# Data analysis for Giant Crab Pulses

**Aoife Brennan**

Supervisor: Dr. Evan Keane

I-LOFAR

## Acknowledgements

I would like to express my very great appreciation to Evan Keane for mentoring me during my internship. I would also like to offer my special thanks to David McKenna, for all the guidance during the internship. I would like to thank Steve Croft and Howard Isaacson for organizing and running the BL internship program. Finally, I would like to thank everyone at I-LOFAR and BL.

## Abbreviations

| | |
|---|---|
| Dispersion measurement | DM |
| Signal to Noise | SN |
| BreakThrough Listen | BL |
| Irish Low Frequency Array | I-LOFAR |
| Fast Radio Burst | FRB |
| Radio Frequency Interference | RFI |
| Interstellar Medium | ISM |
| Bandwidth | BW |
| Fast Fourier Transform | FFT |
| Search for Extraterrestrial Intelligence | SETI |
| Transiting exoplanet Survey Satellite | TESS |
| Polyphase Filterbank | PFB |
| Discrete Fourier Transform | DFT |

# Contents

# 1 Introduction

During the summer of 2020, I undertook an internship with I-LOFAR and the BL initiative. The internship's main aim was to set up the newly installed BL computer backend located on the I-LOFAR site. In preparation, I undertook several data processing exercises to develop the skill set needed to accomplish this goal. This report details the third exercise that I completed. I have processed raw data by completing this exercise, altered a polyphase filter code, and a coherent dedispersion code to the data used in this task.

The data processed in this task contains giant pulsars from the crab pulsar, located in the crab nebula. The crab nebula was discovered in 1731 by John Bevis. The nebula was named by William Rosse in 1840 when observing the nebula with the 36-inch telescope located in Birr. The crab nebula, a supernova remnant, contains a young pulsar (PSR B0531+21), which emits pulses thousands of times more intense than the average pulsar. The Crab Pulsar is one of the very few pulsars to be identified optically. The crab pulsar is roughly 20 kilometer in diameter and has a rotational period of 33 milliseconds. The outflowing relativistic wind from the neutron star produces the bulk of the emission from the nebula.

This report is divided into five sections. The first section describes how the stokes vectors I,Q,U, and V were formed. Then all subsequent data preprocessing is documented in section two. Section three gives a brief description of a polyphase filter and the general procedure for processing a polyphase filterbank. Section four starts with a description of coherent dedispersion, and then outlines how my code performs coherent dedispersion. The last section contains a summary of the results, along with a discussion.

# 2 Stokes Vectors

The data is raw, therefore we need to create a filterbank file, before processing the data further. The data was first read in, then each stokes vector is calculated, by splitting the data into chunks, the stokes I vector is then saved to a new file. This file is the filterbank file. The Stokes parameters are four quantities that fully describe the polarization of an electromagnetic wave. The four Stokes parameters, I, Q, U and V , describe the electric field's amplitudes of the perpendicular components. The parameter I is a measure of the total power in the wave, Q and U describe the linearly polarized components, and V describes the circularly polarized component.

The Stokes parameters have the dimensions of flux density. Below the four Stokes vectors are defined, where Xr is the real component of the X polarisation, Xi is the imaginary component of the X polarisation, Yi is the imaginary component of the Y polarisation, Yr is the real component of the Y polarisation.

$$
\begin{aligned}
Q &= \left[Xr^2 + Xi^2\right] - \left[Yr^2 - Yi^2\right] \\
I &= \left[Xr^2 + Xi^2\right] + \left[Yr^2 + Yi^2\right] \\
U &= 2\left[Xr * Yr\right] + \left[Xi * Yi\right] \\
V &= 2\left[Xr * Yi\right] - \left[Xi * Yr\right]
\end{aligned}
$$

## 3 Pre-processing

After forming the filterbank files containing the stokes I vectors, we need to dedisperse the data to correct for the signal's dedispersion as it travels through the ISM medium. This is done by adding time delays, as described in my first report. Unfortunately, the file is currently too large to dedisperse using the sigproc command. Hence before dedispering, the data is decimated, using the sigproc command. This means that adjacent time channels are added together to reduce the size of the original data. After this, the data can be downsampled using the the sigpyproc command. The file can now be incoherenty dedisperesed for a value of 56.75, using the sigproc command. Viewing the time series, revealed RFI, this can be removed by identifying and removing the channels that contain RFI. After removing channels containing RFI, the bandpass is much clearer, as seen below. As the data has be dedispered and RFI has been removed, we should be able to view the pulses in the time series. I plotted the time series, then zoomed in on three possible pulses, as seen below.

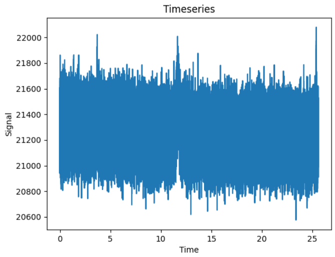


Figure 1: Time Series

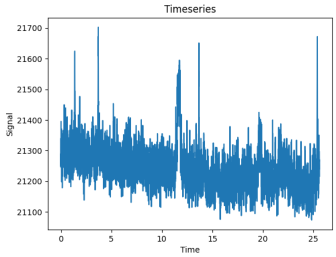


Figure 2: Time Series

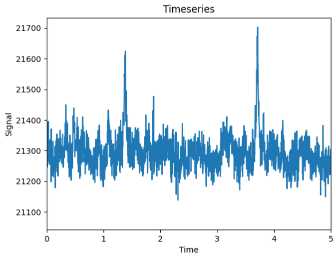


Figure 3: First and Second Pulse

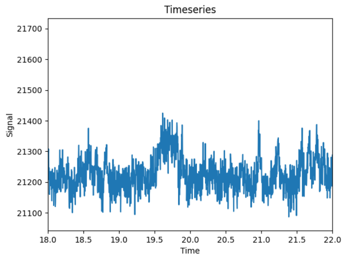


Figure 4: Third Pulse

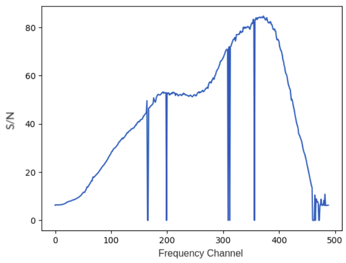


Figure 5: Clean Bandpass

The three pulses in the time series are clearly visible, to investigate this further a single pulse search can be carried out. In the diagram below, I have plotted SN versus time, we can clearly see that our search has found the first two pulses but not the third, which is quite weak.

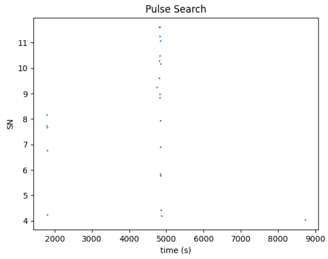


Figure 6: Single Pulse Search

# 4 Polyphase Filterbank

A spectrometer is a device used to record and measure the power spectral density of a radio signal. A PFB is a version of a filterbank which is less costly comparatively, a PFB is constructed from an FFT preceded by a prototype polyphase FIR filter frontend. I adapted code written by Danny Price to apply a polyphase filter to my filterbank, which I created before, as described above. The polyphase filter consists of two steps. The first step is to apply a linear filter, which combines previous time domain spectra, into one filtered spectra.

In the PFB code, the linear filter combines samples from the single channel of raw spectra. Which is described by a finite impulse response filter. The second step is to apply a DFT. The DFT is applied on a single filtered spectra which produces a single frequency spectra. The finite impulse response filter is a convolution of samples within a single channel. The number of past samples that the finite impulse response filter operates on is called taps, denoted by T in the code. The choice of a response function b depends on the polyphase filter's desired features.

The polyphase filter reduces errors introduced by a discrete Fourier transformation. The main errors corrected for by the PFB, are DFT leakage and DFT scalloping loss. The polyphase filter can also serve for sample rate conversions and as a bandpass filter. The polyphase filterbank is shown to suppress the spectral leakage effect of the DFT in a more effective way than the traditional FFT. Hence polyphase filterbanks are widely used in radio astronomy signal processing. In the diagram below, my implementation of the adapted PFB code is shown.

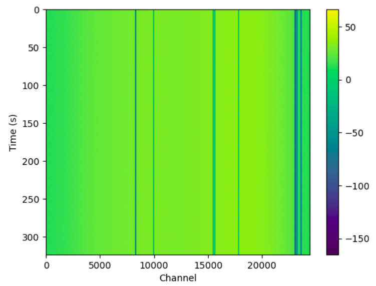


Figure 7: Polyphase Filterbank

# 5 Coherent Dedispersion

As a signal propagating through the ISM it undergoes dispersion, causing a frequency dependent time delay, which is related to the column of electrons between the source and observer. The ISM can be modeled as a filter, as the medium modifies the phase of the signal as it passes through it. The model is referred to as a transfer function. Applying this filter to the data coherently removes the dispersion delay. The result is a timeseries with the same time resolution as before coherent dedispersion is applied. This method is referred to as coherent dedispersion. The only real disadvantage of coherent dedispersion is that the process is computationally costly.

Incoherent dedsipersion normally is used when searching for new pulsars. This is achieved by adding appropriate time delays, as described in my first report concerning FRB's. Incoherent dedispersion corrects for dispersion between channels to a high degree. But the dispersion for the finite bandwidth of the individual channels is not corrected for. Coherent dedisperion can be used instead and effectively removes all dispersion. I used the existing code written by Griffin Foster to carry out coherent dedispersion. The code was slightly modified to handle our data but still follows the normal methodology of coherent dedispersion. The graph below compares the accuracy of incoherent and coherent dedispersion. The dash line is incoherent dedispersion, and the unbroken line is coherent dedispersion.

Coherent dedispersion recovers the intrinsic voltage as it originated from the source. This is achieved by convoluting the raw voltages with the inverse of the transfer function of the ISM. The modification of the propagating signal by the transfer function has a very simple relationship in the frequency domain, given by the first equation below. Hence the convolution is most efficiently performed as a multiplication in the frequency domain. In addition to

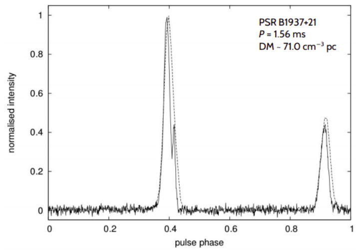


Figure 8: Comparison of Coherent and Incoherent Dedispersion

the convolution, the convolving filterbank trades time resolution for frequency resolution to create a user-defined number of channels. This channelization step is combined with coherent dedispersion by performing a large forward Fourier transform.

$$V\left(f0+f\right)=Vint\left(f0+f\right)H\left(f0+f\right)$$

Where the transfer function is given by,

$$H\left(f0+f\right)=exp\left[-ik\left(f0+f\right)\right]$$

The transfer function can be rewritten as,

$$H\left(f0+f\right)=exp\left\{\frac{-2id\pi}{c}\left[\left(f0-\frac{fp^2}{f0^2}\right)+\left(1+\frac{fp^2}{f0^2}\right)f\right.\right.$$
$$\left.\left.-\left(\frac{fp^2f^2}{2\left(f+f0\right)f0^2}\right)\right.\right.$$

This form of the transfer equation has a constant phase offset (first term), it also has a constant time delay (second term). The third term describes the phase rotation of the signal and is used to recover the undispersed signal, then the inverse transfer function is applied. In summary, the code determines the transfer function for the data and applies its inverse to the measured Fourier transformed voltage. The result is then transformed back into the time domain by applying the transfer function's inverse to the sampled voltage. To do this, we need to modify the phases of the complex Fourier components. This requires sampling and digitization such that both amplitude and phases are measured using the method called baseband sampling. A set of data of length n samples is taken, then compute a discreetly sampled chirp function for n samples. Fourier transform the data set of n points and multiple the resulting back into the time domains. Then take the next n set of points and add them to the last n set of points, this is then repeated.

## 6 Results

In summary the raw data taken by I-LOFAR containing bright pulses, was processed so that the three pulses are clearly visible in the time series. This was accomplished by forming the Stokes vectors, decimating and downsamplying the data appropriately. The bandpass was also examined, revealing channels that were contaminated by RFI, these channels were removed. Using a polyphase code written by Danny Price, and coherent dedispersion code written by Griffin Foster, I tried to adapt both codes. The polyphase code was relatively successful, as it does apply a linear filter, and then applies a discrete Fourier transformation. Unfortunately this code isn't quite complete as the dynamic range of my plot seems to be off. Lastly the coherent dedispersion code, wasn't very successful, but hopefully can be improved upon in future.

## References