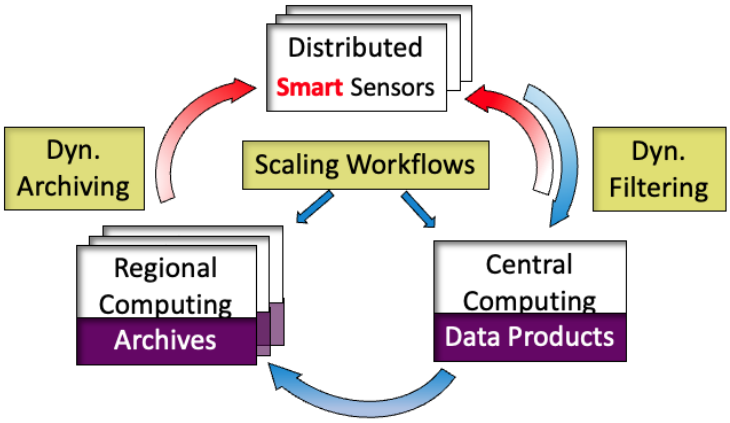

Figure 1: Dynamic Life Cycle Model [3]

The Square Kilometer Array Observatory (SKAO) will produce so much data that, at best, fractions of it can be archived. A selection of "relevant information" from large data streams will require the development of new concepts. As a possible strategy, two feedback loops are proposed, see the red arrows in, see Fig. 1. In the online computing phase, there is not enough time to analyze incoming data streams fully with existing pipelines as they are too complex, i.e. decisions about what to archive are based on incomplete information and exactly this feature inevitably leads to irreversible information loss. To minimize "Data Irreversibility", quick preliminary analyses should be performed during data acquisition to recalibrate control parameters if necessary (see the red arrow on the right hand side of Fig. 1). Further verification of the quality of the data should be done in the off-line phase, especially by comparison with simulation data. If appropriate findings are available, offline computing should also be given the opportunity to optimize control parameters as promptly as possible (see the second red arrow in Fig. 1).

Existing frameworks in radio astronomy can be used for parallel computing, however not efficiently on more than about 8 nodes [4]. Scaling workflows is most relevant not only for offline computing but also for online computing (see the small blue arrows in Fig. 1).

An important goal of this project is to show that significantly better scaling behavior in radio astronomy is possible, in principle, if the design of a pipeline is optimized. The overall software architecture follows the 3-layer architecture of the PUNCH4NFDI consortium [1], see Fig. 2. The top layer provides interfaces for the user to develop pipelines by combining tools and algorithms that are held in the middle layer. The bottom layer enables the creation of containers that can be distributed to data centers where the built-in pipelines can be used to analyze or generate data.

Our project focuses on a limited task: the identification of pulsar signals

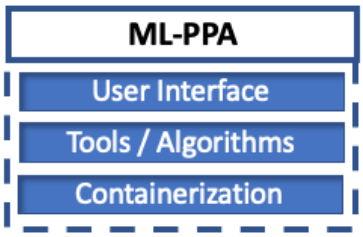

Figure 2: Three–layer architecture of ML–PPA

during data acquisition, see Fig. 3. The very first version (0.1) of our framework includes

- classified real data (from the Effelsberg telescope): pulsar signals from the Crab Nebula and different RFI signals,
- classified synthetic data (digital twins) of pulsar and RFI signals,
- a pipeline for generating synthetic pulsar signals based on a simplified model (see Sec. 3 for details),
- a layered software architecture (see Fig. 2 and the right hand side of Fig. 3): a Python-based user interface on the top of the medium layer with various modules (some of them are written in C++),
- tools for generating a container with all the necessary pipeline components to perform a data analysis, training of Neural Network Models or generation of synthetic data (digital twins).

Realizing a feedback loop from ML–PPA to the sensors (e.g. the telescopes of SKAO) is intended for later versions. The framework ML-PPA has considerable generalization potential and should be interesting for similar, time-critical ML pipelines. The ML approach pursued here - efficient code and its provision with the help of containers - could serve as a blueprint for projects with comparable challenges.

This document is organized as follows. Section 2 reviews important fundamentals of pulsar signals and shows the characteristics of radio frequency interference (RFI). In addition, important aspects to be considered when generating synthetic pulsar and RFI signals are explained. Section 3 describes the currently used (simplified) model for simulating the propagation of pulsar signals from its source through the interstellar medium (ISM) to a telescope. Furthermore Convolutional Neural Networks (CNN) for separating pulsar signals from noise signals are described. Section 4 provides details on the software architecture of the framework ML–PPA. First results on the performance of selected components of the simulation model of Sec. 3 are shown. Section 5 gives a general overview of noise signals from telescope electronics. It is expected that an understanding of this type of noise signal will become important for large telescope systems (as in SKAO).

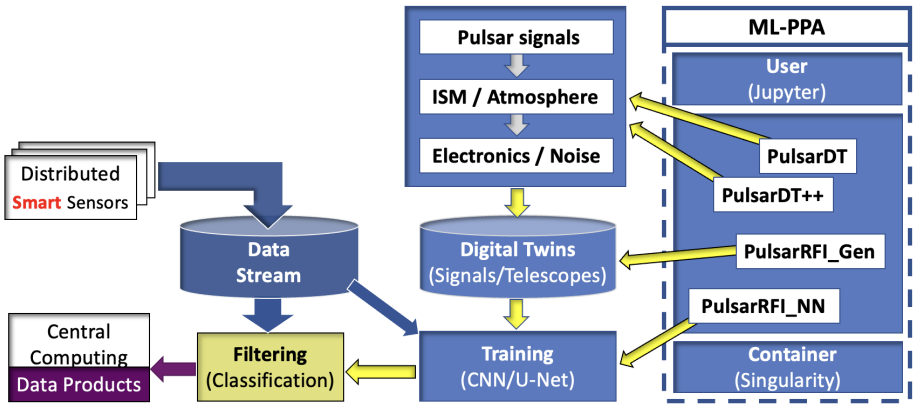

Figure 3: Integration of ML–PPA into the Dynamic Life Cycle Model.

2 Generation of Synthetic Datasets

2.1 Main Observational Features of Individual Pulses of Radio Pulsars

Pulsars are swiftly rotating and highly magnetized neutron stars. Similar to numerous objects in the cosmos, pulsars emit radiation over an extensive range of wavelengths. However, most pulsars have been detected within the radio spectrum. The ATNF catalog[1] has records about 3,389 radio pulsars. Radiation emitted by pulsars possesses observational features that enable us to differentiate their signals from, for instance, any anthropogenic radiation. It's important to highlight that these unique traits stem from the pulsating nature of pulsar emissions and are also common to other astronomical phenomena, such as Fast Radio Bursts.

One of the fundamental observational feature of pulses of extraterrestrial origin is the dispersion delay. An example of such a dispersion delay, as exhibited by the J1800+5034 pulsar, is illustrated in Figure 4. Its nature is explained by the frequency-dependent velocity of the wave packet as it propagates in the ionized interstellar medium. A signal at a higher frequency arrives earlier than one at a lower frequency. The time delay Δt between two frequency channels is described by

$$\Delta t = 4.12 \text{ ms } \left((f_{\rm LO}[\text{GHz}])^{-2} - (f_{\rm HI}[\text{GHz}])^{-2} \right) \text{DM} \, [\text{cm}^{-3}\text{pc}], \qquad (1)$$

where $f_{\rm LO}$ is the low frequency, $f_{\rm HI}$ the high frequency (both units specified in GHz), and DM the dispersion measure (its unit pc cm^{-3} is tailored to astronomy).

Signals with a characteristic dispersion time delay for a group of DM values are shown in Fig. 5.

[1] https://www.atnf.csiro.au/people/pulsar/psrcat/

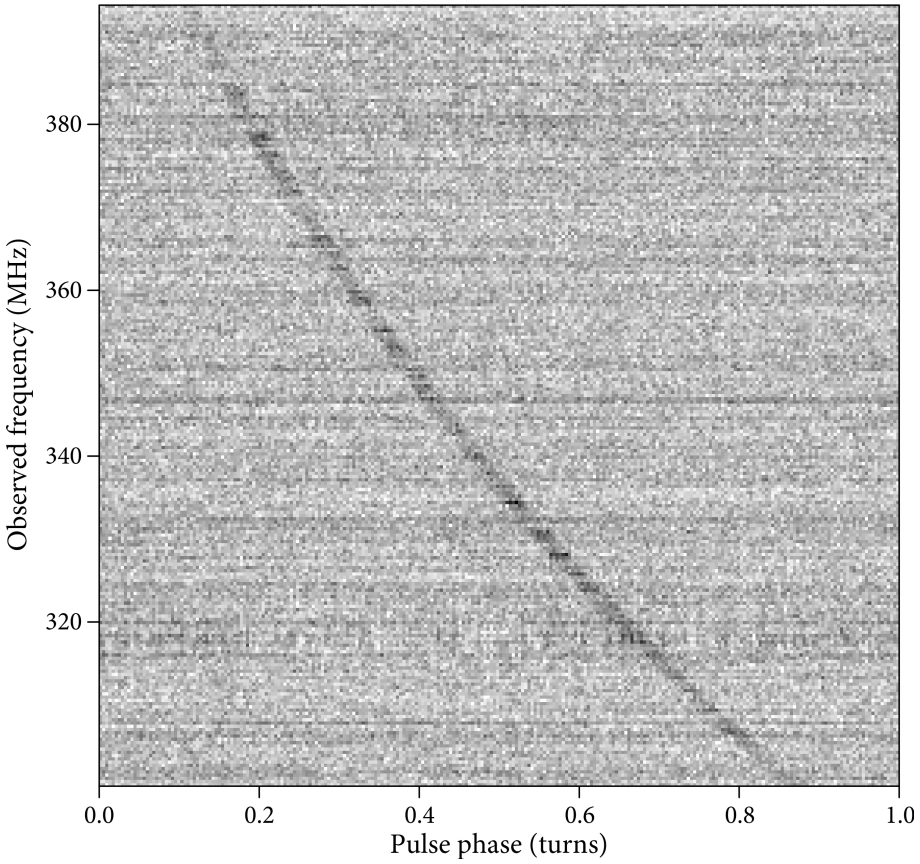

Figure 4: Example of dispersion delay for an individual pulse from J1800+5034 taken from [5].

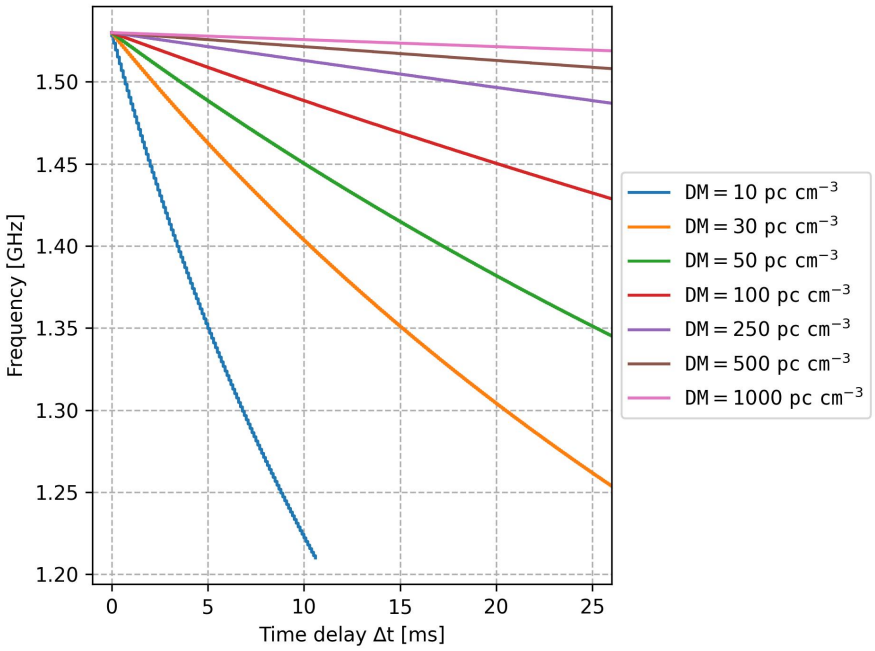

Figure 5: Dispersion–time delays for different DM (with fixed $f_{HI} = 1.53$ GHz).

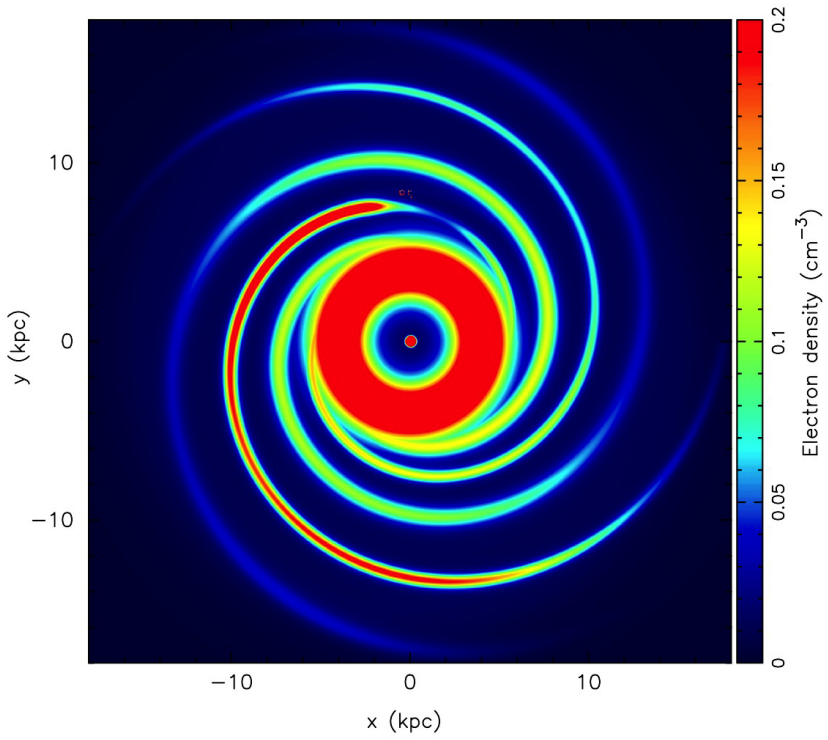

Figure 6: Electron density in the Galactic plane according to the YMW16 model (taken from [6]).

The physical meaning of the dispersion measure is the total number of electrons along the line of sight to the pulsar (or to another source of cosmic pulse radiation, e.g. Fast Radio Bursts):

$$\mathrm{DM} = n_e \ [\mathrm{cm}^{-3}] \ D \ [\mathrm{pc}] \tag{2}$$

where n_e is the mean electron density (i.e. the number of electrons per cm^3), and D is the distance to the source in parsecs [pc].

For preliminary estimations, an average electron density $n_e = 0.03$ cm^{-3} can be employed. For more accurate calculations, however, it is advisable to use models for the distribution of free electrons in the Galaxy, e.g., the model YMW16 [6].

Individual pulses, even those originating from the same pulsar, display considerable variation in shape and intensity. This concept is well demonstrated in recent studies on pulsars J0211+4235 and J0553+4111 [7]. Such significant variability precludes the ability to predict the final shape of an individual pulsar pulse in advance. However, by averaging a substantial number of individual pulses from a single pulsar, a so-called average profile (or mean profile) can be

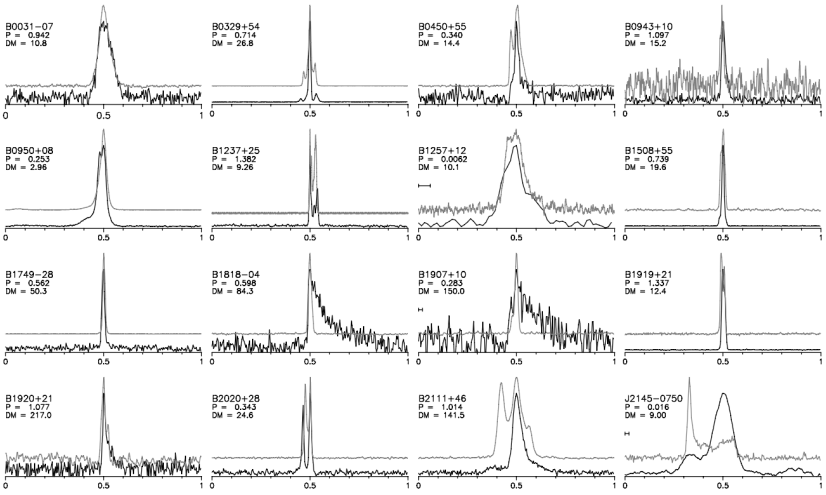

Figure 7: A sample of average pulse profiles from LOFAR observations of various pulsars taken from [10].

obtained. This profile remains stable over a long time period [8]. The average profiles of radio pulsars can be quite intricate and may include multiple profile components (see Fig. 7). However, despite the external complexity of the mean profiles, for a number of scientific tasks, the mean profiles of radio pulsars can be satisfactorily fitted by a Gaussian function or a sum of Gaussian functions in the case of a multicomponent profile [9].

When a broadband pulse passes through the interstellar medium, the pulse experiences scattering, which causes an intrinsically narrow pulse to become broadened in time as offaxis radiation is scattered back into the line of sight and arrives later at the observer due to the extra path length. Depending on the frequency of observations, the mean profile of the pulsar may acquire a tail in the right part of the profile (see Fig. 8). The value of the scattering time t_s depends not only on the frequency of observation ν but also on the value of the dispersion measure DM. For low values of the dispersion measures (DM $\leq$ 22.7 pc cm^{-3}), according to [11], the dependence is described by the formula

$$t_s[\text{ms}] = 3.5 \cdot 10^5 \left(\text{DM } [\text{pc cm}^{-3}]\right)^{2.2} (\nu \text{ [MHz]})^{-4.4} \tag{3}$$

For the other DM values, the dependence is described by the following formula:

$$t_s[\text{ms}] = 6.6 \cdot 10^{12} \left(\text{DM } [\text{pc cm}^{-3}]\right)^{4.2} (\nu \text{ [MHz]})^{-8.4} \tag{4}$$

The shape of the scattered profile can be calculated by convolution of the profile function and the thin screen model [12]

$$u(t) = \frac{1}{t_s} e^{-\frac{t}{t_s}} H(t), \tag{5}$$

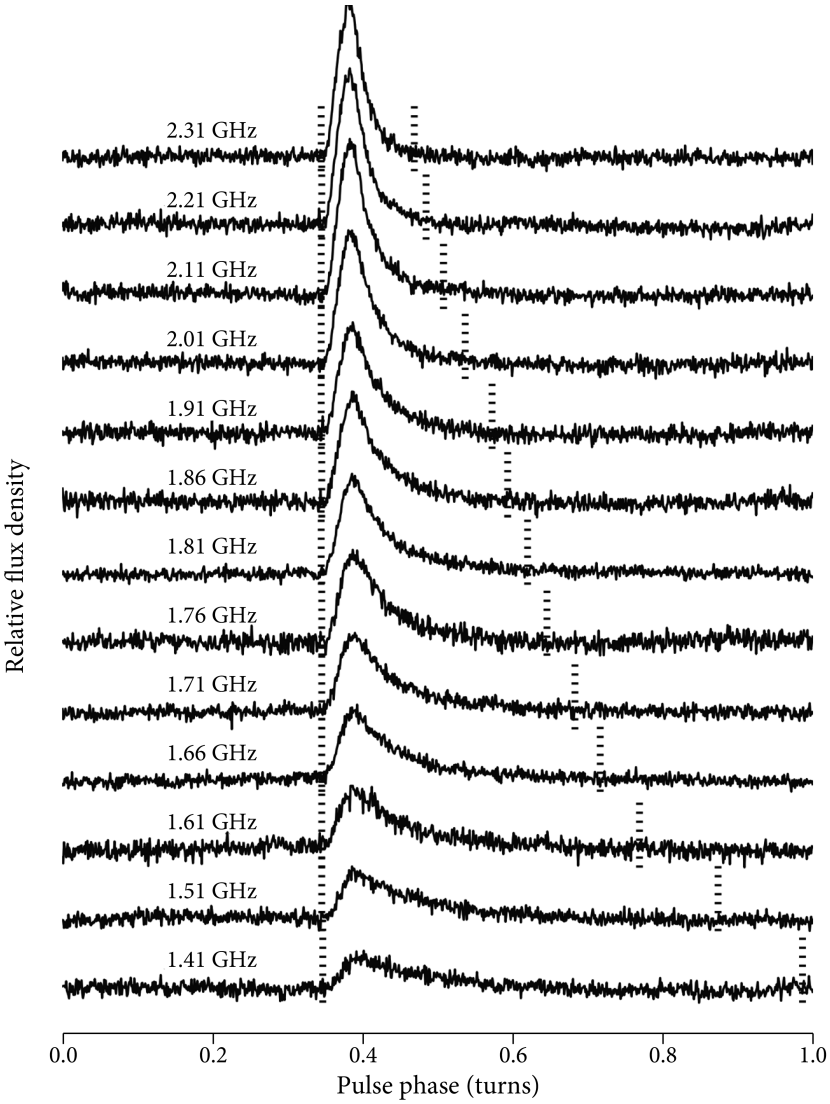

Figure 8: Examples of L- and S-band averaged profiles taken from [13].

where t_s is the scattering time, and $H(t)$ the Heaviside step function.

Another crucial factor that significantly affects the appearance of an individual pulsar pulse on a spectrogram is the pulsar's spectrum. This becomes particularly significant when observing over a broad frequency range, as pulses exhibit different peak flux densities depending on the observation frequency. An analysis of the spectra of 441 pulsars, as conducted in [14], reveals that for the vast majority of pulsars (79 %), the spectrum follows a simple power law model with a median exponent of -1.65.

2.2 Nature of Radio Frequency Interference

Before delving into the nature of radio frequency interference, it's important to note that the radio signals from space received by radio telescopes have significantly lower flux densities compared to sources radiating from Earth or from its orbit. A variety of interference sources, at least for 2010, is represented by the table presented in the Very Large Array (VLA) observation guide[2], see Tab. 1.

The existing radio interferences (RFI) can be roughly divided into two major types: narrowband (NBRFI) and broadband (BBRFI).

Narrowband RFI, alternatively known as Frequency Domain or Spectral Domain RFI, is a form of radio interference that occurs at a specific frequency (single-frequency RFI) or within a relatively narrow frequency range. The genesis of this interference can be traced back to anthropogenic or natural events emitted at a singular frequency or within a narrow frequency band, such as Wi-Fi, radio or TV stations, and cell phones, among others. Often, these radio interferences persist for a long period of time. This duration can be attributed to their sources producing a continuous broadcast, leading to a sustained interference registration as long as the source remains within the telescope's beam. An example of such RFI is shown on the left bottom panel of Fig. 10.

Broadband RFI, otherwise known as Time Domain RFI, typically stems from unintentional radiation emitted by sources such as inductive load switching. This form of RFI can be traced back to a variety of origins. Common sources encompass electric power transmission lines, electrical appliances, lighting systems, and other electronics. In certain instances, natural phenomena like solar flares and lightning strikes can also contribute to broadband RFI. A distinguishing aspect of this interference, as compared to its narrow frequency counterpart, is its span over almost the entirety of the utilized frequency band in addition to its short duration. The duration's extent cannot be definitively determined, as it is primarily dependent on the process that generates the interference. Given that the events causing this type of Radio Frequency Interference (RFI) are largely unpredictable, the signature of such interference tends to be quite random and varied. An example of such RFI is shown on the top right panel of Fig. 10.

[2] https://science.nrao.edu/facilities/vla/docs/manuals/obsguide/rfi

Frequency (MHz)	**Source**	**Comment**
1025–1150	Aircraft navigation	Very strong
1200	VLA modem	
1217–1237	GPS L2	Very strong
1243–1251	GLONASS L2	
1254	Aeronautical radar	
1268	Aeronautical radar	
1268	COMPASS E6	
1310	Aeronautical radar	
1317	Aeronautical radar	
1330	Aeronautical radar	
1337	Aeronautical radar	
1376–1386	GPS L3	Intermittent
1525–1564	INMARSAT satellites	
1564–1584	GPS L1	Very strong
1598–1609	GLONASS L1	
1618–1627	IRIDIUM satellites	
1642	2nd harmonic VLA radios	Sporadic
1683–1687	GOES weather satellite	
1689–1693	GOES weather satellite	
1700–1702	NOAA weather satellite	
1705–1709	NOAA weather satellite	
1930–1990	PCS cell phone base stations	
2178–2195	Satellite Downlink	Very strong*
2320–2350	Sirius/XM Satellite radio	Very strong*
3700–4200	Satellite Downlinks	Very strong*

Table 1: Some Examples of Strong RFI at the VLA Between 1 and 4 GHz

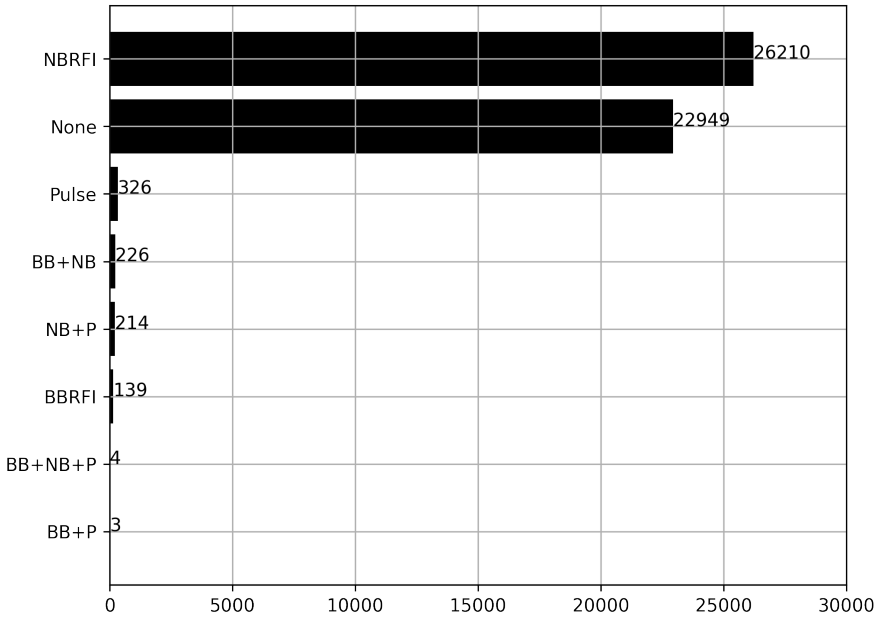

Figure 9: Statistics of real class exemplars taken from real Crab pulsar observations.

2.3 Conception of a Synthetic Dataset

Constructing machine and deep learning models in pulsar astrophysics can be complicated by the unbalanced classes recorded within a single session of real observations. For instance, observing the pulsar in the Crab Nebula will yield varying classes of pulsars (see Fig. 9). These classes can conventionally be segregated into two distinct groups. The first group, referred to as "pure" instances, comprises Pulse, Narrowband RFI (NBRFI), and Broadband RFI (BBRFI). The second group encompasses mixed classes, signifying multiple classes concurrently present on a spectrogram: Narrowband RFI and Pulse (NB + P), Broadband RFI and Pulse (BB + P), Broadband RFI and Narrowband RFI (BB + NB), and Broadband RFI, Narrowband RFI, and Pulse (BB + NB + P). The 'None' class, indicating the absence of any discernible signal on the spectrogram, merits separate mention. It should be noted that the label 'None' does not imply a complete absence of a signal. Instead, it indicates that, without the application of any additional techniques, it is not possible to unambiguously identify one of the pure classes or their combinations on the spectrogram. Illustrations of the "pure" or fundamental classes, inclusive of the 'None' class, are depicted in Fig. 10. Instances of mixed classes are demonstrated in Fig. 11.

Circling back to the fact that within one observation session, the recorded instances of different classes are highly imbalanced, the assembly of comprehen-

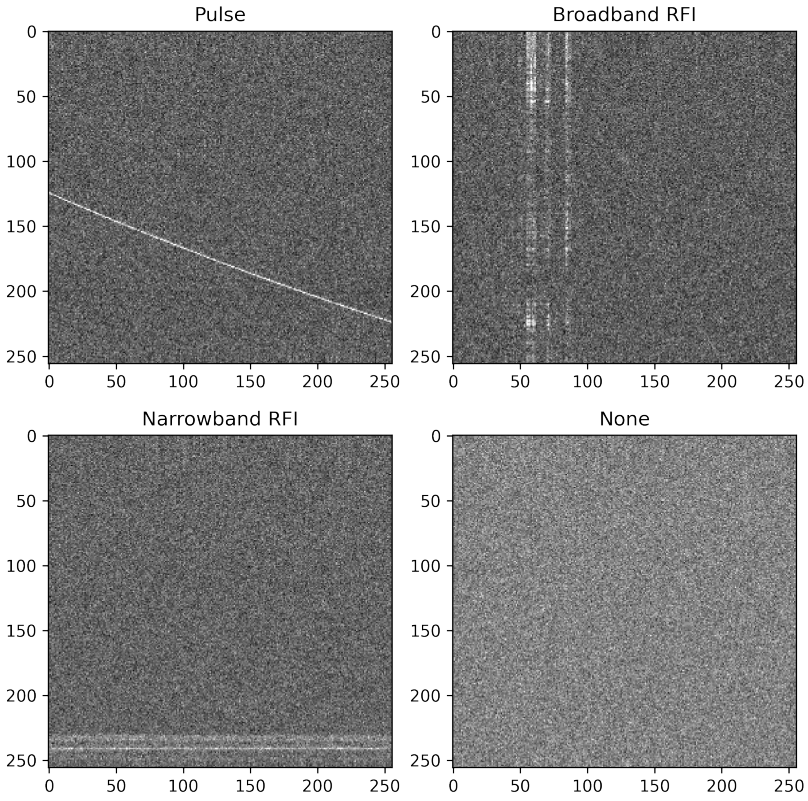

Figure 10: Examples of real basic class exemplars taken from real Crab pulsar observations carried out on 2020.05.31 with the Radio Telescope Effelsberg.

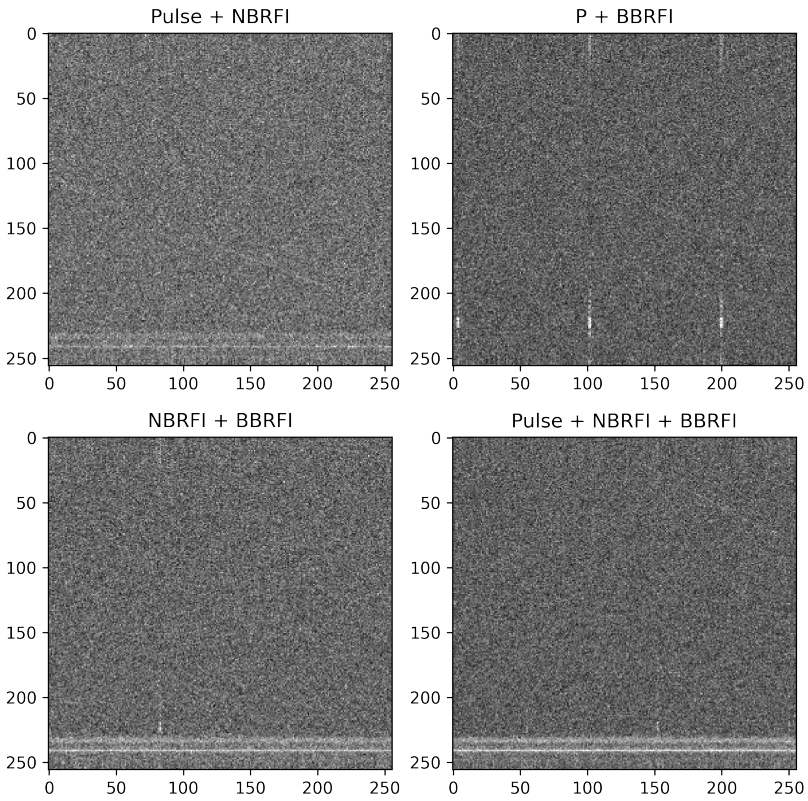

Figure 11: Examples of real mixed class exemplars taken from real Crab pulsar observations carried out on 2020.05.31 with the Radio Telescope Effelsberg.

sive training and validation samples—each containing a comparable number of instances—poses a particular challenge.

Furthermore, even within a single class (in this context, pulsar pulses and radio interference), there can be some heterogeneity, particularly in terms of signal brightness and its registration phase. In the case of the Crab pulsar, the pulse amplitude distribution follows a complex law as a combination of a log-normal and a power-law [15], which results in a limited number of pulses with high signal-to-noise ratios compared to the fainter pulses registered within a single observation session.

The issue of unbalanced classes in the training sample can be addressed by various techniques for handling unbalanced samples (Random Over-Sampling, Random Under-Sampling, Synthetic Minority Over-sampling Technique [16], or Adaptive Synthetic Sampling [17]), Generative Adversarial Networks (GAN) [18], and the generation of synthetic or artificial instances of corresponding classes. The latter approach possesses several advantages and a few disadvantages. Among the advantages is a high degree of control over the created datasets, signifying that the variation of all class exemplar parameters is directly determined by the developer, thereby offering the opportunity to balance the selection based on the task at hand. The major drawback is the inherent impossibility of generating synthetic data that perfectly replicates real data due to the multitude of factors contributing to real observational data. In the case of pulsars and interference, these can be effects associated with the pulsar itself, the telescope and receiving equipment, and the surrounding environment of the telescope.

Despite these challenges, the extensive period of pulsar research and the substantial knowledge accumulated allows us to argue that the primary effects contributing to the final form of the pulsar pulse are well understood, while other effects are of a lesser order. As for interference, despite the varied sources of generation, their final signature is relatively uniform, enabling the generation of convincingly synthetic data.

2.4 Generation of a Synthetic Pulsar Pulses

The concept of generating artificial pulsars is the strict or as close as possible to reality repetition of the basic observational features described in the section 2.1.

The synthetic individual pulse shape for a single frequency channel is represented as a Gaussian function with randomly superimposed masks to simulate the unevenness of the pulse profile. When accumulating a significant number of such pulses, an average profile mimicking the classic Gaussian function will emerge, which is consistent with the average profiles of known pulsars [9].

```
...
@staticmethod
@jit(nopython=True)
def gauss(x, amp, mu, sigma):
    return amp * np.exp(-(x - mu)**2 /( 2. * sigma**2))
...
```

```
@staticmethod
@jit(nopython=True, parallel=True)
def generate_pulses(gauss, amp, hw, loc, size, coeffs):
    pattern_pulses = np.empty((4096, size))

    for i in prange(4096):
        hw *= np.random.uniform(0.5, 1)
        pulse = gauss(np.arange(size), amp, loc, hw)
        pulse = pulse * np.random.uniform(0, 1, size)
        pattern_pulses[i] = coeffs[i] * pulse

    return pattern_pulses
...
```

In the code, *coeffs* is an array of coefficients used for adjusting the pulse intensity in accordance with the pulsar spectrum [14].

```
...
@staticmethod
@njit()
def spectra_func(array):
    return array**-1.65

freq_list = np.linspace(self.f_hi, self.f_lo, self.default_n_channes)
spectr = self.spectra_func(freq_list)
coeffs = (spectr / max(spectr))
...
```

The next step in creating a synthetic spectrogram of the pulsar pulse is the scattering procedure. This procedure is performed by convolution of the thin-screen model with each of the spectrogram sub-pulses, the number of which is set to 4096 by default. The scattering time is calculated according to the pulsar DM by the formulas (3), and (4). The results obtained by testing for a scattered pulse with DM = 73.3 pc cm^{-3} in the frequency range from 128.0 MHz to 230.4 MHz (see Fig. 12) are consistent with real pulsar observations with a similar DM in a similar frequency range[12]. It's crucial to remember that in section 2.1, we referred to an effect known as pulsar scattering. This does significantly impact the shape of the pulse, however, according to formulas 3 and 4, there are combinations of the dispersion measure and frequency where the scattering is much less than the used temporal resolution. Given that the procedure for creating a scattered pulse is computationally expensive, it hasn't been executed for the scenarios described.

After the pulse scattering procedure for each individual frequency channel, the pulse dispersion procedure takes place. Each pulse is shifted according to the time delay calculated by the formula (1) (see upper panels of Fig. 13). The next step is to reduce the number of frequency channels of the final spectrogram.

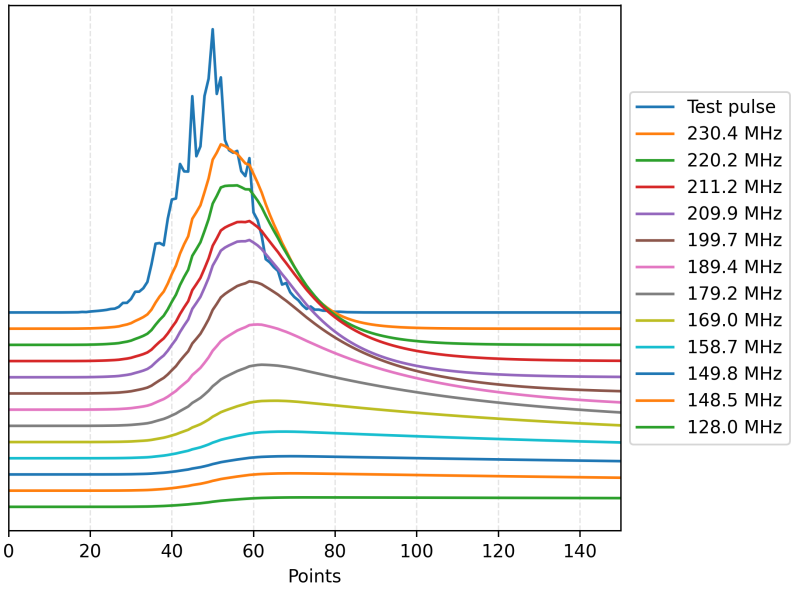

Figure 12: Example of scattered test pulse with DM = 73.3 pc cm^{-3} in frequency range from 128.0 MHz to 230.4 MHz.

This final number of frequency channels depends entirely on the task for which the synthetic data are generated. All frequency channels are divided into sub-bands and all frequency channels within the sub-band are summarized (see left bottom panel of Fig. 13). The obtained spectrogram can be used as a whole, or only its parts can be taken to form the spectrograms necessary for a particular task (see right bottom panel of Fig. 13).

The final step is to add noise to the spectrogram. The noise structure of real observations is described in detail in section 5. In the case of synthetic data creation, homogeneous Gaussian noise is used as background. By setting different noise amplitude values, spectrograms with different signal-to-noise ratios can be obtained (see Fig. 14).

Figures 15 and 16 show a comparison of the artificial pulsars obtained according to the method described above and the real pulsars of the pulsar in the Crab Nebula obtained from real observations at the Effelsberg radio telescope.

2.5 Generation of a Synthetic Narrowband Radio Frequency Interference

As previously discussed, Narrowband RFIs are interferences that predominantly emit in a significantly narrow band of frequencies when compared to the full bandwidth of the observation. The intensity of these interferences can fluctu-

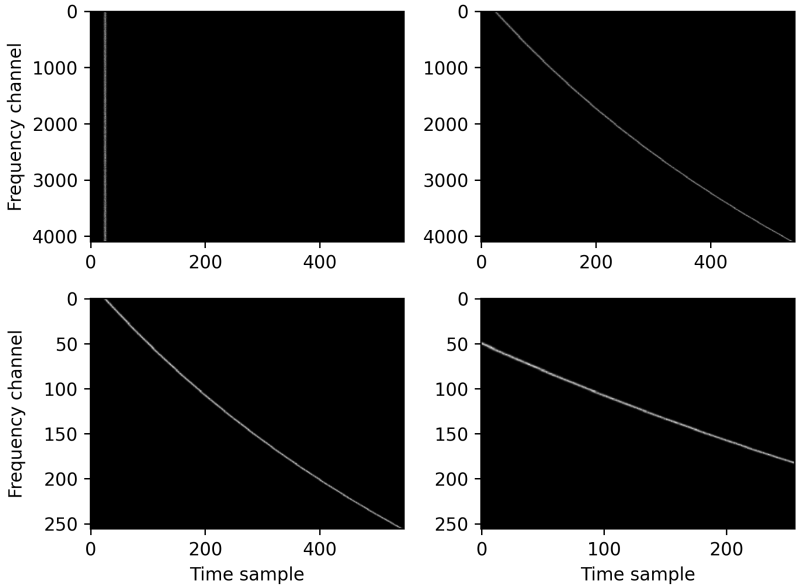

Figure 13: Stages of synthetic spectrogram formation with an individual pulse.

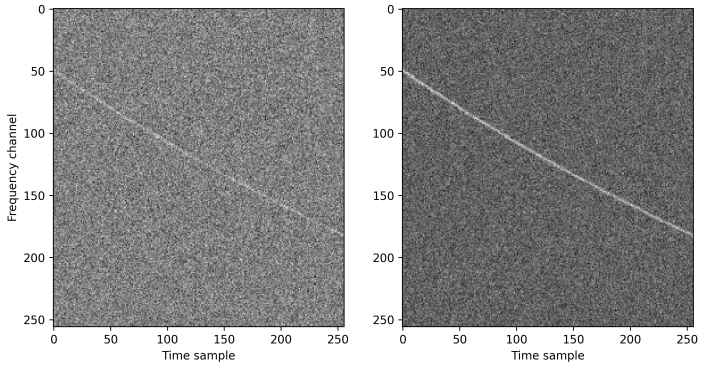

Figure 14: Examples of synthetic pulses with different signal-to-noise ratios.

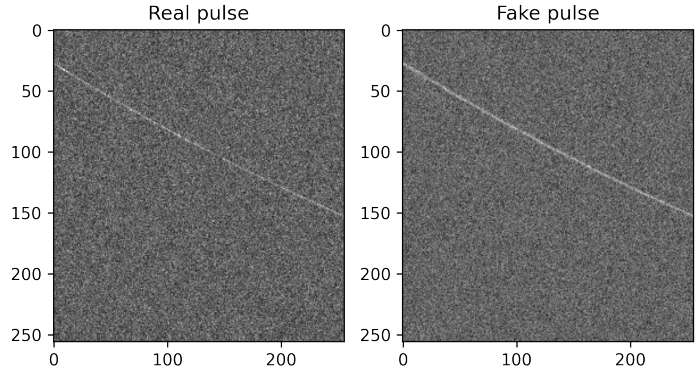

Figure 15: Comparison of the real pulse from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic pulse with Crab pulsar DM.

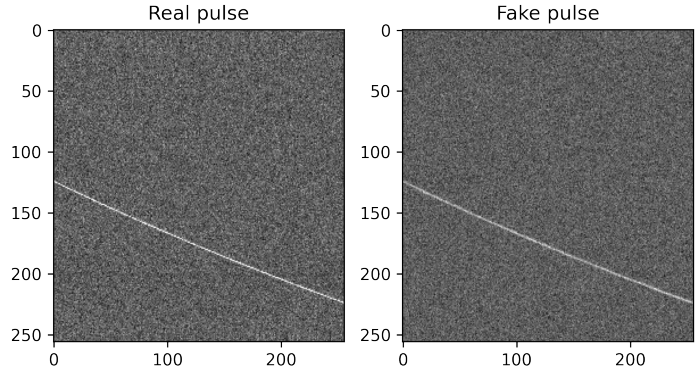

Figure 16: Comparison of the real pulse from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic pulse with Crab pulsar DM.

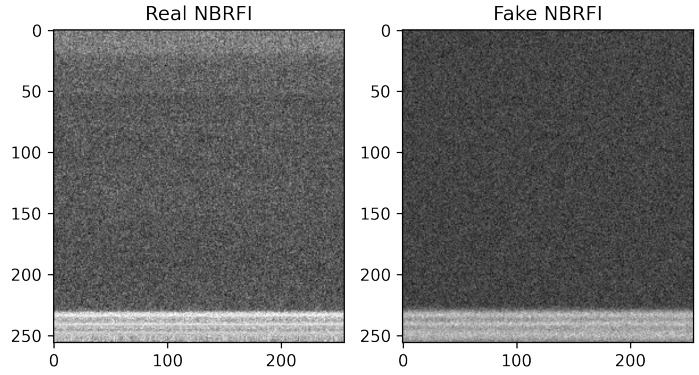

Figure 17: Comparison of the real NBRFI from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic NBRFI.

ate over time, dependent on the position of the interference source within the telescope's field of view and on the characteristics of the interference source itself. Nevertheless, within short time frames comparable to the observed pulsars' period, the interference intensity can be considered as having a constant value over time. In this scenario, the synthetic narrowband interference can be depicted as a Gaussian function within the frequency domain and expanded across the entire spectrogram. In this context, the Gaussian position represents the central frequency emission of the interference, while the Gaussian sigma signifies the emission bandwidth. It should also be noted that in real-world scenarios, a spectrogram may contain multiple instances of narrowband interference, each potentially possessing the same or different levels of brightness and emission bands. Simulating such interference can be achieved by combining previously described Gaussian functions with appropriate positions, widths, and amplitudes. Examples of real interferences of varying complexity and their synthesized counterparts are presented in Figures 17, 18, and 19.

2.6 Synthetic Broadband Radio Frequency Interference Signals

The challenge in creating synthetic broadband interference lies in the high level of randomness associated with its generation. This fact precludes the development of a single template for such interference, apart from the fact that on the spectrogram, this interference would resemble a vertical line of varying intensity across frequency channels and durations. Within a certain level of abstraction, broadband interference can be represented as a Gaussian function in the time domain, extended over all or a portion of the frequency channels. This mirrors the initial step of artificial pulse generation we described in section 2.4, except that the interference amplitude for each individual frequency channel experi-

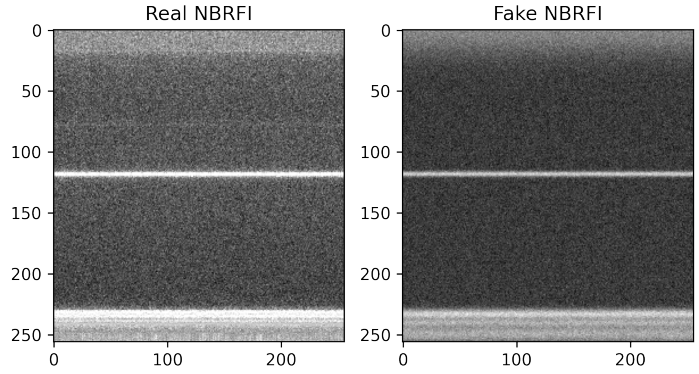

Figure 18: Comparison of the real NBRFI from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic NBRFI.

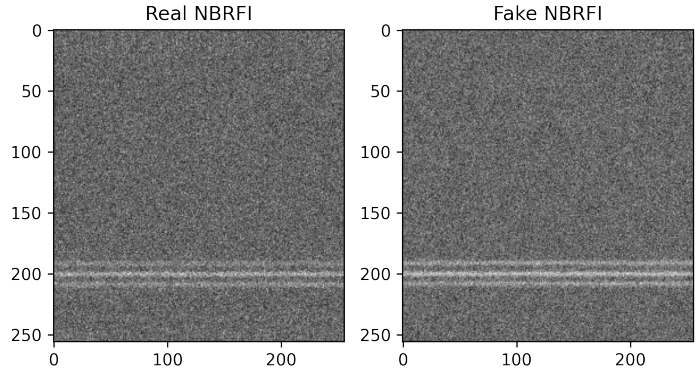

Figure 19: Comparison of the real NBRFI from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic NBRFI.

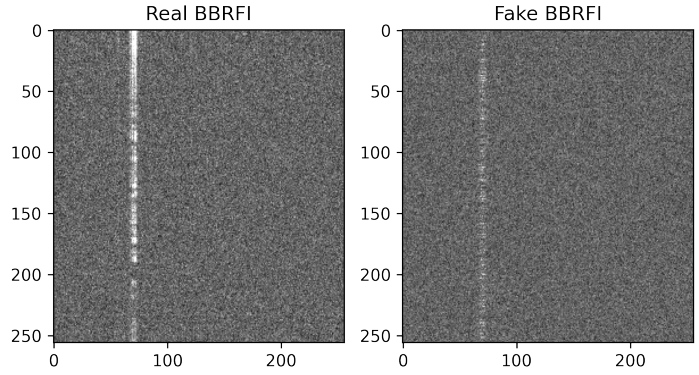

Figure 20: Comparison of the real BBRFI from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic BBRFI

ences greater variation. Examples of real interferences of varying complexity and their synthesized counterparts are presented in Figures 20, and 21.

3 Simulation of Pulsar Signals

3.1 Simulation of Pulsar Animators

3.1.1 Physics of the Pulsar Animator

The objective of this project is to provide astronomers and physicists with a tool to design and visualize various pulsar configurations, which would result in specific signal patterns. These patterns can then be utilized to train neural network models capable of identifying potential pulsar candidates from vast data streams in near real-time. To simplify the initial software release, we assume the pulsar to be a distant, rigid rotating body. By employing the basic kinematic equation of a rotating object and having knowledge of the pulsar's initial state at a particular time, its rotation can be predicted. The magnetic axis $\vec{m}_t$ of the pulsar is constrained by the rigid body and passes through its two distinct magnetic poles. As a result, the magnetic axis can be represented as a three-dimensional vector at any given time

$$\vec{m}_t = \boldsymbol{R}\,\vec{m}_{t=0}, \boldsymbol{R} = \boldsymbol{R}_{\text{trans}}\,\boldsymbol{R}_t \tag{6}$$

where $\boldsymbol{R}_{\text{trans}}$ is the transformation matrix of the pulsar to Earth's observation plane accounting for the pulsar rotation axis tilt with respect to the antenna viewing direction, and $\boldsymbol{R}_t$ is the time dependent rotation matrix to transform the magnetic axis into a new direction based on time t. The method described above is implemented in the file "src.pulsar_simulation.synthetic_dataGen.py". This

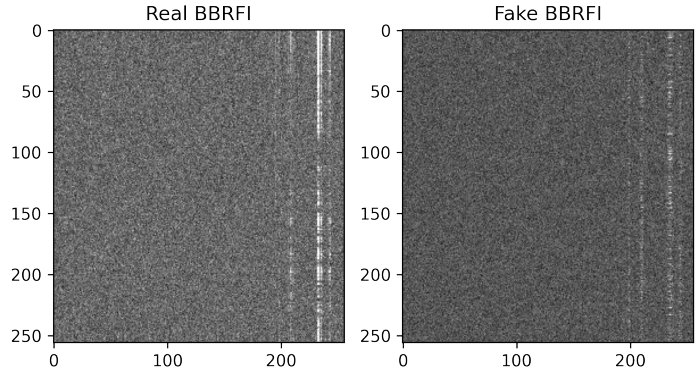

Figure 21: Comparison of the real BBRFI from Crab pulsar (2020.05.31, Radio Telescope Effelsberg) and synthetic BBRFI

file handles the generation of synthetic data for the project. The rotation mechanism is further illustrated in the Jupyter notebook "synthetic_dataGen.ipynb". This notebook provides a visual demonstration and explanation of how the rotation of the pulsar affects the generated data. Implementation of the code is shown below

```
from src.pulsar_simulation.synthetic_dataGen import synthetic_dataGen
t = 0
pulsar_obj = synthetic_dataGen(time_ticks=time_ticks).pulsar(pos=pulsar_pos)
rot_vect_from_earth = \
pulsar_obj.motion_physics(
            rot_params=rot_params).calc_rotation_axis_inEarthsframe()
magAxis_vect_from_earth = \
pulsar_obj.motion_physics(
            rot_params=rot_params).calc_pulsarBeam_orientation(t=t)
mag_axis_vect_mag2wavelettop,mag_axis_vect_mag2wavelet = \
pulsar_obj.motion_physics(
            rot_params=rot_params).calc_poynting_vectWaveLetFrame(t=t)
BeemCenter_dis,distance_travelled = \
pulsar_obj.motion_physics(
            rot_params=rot_params).calc_BeamProperties(t=t)
```

The radio signals emitted by the pulsar propagate through the interstellar medium, following the path of the magnetic axis (approximated, in reality it travels along the magnetic field lines which will be incorporated in the future release). For the purpose of this project, we approximate the interstellar medium as homogeneous. To simplify the model, we further approximate the radio pulse to have a Gaussian profile, as depicted in Fig. 22.

To facilitate calculations of the pulsar flux, we define a radio pulse plane. This plane is perpendicular to the rotation axis of a non-wobbling pulsar, and the antenna is positioned at the origin of this plane. The main motivation behind defining this plane is to simplify the pulsar flux calculation within a highly simplified model. In this model, the intensity or flux $f(d)$ received by

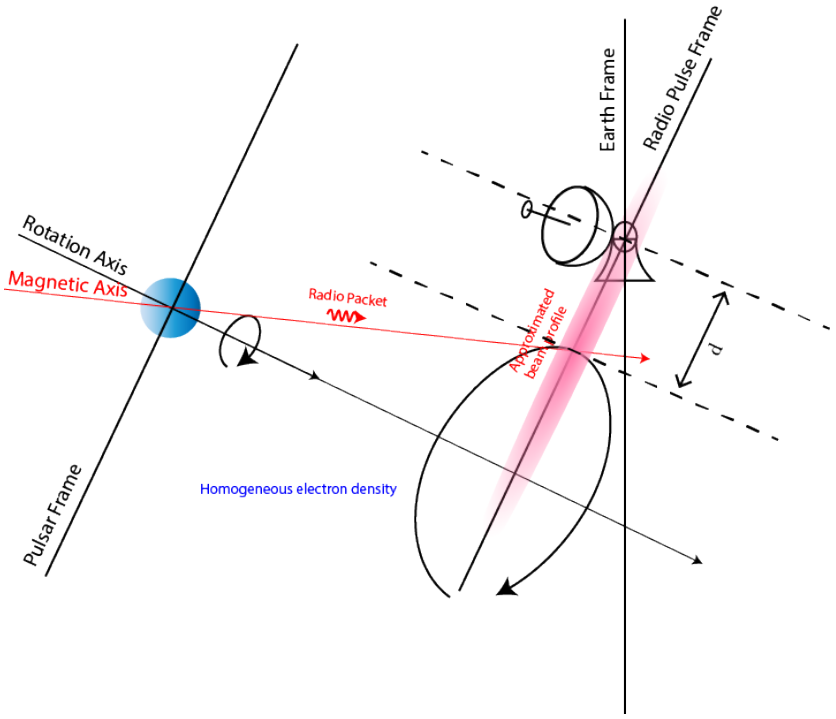

Figure 22: **Geometry of the Pulsar Animator Model**. *The Pulsar animator is employed as an oscillator source to generate pulsar signals, aiming to simulate realistic patterns based on the fundamental characteristics of pulsars. To depict the pulsar in the simulation, a blue object is used, representing a rigid rotating body, following simple rotation kinematic equations. The rotation axis is arbitrary but depicted based on the Earth's frame of reference. A red line represents the magnetic axis, which is constrained with fixed poles on the blue sphere. The intensity or flux of the pulsar signal is approximated by calculating the distance between the Poynting vector coordinate of the magnetic axis and the origin*

the antenna shows a Gaussian profile

$$f(d) = f_0 \cdot \exp\left(-(d/\sigma)^2\right), \tag{7}$$

where d is the distance of the magnetic axis point in the wavelet plane from its origin, d represents the distance of a specific point along the magnetic axis in the radio pulse plane from the origin i.e. d represents the distance of the telescope from the intersection of the magnetic axis with the radio pulse plane. The factor f_0 denotes the flux if the pulsar is pointing directly to the antenna, and σ represents the beam cross section width of the radio pulse (see Fig. 23).

The propagation of radio waves through the interstellar medium leads to variations in their speeds of propagation, causing time delays and frequency-

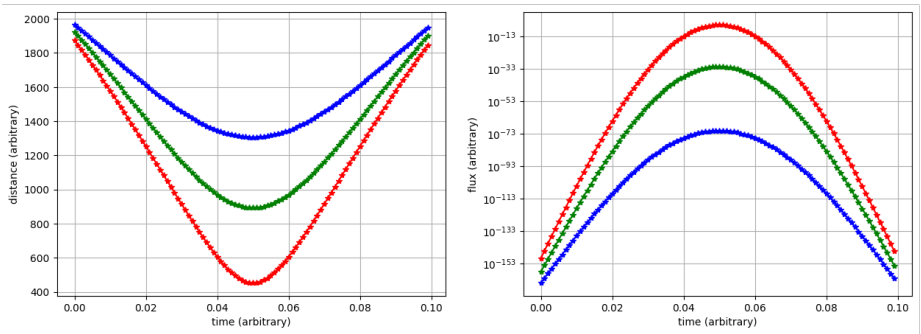

Figure 23: **Pulse Generation** (Left): Distance plot of three different configurations of the poynting vector or the magnetic axis in the wavelet frame away from the origin. (Right): Corresponding flux time series data of the pulsar configurations

based modulation in the broadband radio signals emitted by pulsars. This phenomenon, known as dispersion, occurs because different frequencies within the pulsar signal travel through the interstellar medium at different speeds. Consequently, the signal becomes stretched or spread out in time.

The frequency modulation resulting from dispersion provides a distinct characteristic that can be employed to classify pulsar signals, enabling their differentiation from other types of radio frequency interference or background noise. By analyzing the modulation pattern and examining the timing relationships among different frequency components, astronomers can accurately identify and classify pulsar signals.

To account for dispersion effects from the observer's perspective, the pulsar axis appears to be in a state that occurred δt time earlier when observed at a lower frequency band. This strategy is employed to calculate the associated flux for each frequency band, simulating the dispersion effect in the generated data. The time delay δt of a radio frequency wave denoted by f is calculated using the following formula (see the equivalent formula (1))

$$\delta t = 4.12 \frac{n_e \cdot d}{f^2} - t_H \tag{8}$$

$$t_H = 4.12 \frac{n_e \cdot d}{f_H^2}$$

where n_e is the electron density, d is the distance traveled by the radio signal through interstellar space (in case of our over-simplified model we considered d to be the perpendicular distance from pulsar to the radio pulse frame). f_H, and t_H are the high frequency of the observation band and the time of arrival of that frequency signal, respectively. The important aspects of these effects are simulated using the methods shown in Fig. (24).

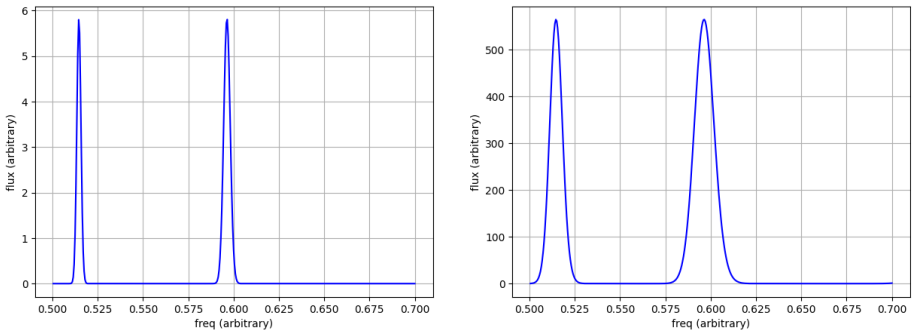

Figure 24: **Dispersion effect** *(Left): Flux magnitude for different frequency channels for an arbitrary beam cross-section. (Right): Flux magnitudes for 3 times the beam cross-section value of the radio pulse.* (**Note**: The broadening of the flux curve for high frequency region is due to the higher velocity of the radio waves through the ISM compared to the low frequency region.)

3.1.2 Generation of Synthetic Data Set

Using the methods described earlier, it is possible to generate a characteristic frequency-time graph that represents the behavior of a simple pulsar. This graph is created by considering the frequency channels associated with the pulsar's signals. By simulating the effects of dispersion and incorporating the appropriate time delays and frequency modulations, the generated graph closely resembles the expected behavior of a pulsar signal.

To make the generated data more realistic, Gaussian white noise is added. While this noise provides a basic approximation of the inherent noise present in observational data, it is important to note that more accurate noise models will be incorporated in future iterations of the project. These advanced noise models will consider various factors, including the characteristics of the antenna electronics and other terrestrial noise sources. By incorporating these factors, the generated data will more accurately reflect the noise characteristics encountered in real-world observations.

Simultaneously with the frequency-time graph, a corresponding binary mask is generated. This mask serves as a segmentation map, where the pulsar signal is differentiated from the background noise. The binary mask essentially traces the presence of the pulsar signal, enabling the distinction between signal and noise. For example, 2000 random such data sets can be generated using the code snippet from the jupyter notebook as shown in Fig. (25).

3.1.3 Training of the UNet for Semantic Segmentation of Pulsar Signals

The generated dataset, which includes the frequency-time graph and its corresponding binary mask, serves as the training data for a UNet model [19]. The

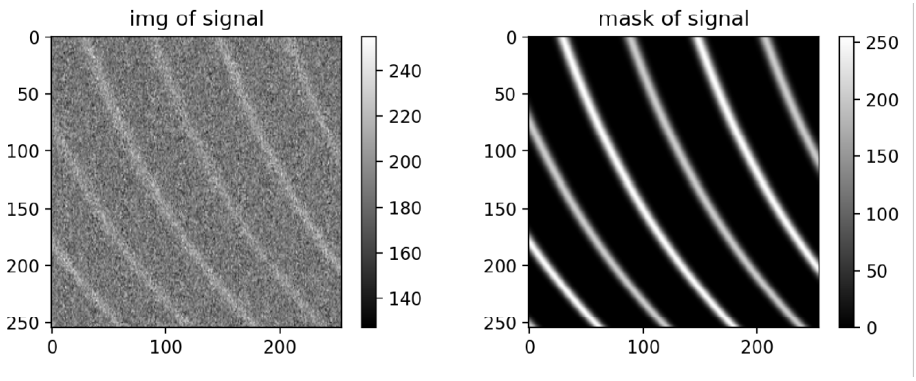

Figure 25: **Example of a Training set** *(Left): A representative image of synthetic freq-time graph/image generated using the module. (Right): Corresponding mask/segmented image of pure pulsar signals without noise.*

goal of this training is to enable the UNet model to perform semantic segmentation, accurately recognizing and delineating the presence of the pulsar signal within the frequency-time graph.

Semantic segmentation involves assigning specific labels to individual pixels or segments within an image. In this case, the UNet model is trained to distinguish between the pulsar signal and the background noise in the frequency-time graph. By training on a batch of these datasets, the UNet model learns the patterns and characteristics of pulsar signals, enabling it to effectively segment pulsar signals in similar, real-world data.

During the training process, the UNet model iteratively learns to map the input frequency-time graph to the corresponding binary mask. It gradually improves its ability to differentiate and classify pulsar signal regions accurately. The model learns the intricate features and structures specific to pulsar signals, allowing it to identify and delineate the pulsar signal boundaries.

Ultimately, once the UNet model is trained and achieves satisfactory performance, it can be used to analyze and segment pulsar signals in new, unseen data. This segmentation capability is crucial for identifying and studying pulsar signals within a large volume of radio frequency data, enabling astronomers and physicists to further understand and explore the nature of pulsars, see Fig. (26).

3.1.4 Legacy ML Tools for Pulsar Classification

After semantic segmentation process of the freq-time graph, it is further processed using a line detection model based on the Hough transform. The Hough transform is a technique used to identify and extract lines in an binary image [20].

By applying the Hough transform to the binary mask, an accumulation

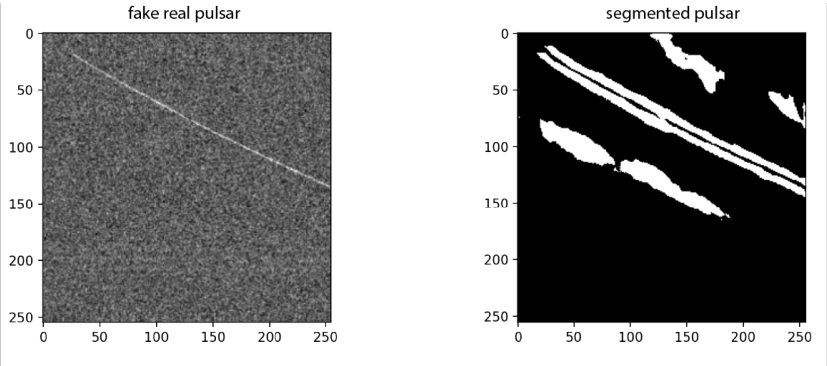

Figure 26: **Segmentation by trained UNet** : *(Left): Fake real pulsar data for testing the code using some data augmentation as described in previous section. (Right): Segmented pulsar data using the trained UNet.*

matrix is generated. This matrix contains the parameters that describe the detected straight lines within the data. These parameters include attributes such as line position, orientation, and length.

The accumulation matrix represents a parameter space where the characteristic shape of the blobs can be analyzed. Different types of signals in the frequency-time graph can be classified based on the distinct patterns exhibited by these blobs. By studying the shape, size, and distribution of the blobs, astronomers can gain insights into the various types of signals present in the data.

This approach allows for a more sophisticated classification of different types of signals within the frequency-time graph, leveraging the inherent characteristics captured by the Hough transform. By identifying and classifying these signals, astronomers can distinguish pulsar signals from other forms of noise or interference, leading to a more accurate and efficient analysis of the data, see Fig. (27).

3.2 Summary

In this article release, our primary objective is to showcase the effectiveness and potential of the method described above in developing a robust platform or pipeline for astronomers. This platform aims to facilitate the real-time detection of pulsar signals from massive data streams, while also integrating the best available physical models of pulsars in a single software package.

By combining advanced signal processing techniques, accurate dispersion modeling, and realistic noise incorporation, the proposed platform offers a powerful solution for detecting pulsar signals in quasi real time. As an example, out of many other characteristic signal properties, it leverages the characteristic frequency-time graphs and corresponding binary masks to accurately identify and segment pulsar signals from the surrounding noise for the current state of

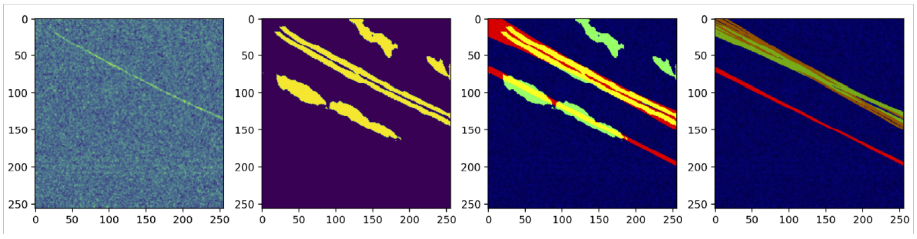

Figure 27: **Classification of pulsar signal after semantic segmentation** *Showing classification of the semantically segmented fake pulsar signal using Kmeans algorithm applied on the parametric space of the Hough transform. First Hough transform was applied on the semantically segmented mask of the fake pulsar signal. A parametric space was defined based on the X and Y coordinates of the midpoint of each Hough lines. Those points were then clustered using Kmeans algorithm. More advanced clustering algorithm will be released in future versions*

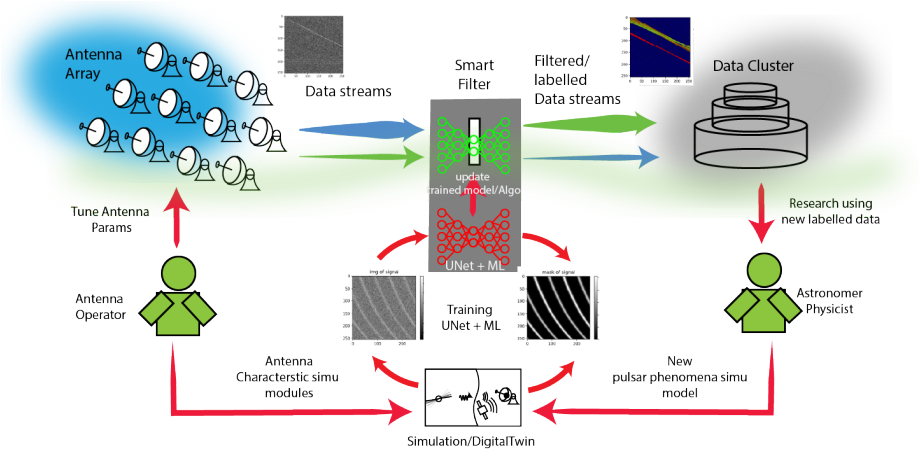

Figure 28: **Summary of the project goal** : *Illustrating the data flow from antenna array to data archiving centers through smart filtering. Efficiently managing the vast data streams generated by radio antennas is crucial, as only a small fraction contains valuable information. To address the challenge of data irreversibility, intelligent filtering methods employing sophisticated algorithms and neural networks play a vital role. Astronomers and physicists analyze labeled signals to gain insights and can integrate new theories into simulation modules. These modules are subsequently employed to train Deep Convolutional Neural Networks (DeepCNNs), such as UNet, for intelligent filtering. Additionally, antenna operators have the opportunity to develop simulation modules that train Deep CNNs to effectively eliminate antenna noise from the signals.*

the release.

Moreover, this platform will establish a positive feedback loop by incorporating the best physical models of pulsars. By integrating our understanding of the physics governing pulsar behavior, the platform gains a deeper insight into the expected signal patterns. This, in turn, enhances the efficiency and accuracy of the pulsar detection process, which in turn provides room for new undiscovered pulsar phenomena.

The synergy between the data-driven approach and the incorporation of physical models enables a more comprehensive and effective pulsar detection pipeline. Astronomers can leverage the platform to efficiently sift through vast amounts of data, swiftly identifying potential pulsar candidates while gaining a simultaneous understanding of the underlying physics driving their behavior.

Ultimately, this article release serves to demonstrate the significant impact of the proposed platform, paving the way for more efficient and insightful exploration of pulsars in the vast sea of observational data.

4 Scaling–oriented Implementation

Astronomical pipelines are not designed for massively parallel processing of huge data streams. Furthermore, the Big Data principle "software to the data" requires the use of virtualization techniques. A key goal of this project is to explore ways to address the associated challenges.

As a specific use case, the project focuses on the analysis of pulsar signals and their simulation. An important question is whether workflows can be distributed across multiple worker nodes and whether speedup can be improved as a result.

Speedup $S(n) = T_1/T_n$ in parallel processing is measured by dividing the time T_1 necessary for completing a given task when using one node by the time T_n needed with n nodes. A performance analysis of the astronomical pipeline CASA showed that its speedup drops when running on more than 8 nodes [21].

The need to process huge amounts of data in near real-time will get more relevant at SKAO. The challenge is to extract "relevant information" out of huge data streasm. Our efforts are focused on pulsar signal identification and noise signal detection.

The simulation outlined in the previous section is developed in the programming language Python. Subsequently, the code is transferred to C++ for performance reasons. This approach is intended to create ready-to-use tools that enable scientists to process large amounts of data in a scalable way. The schematical structure of this tool is shown in Fig. 29. At its core there is a C++ library for all resource intensive computations. To make it more comfortable to use, this library will be compiled and included into a Python environment. This makes it possible to use Jupyter notebooks as a user interface, which are commonly used by scientists working in this field. Python offers many tools for data analysis, but the power of this programming language in parallel computing is limited. This is largely due to the *Global Interpreter Lock*[3], or GIL for short,

[3] https://wiki.python.org/moin/GlobalInterpreterLock

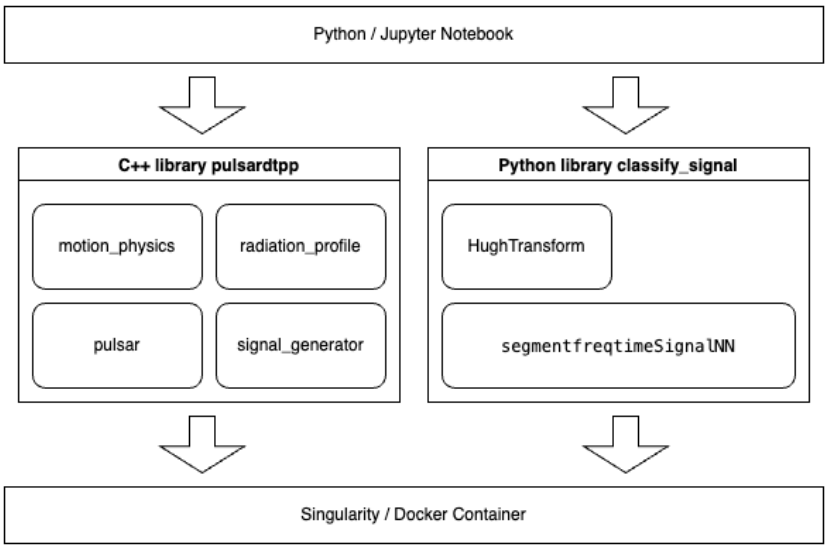

Figure 29: Layered software architecture of the framework ML–PPA. The top layer shows the user interface that can be accessed via Jupyter notebooks or Python. At the center two packages are indicated: methods of the C++ library *PulsarDT++* (left), and methods of the library *classify_signal* of the Python package *PulsarRFI_NN* (right). Libraries and their components can be assembled into a container as needed (bottom layer).

which "locks" multithreading: only one thread can execute at a time. Since parallel processing will be crucial to achieve the desired performance, using a low level language like C++ is necessary for time critical computations. Finally, the entire tool is containerized. The framework used for this is Singularity from Sylabs, which is quite commonly used by data centers in the academic field and has the advantage that no admin rights are needed for execution. All necessary files are copied into the container during the build process, so that the C++ library can be compiled inside of it. To enable all Python functions, the required packages are extracted from the development environment and provided in a file, listing package name and version number. Containerization is great to ensure portability across systems but it needs to be kept in mind that there is always a trade-off between portability and performance. For example, development for a particular system would allow the use of intrinsics, while the program would crash on a system that did not support them. Our tool is in a very early stage and some architeture decisions will be made only in the coming development cycles.

Access to ML-PPA functionalities is provided by a set of module files. These files use the C Python Extension library that allows to define functions and their parameters and return types. From there, the classes and functions that were translated from the Python implementation into C++ can be accessed. This is

the entry point to the actual structure of the simulation, the naming of classes and functions is kept as close as possible to the original implementation. The classes *motion_physics*, *radiation_physics*, *pulsar* and *noise_generator* also exist here with their respective functionality. Some additional algorithms for calculations had to be provided because the convenience of some Python packages like Numpy could not be used. These functions include matrix multiplications, Gaussian normal distribution and discrete convolution. There are a number of ways to integrate C/C++ functions into Python. The build tool chosen for this project is *scikit-build* because it functions as a bridge between the Python packing module *setuptools* and the C++ build tool *CMake*. Having the option to use *CMake* enables this project to grow to a large and complex codebase. The Python standard library *ctypes* can also be used together with *CMake* and would have the advantage of using a standard Python library and thus ensuring future support. The downside of *ctypes*, however, is that the C functions are only called and not completely packaged, which leaves the types of the arguments and return types undefined. Those types would have to be set within a Python script by the user and would leave more work in the hands of the user instead of the developer. This would also become most likely a common point of error. Therefore, the option using *scikit-build* was chosen to have all types defined within the C code and spare the user this step. Should support for the *scikit-build* library drop at some point would a transition to *ctypes* also mean not too much work changing the code base. Future iteration will add an analysis module and functionality to distribute data and instructions across a cluster. This will then take over the role of analyzing and recognizing pulsars with machine learning and image processing algorithms.

4.1 A First Runtime Analysis

The simulation of pulsar signals integrated in ML-PPA already provides an impression of the performance differences between Python and C++. The plot in Fig. 30 compares the execution times of four major simulation functions. Those functions work essentially the same in both implementations with some adjustments specific to the programming language. The C++ implementation of the code is significantly faster by at least a factor 200. That the program performs better in a lower-level language is no surprise, but it shows that common solutions that rely solely on Python are limited in processing data efficiently. The performance of the components of the framework for detecting and analyzing pulsars is a key criterion for determining their quality. In the future, further suitable runtime measurements will be designed and carried out.

5 Noise Analysis

Once we have modeled the signal coming from the source of interest and how it is changed on its way to the Earth through the interstellar medium, we must understand the changes that happen as the signal enters the atmosphere, picked

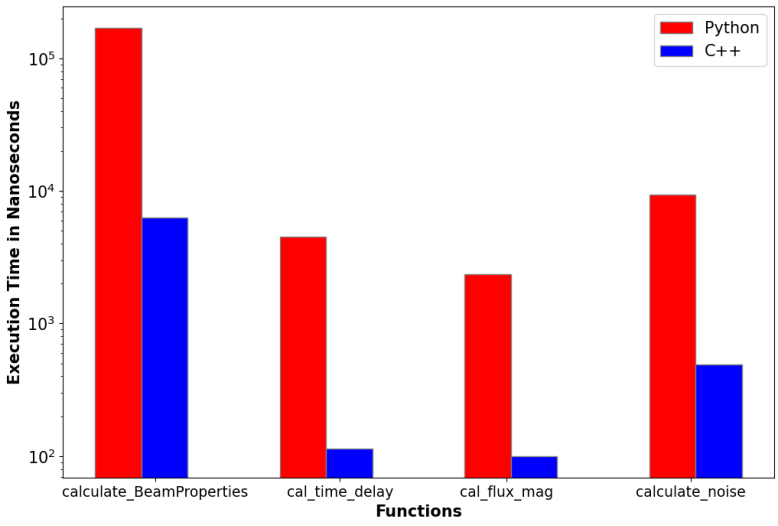

Figure 30: Execution times (logarithmic scale) of four functions ls implemented in Python and C++, respectively, for the simulation of pulsar signals.

up by the radio telescope's antenna, and then amplified, processed and analyzed by the telescope's electronics. This part seems very difficult to formalize and depending on multitude of factors, but, in fact, with certain reasonable assumptions a theory can be constructed that leads to a rather simple model of this whole stage, that can be described with just a few parameters. In the current section we present this theory and apply it to data from the Effelsberg telescope. The main conclusion is that the results are consistent with the white noise hypothesis and a Gaussian model can indeed be used for the noise in simulated data.

5.1 Idealized Superheterodyne Receiver and Its Sensitivity

This subsection describes an idealized radio telescope receiver theory, which has been developed since the mid-20th century and now is usually included in introductory parts of university courses and summer schools for radio astronomers (the account here is based on [22, 5, 23]). Practicing radio astronomers rarely have much use for it, since they deal with more specific problems of complex real telescope configuration. But as we build a digital twin of a radio telescope, it is a convenient starting point, as it offers a holistic approach to the setup, which later can be built upon.

5.1.1 Telescope Signals as Johnson-Nyquist Noise

There is one principle that is in the heart of all the radio astronomical signal and noise analysis: all signals can be treated as thermal noise and characterized with temperature.

It can be explained with the following logic. Let us consider black-body radiation in the Rayleigh–Jeans limit $h\nu \ll k_{\mathrm{B}}T$ (true for most radio astronomy frequencies with the exception of the mm wavelengths). Its sky brightness distribution, or spectral radiance, i. e. the power per solid angle, per area, per frequency (W Sr^{-1} m^{-2} Hz^{-1} is SI units) is expressed as:

$$I_\nu(\theta,\varphi) = \frac{2k_{\mathrm{B}}T(\theta,\varphi)\nu^2}{c^2} = \frac{2k_{\mathrm{B}}T(\theta,\varphi)}{\lambda^2}. \tag{9}$$

Here h is Planck's constant, k_{B} is Boltzmann's constant, c is the speed of light in vacuum. $T(\theta,\varphi)$ is some (equivalent rather than physical) black-body radiation temperature distribution for a sky source.

On the other hand, based on its basic properties, the power per unit of frequency received by a radio antenna with effective area A_e and normalized power response pattern $B_n(\theta,\varphi)$ (also known as the beam) is:

$$P_{\mathrm{rec}} = \frac{A_e}{2}\int_\Omega I_\nu(\theta,\varphi)B_n(\theta,\varphi)d\Omega. \tag{10}$$

We divide by 2 to account for the unpolarized wave coupled to the single-polarization transmission line and receiver. The integration is performed over the relevant solid angle range Ω (ideally over the whole 4π, but given that $B_n(\theta,\varphi)$ normally drops fast with θ, it can be approximated by a cone close to the axis). Substituting Eq. (9) into (10) gives:

$$P_\nu = \frac{k_{\mathrm{B}}A_e}{\lambda^2}\int_\Omega T(\theta,\varphi)B_n(\theta,\varphi)d\Omega. \tag{11}$$

Now, if we integrate the normalized power response pattern, we get the effective antenna (or beam) solid angle:

$$\Omega_{\mathrm{ant}} = \int_\Omega B_n(\theta,\varphi)d\Omega. \tag{12}$$

.

There is an intuitive fact (so called "weak reciprocity theorem"), which can also be directly proven based on thermodynamic considerations or time-reversibility of Maxwell's equations (see [5] for more detailed explanations and references), that the power response pattern $B_n(\theta,\varphi)$ of an antenna in an isotropic medium is the same for transmitting and receiving. Based on this we can conduct a thought experiment of two different lossless antennas, located far enough from each other for the far-field approximation to work, transmitting and receiving at the same wavelength λ. The power transmitted must be the same for the first one transmitting and the second receiving, and vice versa. A simple geometric consequence of this is $A_e\Omega_{\mathrm{ant}} = const$ for *any* lossless antenna.

Choosing a convenient antenna (say, a large uniformly illuminated square) this constant can be directly calculated [24]:

$$A_e \Omega_{\text{ant}} = \lambda^2. \tag{13}$$

This is known as the antenna theorem. It can actually be generalized [24] even for an antenna with ohmic loss, or other similar loss that does not alter the beam pattern: if the antenna absorbs $1 - \epsilon_r$ of the incident power, then the constant becomes $\epsilon_r \lambda^2$. However, most radio telescope antennas have negligible ohmic losses in the relevant frequency range. The physical meaning of this theorem is straightforward: large effective area antenna must have a narrow beam, and a wide beam antenna must have a small effective area, and the exact relation is determined by the wavelength, which is just another twist on the diffraction limitation.

But now in the light of (13) Eq. (11) becomes just

$$P_\nu = \frac{k_{\text{B}}}{\Omega_{\text{ant}}} \int_\Omega T(\theta, \varphi) B_n(\theta, \varphi) d\Omega. \tag{14}$$

We have succeeded in eliminating from this expression all the parameters except for the beam and source angular structure. Thus from the physical point of view it turns into:

$$P_\nu = k_{\text{B}} T_A, \tag{15}$$

where

$$T_A = \frac{1}{\Omega_{\text{ant}}} \int_\Omega T(\theta, \varphi) B_n(\theta, \varphi) d\Omega. \tag{16}$$

It is a beam-averaged equivalent black-body temperature of the observed source, usually referred to as the antenna temperature. E. g. if we have an extended source with $T(\theta, \varphi) = T_{\text{Source}} = const$ within the telescope beam, Ω_{ant} is canceled out and $T_A = T_{\text{Source}}$.

This result is much deeper than it may seem. Making a few reasonable assumptions, an antenna observing a sky source, a rather complex system with properties that are not immediately obvious, has been shown to have the same power output as the one available at terminals of a simple resistor at temperature T_A. This is not a mathematical trick, the equivalence can be confirmed by another thought experiment, coupling an antenna in one isolated cavity with a matched resistor in another, connecting them to each other with a transmission line, that lets only a narrow range of frequencies from ν to $\nu + d\nu$ to go through. Since in thermodynamic equilibrium no power can pass between the cavities, the power per unit frequency received by the antenna must be equal to the power per unit frequency detected at the resistor.

The signal at the resistor is known as the Johnson–Nyquist thermal noise. It has a nearly uniform power spectral density across most frequency bands, which means it is almost identical to the simplest noise type known as the "white" noise (defined as a noise with flat power spectrum). When limited to a finite bandwidth at frequencies relevant for radio astronomy the noise voltage has a nearly Gaussian amplitude distribution with zero mean.

5.1.2 Receiver Structure

A typical idealized radio astronomy receiver is shown in Fig. 31 and can be described as follows.

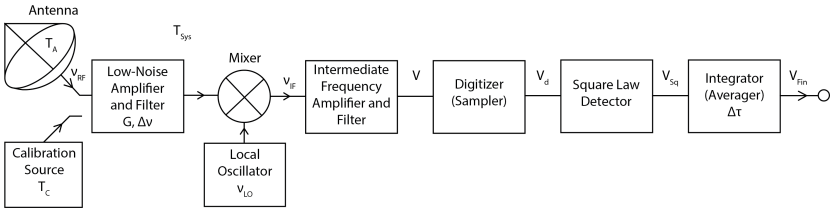

Figure 31: A block diagram of an idealized superheterodyne receiver or total-power radiometer, which is a basic representation of most single-antenna radio telescopes. For details, see text.

Its first element is the Low-Noise Amplifier (LNA) which receives input from the antenna, the incoming signal at frequency $\nu_{\rm RF}$ can be characterized by the antenna temperature $T_{\rm A}$. For calibration purposes another input source can be switched on, with known standard temperature $T_{\rm C}$. The LNA is characterized by gain G and has a filter that limits the input signal bandwidth to $\Delta\nu$. The gain must be high and as close to constant as possible (can be assumed exactly constant for model purposes), the LNA typically dominates the noise of the whole system as so is cooled down to very low temperatures.

Next the signal goes to the mixer where the amplified signal is multiplied by a signal of the local oscillator (LO) at frequency $\nu_{\rm LO}$. This is what is historically called a "superheterodyne" setup, it just follows a simple trigonometric relation:

$$2\sin(2\pi\nu_{\rm RF}t)\sin(2\pi\nu_{\rm LO}t) = \cos[2\pi(\nu_{\rm RF}-\nu_{\rm LO})t] - \cos[2\pi(\nu_{\rm RF}+\nu_{\rm LO})t]. \quad (17)$$

That is by mixing the LO signal with the original one at the "sky frequency" $\nu_{\rm RF}$ we get a sum of two signals at frequencies that are the sum and the difference of the frequencies of the mixed signals. When $\nu_{\rm RF}$ and $\nu_{\rm LO}$ are close to each other, their sum and difference fall into two very different frequency bands, and one of them can be easily filtered out (this is known as a "single-sideband" superheterodyne), so the operation is equivalent to a frequency shift of the signal. E. g. if we take the simplest case of $\nu_{\rm LO} = \nu_{\rm RF}$ and filter out the frequencies of order $2\nu_{\rm RF}$, the original incoming band from $\nu_{\rm RF}$ to $\nu_{\rm RF}+\Delta\nu$ is shifted to the band from 0 to $\Delta\nu$. Of course in real life shifting to such extremely low frequencies is impractical, so instead the band is transformed from $[\nu_{\rm RF}, \nu_{\rm RF}+\Delta\nu]$ to $[\nu_{\rm IF}, \nu_{\rm IF}+\Delta\nu]$, with $\nu_{\rm IF} = \nu_{\rm RF} - \nu_{\rm LO}$, where IF stands for the "intermediate frequency". $\nu_{\rm IF}$ is normally chosen to be (much) lower than $\nu_{\rm RF}$.

There is a number of benefits of the superheterodyne setup, the main of them are:

- Lower frequencies are easier to work with, i. e. amplify, filter, transmit

and digitize, and typical ν_{RF} of radio astronomy ranging from a few to tens of GHz, or even higher, lie above this radio technical "comfort zone".

- Radio astronomical antennas are normally designed to work over many different frequency bands, and designing the whole separate chains of electronics to deal with each band would have been severely impractical.
- Superheterodyne allows to tune just one element of the system, LO, and a wide range of ν_{RF} is translated to the same narrow ν_{IF} band.
- As a result, all the components down from the mixer can be designed to handle only ν_{IF} band.

After the mixer the signal enters the Intermediate-Frequency Amplifier (IFA). It is also characterized by a its gain and in real life its filter is narrower than the filter of the LNA. However in an ideal model one can disregard its gain in comparison with that of the LNA and assume that $\Delta\nu$ stays the same.

Next the signal is converted to a digital data stream in the Digitizer or Sampler. We assume that it samples the signal at the Nyquist rate:

$$\Delta\tau_{\mathrm{Nyquist}} = \frac{1}{2\Delta\nu}. \tag{18}$$

The Nyquist sampling theorem states that a band-limited signal is fully represented by such sampling and can be reconstructed from the resulting digital sequence. An ideal digitizer has no quantization errors or other distortions or biases and so the digital signal has exactly the same statistical properties as the analog one.

Next a square-law detector multiplies the input voltage by itself, so that its output voltage is proportional to the input power. Then the resulting signal is integrated over a relatively long time $\Delta\tau$ and that is what we assume to be the final output of the receiver. These last two stages are introduced for illustration purposes. Although this type of setup can be used for simple continuum observation, mostly a more sophisticated backends are used depending on the type of observation (like spectrometers or spectral processors for spectral line observations etc.)

5.1.3 System Temperature

Traditionally we designate as T_A only the temperature detected from the intended target of observation. This signal is mixed with other signals coming from other sky sources, atmosphere and the ground. Since for their signals the same consideration of equivalence to thermal noise is true, they can be characterized by temperatures of their own. When the signal is processed with the electronics of the receiver, other random noise signals are introduced, but they are also close to thermal. So, in general, we have a number of random thermal signals that are added to each other. Since they are close to Gaussian random variables, the resulting pattern is also a Gaussian random variable, and the

thermal noise pattern can be characterized by a sum of the temperatures of the contributing components.

The final signal that comes out of the receiver is characterized by a total temperature of $T_A + T_{\rm Sys}$, where $T_{\rm Sys}$ is the sum of all the contributions from sources other than the target source of the observations and is known as the system temperature. If we imagine that the target source blinks out (as happens with periodic and other transient sources), the telescope's signal temperature drops to $T_{\rm Sys}$. In majority of cases $T_A \ll T_{\rm Sys}$[4].

In the most general form $T_{\rm Sys}$ can be expressed as follows:

$$T_{\rm Sys} = T_{\rm rad} + T_{\rm cmb} + T_{\rm bg} + T_{\rm atm} + T_{\rm spill} + T_{\Omega}. \tag{19}$$

The different components of the system temperature $T_{\rm Sys}$ include:

- $T_{\rm rad}$ is the radiometer temperature, that is the contribution of all the electronics of the receiver. If the gain of the LNA is high enough, which is usually the case, it determines the contribution of the whole system. For this reason LNA is typically cooled down to ≈ 15 K. It is not a purely thermal system, so this can translate to $T_{\rm rad}$ from several to a few tens of K.

- $T_{\rm cmb}$ is the temperature of cosmic microwave background radiation equal to 2.73 K. It is almost ideally isotropic and for a wide range of frequencies above 1 GHz is the main contribution of the sky into $T_{\rm Sys}$.

- $T_{\rm bg}$ is the contribution of all the other cosmic "background" sources (i. e. those within the telescope's primary beam together with the target), both Galactic and extragalactic. This component strongly depends on where the antenna is pointed, since the distribution of bright radio sources is far from isotropic. For frequencies $\gtrsim 1$ GHz it is mostly negligible, but below 1 GHz the diffuse Galactic synchrotron emission along the disk and various "loops" starts to dominate.

- $T_{\rm atm}$ is the temperature due to the Earth atmospheric emission. For higher frequencies this term should be written as $(1 - e^{-\tau_z / \cos z})T_{\rm atm}$ to take into account current optical depth, where τ_z is the zenith opacity for the current frequency band and z is the zenith angle of the antenna at the given moment. Above 10 GHz not only increasing thermal emission of the atmosphere ($\propto \nu^2$) plays a significant role, but also the water vapor lines start to contribute. By the mm wavelengths water vapor becomes the main contributor to $T_{\rm Sys}$.

- $T_{\rm spill}$ is the "spillower" contribution. Since the telescope is surrounded by ground at $\approx$ 300 K, even small imperfectness of its optical system

[4]It must be noted though, that our case of a single-dish telescope observing the Crab pulsar is an exception of this rule: because the pulsar is embedded in a very radio-bright nebula (the Crab Nebula) and the telescope beam is wide enough to encompass the pulsar environment, T_A is significantly larger than $T_{\rm Sys}$ even when pulsar is off. But this does not change much in the derivations of this section and their applicability to our data.

(sidelobes, reflections from support elements) can add up to a few K to the system temperature. This component strongly depends on the design of the antenna in question.

- T_Ω is due to ohmic losses in antenna and waveguides. Mostly it is no more than a few tenths of K, but depending on the configuration of the particular telescope at the given frequency it may be noticeable.

Since different terms of this equation are widely different in magnitude for different frequency bands, it must be rewritten for the frequency band in question including only the dominant terms. For example for a range of frequencies from ∼1 to ∼10 GHz in most cases the above equation is reduced to a much simpler relation:

$$T_{\rm Sys} = T_{\rm rad} + T_{\rm cmb}. \tag{20}$$

5.1.4 Signal Transformation and Resulting Variance

Let us designate the signal's voltage V after it leaves the IFA. Based on its thermal nature, we assume V to be a zero-mean Gaussian random process with a "white", i. e. flat spectrum. Such process is defined by a single parameter, its rms σ (or variance σ^2 if preferred). Its odd moments are zero and even moments are given by

$$\langle V^n \rangle = (n-1)!!\, \sigma^n. \tag{21}$$

Thus $\langle V \rangle = 0$ and the power, taking into account Nyquist power spectral density, gain G, and (ideally square) bandwidth $\Delta\nu$:

$$\langle V^2 \rangle = \sigma^2 = k_{\rm B}(T_A + T_{\rm Sys})G\Delta\nu. \tag{22}$$

Assuming ideal sampling, the statistical properties of the digitized signal are exactly the same, so $\langle V_d \rangle = \langle V \rangle$, $\langle V_d^2 \rangle = \langle V^2 \rangle$ and so on, so we can relate the next step directly to V. And since the square law detector multiplies the signal by itself, we get for $V_{\rm Sq}$

$$\begin{aligned} \langle V_{\rm Sq} \rangle &= \langle V^2 \rangle \\ \langle V_{\rm Sq}^2 \rangle &= \langle V^4 \rangle = 3\sigma^4 = 3\langle V^2 \rangle^2 \\ \sigma_{\rm Sq}^2 &= \langle V_{\rm Sq}^2 \rangle - \langle V_{\rm Sq} \rangle^2 = 2\langle V^2 \rangle^2. \end{aligned} \tag{23}$$

Finally, the integrator averages $N = \Delta\tau/\Delta\tau_{\rm Nyquist} = 2\Delta\nu\Delta\tau$ samples and the properties of the resulting signal are:

$$\begin{aligned} \langle V_{\rm Fin} \rangle &= \langle V^2 \rangle = k_{\rm B}(T_A + T_{\rm Sys})G\Delta\nu \\ \sigma_{\rm Fin}^2 &= \frac{\sigma_{\rm Sq}^2}{N} = \frac{2[k_{\rm B}(T_A + T_{\rm Sys})G\Delta\nu]^2}{2\Delta\nu\Delta\tau}. \end{aligned} \tag{24}$$

The final signal is proportional to temperature with the constant coefficient $k_{\rm B}G\Delta\nu$. By switching to the calibration temperature source T_C this factor can

be eliminated and the final signal recalibrated to temperature units:

$$\begin{aligned} \langle V_{\mathrm{Fin}_T} \rangle &= T_A + T_{\mathrm{Sys}} \\ \sigma_{\mathrm{Fin}_T} &= \frac{T_A + T_{\mathrm{Sys}}}{\sqrt{\Delta\nu\Delta\tau}}. \end{aligned} \tag{25}$$

The useful signal part of $\langle V_{\mathrm{Fin}_T} \rangle$ is only T_A and normally $T_A \ll T_{\mathrm{Sys}}$, thus the signal-to-noise ratio is

$$\mathrm{SNR} \approx \frac{T_A}{T_{\mathrm{Sys}}} \sqrt{\Delta\nu\Delta\tau}. \tag{26}$$

Similarly, σ_{Fin_T} of Eq. (25) can also be rewritten as

$$\sigma_{\mathrm{Fin}_T} \approx \frac{T_{\mathrm{Sys}}}{\sqrt{\Delta\nu\Delta\tau}}. \tag{27}$$

It is called the ideal radiometer equation and determines the noise level of a total-power radiometer based on its three main characteristics: system temperature, bandwidth, and integration time.

As we have mentioned before, in real telescope configuration, in particular when observing pulsars, other backends are used, not just a square-law detector and integrator. But due to the mathematical properties of Gaussian processes the main result of this derivation holds: we are dealing with signals that can be closely characterized by a Gaussian white-noise pattern, described by a single "temperature" parameter, and this temperature can be split into two parts, the (useful source) antenna temperature T_A and system temperature T_{Sys}, summing up all the other contributions. The immediate and very useful consequence of this is that can use a Gaussian noise model for our final product, and it is not an oversimplification, but a well-researched direct result of the radio telescope physics.

5.2 Noise Analysis of the Real Signal

Before we can start adapting this theory as the basis of our work, we must check how well the conclusion about the white noise holds, which in turn will show if all the assumptions we made when developing the model are valid for our purposes.

5.2.1 Colors of Noise and the Allan Deviation

"White" noise is a noise with a flat frequency spectrum, i. e. equal power in every band. If the spectrum is not flat, it can be described using a number of basic "noise colors". Their power spectral density per unit of bandwidth are assumed proportional to $1/f^\beta$. So for the white noise $\beta = 0$, then noise with $\beta = 1$ is called "pink", with $\beta = 2$ "brown" or "red", with $\beta = -1$ "blue", with $\beta = -2$ "violet" (the color names are chosen metaphorically, e. g. white noise is named after white light, where all frequencies are equally represented). Violet noise can be created by temporal differentiation of a white noise, and

brown — by temporal integration of the same. More complex noise patterns can be described by dominant spectral features at different bands, e. g. pink at low frequencies and white at high.

But simply doing a Fourier transform on a noise pattern is not practical, since the result is increasingly unstable at the high frequency end and so the slope may be difficult to assess. Instead certain statistical tests are performed. One of them is the Allan variance and related Allan deviation.

The Allan variance [25], sometimes also called "two-sample variance", is a statistical measurement that characterizes correlation between values of a time series at different time scales. It is often designated as $\sigma^2(\tau)$, where τ is the variable observation time. The Allan deviation, also known as "sigma-tau", is the square root of the Allan variance:

$$\sigma(\tau) = \sqrt{\sigma^2(\tau)}, \tag{28}$$

just like with the common variance and deviation.

It is named after David W. Allan and was originally introduced to characterize the stability of atomic clocks and crystal oscillators [26]. It had been discovered that the phase noise of such systems was not a pure white noise. This presented a problem for the common statistical instruments like the traditional variance and deviation, and so new tools had to be introduced.

The real power of the Allan variance is not in calculating it for a single value of observation time τ, but going through all possible observation times of the given time series. In this case we start with the sampling rate τ and then take the observation time in the calculation equal to 2τ, 3τ etc. The shape of the plot of $\sigma(\tau)$ is directly related to the shape of the frequency spectrum. If the noise pattern is represented by one of the basic colors (i. e. with spectrum $\propto 1/f^\beta$), then $\sigma(\tau) \propto 1/\tau^\alpha$. For the "frequency-type" data pink noise has $\alpha = 0$, white $\alpha = 1/2$, blue $\alpha = 1$, violet $\alpha = 3/2$, and brown $\alpha = -1/2$:

$$\alpha = \frac{1}{2}(1 - \beta).$$

Computation of the Allan deviation has been efficiently implemented in some dedicated statistical libraries. E. g. for Python there is the AllanTools library [27].

5.2.2 Noise in Our Raw Data

We have analyzed the raw noise pattern of the Effelsberg data series[5] (in terms of our ideal diagram on Fig 31 it is V_d just after sampling[6]). An example of this data series, a fragment of 1000 points, is shown in Fig. 32.

[5] The analyzed data set covers about 20 minutes of the Effelsberg 100 m radio telescope observation of the Crab pulsar in the sky frequency interval of 1.21 – 1.53 GHz with the standard L-band pulsar setup. An in-depth discussion of the technical details of this type of observation can be found e. g. in [28]

[6] We cannot look at the signal higher up the processing stream, because this is when it is converted to a digital data sequence. However the noise properties should stay the same as for the analog signal, as it has been explained in subsection 5.1.4.

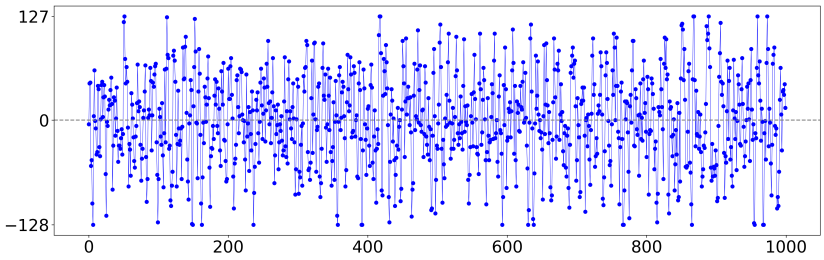

Figure 32: A small sample (1000 points) of voltage raw data (left polarization only), taken right after the digital sampling (V_d in Fig. 31). The value is sampled as an 8-bit signed integer, so natural limits are -128 and 127. The sampling interval is 1.5625 ns. It appears to be zero-mean random pattern, but requires a statistical study to understand its detailed properties.

Because we are covering an enormous range of time intervals from ≈ 1.56 ns to 20 min, or about twelve orders of magnitude, we have to split the computation into three parts, for the shortest, medium, and longest time intervals. The results are shown in Figs. 33, 34, and 35. With the exception of the shortest range 1.56ns $\lesssim \tau \lesssim 2\mu$s, where the deviations to pink and blue noise patterns can be explained with electronic and sampling imperfections inevitable at such short times, all the data is clearly consistent with the white noise hypothesis.

5.3 Signal Transformation and Noise in Observations with Antenna Arrays

In this section we have been discussing the signal transformation and noise as they occur in a single-dish radio telescope, a telescope equipped with a single antenna. But many modern radio telescopes, in particular new generation tools like the MeerKAT, use antenna arrays, that is configurations of multiple antennas.

A detailed analysis of such systems is beyond the scope of this paper, but their general behavior should not be significantly different from what is outlined in this section.

The reason for this is that pulsars are usually observed in a phased array mode, i. e. the separate antennas are connected in such way that their signal is added "in phase" (phase difference resulting from different length of the signal conduits is compensated by special phase shifter devices) and then processed as if it is coming from a single dish (mixed with the LO signal etc.). When treated in this way an array is equivalent to a single very big antenna, with each array element playing a role of an antenna piece (the rest is just "missing"), so that the total area of this antenna is the sum of the areas of the individual elements, and its size is represented by the largest distance between two of the array elements.

As each array element collects radio waves as a single dish, its signal must

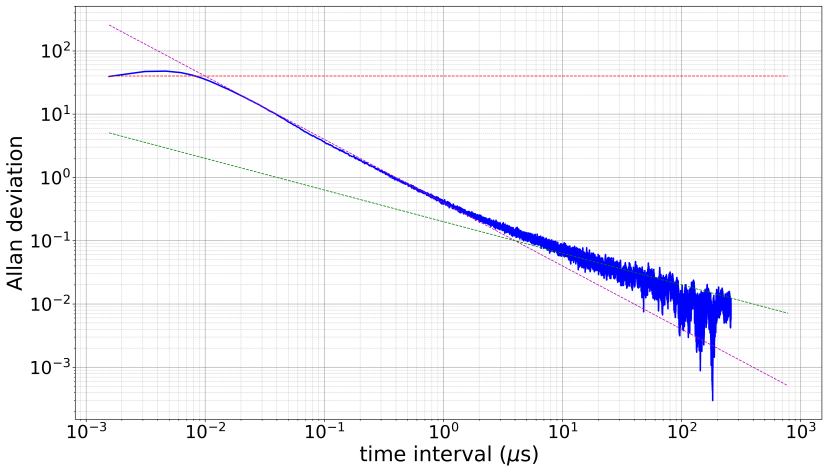

Figure 33: A log-log plot of the Allan deviation $\sigma(\tau)$ for the shortest time intervals. It was calculated with *allantools.adev* [27] in the "frequency-type" data mode for a sample of 500,000 sequential points of V_d in the left polarization of our Effelsberg data sample. The range of the time intervals τ is from the shortest sampled ≈ 1.56 ns to $\approx$ 0.78 ms. At longer time intervals there are fewer pairs available, so a larger statistical error is introduced. In this figure we can distinctly see three different slopes corresponding to three noise colors at different ranges of time intervals. For $\tau \lesssim 10$ ns $\sigma(\tau)$ is close to flat (the red dashed line), corresponding to pink noise, then the slope goes down to $\alpha = 1$, magenta dashed line, i. e. a blue noise pattern for up to $\approx$ 2 μs, and for the longer intervals settles on $\alpha = 1/2$ of the white noise (green dashed line). This means that for $\tau \lesssim 2$ μs the noise deviates from the ideal Johnson-Nyquist thermal one first in the lower-frequency-dominant and then in the higher-frequency-dominant way, due to imperfections of electronics and digital sampling. It is a normal behavior, these time intervals are typically too short to convey meaningful astronomical information and are integrated over at the later processing stages. But starting with ≈ 2 μs our thermal white noise hypothesis is confirmed.

have the same properties, i. e. a thermal signal characterized by antenna and system temperatures with an underlying Gaussian statistics. As it has been explained above, such "white noise" signals produce similar signals when added. So the resulting signal will still be a "white noise" pattern, characterized by combined system and antenna temperatures.

5.4 Unintended Satellite Radiation

There is another new and very dangerous external contribution to the source signal that should be discussed here, although it is only tangentially related to the question of noise.

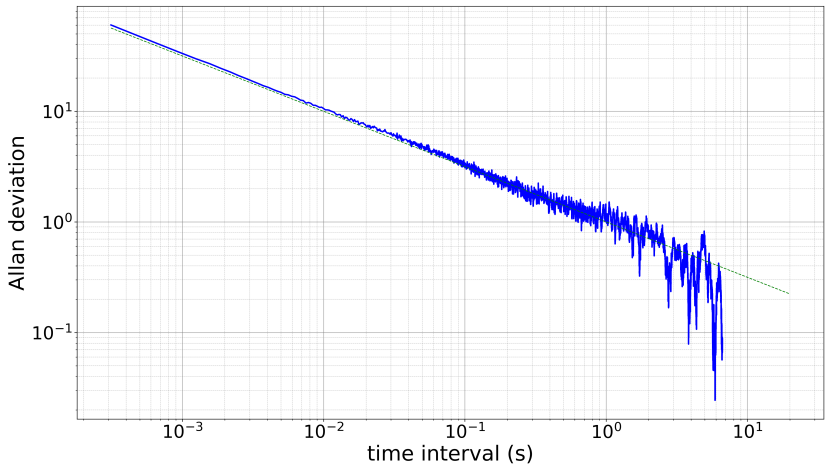

Figure 34: A log-log plot of the Allan deviation $\sigma(\tau)$ for the medium time intervals. It was calculated with *allantools.adev* [27] in the "frequency-type" data mode for a sample of 64,000 sequential points (skipping every 200,000 points) of V_d in the left polarization of our Effelsberg data sample. The range of the time intervals τ is from $\approx$ 0.3 ms to 20 s. At longer time intervals there are fewer pairs available, so a larger statistical error is introduced. We can only see the slope corresponding to $\alpha = 1/2$ of the white noise (green dashed line). Thus our thermal white noise hypothesis is fully confirmed for this range.

With the arrival of Starlink the number of man-made satellites has drastically increased, and so has the probability that one of them may pass within a radio telescope's beam during observations. Although main satellite communication frequencies are by design far from the radio astronomy bands, it has been recently found [29] that Starlink satellites can be detected at $\sim$ 100 MHz frequencies far below their 10.7 to 12.7 GHz downlink band. Such unintentional leakage radiation, negligible for the companies that produce and operate satellites, can be quite strong by radio astronomical standards (flux densities reported in [29] are 0.1-10 Jy for broad-band signals and 10-500 Jy for narrow-band ones, which is enormous).

At this point it is unclear how satellite leakage will influence different types of observations in different bands, but one can reasonably expect brief random spikes of both BBRFI and NBRFI as various satellites cross the telescope beam or even a sidelobe. Low levels of such signals, especially as satellite constellations expand, may also influence the noise component, adding to the sky background.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 460248186 (PUNCH4NFDI [1]) and by

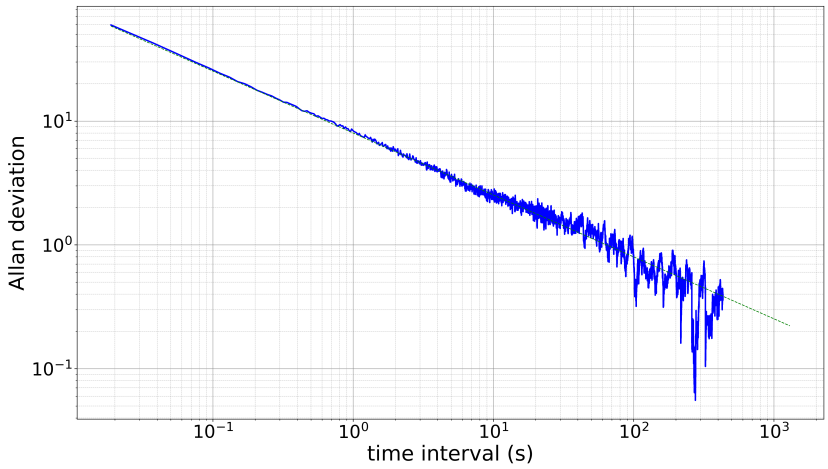

Figure 35: A log-log plot of the Allan deviation $\sigma(\tau)$ for the longest time intervals. It was calculated with *allantools.adev* [27] in the "frequency-type" data mode for a sample of $\approx$ 70,000 sequential points (skipping every 12,000,000 points and covering the whole sample) of V_d in the left polarization of our Effelsberg data sample. The range of the time intervals τ is from $\approx$ 19 ms to 20 min. At longer time intervals there are fewer pairs available, so a larger statistical error is introduced. We can only see the slope corresponding to $\alpha = 1/2$ of the white noise (green dashed line). For the longest time intervals the points must be uncorrelated and naturally form a Gaussian pattern. But the shorter time interval side of this plot is still in the scientifically relevant zone, and the slope is the same over the whole plot, thus also confirming the white noise hypothesis.

the European Union Horizon Europe Programme – Grant Agreement number 101058386 (interTwin [2]).

References

[1] PUNCH4NFDI project, 2025. https://www.punch4nfdi.de/.

[2] interTwin project, 2025. https://www.intertwin.eu/.

[3] H. Heßling, M. Kramer, and S. Wagner. Data Challenges at the Square Kilometre Array (SKA). *EGI Conference 2020*, 2020. https://indico.egi.eu/event/5000/contributions/14366/.

[4] M. Trattner and T. Oelkers. Memory-based computing in astronomy. *Radio2022 Symposium: www.glowconsortium.de/images/Documents/GlowAssembly2022/trattner_oelkers.pdf*, 2022.

[5] J. J. Condon and S. M. Ransom. *Essential Radio Astronomy*. Princeton University Press, 2016.

[6] J. M. Yao, R. N. Manchester, and N. Wang. A new electron-density model for estimation of pulsar and frb distances. *The Astrophysical Journal*, 835(1):29, January 2017.

[7] Ziping Guo, Zhigang Wen, Jianping Yuan, Feifei Kou, Qingdong Wu, Na Wang, Weiwei Zhu, Di Li, Mengyao XUE, Pei Wang, Chenchen Miao, De Zhao, Yue Hu, W. M. Yan, Jiarui Niu, Rukiye Rejep, and Zhipeng Huang. Single-pulse emission variation of two pulsars discovered by FAST. *Research in Astronomy and Astrophysics*, April 2023.

[8] P. R. Brook, A. Karastergiou, M. A. McLaughlin, M. T. Lam, Z. Arzoumanian, S. Chatterjee, J. M. Cordes, K. Crowter, M. DeCesar, P. B. Demorest, T. Dolch, J. A. Ellis, R. D. Ferdman, E. Ferrara, E. Fonseca, P. A. Gentile, G. Jones, M. L. Jones, T. J. W. Lazio, L. Levin, D. R. Lorimer, R. S. Lynch, C. Ng, D. J. Nice, T. T. Pennucci, S. M. Ransom, P. S. Ray, R. Spiewak, I. H. Stairs, D. R. Stinebring, K. Stovall, J. K. Swiggum, and W. W. Zhu. The NANOGrav 11-year data set: Pulse profile variability. *The Astrophysical Journal*, 868(2):122, November 2018.

[9] M. Kramer, R. Wielebinski, A. Jessner, J. A. Gil, and J. H. Seiradakis. Geometrical analysis of average pulsar profiles using multi-component gaussian fits at several frequencies. i. method and analysis. *Astronomy and Astrophysics Suppl.*, 107(1):11, November 1994.

[10] B. W. Stappers, J. W. T. Hessels, A. Alexov, K. Anderson, T. Coenen, T. Hassall, A. Karastergiou, V. I. Kondratiev, M. Kramer, J. van Leeuwen, J. D. Mol, A. Noutsos, J. W. Romein, P. Weltevrede, R. Fender, R. A. M. J. Wijers, L. Bähren, M. E. Bell, J. Broderick, E. J. Daw, V. S. Dhillon, J. Eislöffel, H. Falcke, J. Griessmeier, C. Law, S. Markoff, J. C. A. Miller-Jones, B. Scheers, H. Spreeuw, J. Swinbank, S. ter Veen, M. W. Wise, O. Wucknitz, P. Zarka, J. Anderson, A. Asgekar, I. M. Avruch, R. Beck, P. Bennema, M. J. Bentum, P. Best, J. Bregman, M. Brentjens, R. H. van de Brink, P. C. Broekema, W. N. Brouw, M. Brüggen, A. G. de Bruyn, H. R. Butcher, B. Ciardi, J. Conway, R.-J. Dettmar, A. van Duin, J. van Enst, M. Garrett, M. Gerbers, T. Grit, A. Gunst, M. P. van Haarlem, J. P. Hamaker, G. Heald, M. Hoeft, H. Holties, A. Horneffer, L. V. E. Koopmans, G. Kuper, M. Loose, P. Maat, D. McKay-Bukowski, J. P. McKean, G. Miley, R. Morganti, R. Nijboer, J. E. Noordam, M. Norden, H. Olofsson, M. Pandey-Pommier, A. Polatidis, W. Reich, H. Röttgering, A. Schoenmakers, J. Sluman, O. Smirnov, M. Steinmetz, C. G. M. Sterks, M. Tagger, Y. Tang, R. Vermeulen, N. Vermaas, C. Vogt, M. de Vos, S. J. Wijnholds, S. Yatawatta, and A. Zensus. Observing pulsars and fast transients with LOFAR. *Astronomy and Astrophysics*, 530:A80, May 2011.

[11] Siyao Xu and Bing Zhang. SCATTER BROADENING OF PULSARS AND IMPLICATIONS ON THE INTERSTELLAR MEDIUM TURBULENCE. *The Astrophysical Journal*, 835(1):2, January 2017.

[12] F. Kirsten, N. D. R. Bhat, B. W. Meyers, J.-P. Macquart, S. E. Tremblay, and S. M. Ord. Probing pulsar scattering between 120 and 280 MHz with the MWA. *The Astrophysical Journal*, 874(2):179, April 2019.

[13] T. T. Pennucci, A. Possenti, P. Esposito, N. Rea, D. Haggard, F. K. Baganoff, M. Burgay, F. Coti Zelati, G. L. Israel, and A. Minter. SIMULTANEOUS MULTI-BAND RADIO AND x-RAY OBSERVATIONS OF THE GALACTIC CENTER MAGNETAR SGR 1745–2900. *The Astrophysical Journal*, 808(1):81, July 2015.

[14] F. Jankowski, W. van Straten, E. F. Keane, M. Bailes, E. D. Barr, S. Johnston, and M. Kerr. Spectral properties of 441 radio pulsars. *Monthly Notices of the Royal Astronomical Society*, 473(4):4436–4458, October 2017.

[15] R. Karuppusamy, B. W. Stappers, and W. van Straten. Giant pulses from the crab pulsar. *Astronomy and Astrophysics*, 515:A36, June 2010.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[17] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, June 2008.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[20] eastWillow. Hough line transform, 2016. https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html.

[21] Hermann Heßling, Marco Strutz, Elsa Irmgard Buchholz, and Peter Hufnagl. On divide&conquer in image processing of data monster. *Big Data Research*, 25:100214, 2021.

[22] A. R. Thompson, J. M. Moran, and G. W. Swenson Jr. *Interferometry and Synthesis in Radio Astronomy*, chapter 1 and appendix 1.1. SpringerOpen, 2017.

[23] D. B. Campbell. Measurement in radio astronomy. In S. Stanimirović, D. R. Altschuler, P. F. Goldsmith, and Salter C. J., editors, *Single-Dish Radio Astronomy: Techniques and Applications*, volume 278 of *ASP Conference Series*, pages 81–90, San Francisco, CA, 2002. The Astronomical Society of the Pacific.

[24] P. F. Goldsmith. Radio telscopes and measurements in radio astronomy. In S. Stanimirović, D. R. Altschuler, P. F. Goldsmith, and Salter C. J., editors, *Single-Dish Radio Astronomy: Techniques and Applications*, volume 278 of *ASP Conference Series*, pages 45–79, San Francisco, CA, 2002. The Astronomical Society of the Pacific.

[25] W.J. Riley. *Handbook of Frequency Stability Analysis*, volume NIST Special Publication 1065. National Institute of Standards and Technology, U.S. Department of Commerce, July 2008.

[26] David W. Allan. Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221–230, February 1966.

[27] Anders E. E. Wallin. AllanTools documentation, 2014-2019. https://allantools.readthedocs.io/en/latest/.

[28] Marina Berezina. *Pulsar searching with the Effelsberg telescope*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, February 2020. Available at https://hdl.handle.net/20.500.11811/8250.

[29] F. Di Vruno, B. Winkel, C. G. Bassa, G. I. G. Józsa, M. A. Brentjens, A. Jessner, and S. Garrington. Unintended electromagnetic radiation from Starlink satellites detected with LOFAR between 110 and 188 MHz. *arXiv e-prints*, page arXiv:2307.02316, July 2023.