

RFI Mitigation in Radio Astronomy

Nicholas Christopher Thompson

*Thesis presented in fulfilment of the requirements for the degree
Master of Engineering in the Faculty of Engineering at
Stellenbosch University*



Supervisor: Prof. D.B. Davidson
Department of Electrical and Electronic Engineering

April 2014

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2014/02/21

Copyright © 2014 Stellenbosch University
All rights reserved.

Abstract

RFI Mitigation in Radio Astronomy

N.C. Thompson

*Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag XI, Matieland 7602, South Africa.*

Thesis: MScEng (E&E)

April 2014

Technical advances in electromagnetics, signal processing and processing power have led to a significant increase in sensitivity and accuracy in radio telescopes. With this increase in sensitivity, radio frequency interference (RFI) has become a much larger problem. The notable growth in wireless communication as well as self generated RFI has further escalated this problem. In order to utilise the full capabilities of modern radio telescopes, RFI mitigation is required on the captured signals. With the enormous data rates of modern radio telescopes, managing RFI has become increasingly difficult, and in order to utilise the full captured radio spectrum, more accurate RFI mitigation strategies will be necessary.

The use of different RFI mitigation strategies is studied in the form of online and offline techniques. This includes Spectral Kurtosis, Spectral Flatness and the Var/SumThreshold method. The special case for RFI mitigation in timing pulsars will also be studied. These techniques are well known in the radio astronomy community; here, spectral kurtosis and spectral flatness will be implemented on the raw data as well as the post correlated data. System speed and accuracy will be the deciding factors when testing these methods as possible solutions to this problem.

Uittreksel

Radiofrekwensiesteuring Matiging in Radioastronomie

(*“Radio Frequency Interference Mitigation in Radio Astronomy”*)

N.C. Thompson

*Departement Elektries en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak XI, Matieland 7602, Suid Afrika.*

Tesis: MScIng (E&E)

April 2014

‘n Toename in die sensitiwiteit van hedendagse radioteleskope kan toegedra word aan die tegniese bevordering in elektromagnetika en seinverwerking. Die toename in sensitiwiteit het egter tot die gevolg dat radiofrekwensiesteuring ‘n groter rol speel in hedendaagse radioteleskope. Die groei in die gebruik van radioverbindinge asook die gevolge van self geïnduseerde radiofrekwensiesteuring dra ook verder by tot hierdie probleem. Radiofrekwensiesteuring matiging word toegepas op die opgevangde seine, om sodoende gebruik te maak van die volle kapasiteit van moderne radioteleskope. Die bestuur van radiofrekwensiesteuring word bemoeilik deur die groot hoeveelheid intydse data van die radioteleskope. Meer akurate radiofrekwensiesteuring matigingstegnieke word vereis om die bandwydte ten volle te hanteer.

Daar word op ‘n aantal verskillende matigingstegnieke gefokus. Hierdie tegnieke kan in twee kategorieë verdeel word, naamlik aanlyn- en aflyntegnieke. Onderafdelings van hierdie kategorieë sluit in: spektrale kurtose, spektrale matheid en “Var/SumThreshold”. Daar word ook na ‘n spesiale geval van radiofrekwensiesteuring matiging gekyk, in die opmetting van tydsberekening-pulsars. Alhoewel hierdie tegnieke bekend is in die radioastronomie gemeenskap, word spektrale kurtose en spektrale matheid egter toegepas op die rou data sowel as postgekorreleerde data. Daar sal op stelsel spoed en akuratuur gefokus word, om vas te stel of hierdie metodes wel moontlike oplossings bied tot die probleem bespreek.

Acknowledgements

I would like to express my sincere gratitude to the following people that assisted and supported me throughout this project.

- Prof David B. Davidson for allowing me the opportunity to explore the captivating field of radio astronomy under his guidance and supervision, as well as his continual advice and support.
- Jason Manley for his continual help and guidance.
- Prof Tomas Niesler for introducing me to the theory of spectral flatness.
- Sarah Buchner for inviting me to HartRAO, explaining pulsar timing techniques and the supporting software and providing pulsar data for tests.
- JP Meijers for helping me perform the final tests.
- Sean Passmoor for providing KAT7 test data.
- JP Jansen van Rensburg for providing the interferometer test data, as well as building the system that the radiometer tests were performed on.
- My colleagues for their input whenever I asked for advice.
- My parents, for their never-ending love and support.
- Lastly, to Estelle Du Preez for the constant encouragement, love and support.

Contents

Declaration	i
Abstract	ii
Uittreksel	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
List of Tables	x
Nomenclature	xi
1 Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Overview	2
1.4 Contribution	3
2 Literature Survey	4
2.1 RF and Microwave Operating Bands	5
2.2 Spectrometer and Radiometer	7
2.3 Interferometry	7
2.4 Pulsars	10
2.5 Known Pulsar Observation and Timing	12
2.6 Kurtosis	16
2.7 Conclusion	16
3 Digital Back-end	18
3.1 The Digital Back-end Hardware of Radio Telescopes	18
3.2 Software	23
3.3 Conclusion	25

<i>CONTENTS</i>	vi
4 RFI Mitigation	26
4.1 Online	26
4.2 Offline	33
4.3 Other Known Methods	39
4.4 Conclusion	39
5 Modelling	41
5.1 Spectral Kurtosis	41
5.2 Spectral Flatness	47
5.3 SumThreshold	50
5.4 Conclusion	53
6 Measurement Setup	55
6.1 Spectral Kurtosis	56
6.2 Spectral Flatness	58
6.3 Pulsar RFI Mitigation	59
6.4 Var/SumThreshold	60
6.5 Conclusion	60
7 Measurements	61
7.1 Spectral Kurtosis	61
7.2 Spectral Flatness	69
7.3 Var- and SumThreshold	74
7.4 PSRchive RFI Mitigation	78
7.5 Daisy-Chain	82
7.6 Conclusion	85
8 Conclusion	86
8.1 Summary	86
8.2 Tests and Measurements	87
8.3 Future Development	89
List of References	90
Appendices	94
A Spectral Variability	95
B Variance derivation	100

List of Figures

2.1	Radio Telescope.	4
2.2	Atmosphere Transparency.	5
2.3	Spectrometer Design.	8
2.4	Model of a Pulsar.	11
2.5	Vela pulse example	12
2.6	Vela pulse profile example	14
2.7	Kurtosis example.	17
3.1	Radio Telescope Layout.	19
3.2	The KatADC	20
3.3	The ROACH1	22
3.4	PFB Example.	24
4.1	Narrowband Pulsar RFI Mitigation.	37
4.2	Median-smooth Example.	38
4.3	Broadband Pulsar RFI Mitigation.	38
4.4	Active RFI Cancellation.	40
5.1	Spectral Kurtosis Waterfall Plot.	42
5.2	Spectral Kurtosis Waterfall Slice.	43
5.3	Spectral Kurtosis varied Amplitude.	44
5.4	Spread Spectrum PSD.	45
5.5	Spectral Kurtosis of Broadband signal.	45
5.6	SK of the Spread Spectrum at 1%.	46
5.7	SK of the Spread Spectrum at 100%.	46
5.8	Waterfall plot of the test Spectrum.	47
5.9	Total Power per Frequency Channel.	48
5.10	Spectral Flatness per Frequency Channel.	48
5.11	Power per Time Bin.	49
5.12	Spectral Flatness per Time Bin.	50
5.13	Waterfall Plot of Spectrum for SumThreshold.	51
5.14	Waterfall of VarThreshold.	52
5.15	Waterfall of SumThreshold.	53
5.16	Difference between Var and SumThreshold.	54

<i>LIST OF FIGURES</i>	viii
6.1 ROACH Spectral Kurtosis Radiometer.	57
7.1 Radiometer integrated spectrum.	62
7.2 Radiometer spectral kurtosis estimate.	63
7.3 Cleaned Radiometer Spectrum.	63
7.4 Raw KAT7 spectrum.	64
7.5 Spectral kurtosis estimate.	65
7.6 Clean KAT7 spectrum.	65
7.7 Noise in KAT7 spectrum.	65
7.8 Raw interferometer spectrum.	66
7.9 Raw interference fringes.	67
7.10 Interferometer spectral kurtosis estimate.	67
7.11 Clean interferometer spectrum.	68
7.12 Noise in interferometer spectrum.	68
7.13 SK Cleaned interference fringes.	68
7.14 Raw Radiometer Spectrum.	70
7.15 Radiometer Spectral Flatness.	70
7.16 Cleaned Radiometer Spectrum.	70
7.17 Noise in Radiometer Spectrum.	70
7.18 Spectral Flatness Measure.	71
7.19 Spectral flatness cleaned spectrum.	71
7.20 Spectral flatness noise spectrum.	71
7.21 Interferometer spectral flatness.	72
7.22 Spectral flatness cleaned spectrum.	73
7.23 Spectral flatness noise spectrum.	73
7.24 SF cleaned interference fringes.	73
7.25 VarThreshold cleaned spectrum.	75
7.26 SumThreshold cleaned spectrum.	75
7.27 VarThreshold noise spectrum.	75
7.28 SumThreshold noise spectrum.	75
7.29 VarThreshold cleaned interference fringes.	75
7.30 SumThreshold cleaned interference fringes.	75
7.31 M Variation.	76
7.32 M Variation time.	76
7.33 ρ Variation.	76
7.34 VarThreshold cleaned spectrum.	77
7.35 SumThreshold cleaned spectrum.	77
7.36 VarThreshold noise spectrum.	77
7.37 SumThreshold noise spectrum.	77
7.38 Raw Narrowband Noise Example.	79
7.39 Cleaned Narrowband Noise Example.	79
7.40 Raw Broadband Noise Example.	81
7.41 Cleaned Broadband Noise Example.	81
7.42 Clean interferometer spectrum.	82

LIST OF FIGURES

ix

7.43 SK and SF Cleaned interference fringes.	82
7.44 Clean interferometer spectrum.	83
7.45 SK and SumThreshold Cleaned interference fringes.	83
7.46 Clean interferometer spectrum.	84
7.47 SK and SF Cleaned interference fringes.	84

List of Tables

2.1	Frequency Bands Allocation	6
7.1	Algorithm Results Summery	85

Nomenclature

Constants

$$\pi = 3.141\,592\,654$$

$$e = 2.718\,281\,828$$

Variables

X_k	Frequency domain signal	[V]
x_k	Time domain signal	[V]
f	Frequency	[Hz]
t	Time	[s]

Subscripts

k	Frequency Bin
t	Time

Abbreviations

ADC	Analogue to Digital Converter
BORPH	Berkeley Operating system for ReProgrammable Hardware
BRAM	Block Random-Access Memory
CASPER	Collaboration for Astronomy Signal Processing and Electronic Research
DFT	Digital Fourier Transform
DRAM	Dynamic Random-Access Memory
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPU	Graphics Processing Unit
H1	Hydrogen-1
ISM	Inter Stellar Medium
KAT-7	The seven-dish MeerKAT precursor array

NOMENCLATURE

xii

PCIe	Peripheral Component Interconnect Express
PSD	Power Spectral Density
QRAM	Quad data rate static Random-Access Memory
RAM	Random-Access Memory
ROACH	Reconfigurable Open Architecture Computing Hardware
RFI	Radio Frequency Interference
SK	Spectral Kurtosis
SRAM	Static Random-Access Memory
VLBI	Very-Long-Baseline Interferometry

Chapter 1

Introduction

1.1 Background

Astronomy is one of the oldest sciences, as shown by historical and pre-historical artefacts. For millennia the sky has been observed, driving improvements in observation techniques. This includes the invention of hardware to aid observation, from the light refracting telescope to the modern interferometric radio telescopes. Radio astronomy as a sub field of astronomy is a very recent discovery in the science, dating to the confirmation of radio frequency emissions by Jansky in the 1930s of the Milky Way. Radio astronomy aided in the Big Bang theory with the detection of the cosmic background radiation, the hydrogen line and the discovery of pulsars by S. J. Bell in 1967. Modern radio telescopes rely on interferometry to increase the resolution of measurement, to look at fainter and more distant sources. The large array that are constructed using interferometry increase the collecting area and thus increase sensitivity. This is done to better understand the formation of the universe.

With the advances in electromagnetics and electromagnetic simulation software such as FEKO and CST as well as silicon based electronics, the sensitivity of modern radio telescopes has increased over the century. These advances have also aided in the development of wireless communication technologies and radar. This includes:

- Wi-Fi.
- Cellphones.
- Aeroplane radar navigation.
- Satellite communication
- Satellite navigation.

The advances in these fields have also aided in radio astronomy, as discovery of radio emissions by Jansky was part of a project to locate RFI in telecommunication.

Radio Astronomy in turn, has also directly help with the development of the 802.11 wireless standard by reducing the effects of multipath [1].

The rise in use of these technologies has led to an increase in radio frequency interference (RFI) for radio astronomy. Other sources of RFI include electrical fence sparking that produces broadband noise, as well as lighting that has a similar yet larger effect.

With the increased sensitivity and data throughput in radio telescopes, the problem of RFI has escalated. In order to fully utilise modern radio telescopes, RFI mitigation strategies will have to be implemented on the raw recorded data as well as on the post-processed data.

1.2 Objectives

There are a large collection of RFI mitigation strategies for radio astronomy. Within this project a number of these strategies will be discussed, simulated and tested. The strategies normally follow an online or offline approach in the detecting of RFI. The online approach handles raw data as it is observed by the radio telescope, which results in high data rates. The offline data is usually compressed and thus smaller in size. The list of strategies that will be looked at are:

- Spectral Kurtosis.
- Spectral Flatness.
- Var/SumThreshold.
- Timing-Pulsar specific algorithms.

The focus of each algorithm will be the effectiveness of RFI detection with the least number of false positives, while remaining computational inexpensive. Where possible, the algorithms will be tested on online data as well as offline data. This project will focus only on the flagging of the data and not the removal.

1.3 Overview

The basic background theory of radio astronomy will briefly be discussed as well as the theory of kurtosis. The layout of a modern radio telescope will then be introduced, with the discussion of the hardware and software components that make up the system. The theory behind a number of RFI mitigation techniques will be explained and then tested in simulations. The chosen RFI mitigation algorithms will then be implemented on real radio astronomy data in order to fully test the theory. The data from a two element interferometer in an urban environment, KAT-7 data as well as radiometer data will be used for the testing of the algorithms. The radiometer will be used as a online example and the KAT-7 and two element

interferometer as offline examples. Lastly, a conclusion will be drawn from the final test results by looking at the RFI detection effectiveness of the algorithms and the computational costs.

1.4 Contribution

The theory of spectral kurtosis, spectral flatness as $\text{Var}/\text{SumThreshold}$ is not novel as shown in the work done by [2] [3] [4]. The novelty of the work, however, is described as follows:

- The use of spectral kurtosis on post correlator data in order to detect RFI before the formation of the visibility function.
- The use of spectral flatness to detect RFI in radio astronomy, with the variation of detecting on a per frequency bin basis.

Chapter 2

Background, Definitions and Literature Survey

The radio telescope is a scientific instrument used in a variety of experiments. The experiments range from the mapping of sky to the detection of the formation of the universe. For this reason, the radio telescope has a number of configurations in order to accomplish all of the tasks. The basic layout of a radio telescope is shown in Fig 2.1. The modern approach in radio astronomy is to leave out the IF system and digitise the the full band with no down-conversion. A number of configurations are introduced and explained in the following chapter as well as basic theory behind some of the RFI mitigation techniques. The radio telescope configurations include:

- Spectrometer.
- Radiometer.
- Interferometer.
- Pulsar Measurement.

The basic theory of kurtosis is briefly introduced and will be later expanded on in Sec 4.1.1.

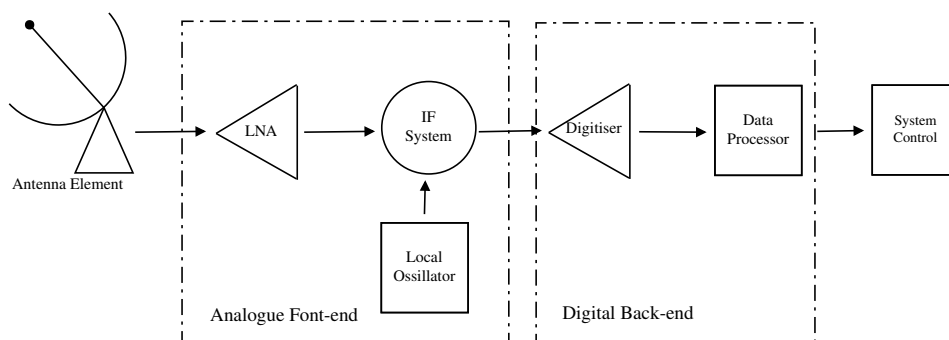


Figure 2.1: The basic layout of a radio telescope.

2.1 RF and Microwave Operating Bands

For earth-based radio astronomy, the radio window is used to determine viable bands to do observations in. The radio window is the frequency band where radio frequencies can pass through earth's atmosphere. The ionosphere determines the lower limit at 30 MHz, where as the upper limit is about 1 THz [5]. The transparency of the atmosphere is shown in Fig. 2.2. From this we can see that the transparency is not only dependent on frequency, but on altitude as well.

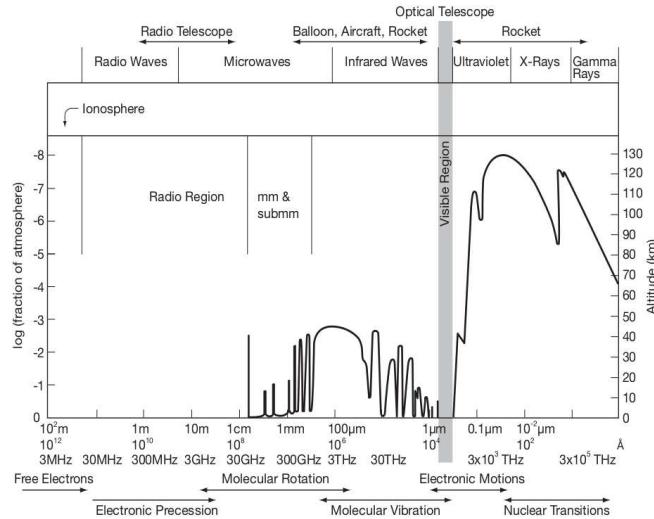


Figure 2.2: The transparency of the atmosphere as taken from [6].

Within the radio window, a number of bands are allocated for radio astronomy in South Africa [7]. The use of these bands are further discussed in [8]. The summary of the allocated bands are shown in Table 2.1. A number of the bands are shared with other services, as shown in [7].

Although a number of frequency bands are allocated for radio astronomy, radio astronomical observations are not limited to these bands. The bandwidth of the observations are increased in order to raise the signal-to-noise ratio of the measurements [9]. This can be seen in the radiometer equation [9]:

$$\text{SNR} = \frac{T_{src}}{T_{sys}} \sqrt{\delta t B} \quad (2.1.1)$$

SNR : Signal-to-noise ratio of the observed source.

T_{src} : The temperature of the source.

T_{sys} : The system temperature of the full system.

δt : The time of the observation.

Frequency Band (MHz)	Shared Service	Scientific Observations
Frequencies below 100MHz		
25.550 - 25.670		RF Bursts (Sun, Jupiter), Interstellar Medium, Steep spectrum sources
37.5 - 38.25	Mobile Communication.	RF Bursts (Sun, Jupiter), Interstellar Medium, Steep spectrum sources
73.0 - 74.6		RF Bursts (Sun, Jupiter), Interstellar Medium, Steep spectrum sources
Frequencies below 1000MHz		
150.05 - 153.0	Mobile Communication except Aeronautical.	RF Bursts (Sun), Interstellar Medium, Steep spectrum sources, Pulsars
322.0 - 328.6	Mobile Communication.	Deutrium line (327.384 MHz), RF Bursts (Sun), Interstellar Medium, Steep spectrum sources, Pulsars
406.1 - 410.0	Mobile Communication except Aeronautical.	RF Bursts (Sun), Interstellar Medium, Steep spectrum sources, Pulsars
608.0 - 614.0	UHF TV Broadcasting.	RF Bursts (Sun), Interstellar Medium, Steep spectrum sources, Pulsars
Frequencies below 5000MHz		
1400.0 - 1427.0	Space Research and Earth Oration-Satellite. (passive)	Galactic and local extragalactic H1 (Hydrogen-1) (1420.406 MHz), Galactic Continuum
1610.6 - 1613.8	Mobile Satellite and Aeronautical Radio Navigation.	OH(Hydroxyl)
1660.5 - 1670.0	Mobile Satellite, Mobile Communication and Space Research.	OH(Hydroxyl)
1710.0 - 1930.0	Mobile Communication.	OH(Hydroxyl)
2655.0 - 2700.0	Mobile Communication except Aeronautical, Mobile Satellite, Space Research and Earth Exploration Satellite.	
4800.0 - 5000.0	Mobile Communication and Space Research.	Formaldehyde, Continuum, VLBI

Table 2.1: The frequency bands that are allocated for radio astronomy between 25 MHz and 5 GHz. This also includes the services that share these bands, as well as the radio astronomical use of the radio bands.

B : The bandwidth of the observation.

The H1 line mentioned in Table 2.1 is used to identify galaxies, due to the abundance of hydrogen in the universe. Due to red-shift, the frequencies of the H1 line of galaxies further away from earth are decreased [9]. This requires the full band from 1420.406 MHz down to 30 MHz to be able map distant galaxies.

The observations made by modern radio telescopes are not always limited to the set, protected, frequency bands and are thus more prone to RFI contamination. For this reason, RFI mitigation is necessary.

2.2 Spectrometer and Radiometer

The basic concept of the Radiometer is to measure the power received by the system. This can be done in terms of total power or power per frequency bin. By definition this includes no phase information. This is useful for determining the total power received from a source.

The frequency spectrum of a given signal is produced by a spectrometer and the output of the spectrometer has polarisations information. Again, the measurements are divided into frequency bins. These bin sizes are a function of the processing capacity of the computer or Field Programmable Gate Array (FPGA). The output for the spectrometer is generally in full Stokes parameters.

A design for a spectrometer has been done on a Reconfigurable Open Architecture Computing Hardware version 1 (ROACH1) board for use with pulsar measurements as well as direct observations. The design for the spectrometer was part of the requirements for AVN project and also included a pulsar-timing system. The help with the design of the digital back-end for the AVN project was done as an introduction into digital signal processing for radio astronomy. The block diagram of the spectrometer design is shown in Fig 2.3.

The basic layout of the ROACH1 implementation is an auto-correlator, where the output is then converted into Stokes parameters. The two polarisations are sent to the Analogue to Digital Converter (ADC) to be digitised, the digital signals are then sent to a Finite Impulse Response (FIR) filter to be windowed before performing a Fourier transform. The windowing of the signal minimises spectral leakage after the Fast Fourier Transform (FFT) [10]. The frequency domain signals of the two polarisations are multiplied with each other, as well as with themselves. This results in the three outputs as shown in the figure $X_i \times X_i^*$, $X_i \times X_j^*$ and $X_j \times X_j^*$. This is sent to a computer to be transformed to Stokes parameters (3.2.1).

2.3 Interferometry

The basic concept of interferometry, as defined in [11], is to increase the resolution of a radio telescope by correlating a number of antenna outputs with each other.

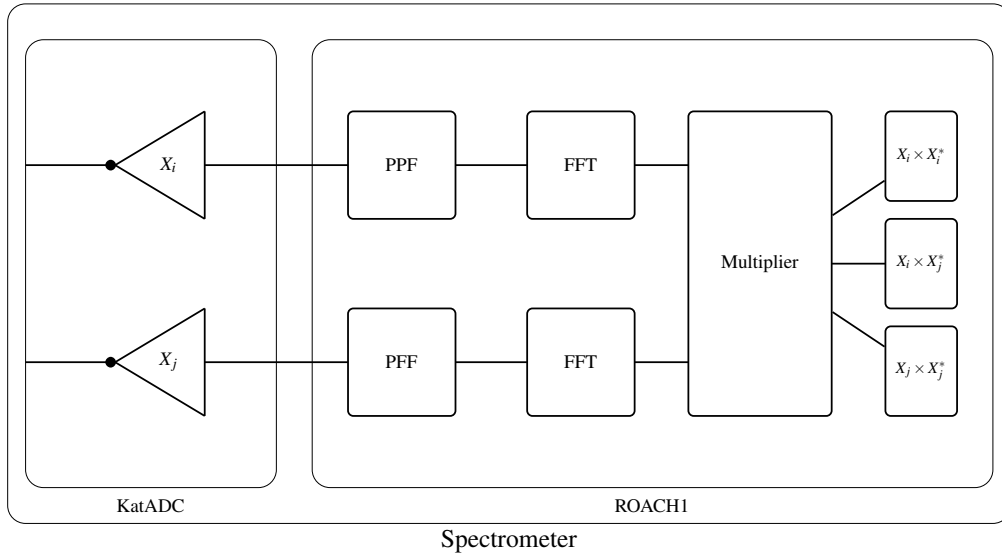


Figure 2.3: The layout of a spectrometer design.

This will allow a number of smaller parabolic dishes to increase their overall resolution in order to achieve the same resolution as a large, unmanageable dish.

2.3.1 Electric Field Propagation

With radio astronomy, the electromagnetic field from sources in space is measured at a point on earth. This means that the summation of electric fields from a random source is measured by an antenna on earth. The signal that is observed from earth is however a time varying signal [12]. Maxwell's equation (2.3.1) in the monochromatic form for an electric field, describes this quasi-monochromatic time-varying observation. It is written in phaser form to deal with time varying signals more elegantly.

$$\mathbf{E}_v(\mathbf{r}) = \iiint P_v(\mathbf{R}, \mathbf{r}) \mathbf{E}_v(\mathbf{R}) dx dy dz \quad (2.3.1)$$

\mathbf{R} : The vector of the source location

\mathbf{r} : The vector location of the observer

$\mathbf{E}_v(\mathbf{r})$: The field effect at the observer as a complex Fourier transform at \mathbf{r}

$P_v(\mathbf{R}, \mathbf{r})$: The propagation and source effect at \mathbf{R} that influence the observer at \mathbf{r}

$\mathbf{E}_v(\mathbf{R})$: The complex Fourier transform of the source at \mathbf{R}

To simplify (2.3.1), a number of assumptions are made by Clark [12]. The first is to ignore polarization on the source signal. This reduces the vector equation to a scalar equation, and the dot product becomes a simple multiplication.

The second assumption is based on the fact that most of the sources are very far away. This means that $|\mathbf{R}|$ becomes very large. The result is that sources can

only be described by a surface brightness, and not by their depth. This also means that space is seen as an extremely large sphere, which implies that there are no “external” sources. All of the sources are from the surface of the “celestial sphere”. The last assumption says that this celestial sphere is empty and that the propagation is directly from the source to the observer with no obstructions. (2.3.1) is then simplified to (2.3.2)

$$E_v(\mathbf{r}) = \int_S \epsilon_v(\mathbf{R}) \frac{e^{2\pi i v \|\mathbf{R}-\mathbf{r}\|/c}}{\|\mathbf{R}-\mathbf{r}\|} dS \quad (2.3.2)$$

$\epsilon_v(\mathbf{R})$: The electric field distribution on the surface of a sphere with radius $|\mathbf{R}|$.

The integral is across S , which represents the surface of the celestial sphere. The equation is written with the physics conversion of $e^{2\pi i v}$, in comparison to the engineering conversion of $e^{-j\omega t}$.

2.3.2 The Correlation of the Electric Field

The correlation of two signals are demonstrated for illustration purposes. These two signal are received at position \mathbf{r}_1 and \mathbf{r}_2 . The correlation is then defined in (2.3.3) as the expected value of a product:

$$V_v(\mathbf{r}_1, \mathbf{r}_2) = \langle \mathbf{E}_v(\mathbf{r}_1) \mathbf{E}_v^*(\mathbf{r}_2) \rangle \quad (2.3.3)$$

\mathbf{E}_v can then be substituted into (2.3.2) from (2.3.3).

$$V_v(\mathbf{r}_1, \mathbf{r}_2) = \left\langle \iint \epsilon_v(\mathbf{R}_1) \epsilon_v(\mathbf{R}_2) \frac{e^{2\pi i v \|\mathbf{R}_1-\mathbf{r}_1\|/c}}{\|\mathbf{R}_1-\mathbf{r}_1\|} \frac{e^{-2\pi i v \|\mathbf{R}_2-\mathbf{r}_2\|/c}}{\|\mathbf{R}_2-\mathbf{r}_2\|} dS_1 dS_2 \right\rangle \quad (2.3.4)$$

It can also be assumed that the source will be spatially incoherent. This implies that $\langle \epsilon_v(\mathbf{R}_1) \epsilon_v(\mathbf{R}_2) \rangle = 0$ if $\mathbf{R}_1 \neq \mathbf{R}_2$. (2.3.4) is then simplified to (2.3.5)

$$V_v(\mathbf{r}_1, \mathbf{r}_2) = \int \left\langle \|\epsilon_v(\mathbf{R})\|^2 \right\rangle \|\mathbf{R}\|^2 \epsilon_v(\mathbf{R}_2) \frac{e^{2\pi i v \|\mathbf{R}_1-\mathbf{r}_1\|/c}}{\|\mathbf{R}_1-\mathbf{r}_1\|} \frac{e^{-2\pi i v \|\mathbf{R}_2-\mathbf{r}_2\|/c}}{\|\mathbf{R}_2-\mathbf{r}_2\|} dS \quad (2.3.5)$$

The final simplification done by Clark is to rewrite (2.3.5) by substituting $\frac{\mathbf{R}}{\|\mathbf{R}\|}$ with the unity vector \mathbf{s} as well as $\left\langle \|\epsilon_v(\mathbf{R})\|^2 \right\rangle \|\mathbf{R}\|^2$ with $I_v(\mathbf{s})$ (the observed intensity). An assumption can be made that most sources will be at a great distance, as most observation are intergalactic. This will allow all the small terms to fall away as well as the replacement of dS with $\|\mathbf{R}\|^2 d\Omega$:

$$V_v(\mathbf{r}_1, \mathbf{r}_2) \approx \int I_v(\mathbf{s}) e^{-2\pi i v \mathbf{s} \cdot (\mathbf{r}_1 - \mathbf{r}_2)/c} d\Omega \quad (2.3.6)$$

This equation shows that the output of the correlation is only dependent on the distance between the two receivers. The source can then be mapped by keeping

one antenna stationary and moving the second antenna to form different baseline lengths. The same result can be achieved by allowing the earth's rotation to change the baseline lengths between the two antennas as the earth rotates. The visibility function or spatial coherence function is the Fourier transform of the source brightness.

2.4 Pulsars

A pulsar is a rapidly rotating, highly magnetised neutron star [13, p.54]. They were first observed by S. Jocelyn Bell in 1967 after observing strange pulsating sources of electromagnetic energy in the sky [14]. The neutron star produces a high energy radio beam that rotates with the star at a set frequency. The star loses energy by the emission of the electromagnetic beam and causes a decrease in angular momentum. The rotation and magnetisation is a result of the formation of the neutron star. This formation process is due to the core of a large star collapsing after a supernova event, leaving most of the angular momentum to the much smaller neutron star. Due to the large decrease in radius, the neutron star's rotation speed is increased. The neutron star has a weight of more than one solar mass, but a radius close to 20km which results in an extremely high density core as well as a strong gravitational pull. For this reason, the gravitational effects, as well as the strong magnetic field emitted by the star, provide a testing platform for physics. This includes gravity, gravitational potential and the magnetic field of the galaxy. A model of the star is shown in Fig. 2.4.

Pulsars can be discovered when the strong radio beam of the neutron star sweeps past an observing radio telescope. The periodic rotation of the beam is seen as frequency dispersed pulses. These pulses are dispersed in frequency because of the interstellar medium (ISM). Looking at the refractive index R of the ISM as shown in (2.4.1), the effect of electron density as well as the observing frequency can be seen in interstellar observations [13, p84].

$$R(f) = \sqrt{1 - \left(\frac{f_p}{f}\right)^2} \quad (2.4.1)$$

where f is the observing frequency and f_p the plasma frequency defined as:

$$f_p = \sqrt{\frac{e^2 \eta_e}{\pi m_e}} \simeq 8.5 \text{kHz} \left(\frac{\eta_e}{\text{cm}^{-3}}\right)^{\frac{1}{2}} \quad (2.4.2)$$

η_e : Electron density where $\eta_e \sim 0.03 \text{cm}^{-3}$

e : Electron charge.

m_e : Electron mass.

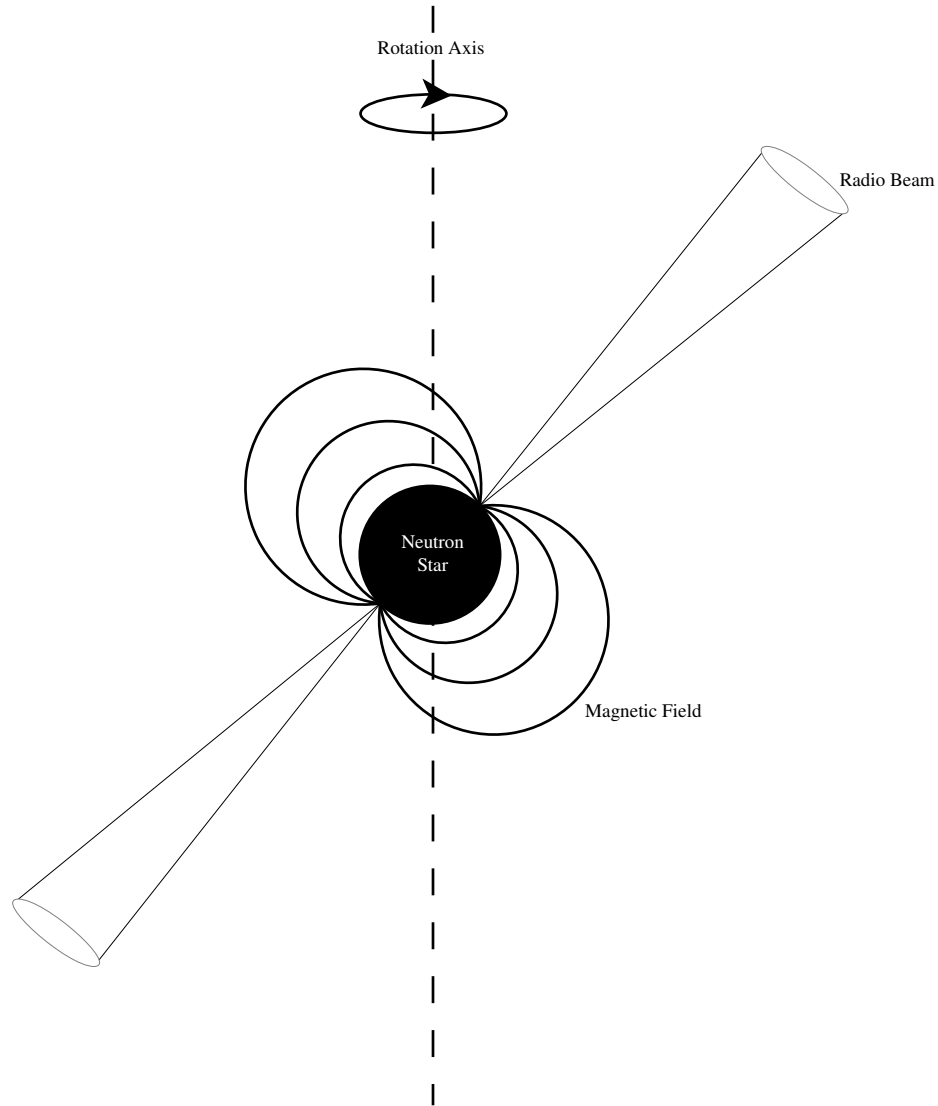


Figure 2.4: A simplified model of a pulsar.

The signal delay from the source to earth can now be computed using the refraction index (2.4.1).

$$t = \left(\int_0^d \frac{dl}{v_g} \right) - \frac{d}{c} \quad (2.4.3)$$

t : Time delay between source and earth.

d : Distance between source and earth.

c : Speed of light in a vacuum.

v_g : Group velocity of the propagating wave, where $v_g = cR$, with R defined as in (2.4.1)

It can be seen from (2.4.3) that the delay in wave propagation is a function of frequency and distance. The result is that the broadband pulses from pulsars are dispersed in frequency. An example of a received dispersed pulse from the Vela pulsar is shown in Fig. 2.5. In radio astronomy, these electromagnetic pulses

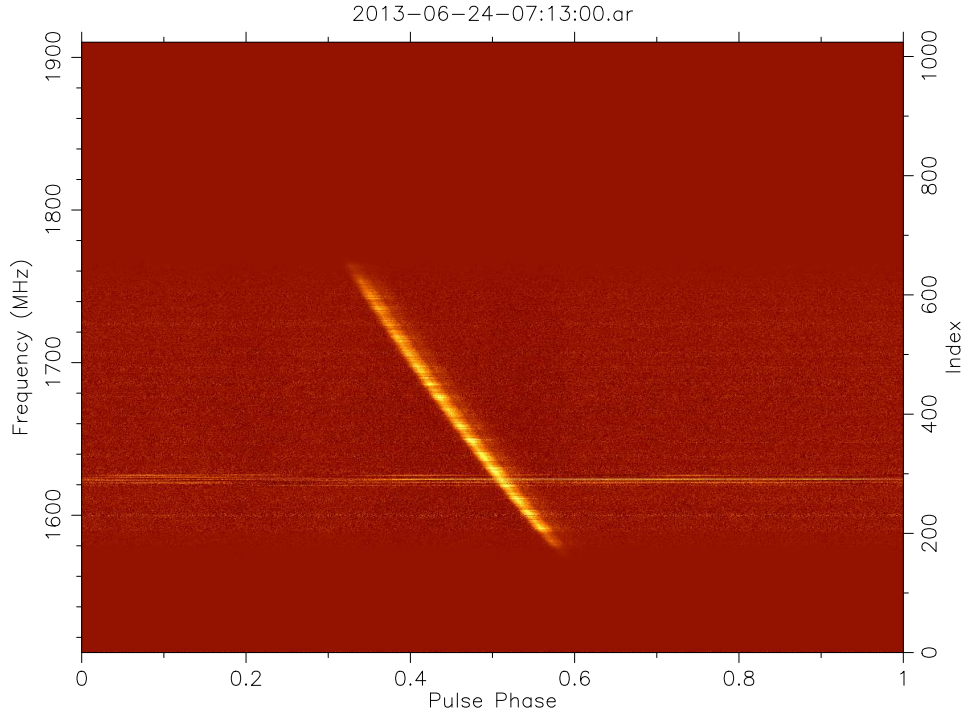


Figure 2.5: An example of a measured pulse from the Vela pulsar as measured at 2013/06/24 07:13:00 using a second of folded data (see Sec 2.5). The graph is drawn using PSRchive [15]. The data was provided by [16].

emitted by pulsars are monitored and classified. The frequency of the pulses can vary from millisecond to a few seconds [13]. The accurate timing of millisecond pulsars is important as they can be used to compare the performance of clocks as well as indicate the internal workings of a pulsar if it glitches [13]. Glitches in pulsars can occur when the pulsar's rotational speed is increased for a brief period. RFI in the measurements, as seen in the Vela measurement (Fig. 2.5) at 1620 MHz, will result in timing of the pulsar varying for each pulse. RFI mitigation is thus necessary for the accurate measurement of known pulsars, to distinguish between glitches and RFI noise.

2.5 Known Pulsar Observation and Timing

The process of discovering new pulsars usually involves de-dispersing chunks of data with a number of trial de-dispersion measures which we then analysed for pe-

riodic signals [13, p127]. The full procedure of finding new pulsars is not discussed in this thesis as the main focus is the RFI mitigation for known pulsars.

For a known pulsar, the data must be analysed and the pulsar classified. The pulse itself has to be de-dispersed in order to extract the true pulse shape. The data is folded in order to increase the signal to noise ratio of the weak observed pulse. The basic steps for folding data are as follow [13, p165]:

- De-disperse the data. This can be done incoherently or coherently.
- The pulse profile is split up in pulse phase bins across the pulse period, in which to save folded data.
- Sample the time series and calculate the pulse phase relative to the pulse period.
- Add the sample to the corresponding phase bin.
- Repeat from step 3 for the next sample.

As mentioned above, the data can be de-dispersed either coherently or incoherently. Incoherent de-dispersion is the simplest way of de-dispersing the incoming signal from the pulsar. The power of each frequency channel relative to time is looked at and the phase of the incoming signal is ignored. Each frequency channel is then delayed by a certain time to de-disperse the overall incoming signal. This technique can be done offline, as it can be used on time averaged data and is also computationally inexpensive to use. However, incoherent de-dispersion does not compensate completely for the dispersion effects of the ISM. Coherent de-dispersion is necessary to remove all the effects of the ISM. The de-dispersion is done on the raw voltage by looking at the ISM as an all-pass filter with transfer function $H(f)$. The recorded and Fourier transformed voltage is described as the original signal passed through the ISM all-pass filter [13, p115].

$$V(f) = V_{int}(f)H(f) \quad (2.5.1)$$

$V(f)$: Fourier transform of the recorded signal.

$H(f)$: ISM transfer function.

$V_{int}(f)$: Original unmodified Fourier transformed signal from the pulsar.

The description of $H(f)$ becomes

$$H(f) = e^{-jk(f)d} \quad (2.5.2)$$

d : The distance between earth and the source.

$k(f)$: $\frac{2\pi}{c}R(f)f$

$R(f)$: The refraction index as defined in (2.4.1).

The process of coherent de-dispersion must be done online. The computational costs as well as the data rates (from the digitiser to the computer) are high.

The result of the de-dispersion and folding is a pulse profile for the pulsar. An example of a pulse profile for Vela can be seen in Fig. 2.6.

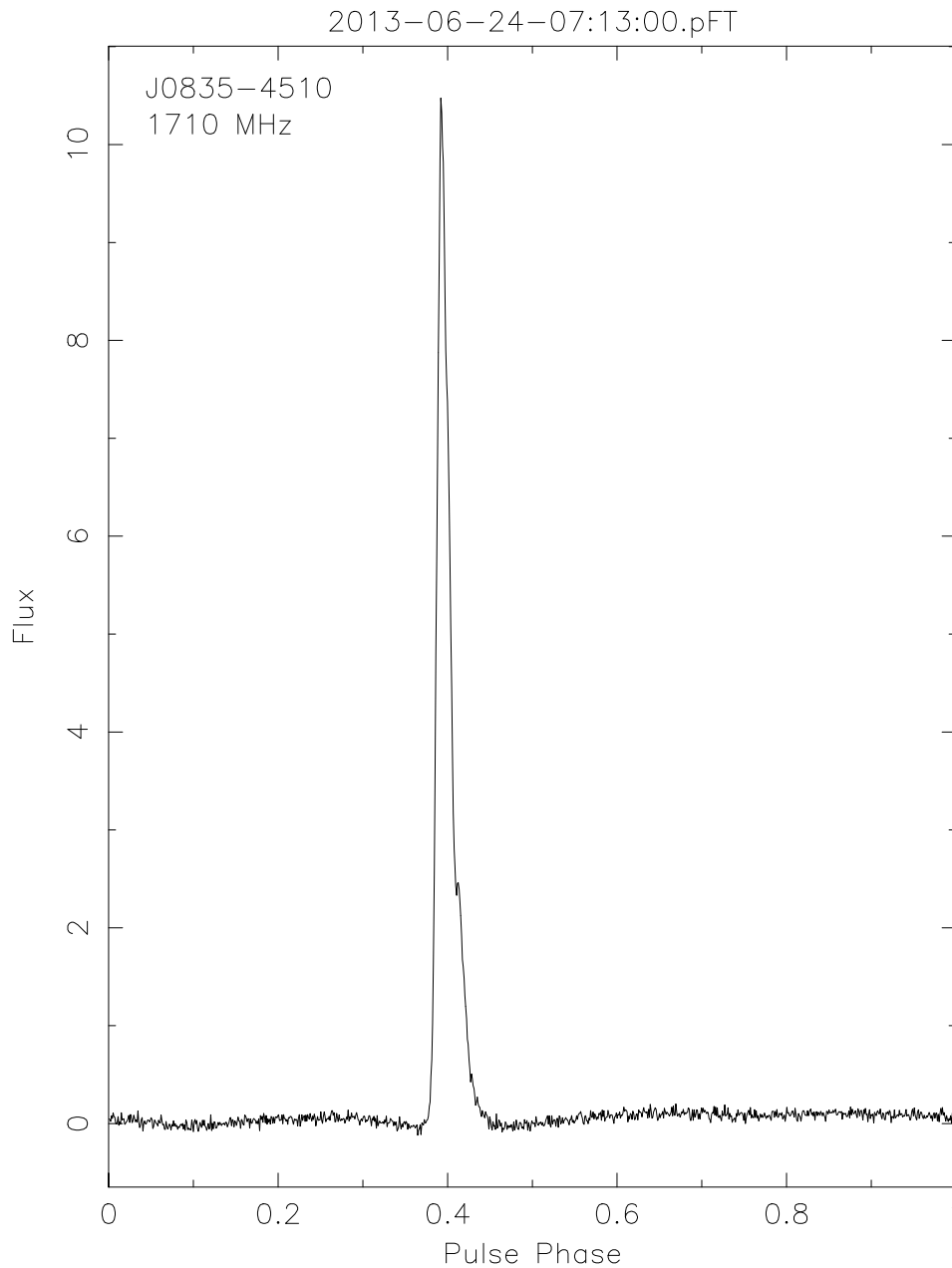


Figure 2.6: An example of a pulse profile from the Vela pulsar as measured at 2013/06/24 07:13:00 using a second of folded data. The graph is drawn using PSRchive [15]. The data was provided by [16].

Once the pulse is folded and de-dispersed, the pulsar can be timed. This is useful

for gathering information about the neutron star itself as well as effects that the ISM have on the propagating signal. Millisecond pulsars are more widely used for this, as they are much more stable timing sources. A known pulsar has a pulse profile built up and each new de-dispersed pulse is then matched to this pulse profile. The time of arrival for the pulse is then normally chosen midway in the pulse profile. The precision of the arrival time is a function of the pulse width and signal to noise ratio of the pulse profile, and is proportional to system parameters.

$$\sigma_{TOA} \simeq \frac{W}{S/N} \propto \frac{S_{sys}}{\sqrt{t_{obs}\Delta f}} \times \frac{P\delta^{\frac{3}{2}}}{S_{mean}} \quad (2.5.3)$$

σ_{TOA} : Uncertainty of pulse arrival time.

W : Pulse width.

S/N : Profile signal to noise ratio.

S_{sys} : System equivalent flux density.

t_{obs} : Integration time for observation.

Δf : Bandwidth of the observation.

P : Pulse period.

δ : $\frac{W}{P}$, the pulse duty cycle.

S_{mean} : The mean flux density of the pulsar.

S_{sys} is defined as the strength of a source with a temperature equal to T_{sys} , and used to characterise the ratio performance of pulsar systems [13]. The signal to noise ratio for the pulse profile is defined as:

$$S/N = \frac{1}{\sigma_p \sqrt{W_{eq}}} \sum_{i=1}^{n_{bins}} (p_i - \bar{p}) \quad (2.5.4)$$

σ_p : Pulse variance.

\bar{p} : Mean Pulse.

p_i : pulse bins.

W_{eq} : The equivalent width (in bins) of a pulse of near-uniform energy density, with the same area and peak height as the observed profile.

Unwanted noise, like RFI, in the system will cause a greater uncertainty in pulse time of arrival and skew results.

2.6 Kurtosis

Kurtosis was introduced by Pearson to measure the shape of a probability distribution function, specifically looking at peak and tails of the function [17, p109]. The formal definition is the ratio of the fourth moment to the square of the variance. It is used to categorise probability density functions based on their shapes. For radio astronomy this is useful as most radio astronomical signals are Gaussian; kurtosis can then be used to determine if the signal is RFI or valid data. The practical implementation of kurtosis is discussed in Sec. 4.1.1. For now, the following definitions are introduced:

$$\beta_2 = \frac{\mu_4}{(\sigma^2)^2} \quad (2.6.1)$$

β_2 : Kurtosis.

μ_4 : Fourth moment of the probability distribution function.

σ^2 : Variance of the probability distribution function.

The value of the kurtosis for a Gaussian distribution function is 3. This will increase or decrease depending on the shape of the function. For this reason γ_2 is defined in [17] as the following.

$$\gamma_2 = \beta_2 - 3 = \frac{\mu_4}{(\sigma^2)^2} - 3 = \frac{\kappa_4}{\kappa_2^2} \quad (2.6.2)$$

γ_2 : Excess kurtosis.

κ_n : The nth cumulant of the probability distribution function.

By this definition γ_2 will be zero for a Gaussian function. All functions for which $\gamma_2 = 0$ are said to be mesokurtic, $\gamma_2 > 0$ leptokurtic and $\gamma_2 < 0$ platykurtic. An example of the kurtosis for a number of common probability density functions is shown in Fig. 2.7.

The coefficient of dispersion as shown in (2.6.3) is related to kurtosis. This relation is discussed in Sec. 4.1.1 and is later used to derive the spectral kurtosis estimate.

$$V_k^2 = \frac{\sigma_k^2}{\mu_k^2} \quad (2.6.3)$$

2.7 Conclusion

In conclusion, the concept of radio astronomy was explained. This included the basic layout of a radio telescope as well as the building blocks thereof. The theory on the chapters to follow will make use of these subsystems in order to accomplish

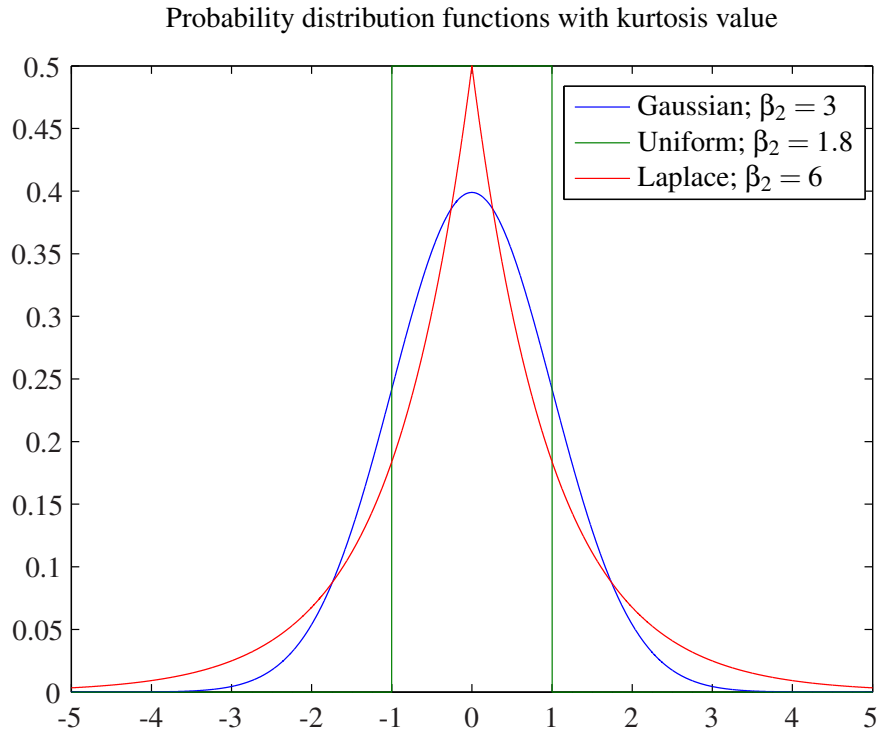


Figure 2.7: The kurtosis, β_2 , for three probability distribution functions; Gaussian/Normal distribution, Uniform distribution and the Laplace distribution

RFI mitigation. Pulsars as well as pulsar timing was also introduced, as this was part of the AVN design specification. The theory on pulsar timing is then later used in order to inspect RFI mitigation techniques specifically for timing-pulsars. The basic theory of Kurtosis is briefly looked at, and will later be expanded on.

Chapter 3

The Basic Digital Back-end Components of a Radio Telescope

As with most technologies over the past century, the radio telescope has shifted from a mostly analogue to a more digital instrument. The main advantage this brings, is the predictability and repeatability of the system, compared to an analogue system. Current FPGA and Graphics Processing Unit (GPU) based system have the added advantage that the instrument itself has become much more fluid in its configuration, as a simple change in software can allow it to change configuration. In this chapter the main components of the digital back-end of a radio telescope will briefly be introduced and explained, as they relate to current multi-application telescopes such as MeerKAT. This includes the physical hardware that is used, the software to set up an application of choice and the layouts for different systems.

3.1 The Digital Back-end Hardware of Radio Telescopes

The first step in the digital back-end of the radio telescope is to digitise the signal, as shown in Fig. 3.1. In the traditional setup, the RF signals are fed to a central point and then digitised. This is true for the seven-dish MeerKAT precursor array (KAT-7), as the RF signal is fed via Analogue Over Fiber links to the control room where it is digitised using a ROACH1 board and KatADCs. The ROACH1[18] board is an open source FPGA based processing board developed by the Collaboration for Astronomy Signal Processing and Electronic Research (CASPER), and the KatADC is an analogue to digital converter developed by SKA-SA. The full specifications of the ROACH1 as well as the KatADC is described later in this chapter. MeerKAT will implement direct digitisation at the antennas and feed the digital data via Ethernet links to the control room.

After the data is in the digital domain, it is then usually passed through as polyphase filter bank (PFB), which comprises of a FIR filter and an FFT. This process is also referred to as the F engine. The Fourier transformation can either be com-

pletely performed on a FPGA, or only coarsely channelised by the FPGA and finely channelised on a GPU. Most of modern radio astronomy is done using some form of spectral analysis. The frequency domain signal can be passed to a multiplier, called the X engine, to form a correlator, or used as a spectrometer where the data can be sent to a pulsar analysis system for monitoring and timing. This spectrometer can be performed per dish or as a tied-array over a number of dishes. The correlation is also done either on a FPGA based system, or on a GPU based one. The data can then be stored to do Very-Long-Baseline Interferometry (VLBI) or for further analysis.

In each step the signal must be time stamped to ensure the accuracy of the measurements, as all the analyses are time sensitive. In the case of pulsars, the arrival time of each pulse must be accurately labelled to ensure the precise timing of the pulsar. As correlation makes use of the phase differences between antennas, to effectively increase the capture area by combining a number of antennas, any timing jitter will result in phase noise in the correlation process. To ensure microsecond timing accuracy, a combination of masers, atomic clocks and GPS time servers are implemented.

The reprogrammable digital radio telescope setup allows for a much more dynamic system, as it is not limited to a single application. It also allows the use of real time calibration techniques and RFI mitigation. Each hardware component is discussed in more detail in the following subsections.

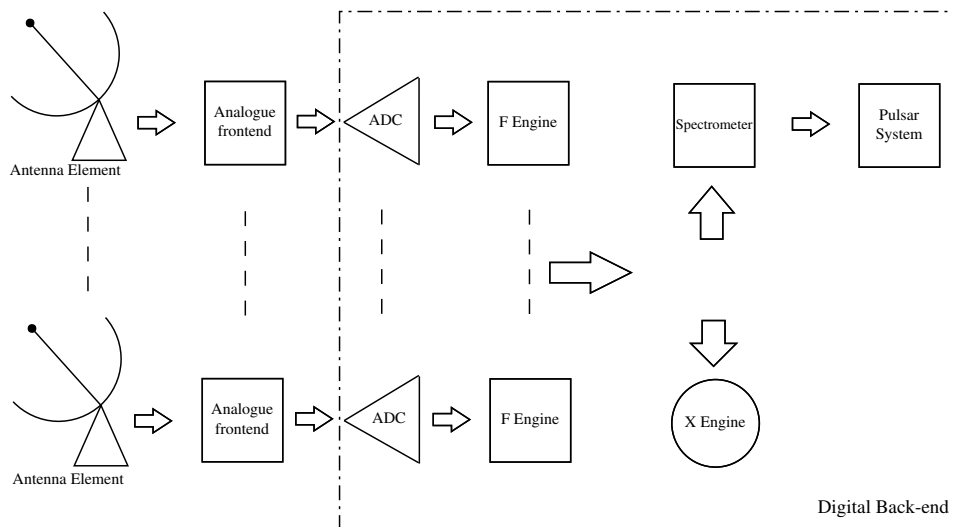


Figure 3.1: The basic components of a radio telescope, specifically the digital back-end.

3.1.1 The Analogue to Digital Converter

The first step in the digital chain is to digitise the input signal. The two main specifications to look at are the input bandwidth of the ADC and the number of bits to which it digitises. These are not the only factors to consider, since the internal noise of the ADC is a major influence on the dynamic range of the telescope. The internal structure of the ADC can lift the noise floor when a number of cores are interleaved to increase the bandwidth. This will cause spurious signals in the output and increase the noise floor. The best solution is to use a monolithic core to ensure a pure digitised signal. Furthermore, timing jitter in the ADC itself can include additional phase noise in the measurements. Non-linearity of the ADC can also affect the signal if it cannot be compensated for correctly, as the power measurements of the frequency bins result in a non-flat spectral response.

There are a number of ADCs available that are compatible with ROACH boards, that make use of the Z-DOK connector. The measurements done in this thesis make use of the KatADC shown in Fig. 3.2, a 8 bit analogue to digital converter with a maximum of 1.5 GSPS per channel[19]. The analogue signal is firstly amplified by 20 dB. The signal can then be attenuated by 0 to 31.5 dB in intervals of 0.5 dB. The attenuation is software controlled via an I²C bus on the ADC. The frequency range of the amplifier at the front end of the board is between 50 MHz and 850 MHz. This can be upgraded to 50 MHz to 6 GHz with a few adjustments on the ADC itself. The output is then demuxed in four lines per channel. This allows for the FPGA to run at quarter of the ADC clock speed, as the data rate is divided by four.

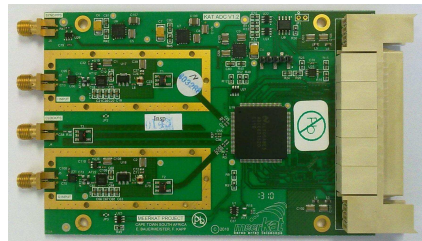


Figure 3.2: A picture of the KatADC as shown in the Casper wiki [19]

3.1.2 FPGA computation

The advantages of the FPGA are the high IO it can handle as well as the low power consumption. The high IO makes it the only practical solution at present to handle the high data rate from the ADC. The FPGA can be used to only channelise and packetise the data, to be sent via Ethernet to an alternative computation processor. The FPGA can also be used to implement the F as well as X engines. In the case of MeerKAT, a smaller FPGA will be used to packetise the time domain samples. The samples will then be sent to a processing bank consisting of ROACH2 boards which are Vertex 6 FPGA based systems. The F engines as well as the X engines

will be implemented on the ROACH2 boards. Other current systems have been implemented where only the F engine is on a FPGA based system, and the X implemented on a GPU. This is done in LEDA [20] where X engines is implemented on a GPU based system. The disadvantages of the FPGAs include:

- Expensive.
- Difficult to program.
- Historically do not support floating point arithmetic.

Newer versions of Xilinx' Vertex range, however, are starting to support floating point operations. The CASPER development chain also simplifies the programming of the ROACH system with the implementation of drag-and-drop boxes in Matlab Simulink.

For the measurements done in this thesis a ROACH1 (see Fig. 3.3) is used for the F and X engine where appropriate. The output from the ADC is fed directly to the Vertex 5 FPGA via the Z-Dok connector, which enables interchangeability between other supported ADC's. The FPGA allows for the mathematical manipulation of the real-time data. More specifically, it allows for real-time FFT and multiplication of the ADC data. The FPGA is programmed and controlled via the on-board PowerPC [18].

The PowerPC can boot from an on-board SD card or netboot. For a stand alone setup, booting from the SD card is the easiest. For larger setups, such as MeerKAT and KAT-7, netboot is a better option as this ensures that all of the boards are running the same operating system version. The PowerPC also has a serial connection as well as an Ethernet adaptor.

To allow for sample buffering and averaging, the ROACH1 has 3 memory types.

- Dynamic Random-Access Memory (DRAM)
- Quad data rate static Random-Access Memory (QRAM)
- Block Random-Access Memory (BRAM)

The DRAM is standard DDR2 RAM that is connected directly to the FPGA. The external RAM of 1 GB allows for a large sample size to be integrated. The PowerPC interface speed is extremely slow, which in turn decreases the copy speed of the DRAM data.

The QRAM module is a Static RAM block that is also connected directly to the FPGA. The interface speed is faster than the DRAM block, but the storage space is limited compared to the DRAM.

The BRAM is block RAM, addressed directly on the FPGA. The space is extremely limited as the memory is allocated on the FPGA's as primitives; however, this is fastest memory access method.

The data can then be extracted from the FPGA via two methods. Firstly, the PowerPC can access the FPGA data via registers. The data is then extracted via the

1 Gig Ethernet adaptor. The ROACH also has four 10-gigabit Ethernet ports that are connected directly to the FPGA. This allows for the data to be directly extracted from the FPGA without the PowerPC slowing it down. The 1 gigabit Ethernet port is sufficient for time averaged observations, where as the 10 gigabit ports are required for real-time data output.

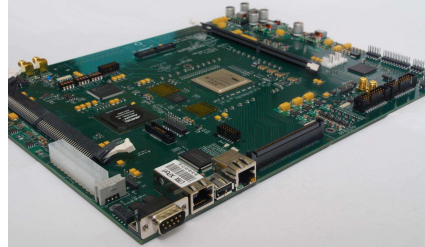


Figure 3.3: A picture of the ROACH1 board as shown in the Casper wiki [18]

3.1.3 GPU Computation

The GPUs are less expensive and easier to program than FPGA based systems, although more complicated than normal x86 based system. Writing code for a GPU system makes use of CUDA for Nvidia GPUs, or OpenCL for cross platform compatibility. GPUs also support floating point arithmetic and are extremely effective with parallel processing. This is a great advantage for radio astronomy as the F and X engines are parallelizable. Implementating an FFT on a GPU permits up to a million point frequency spectrum. The FPGA will do a coarse channelisation of 1024 channels and send it to the GPUs. The GPUs can then use 1024 coarse spectra to produce a 1024×1024 spectrum. With these narrow channels, locating narrowband RFI becomes much simpler. The affected channels can be removed without compromising too much unaffected data.

When the F engine is implemented on a FPGA, the output can then be sent to a X engine GPU cluster. This can reduce the startup and running cost of the system, but only becomes practical on radio telescopes with a large amount of antennas. This is due to the fact that the PCIe bus limits the connection speed to the GPU. The processing requirements are usually of order $O(N^2)$ where as the IO of the system is of order $O(N)$. In a smaller telescope the GPUs can sit idle while waiting for new data, wasting energy in the process. The GPUs are also more power hungry than FPGA.

GPUs are also used for coherent de-dispersion in pulsar analysis as is done in GAPPI [21]. GPUs are not used in the RFI mitigation in this report, as the system is too small to require implementation.

3.1.4 CPU Computation

The use of the CPU is mostly for monitoring, graphing and data transfer/buffering, as the bulk of the data is either handled by GPUs or FPGAs. The incoherent de-dispersion of PSRchive is done on the CPU and the buffering done by PSRdada [22] as well. The monitoring of the system usually includes sensors on each antenna element, which include data such as system temperature and cryogenic system pressure. On the GPU clusters, the CPU transfers the data from the network interface card, to the GPU and back. The CPU load can be reduced with the use of Infini-bands' Remote Direct Memory Access (RDMA) and the newer implementation of RDMA on Ethernet.

3.2 Software

The supporting software for the digital back-end is dependent on the application of the radio telescope. Most of the software is generally custom made for the specific radio telescope, due to the fact that a lot of the hardware is developed for the radio telescope. [23] has brought some conformity to radio astronomy with the development of the open source hardware and supporting tool chain. Other open source projects have also developed packages for radio astronomy, such as PSRchive and PSRdada for pulsar analysis.

3.2.1 Spectrometer

As mentioned in Sec. 3.1, the spectrometer can be implemented on a FPGA-exclusive system or in combination with a GPU cluster. The signal is split in to a number of down-sampled signals, as the frequency of the FPGA is slower than that of the ADC. In the FPGA-only system, the signal is passed through a polyphase filter bank. The FIR minimises spectral leakage between channels once the FFT is performed. Fig. 3.4 shows the effects of pre-filtering compared to a standard square window. The multiple FFT windows of the PFB reduces the width of the channel response, where normal windowing can broaden the channel response.

The Stokes parameters are then calculated if necessary [9]. For a linear polarised system the Stokes parameters are defined as follows:

$$\begin{aligned}
 I &= |X_i|^2 + |X_j|^2 \\
 Q &= |X_i|^2 - |X_j|^2 \\
 U &= 2\Re(X_i X_j^*) \\
 V &= 2\Im(X_i X_j^*)
 \end{aligned}
 \tag{3.2.1}$$

X_i : The i axis of the FFT of the recorded signal X .

X_j : The j axis of the FFT of the recorded signal X .

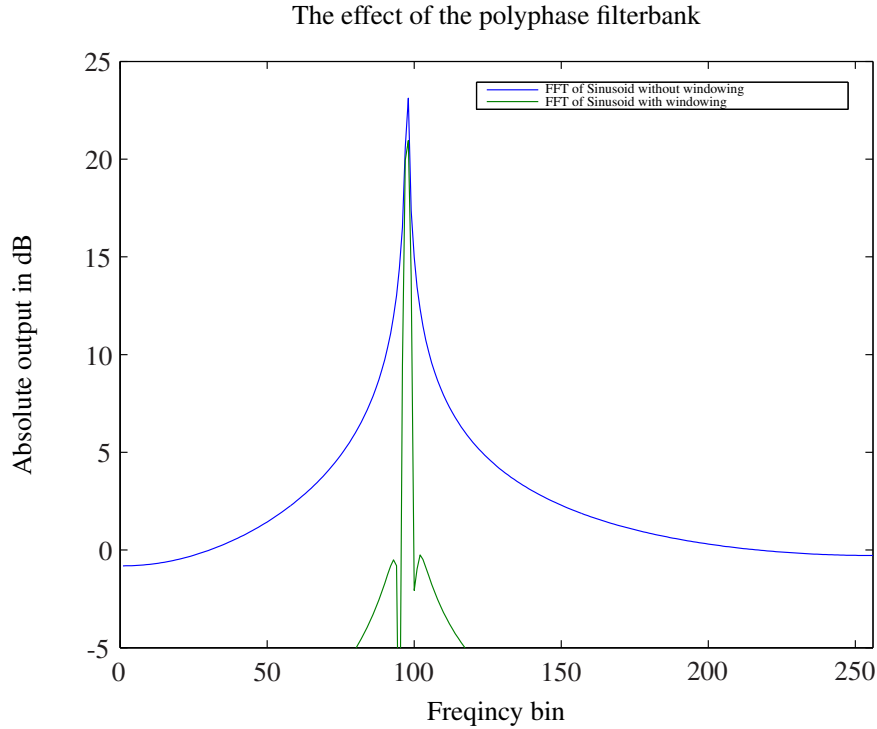


Figure 3.4: An example of the effect of a Hamming window on a digital signal compared to a square window.

This is performed on the FPGA by multiplying the two polarisations with each other as well as with themselves. The multiplication for the computation of the Stokes parameters is done on the FPGA and then fed to a computer or other secondary system. At the next stage, multiple streams of down sampled data are recombined as interleaved data. For the implementation on the ROACH1 with a KatADC, the input signal is split into four down sampled signals. The ADC is then sampling at 800MHz and the ROACH is running at 200MHz.

In the GPU implementation, the data is sent to a GPU cluster after the polyphase filtering. The FPGA is then used to do a coarse channelisation and package the data into UDP packets. Custom software is then run on the GPU cluster to handle the incoming UDP stream and perform the FFT and Stokes parametrisation.

3.2.2 Pulsar Timing

The pulsar timing is done with pre-developed packages such as PSRchive and PSRdada. PSRdada is a ring buffer software that handles the high data rates from the spectrometer. Other software such as PSRchive can then couple to the ring buffer and extract data to be processed. PSRchive can handle the folding, dedispersion pulsar profile and RFI mitigation. At HartRAO, the sigproc package is used for the pulsar timing, to determine pulsar jitter.

3.3 Conclusion

The FPGA is suited for IO intensive operations, as the PCIe bus of GPU results in a bottleneck for data transfers. The GPU, however, implements floating point arithmetic, which in turn simplifies the implementation of the FFT and multiplication operations. The CPU as a computational platform does not compete with either the GPU or FPGA in radio astronomy.

The majority of the software used in radio astronomy is custom built for a specific radio telescope. The development tools and open source hardware created by CASPER simplifies the creation of custom software.

Chapter 4

RFI Mitigation

The increase sensitivity of modern radio telescopes result in a higher influence of external as well as internal RFI on measurements. The RFI contaminated data can cause distortions, artefacts or drown out all the weaker intergalactic radio emissions. In order to fully utilise the radio telescope, the RFI has to be flagged and removed from the final data.

This can be done in a number of ways:

- Removing and blanking all the affected bins.
- Removing a whole time series that has been affected (useful for broadband RFI).
- Removing all bins with the same frequency (useful when removing narrow-band RFI).

In the following section we will mostly be looking at the flagging process for RFI, to indicate possible data corruption. This is done either online on the raw data as it is captured, or offline on stored data as part of the data processing system. The online methods include Spectral Kurtosis and Spectral Flatness. The offline methods include the Sum Threshold as well as pulsar focused methods that are included in the PSRchive package.

4.1 Online

The online methods are more focused on handling raw data with little computational costs, but are not limited to online usage.

4.1.1 Spectral Kurtosis

The general theory of Kurtosis has been discussed in Sec 2.6. The theory of Spectral Kurtosis by Nita [2] builds on kurtosis as discussed in Sec. 2.6 to produce an estimation.

A number of assumptions are made about the digital time domain signal x_n :

- The time domain samples are Gaussian noise with the mean $\langle x_n \rangle = 0$ and a constant variance $\sigma_x^2 = \langle x_n^2 \rangle$.
- The time domain samples are uncorrelated noise, with zero covariance.
- The signal is bandwidth limited to the Nyquist frequency f_c by the sampling frequency, where $f_c = \frac{1}{2}f_n$ and f_n the sampling frequency.
- The test signal is a band limited Gaussian signal (white noise).

Spectral Kurtosis is a frequency domain analysis of the recorded signal, therefore the first step is to transform the time domain signal to a frequency domain signal with the standard DFT as shown in (4.1.1).

$$X_k = \sum_{n=0}^{N-1} w_n x_n e^{-2\pi i k n / N} \quad (4.1.1)$$

X_k : Frequency domain signal.

k : Frequency bin number.

N : Window size of the DFT.

n : Time domain sample number.

w_n : Time domain windowing function coefficients to reduce spectral leakage in the DFT.

x_n : Time domain samples of the recorded signal.

For spectral kurtosis, we are interested in the power of each frequency bin. The frequency bins are then defined in the following (4.1.2).

$$f_k = \frac{2k}{N} f_c \text{ where; } k = 0, 1, \dots, \frac{N}{2} \quad (4.1.2)$$

k : Frequency bin number.

N : Window size of the DFT, number of samples.

f_k : Frequency of frequency bin k .

f_c : Nyquist frequency of the system.

The windowing function has an influence on the statistics of the frequency domain analysis of each frequency bin, although it cannot change the Gaussian nature of the signal [24, p89]. In order to simplify the derivation, a standard symmetrical windowing function will be assumed e.g. Hamming or Blackman window. The power spectral density (PSD) of the signal can now be estimated using the derivation in [25, p657]. The power per frequency bin is defined as follow.

$$\hat{P}_0 = P(f_0) = \frac{1}{W_{ss}} |X_0|^2 \quad (4.1.3)$$

$$\hat{P}_k = \hat{P}(f_k) = \frac{1}{W_{ss}} [|X_k|^2 + |X_{N-k}|^2] \quad (4.1.4)$$

for $k = 1, 2, \dots, (\frac{N}{2} - 1)$

$$\hat{P}_{\frac{N}{2}} = \hat{P}(f_c) = P(f_{\frac{N}{2}}) = \frac{1}{W_{ss}} |X_{\frac{N}{2}}|^2 \quad (4.1.5)$$

\hat{P}_k : Power spectral density estimation per frequency channel f_k .

X_k : Frequency domain of recorded signal as defined in (4.1.1).

W_{ss} : Window squared and summed, see (4.1.6).

$$W_{ss} = N \sum_{n=0}^{N-1} w_n^2 \quad (4.1.6)$$

The input signal is assumed to be real, thus the Fourier transform of the signal will be symmetric around f_c . For this reason $|X_k|^2 = |X_{N-k}|^2$ and (4.1.4) can be simplified to the following.

$$\hat{P}_k = \frac{2}{W_{ss}} |X_k|^2; k = 0, 1, 2, \dots, \frac{N}{2} \quad (4.1.7)$$

(4.1.7) takes into account the power at $k = 0$ and $k = \frac{N}{2}$, as we are regarding \hat{P}_k as the true power density (i.e. power divided by the unit frequency interval) and is thus normalised in the equation.

The statistical analysis of (4.1.7) is done in [2] to produce the following results (see App. A). The equation is split into real and imaginary parts to help with the derivation. The result is that at $k = 0, \frac{N}{2}$, the probability function is purely real.

$$p(\hat{P}_k) = \frac{1}{\sqrt{2\pi\sigma_{A_k}^2 \hat{P}_k}} e^{\left(-\frac{\hat{P}_k}{2\sigma_{A_k}^2}\right)} \quad \text{where; } k = 0, \frac{N}{2} \quad (4.1.8)$$

$p(\hat{P})$: probability function of \hat{P} .

\hat{P} : PSD estimate.

μ_k : Mean value of the PSD estimate.

(4.1.8) is used to evaluate the moments of \hat{P} for $k = 0, \frac{N}{2}$. This is done using (A.0.12), with the results shown next.

$$\sigma_{\frac{N}{2}}^2 = \sigma_0^2 = \frac{8\sigma_x^4}{N^2} \quad (4.1.9)$$

and

$$\mu_k = \langle \hat{P}_k \rangle = \frac{2\sigma_x^2}{N} \quad (4.1.10)$$

With these two results the coefficient of dispersion V_k^2 can be computed for $k = 0, \frac{N}{2}$.

$$V_k^2 = \frac{\frac{8\sigma_x^4}{N^2}}{\left(\frac{2\sigma_x^2}{N}\right)^2} = 2 \quad (4.1.11)$$

When a standard square window, $w_n = 1$ for $k = 0, 1, \dots, \frac{N}{2}$, is used, the real and imaginary parts of \hat{P} are shown to be independent. This simplifies the distribution function for $k \neq 0, \frac{N}{2}$.

$$p(\hat{P}_k) = \frac{1}{\mu_k} e^{-\frac{\hat{P}_k}{\mu_k}} \quad (4.1.12)$$

The first and second moment can now be computed quite easily.

$$\langle \hat{P}_k^i \rangle = n! \mu_k^2 \quad (4.1.13)$$

This result is then $\langle \hat{P}_k \rangle = \mu_k$ and $\langle \hat{P}_k^2 \rangle = 2\mu_k^2$, and the value of V_k can then be computed.

$$V_k = \frac{2\mu_k^2 - (\mu_k)^2}{\mu_k^2} = 1, \text{ for } k \neq 0, \frac{N}{2} \quad (4.1.14)$$

This is however not a valid result if a more complex window is used, as A_k and B_k are not statistically independent any more. The Gaussian-Bivariate function shown in (4.1.15) is necessary to compute V_k as shown in [24, p440].

$$p(A_k, B_k) = \frac{1}{2\pi\sigma_{A_k}\sigma_{B_k}\sqrt{1-\rho_k^2}} \times \exp\left[\frac{1}{1-\rho_k^2}\left(-\frac{A_k^2}{2\sigma_{A_k}^2} - \frac{B_k^2}{2\sigma_{B_k}^2} - \rho_k \frac{A_k B_k}{\sigma_{A_k}\sigma_{B_k}}\right)\right] \quad (4.1.15)$$

where

$$\rho_k = \frac{\sigma_{A_k B_k}^2}{\sigma_{A_k}\sigma_{B_k}} \quad (4.1.16)$$

The full derivation of V_k is shown in App. A, but the result is a simple equation.

$$V_k^2 = 1 + |W_{2k}|^2 \quad (4.1.17)$$

where

$$W_{2k} = \frac{1}{\sum w_n^2} \sum_{n=0}^{N-1} w_n^2 e^{-2\pi i k n / N} \quad (4.1.18)$$

W_{2k} : The normalised DFT of the squared time domain windowing function.

V_k^2 : The spectral variability

$2k$: $2f_k$, the even indexed frequency bins.

As W_{2k} is normalised, the value of $|W_{2k}|$ can only be between zero and one, and the result is that V_k^2 is bounded between one and two. For standard symmetrical windows, such as the Hamming or Blackman window, the windowing function is well behaved. The value of w_n slowly approaches zero as n reaches 0 and $\frac{N}{2}$. This causes the value of $|W_{2k}|$ to be zero for most of the values of k . It slowly climbs up to one at the edges of the window, which in turn implies that V_k is one for most of the spectrum, except at the edges. Although the effect of the windowing function is minimal, the influence is further investigated in [2].

A number of previous authors have also produced results similar to (4.1.17) in order to use frequency statistics to detect non-Gaussian signal. In the paper by Dwyer [26], frequency domain analysis is used in order to detect non-Gaussian signals in an underwater acoustic experiment and he obtains the following equation.

$$K(f_k) = \frac{\langle A_k^4 \rangle}{\langle A_k^2 \rangle^2} \quad (4.1.19)$$

$K(f_k)$: The frequency domain kurtosis estimate.

$\langle A_k^n \rangle$: The moment n about the origin, as defined in Appendix A.

The value $K(f_k)$ will be three for a Gaussian signal, except for the DC and Nyquist frequencies where the value becomes 2 as seen in (4.1.17) as well. The paper, however, only uses a standard square window and thus treats the real and imaginary part as two zero-mean variables.

[27] then defines a spectral kurtosis estimate, and is 0 for a Gaussian time domain signal. The definition given by [27] is shown in (4.1.20).

$$SK(F_k) = \frac{\langle |X_k|^4 \rangle - 2 \langle |X_k|^2 \rangle^2}{\langle |X_k|^2 \rangle^2} \quad (4.1.20)$$

$SK(F_k)$: Spectral kurtosis estimate of frequency f_k .

$\langle |X_k|^n \rangle$: The n th moment of the PSD of X_k .

We know that $|X_k|^2$ has the same PSD as \hat{P}_k , which implies a relation between the spectral kurtosis estimate (4.1.20) and (4.1.17) for all but $k = 0, \frac{N}{2}$.

$$V_k^2 = SK(f_k) + 1 \quad (4.1.21)$$

In [28], the values for $SK(f_k)$ at $k = 0, \frac{N}{2}$ are redefined to accommodate the purely real result of the spectral kurtosis estimate. The work done by Nita [2] then

combines previous work in the field of spectral statistics, as previously described in this Sec 4.1.1, in order to produce a usable model.

The theory behind spectral kurtosis is then transformed into an equation that can easily be used on hardware. To be able to do this more easily, (2.6.3) is redefined to be an estimate. As \hat{P}_k is an unbiased linear estimate of the PSD, it can directly be used to compute the kurtosis estimate.

$$\hat{V}_k^2 = \frac{\hat{\sigma}_k^2}{\hat{\mu}_k^2} \quad (4.1.22)$$

\hat{V}_k^2 : The spectral kurtosis estimate of frequency bin k .

$\hat{\sigma}_k^2$: The estimate of the variance computed from \hat{P}_k , where $\langle \hat{\sigma}_k^2 \rangle = \sigma_k^2$.

$\hat{\mu}_k$: The estimate of the mean computed from \hat{P}_k , where $\langle \hat{\mu}_k \rangle = \mu_k$.

The estimate is computed taking a number of spectral samples M , each $\frac{N}{2}$ wide. The samples form the PSD estimate \hat{P}_{ik} where $k = 0, 1, \dots, \frac{N}{2}$ and $i = 0, 2, \dots, M$. As we assume the signal x_n is a real random time domain signal, half of the frequency spectrum can be discarded due to symmetry. The values of $\hat{\mu}_k$ and $\hat{\sigma}_k^2$ are then defined as follow from [17, p351].

$$\hat{\mu}_k = \frac{S_1}{M} \quad (4.1.23)$$

$$\hat{\sigma}_k^2 = \frac{MS_2 - S_1^2}{M(M-1)} \quad (4.1.24)$$

where

$$S_1 = \sum_{i=0}^M \hat{P}_{ki} \quad (4.1.25)$$

$$S_2 = \sum_{i=1}^M (\hat{P}_{ki})^2 \quad (4.1.26)$$

The derivation of the variance of the SK estimator is done in full in App. B. The final result is shown in (4.1.27).

$$\text{Var}(\hat{V}_k^2) = \frac{4M}{(M-1)^2} (1 + 5|W_{2k}|^2 - |W_{2k}|^4 + |W_{2k}|^6) \quad (4.1.27)$$

If we take the value of $|W_{2f}|$ at $k = 0, \frac{N}{2}$ as 1 and assume most of the rest of the bins $|W_{2f}| \approx 0$, the first order approximation of the variance can be simplified quite easily.

$$\text{Var}(\hat{V}_k^2) \approx \begin{cases} \frac{24}{M} & \text{for } k = 0, \frac{N}{2} \\ \frac{4}{M} & \text{for } k = 1, 2, \dots, \frac{N-2}{2} \end{cases} \quad (4.1.28)$$

The appropriate values for M are discussed in Sec 5.1.

The computational cost of spectral kurtosis is $O(N)$ where N is the number of spectral samples M . The computation of the parameters, for the M iterations (the power per frequency bin) are already done in a running system, this makes spectral kurtosis essentially a minor calculation of $O(1)$.

4.1.2 Spectral Flatness

Spectral flatness is used to detect an information carrying signal in noise and the derivation assumes communication. The use of spectral flatness for RFI in radio astronomy will be used in reverse to ensure no information carrying signal is detected.

The derivation of spectral flatness is shown in [3] with the use of entropy. For a given probability density function $f(x)$, the entropy is defined as

$$H(x) = - \int f(x) \log f(x) dx \quad (4.1.29)$$

The entropy for 2 variables is then defined as:

$$H(x_1, x_2) = - \int f(x_1, x_2) \log f(x_1, x_2) dx_1 dx_2 \quad (4.1.30)$$

From (4.1.30) and (4.1.29), the mutual information between x_1 can be defined as [25, p760]:

$$I(x_1, x_2) = H(x_1) + H(x_2) - H(x_1, x_2) \quad (4.1.31)$$

Rewriting x_1^n as a n sample vector (x_1, x_2, \dots, x_n) , the general mutual information equation can be written as:

$$I(x_1^n) = \sum_{i=1}^n H(x_i) - H(x_1^n) \quad (4.1.32)$$

(4.1.32) defines the common information in the x_1^n vector. The marginal information can then be defined as the difference between the common information in the whole set x_1^n and the subset x_1^{n-1} .

$$\rho(x_1^n) = I(x_1^n) - I(x_1^{n-1}) \quad (4.1.33)$$

In the physical system, $\rho(x_1^n)$ will be the measure of information that is added to the system as time progresses, known as the redundancy. Redundancy is then related to entropy through (4.1.33) as follows:

$$\rho(x_1^n) = H(x_n) - H(x_n | x_1^{n-1}) \quad (4.1.34)$$

For $n \rightarrow \infty$ (4.1.34) reduces to the difference between the marginal entropy of process x and the entropy rate. The entropy rate $H_r(x)$ of a Gaussian signal with a PSD $P(\omega)$ from [29, p568] is given as:

$$H_r(x) = \frac{1}{4\pi} \int_{-\pi}^{\pi} \ln P(\omega) d\omega + \log_2 \sqrt{2\pi e} \quad (4.1.35)$$

and the marginal entropy is shown to be:

$$H(x) = \frac{1}{2} \ln \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) d\omega \right) + \log_2 \sqrt{2\pi e} \quad (4.1.36)$$

From (4.1.35) and (4.1.36) the redundancy for a Gaussian signal can be computed.

$$\rho = H(x) - H_r(x) \quad (4.1.37)$$

The spectral flatness is then defined as $e^{-2\rho}$

$$SF = e^{-2\rho} = \frac{\exp \left(\int_{-\pi}^{\pi} \ln P(\omega) d\omega \right)}{\frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) d\omega} \quad (4.1.38)$$

$\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln P(\omega) d\omega$ is then total power in the signal as integrated over frequency. For a digital signal this simplifies to $\frac{1}{N} \sum_{k=0}^{N-1} X_k$.

$$SF = \frac{\exp \left(\frac{1}{N} \sum_{k=0}^{N-1} \ln X_k \right)}{\frac{1}{N} \sum_{k=0}^{N-1} X_k} = \frac{\sqrt[N]{\prod_{k=0}^{N-1} X_k}}{\frac{1}{N} \sum_{k=0}^{N-1} X_k} \quad (4.1.39)$$

X_k : Fourier transform of the signal x_n with frequency bin k

SF : Spectral Flatness of signal X .

N : Size of the FFT.

For a real signal only $\frac{N}{2}$ will be used, because of the symmetry in the frequency spectrum. The spectral flatness is then the geometric mean of the power spectrum divided by the arithmetic mean. This will be 1 for a pure Gaussian signal.

The computational cost of spectral flatness is $O(N)$, where N the size is of the FFT.

4.2 Offline

The offline methods are in general more computationally expensive and used on data that has been compressed in some form or another.

4.2.1 Sum Threshold

In the online RFI detection methods, the statistical shape of the probability distribution function is investigated to determine if RFI is detected. This can be done in real-time on the incoming samples. An alternative is to look at recorded samples and filter RFI out by determining whether the power in a region is disproportionately high. This process can be done post-correlation. This is the basic idea for the sum threshold method as described by [4]. The method is run on a matrix representation of the recorded values of the correlated visibility $V(f, t)$ as shown in (4.2.1).

$$V(f, t) = \begin{pmatrix} v_{f,1} & v_{f,2} & \cdots & v_{f,t} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,0} & v_{1,1} & \cdots & v_{1,t} \\ v_{0,0} & v_{0,1} & \cdots & v_{0,t} \end{pmatrix} \quad (4.2.1)$$

Sum thresholding uses a combinatorial thresholding technique to determine whether the data sample contains RFI or not. This is done by looking at each sample and determining if it is above a threshold χ_1 and if the threshold is triggered, the bin becomes flagged. A second pass is then made with a lower threshold of χ_2 , but now looking at two adjacent bins. When both bins exceed the threshold both bins are flagged, but if only one of the bins exceed the threshold, no bins are flagged. A third pass through is made with an even lower threshold χ_3 , now looking at three adjacent bins and so on.

This algorithm is run on both the frequencies as well as time axis. The number of iterations pre-set to N , with each iterations i threshold as $\{\chi_i\}_{i=1}^N$.

For an example of the basic algorithm, look at the following data set.

$$V_{exp} = \begin{pmatrix} 5 & 3 & 2 & 7 \\ 7 & 4 & 9 & 7 \\ 5 & 4 & 2 & 8 \end{pmatrix} \quad (4.2.2)$$

Assume $\chi_1 = 8$ and $\chi_2 = 6$. The flag matrix of the sample set will then look as follows, with each flag shown with appropriate flagging.

$$\text{Flag}_{V_{exp}} = \begin{pmatrix} 0 & 0 & 0 & f_{\chi_2} \\ 0 & 0 & f_{\chi_1} & f_{\chi_2} \\ 0 & 0 & 0 & f_{\chi_2} \end{pmatrix} \quad (4.2.3)$$

From this limited characterisation of the algorithm, VarThreshold can be fully defined. The full definition requires a number of parameters to be explained in more detail, including:

- The subsets 1 to N to compare.
- The threshold for $\{\chi_i\}_{i=1}^N$.
- The false positive rate of the algorithm.

The threshold itself is only loosely described, which makes implementation difficult. A set formula is defined in [4]

$$\chi_i = \frac{\chi_1}{\rho^{\log_2 i}} \quad (4.2.4)$$

and from empirical evidence $\rho = 1.5$ is found work for VarThreshold and the later defined SumThreshold. The start value χ_1 will have to change for each situation, depending on the power of the RFI as well as the observing signal. This exponential function works well as it starts off by flagging strong RFI and can then filter out broadband RFI on the higher iterations without removing too much usable data. This is a major problem in the higher iterations, as more samples are flagged on each turn.

The full definition of each search subset, for either a search with constant time or constant frequency is

$$\mathcal{M} = \{i \in \mathbb{Z} : 1 \leq i \leq \max(N_f, N_t)\} \quad (4.2.5)$$

N_f : The maximum number of frequency bins.

N_t : The maximum number of time domain bins

Although it is not specified, the number of iterations is also something to consider, though after five iterations diminishing returns are seen.

To compute the false positive ratio, a number of assumptions are made. The residual R of the complex correlated visibility V and surface fit \hat{V} is related to a random distortion $D(\sigma_{N_s})$ with the average power of σ_{N_s} . The probability of a non-contaminated bin from the residual R flagged as RFI, by being larger the χ , is given in (4.2.6).

$$\forall f \forall t : P(|R(f, t)| \geq \chi) = \int_{-\infty}^{-\chi} \varphi_{\sigma}(x) dx + \int_{\chi}^{\infty} \varphi_{\sigma}(x) dx \quad (4.2.6)$$

$P(|R(f, t)| \geq \chi)$: The probability of the residual being flag as RFI.

$\varphi(x)$: PSD of the random distortion $D(\sigma_{N_s})$.

The answer is computed numerically to simplify the solution and a value for $M = 2$ is chosen. As the input signal of uncontaminated data is assumed to be Gaussian noise, the two samples $R(f_1, t_1)$ and $R(f_2, t_2)$ are statistically independent. The probability of a false positive can the be computed using (4.2.7) with all the neighbours $R_{2...5}$.

$$\begin{aligned} P_{\text{false}}(f, t) &= P(\text{flag}_{fM=2}(f, t) \vee \text{flag}_{tM=2}(f, t)) \\ &= P(|R_1| > \chi \wedge \exists i \in [2...5] : |R_i| > \chi) \\ &= P(|R| > \chi) - P(|R| > \chi)(1 - P(|R| > \chi))^4 \end{aligned} \quad (4.2.7)$$

With larger sample sizes, the result becomes a lot more complex.

An improvement can be made to the VarThreshold method to more accurately flag RFI. The basic algorithm is still the same as in VarThreshold, except for the comparison made. In SumThreshold, the sum of the samples are compared instead of looking at each sample individually.

$$\sum_{i=1}^{\mathcal{N}} \mathcal{M}_{\mathcal{N}} > \mathcal{N} \frac{\chi_1}{\rho^{\log_2 \mathcal{N}}}, \text{ for } \mathcal{N} = 1, 2, \dots, N \quad (4.2.8)$$

The problem occurs when a strong RFI signal is introduced to a sample set. As an example, look at the following sample set $[0, 9, 7, 0, 0, 0]$ with $\chi_1 = 7$. This implies a value for $\chi_6 \simeq 2.45$ using (4.2.4) with $\rho = 1.5$. The problem is that the sum of all the values in the sample set is 16, while $6\chi_6 = 14.7$. This will flag all six values as RFI even though there are clearly only 2 bins that are RFI contaminated. The solution is to leave out the previously flagged data and replace it with the current RFI threshold value. For the sixth iteration of the sample set, this will become $(0 + \chi_6 + \chi_6 + 0 + 0 + 0) = 2\chi_6$. The only flagged data will then be 9 and 7.

For the false positive ratio of the SumThreshold method for $M = 2$, the probability of two random variables being larger than χ are given in (4.2.9).

$$\begin{aligned} \forall f_1 f_2 t_1 t_2 : P(T_{\chi,1,2}) &= P[R(f_1, t_1) + R(f_2, t_2) \geq \chi] \\ &= P(\mathcal{D}(2\sigma_{N_s}) \geq \chi) \\ &= \int_{\chi}^{\infty} \phi_{2\sigma}(x) dx \end{aligned} \quad (4.2.9)$$

Assuming the number of time bins are equal to the number of frequency bins, N , the computational cost of both Var- and SumThreshold are $O(N^2)$.

4.2.2 Basic Pulsar RFI Mitigation Techniques

There are two basic techniques to either deal with narrowband RFI or broadband RFI; these are part of PSRchive and described in [30]. For the narrowband RFI mitigation, the algorithm is quite simple. Compute a median-smooth spectrum by looking at a number of adjacent frequency bins. This window is swept over the full frequency spectrum and the median is computed for each bin to form the median-smoothed spectrum as shown in Fig 4.1 and 4.2. The default window size of PSRchive is 21 frequency bins. Compute the standard deviation using the original spectrum, whilst ignoring previously flagged bins. Compare each bin, with the equivalent media-smoothed bin. If a bin differs by more than 4 times the standard deviation, null that bin. Once done with the whole spectrum, check if any bins were flagged. If bins where flagged, recompute the standard deviation leaving out the newly flagged bins and sweep the spectrum again. Repeat this process until no bins are flagged. The value of the window size as well as the tolerance of 4 times the deviation can be adjusted for each situation. The flagged bins are usually zeroed as the loss of a small frequency band does not effect the final results greatly, especially if the bandwidth is much larger than the individual frequency bins.

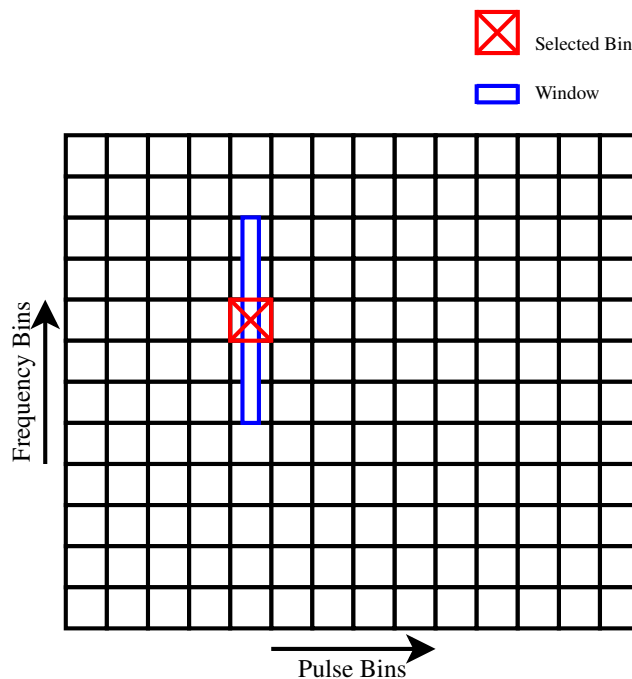


Figure 4.1: Example of the narrowband median-smoothing.

The idea of the broadband RFI mitigation is not to throw away a lot of data because of corruption caused by, for example, a lightning strike. To start off, each phase for the pulse is integrated for all frequencies to return the total flux for each phase bin. This is done before the signal is de-dispersed. The median windowing is again done, but on the total flux of the phase bins as shown in Fig. 4.3. This

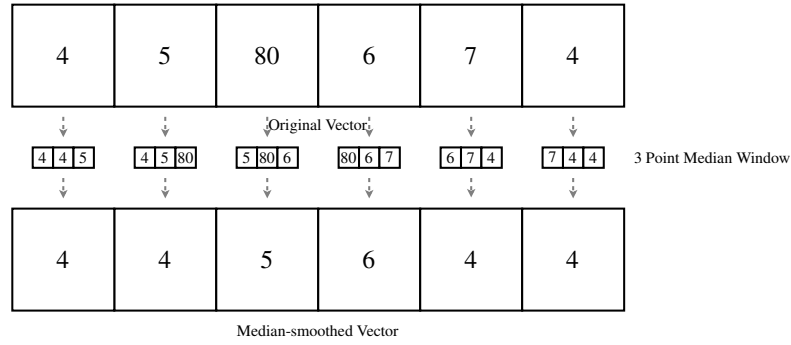


Figure 4.2: Example of the median-smoothing of a vector.

time, however, the median absolute deviation is computed recursively instead of the standard deviation, as it adds robustness to the filter. After all of the affected phase bins are flagged, each frequency bin for each phase bin is corrected independently. The value of each frequency bin of a flagged phase is set to the locally computed median plus uncorrupted noise. The noise is taken as the difference between the observed pulse profile and the local median of a randomly selected un-flagged phase bin.

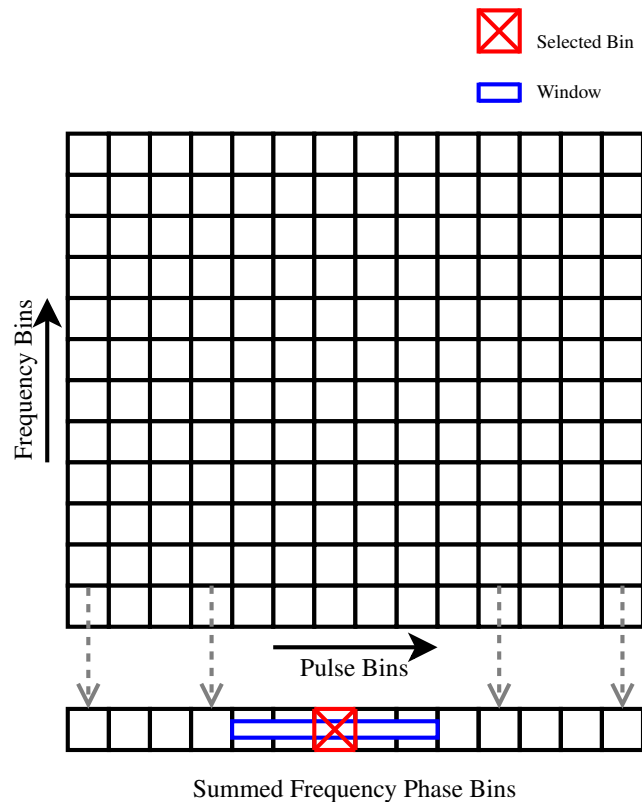


Figure 4.3: Example of the broadband RFI mitigation, frequency integration and phase bin windowing.

4.3 Other Known Methods

The following methods are also used for RFI mitigation, but not studied in this thesis. A brief summary of each method will be given.

4.3.1 MAD - Median Absolute Deviation

A real-time RFI mitigation system is implemented at the GMRT in Pune India that makes use of MAD to detect RFI [31]. MAD is more robust than looking at the standard deviation of a signal, as it can handle outliers more easily. With the system at the GMRT, the median absolute deviation is computed for the spectrum after it has been Fourier transformed. The strong sources can then be flagged if the frequency bin is more than the given threshold from the computed MAD value, similar to the pulsar technique shown in Sec. 4.2.2. The threshold is set as the median plus N times the standard deviation σ , where N is chosen by the user. The values of the flagged bins are then replaced with Gaussian noise with the same power as signal background noise.

4.3.2 Active RFI Cancellation

An alternative to using statistics to determine if a signal contains RFI, is to use an active RFI cancellation method. This is shown in [32] where an extra antenna is pointed to a known source for RFI cancellation. This method is very suitable when the source is known, but obviously of limited value when this is not the case. The standard method makes use of a simple reference antenna, such as a Yagi-Uda or Log-Periodic, that is pointed at the source of the RFI. The signal for the telescope, which contains an astronomical signal as well as RFI, is then cross correlated with the reference signal to determine the power of the RFI. The gain of the reference signal is then adjusted to null RFI from the telescope if subtracted, as shown in Fig. 4.4.

The gain and phase of the RFI can be extracted from the covariance matrix of the interferometer. In the setup of [32], 10 telescopes are used with 2 reference antennas which results in a 12×12 covariance matrix \mathbf{C} . Eigen decomposition is then done on \mathbf{C} to form $\mathbf{C} = \mathbf{G}\mathbf{P}\mathbf{G}^H$, where \mathbf{G} then forms the gain matrix and \mathbf{P} , the Eigen matrix that represents the power from each source. The power of the RFI source in \mathbf{P} is then not completely zeroed out, but only partially as a zeroed matrix could potentially distort weak celestial signal.

4.4 Conclusion

Both the spectral flatness and spectral kurtosis algorithms are simple to compute. This allows for these methods to be implemented in real-time, as the algorithms will not cause a bottleneck. The Var/SumThreshold as well as the pulsar RFI mitigation

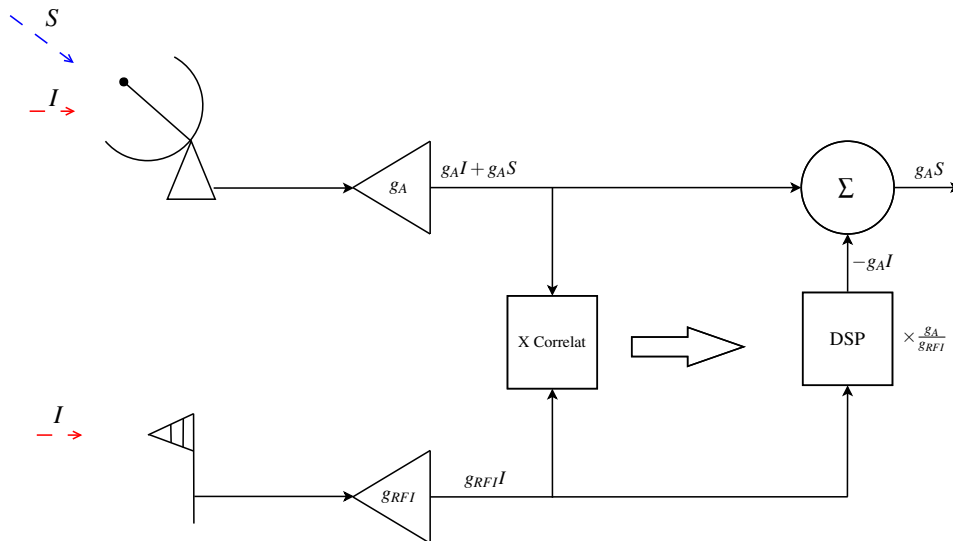


Figure 4.4: Example of the active RFI cancellation method used by [32], with I as the interference and S as the celestial signal.

techniques require a number of iterations to flag RFI. This does not allow for the methods to be implemented on real-time data, but only on the compressed post correlation data.

Chapter 5

Modelling

Before the algorithms are tested on physical system, the theory discussed in chapter 4 will be simulated using tools such as Matlab, SciPy and Simulink. The following chapter will look at most of the simulations of the techniques discussed. The cases that are looked at are:

- Continuous narrowband RFI, as seen on TV broadcast.
- Random narrowband RFI, such as cellphone transmissions.
- Broadband bursts, such as sparks and lightning strikes.

5.1 Spectral Kurtosis

The simulation of Spectral Kurtosis as described in Sec. 4.1.1 is done in Matlab. The cases that are simulated using Matlab are:

- A single tone, varied in amplitude
- A single tone, varied in duty cycle.
- A single tone, varied in M .
- A broadband signal, varied in duty cycle.

For the varied duty cycle, M amount of spectra are summed to produce the PSD estimate. This is done for each change in duty cycle and the resulting waterfall plot can be seen in Fig. 5.1. All of the simulations are done with $M = 1000$ and a window size of 512, but the minimum for M is shown in [2] for stationary deterministic signals. The equation given is

$$M(\eta_k) \geq 36\left(1 + \frac{1}{\eta_k}\right)^4 \quad (5.1.1)$$

M : The number of spectra that are summed to form the PSD estimate.

η_k : The signal to noise ratio of the RFI to celestial signal.

From (5.1.1) we can see that no RFI will be detected when M is made less than 36, irrespective of the signal to noise ratio. It also implies that M must be greater or equal to 576 in order to detect RFI with the same power as the celestial signal. To accurately see what is happening at the tone frequency, a slice of the waterfall plot is shown in Fig. 5.2. The signal to noise ratio for references at $M = 1000$ is then:

$$\eta_k = \frac{1}{\sqrt[4]{\frac{M}{36} - 1}} \simeq 0.772 \quad (5.1.2)$$

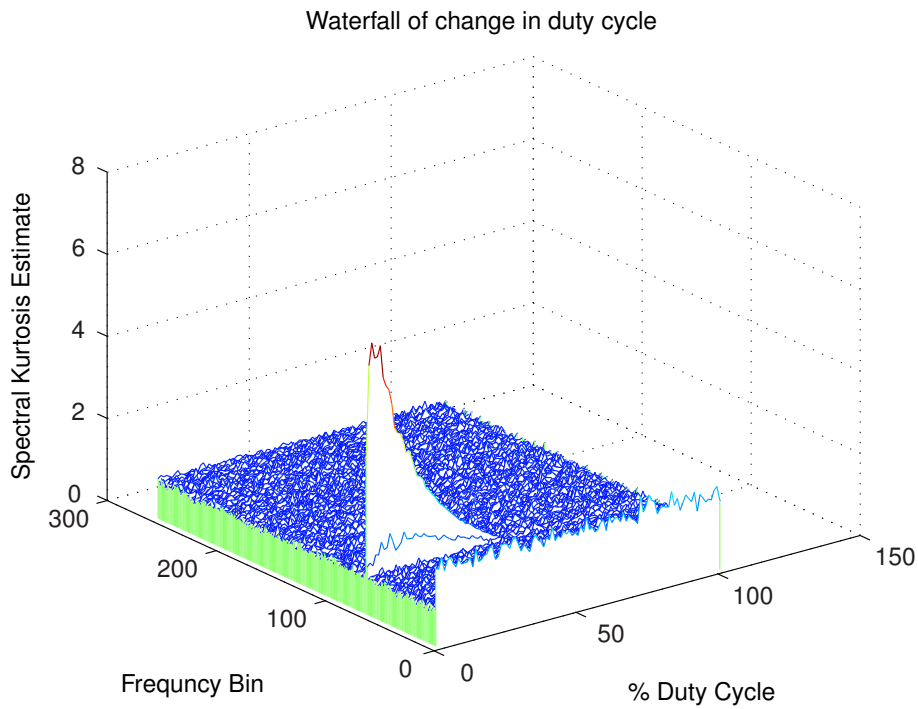


Figure 5.1: A Waterfall plot of the spectral kurtosis of a varied duty cycle sine wave.

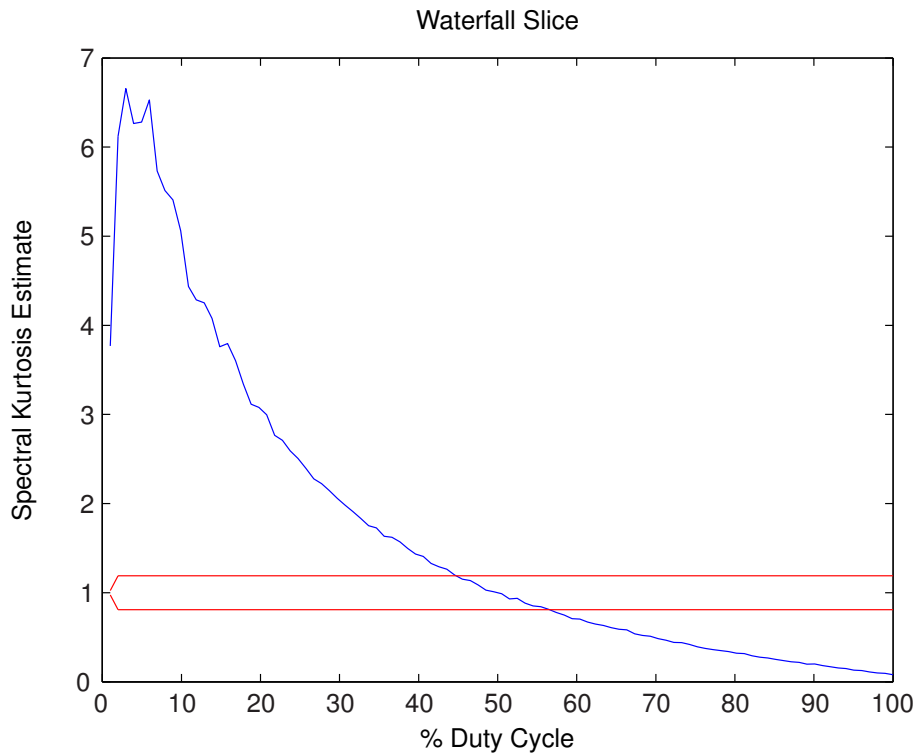


Figure 5.2: A slice of the waterfall plot at the RFI frequency.

The amplitude of the tone is also varied, to see the effectiveness of spectral kurtosis on different signal to noise levels. This can be seen in Fig. 5.3. The signal is a random Gaussian noise source, similar to the electromagnetic signal from a radiating black body. The continuous wave tone is used as the corrupting noise. The upper and lower references, indicated by the blue lines, are set by (4.1.28). It is clear to see in all of the above simulations that, for a 50% duty cycle sine wave, the spectral kurtosis estimate is 1. This is always true, irrespective of the power of the contaminating RFI signal and thus masks the RFI.

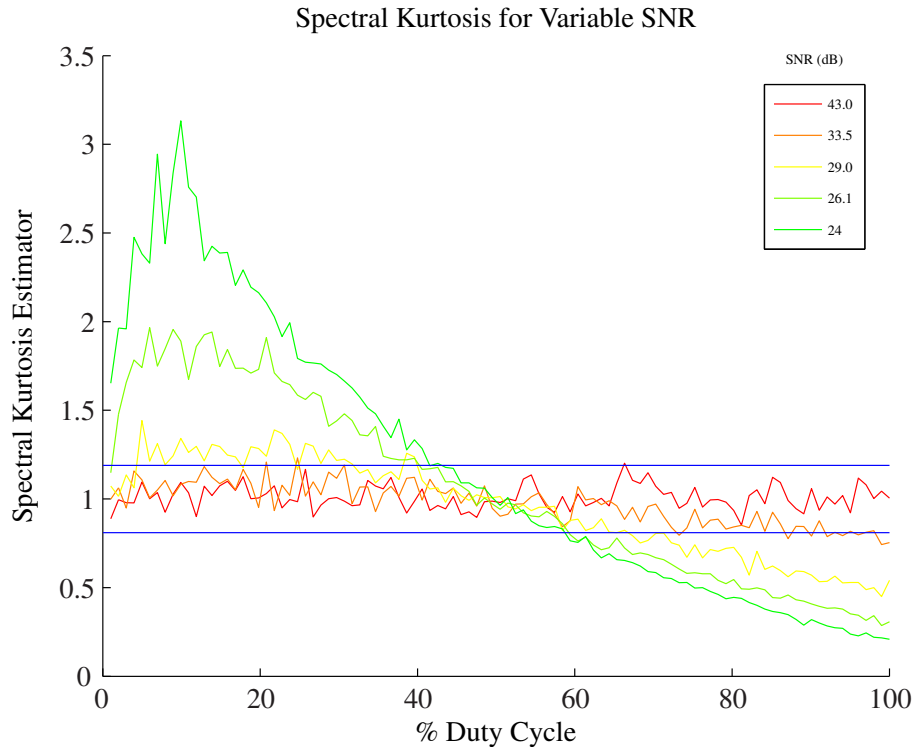


Figure 5.3: A sine wave varied in amplitude with the duty cycle of the wave changed from 0% to 100%.

The effect of a broadband signal will also be looked at. The sample PSD of the spread spectrum signal is shown in figure 5.4. The spread spectrum used for the spectral kurtosis estimate is generated using the code of [33].

The duty cycle of the spread spectrum signal is also varied from 0% to 100% with the power in the signal 10dB greater than the noise. The waterfall plot is shown in figure 5.5. The DC offset seen in figure 5.1 is removed in the waterfall. The effect of the DC value can be neglected in the analysis as the roll-off of the KatADC is 30MHz before DC, and the spectrum near bin 0 is ignored.

To see the effects of the spread spectrum signal at the limits of the duty cycle, a slice is taken from the waterfall plot at 1% duty cycle and at 100% duty cycle. This is shown in Fig. 5.6 and 5.7 respectively. The signal to noise ratio of the spread spectrum signal at 1% duty cycle is 10 dB, but the maximum value of the kurtosis estimate for this signal is 2.1. This can be seen at Fig. 5.6 at frequency bin 0.14. The spectral kurtosis estimate of a single tone at a duty cycle of 1% is shown in Fig. 5.3. The spectral kurtosis estimate of the tone at a duty cycle of 1% is 3.4. The spectral kurtosis estimate is larger for the weaker signal tone RFI, than for the stronger spread spectrum signal. This is due to the fact that the power is dispersed in frequency and not concentrated at a single tone. The broadband noise generated in this instance is an extreme case, as the bandwidth usage of this example for the KatADC will be the equivalent of a 200MHz spread spectrum signal.

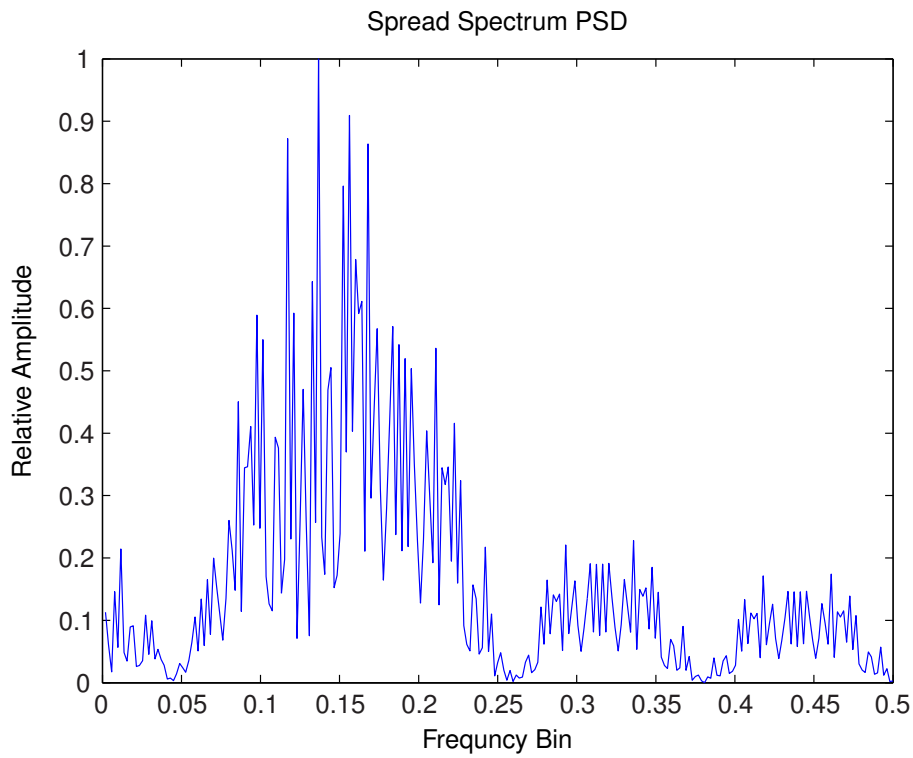


Figure 5.4: The PSD of the spread spectrum used for the spectral kurtosis estimate.

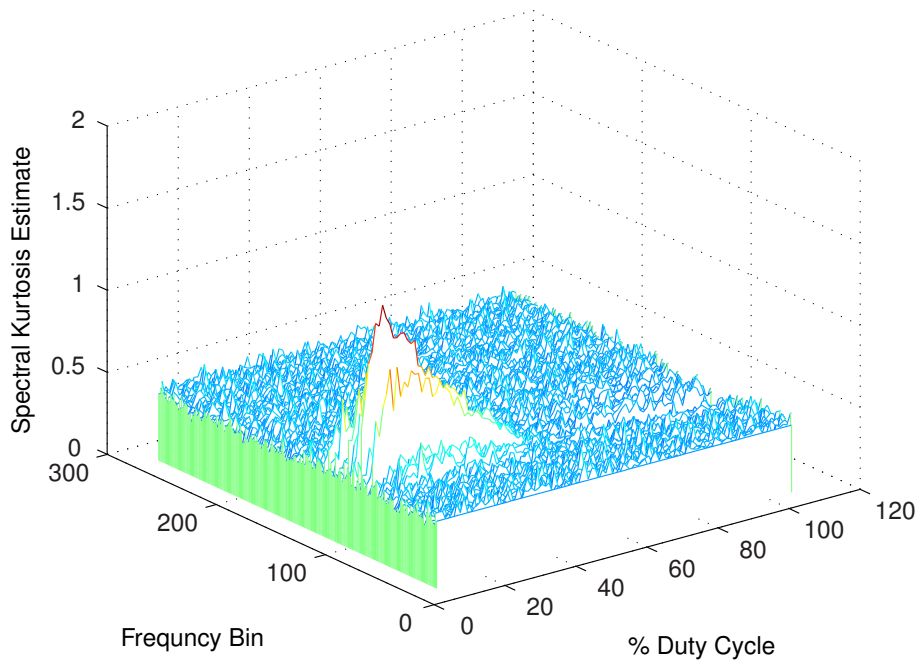


Figure 5.5: The spectral kurtosis estimate of a broadband signal, varied in duty cycle.

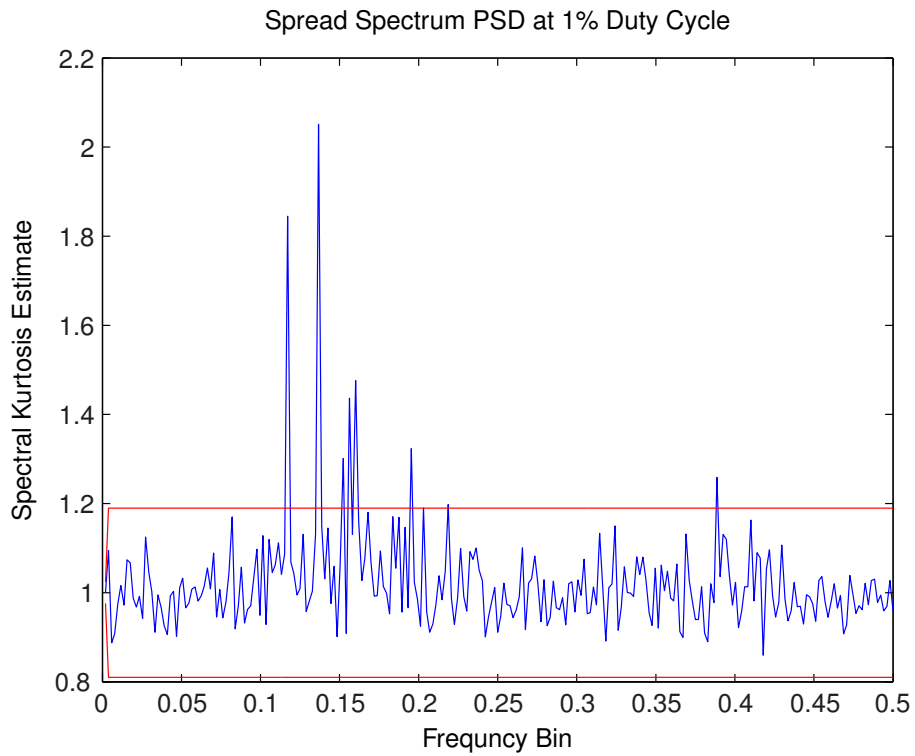


Figure 5.6: The spectral kurtosis estimate of a broadband signal at 1% duty cycle

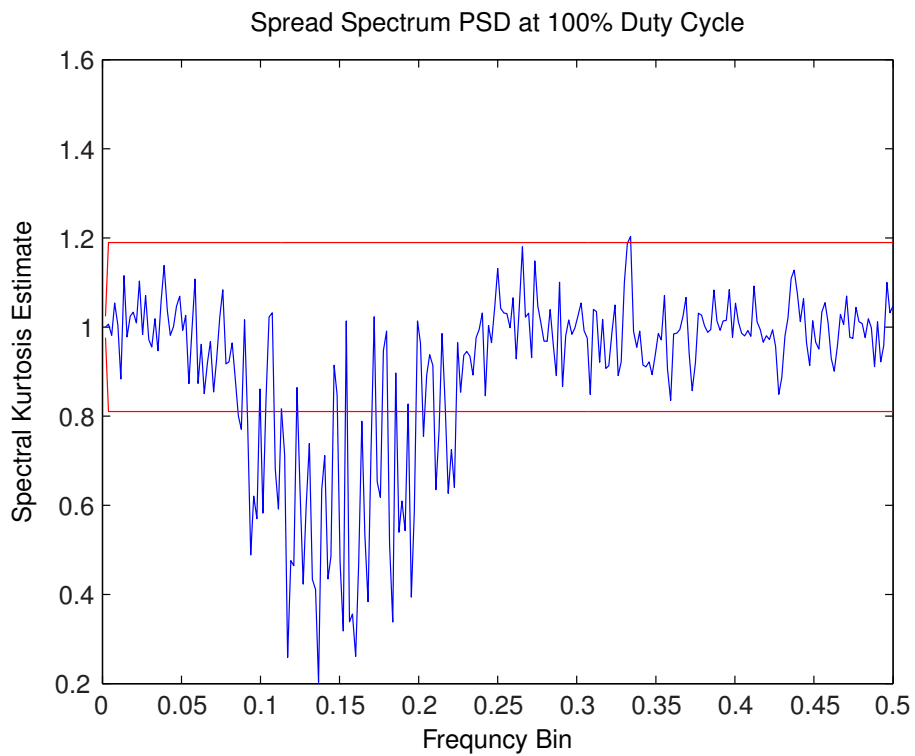


Figure 5.7: The spectral kurtosis estimate of a broadband signal at 100% duty cycle.

5.2 Spectral Flatness

The simulation of Spectral Flatness is only done in Matlab as the FPGA on the ROACH 1 does not support floating point arithmetic. This is due to the fact that the calculations of the spectral flatness makes use of logarithms. This is done in the time domain, per time sample as well as in the frequency domain per frequency bin.

The spectral flatness is tested on a single tone as well as a broadband signal. The spectrum with noise generated is shown in Fig. 5.8. The tone is on a single frequency for all time samples. The broadband noise is Gaussian noise added to the spectrum, with a bandwidth of 0.56 of the full spectrum. The broadband noise is shaped using a hamming window for a slower increase in power. The shaping is done in time and frequency. The total power per frequency bin is computed over all time bins of the waterfall plot and shown in Fig. 5.9. The tone is visible at frequency bin 0.3, but the broadband noise disappears in the background noise.

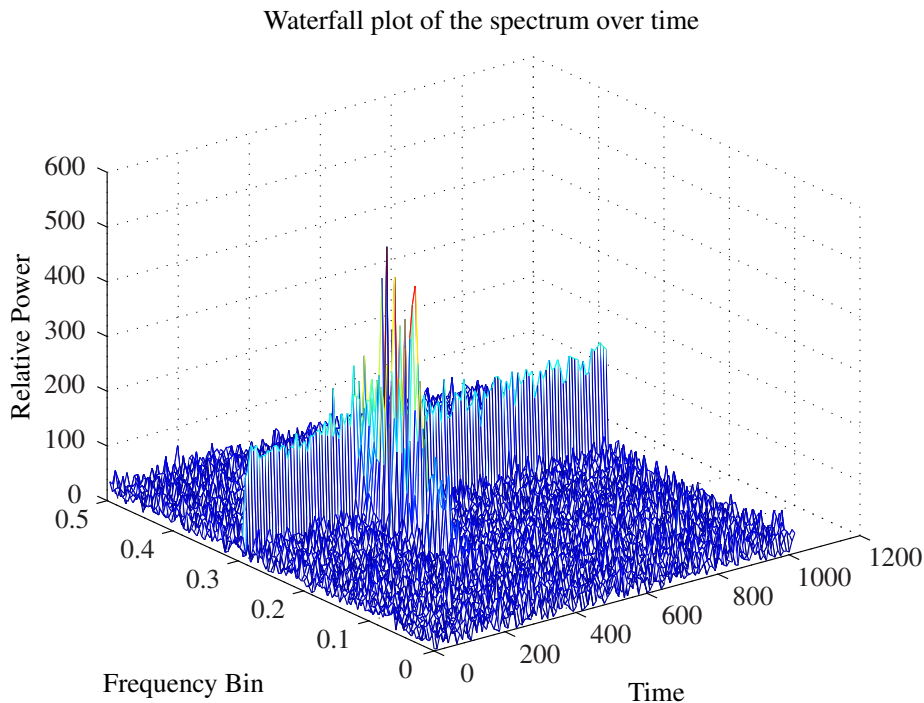


Figure 5.8: A waterfall plot of the test spectrum used.

The power of the spectrum is then computed and the spectral flatness of each frequency bin is computed. This is done by calculating the arithmetic mean of each frequency bin, as well as the geometric mean. The values are then divided for each frequency channel. The result is shown in Fig. 5.10. The broadband Gaussian noise is shown in the red rectangle and the single tone is visible between the Gaussian noise.

The same procedure is followed as for Fig. 5.10, but for all time samples. The geometric and arithmetic mean is computed for each time sample, i.e. for each

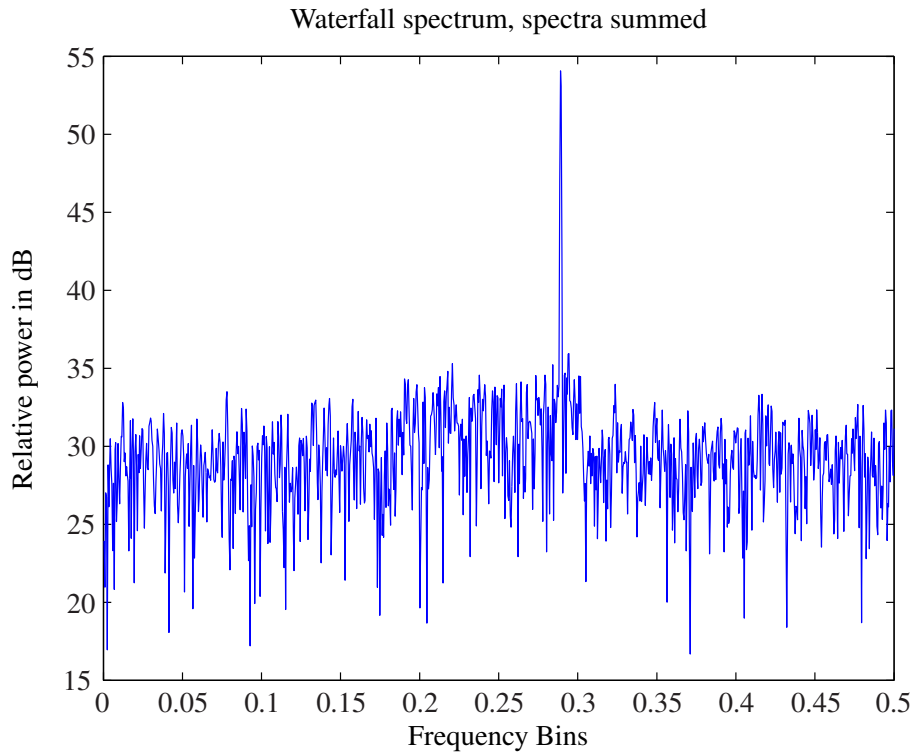


Figure 5.9: The total power in each frequency channel when summed for all time.

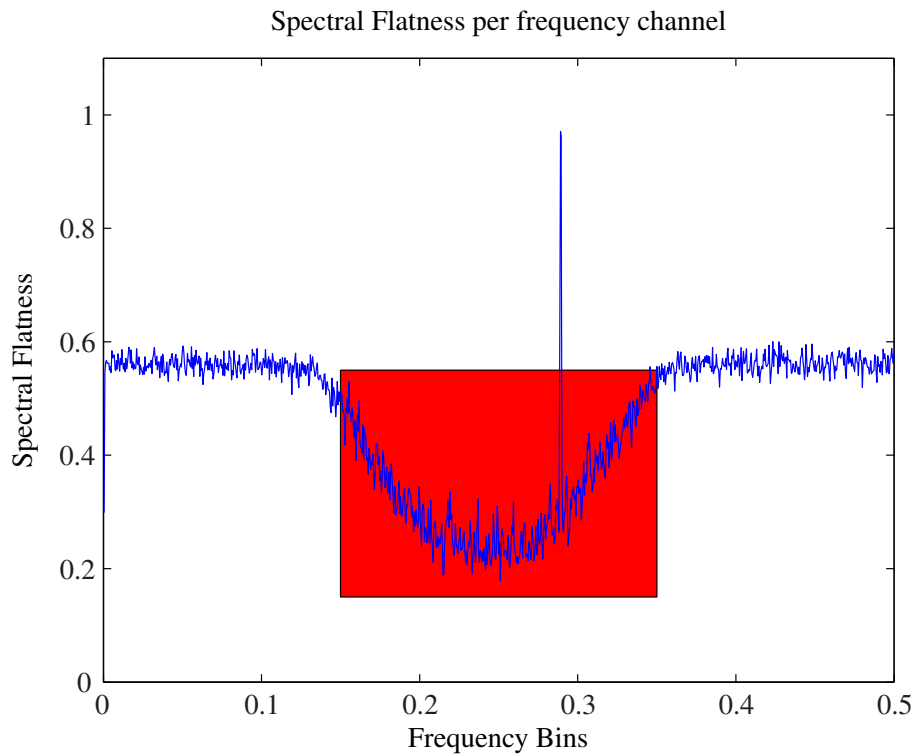


Figure 5.10: The Spectral Flatness for each frequency channel.

spectrum. The result is that the spectral flatness is shown for each time sample and not per frequency bin. The power of each time sample is shown in Fig. 5.11 and the spectral flatness result is shown in Fig. 5.12. The single tone will not be visible in the total power graph and the broadband noise is just visible at time bin 400. The the Spectral Flatness of the time domain shows the broadband noise clearly between bins 300 and 400. It also has an offset from one for all time bins because of the tone.

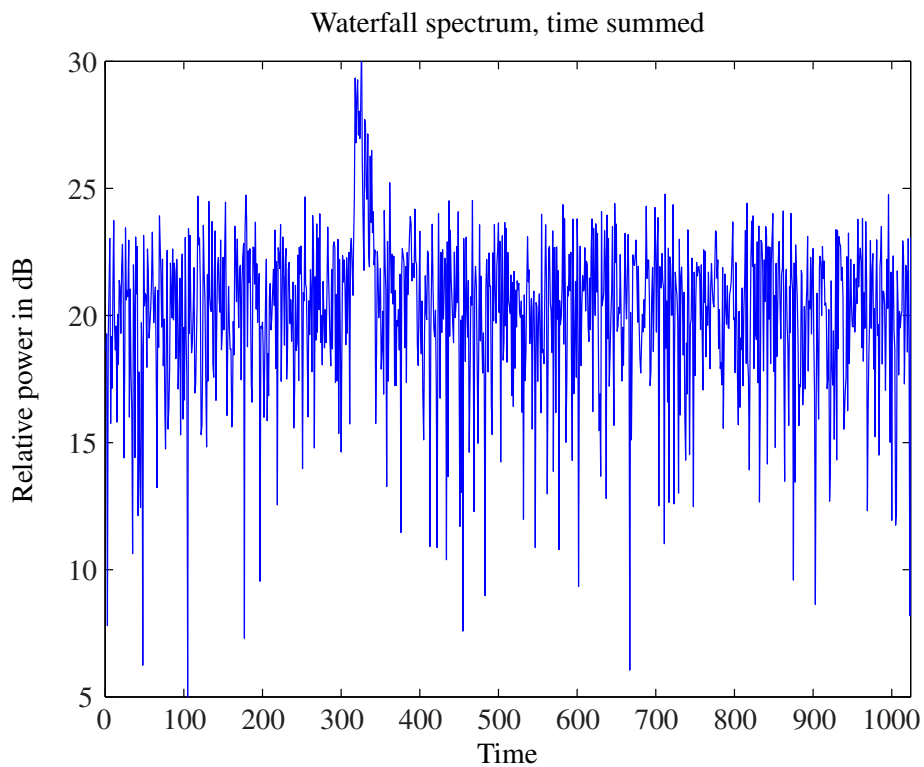


Figure 5.11: The total power in each time bin for the whole frequency band.

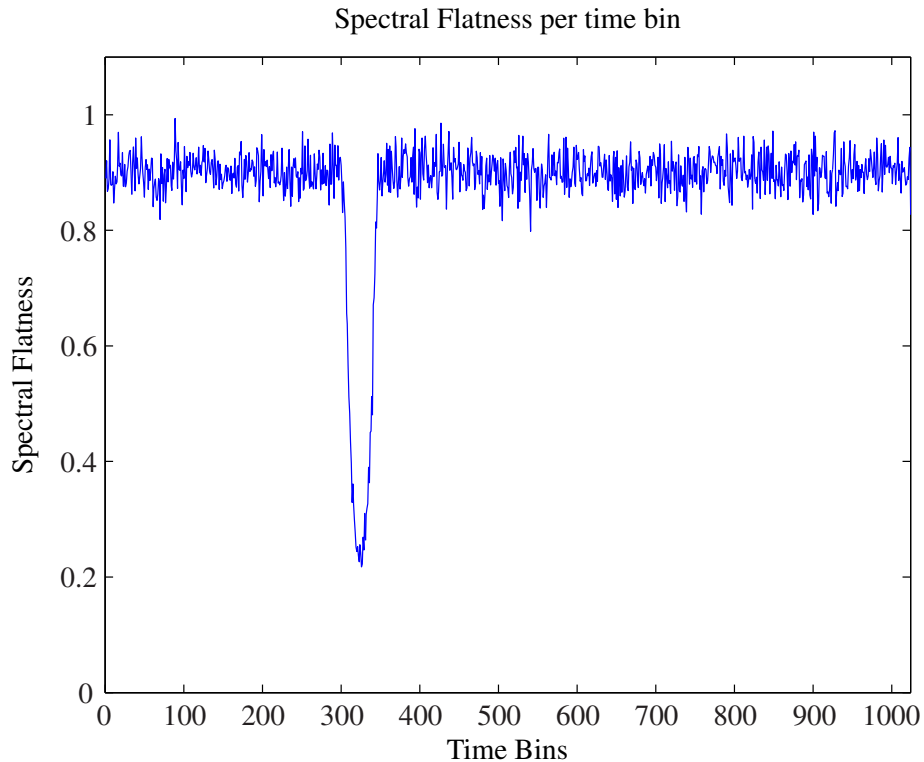


Figure 5.12: The Spectral Flatness for each time bin, for the whole frequency band.

5.3 SumThreshold

For both the VarThreshold and SumThreshold, the simulated RFI generated in Sec. 5.2 for spectral flatness is used again. The simulations are done using Matlab to generate the data and test the algorithm. The spectrum for the input signal over time is again displayed using a waterfall plot. This is shown in Fig. 5.13. There is again a single tone, as well as a broadband noise source in the generated spectrum.

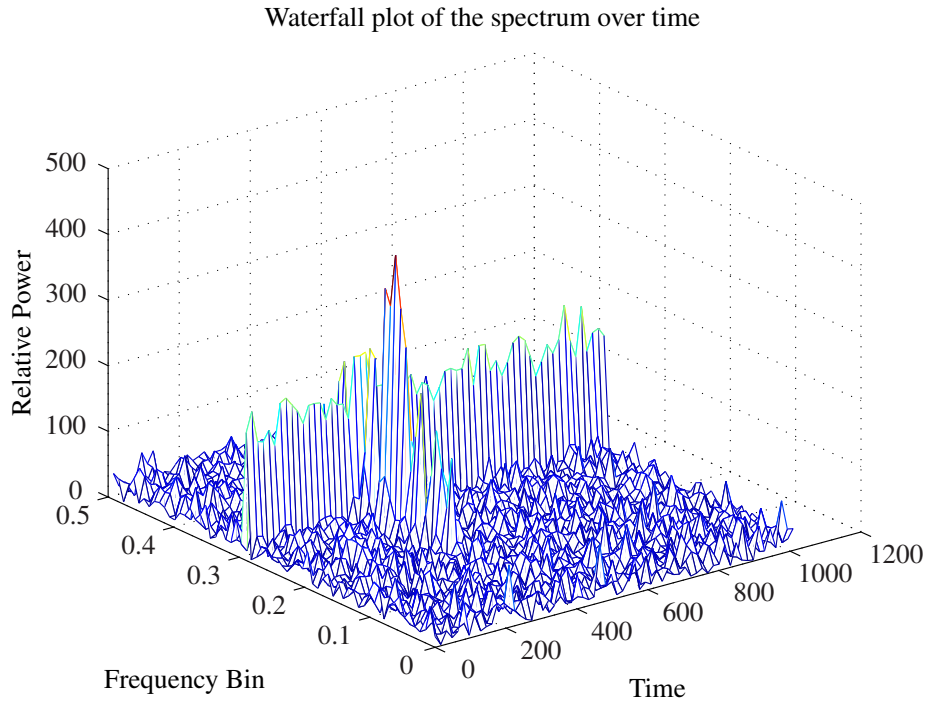


Figure 5.13: The waterfall plot of the spectrum used for the VarThreshold and SumThreshold.

The first method that is tested is VarThreshold and the threshold value for χ_1 is set empirically to $\chi_1 = 97$. The resulting graph for the threshold is shown in Fig. 5.14. The white areas indicate flagged bins while the black areas indicate unflagged bins. The tone is visible at frequency bin 0.3 and the broadband burst is at time 300, between frequency bins 0.1 and 0.4. There are also a number of false positives in the resulting graph. The more stable SumThreshold method is used on the same

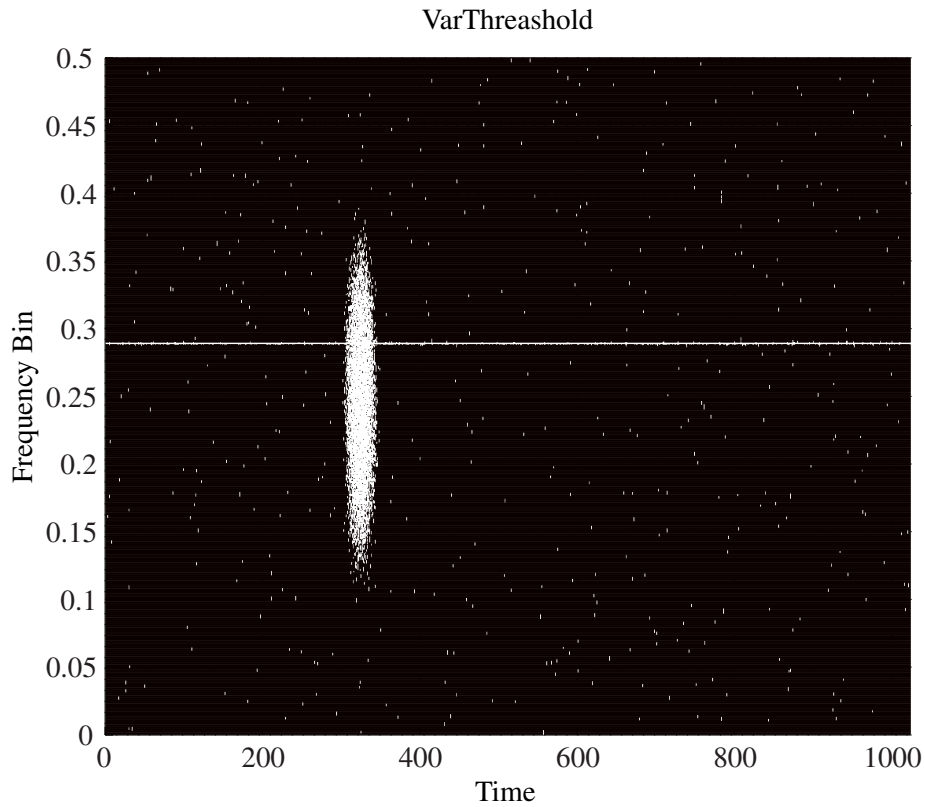


Figure 5.14: The waterfall plot of the spectrum as flagged by VarThreshold.

data as the VarThreshold method. The first threshold is set to $\chi_1 = 120$ as it was found that the SumThreshold method is much more sensitive than the VarThreshold method. The threshold was set until the resulting number of false positives was equal to that of the VarThreshold method. The main focus was on the edge cases at the tone, as well as the broadband noise. The result is shown in Fig. 5.15.

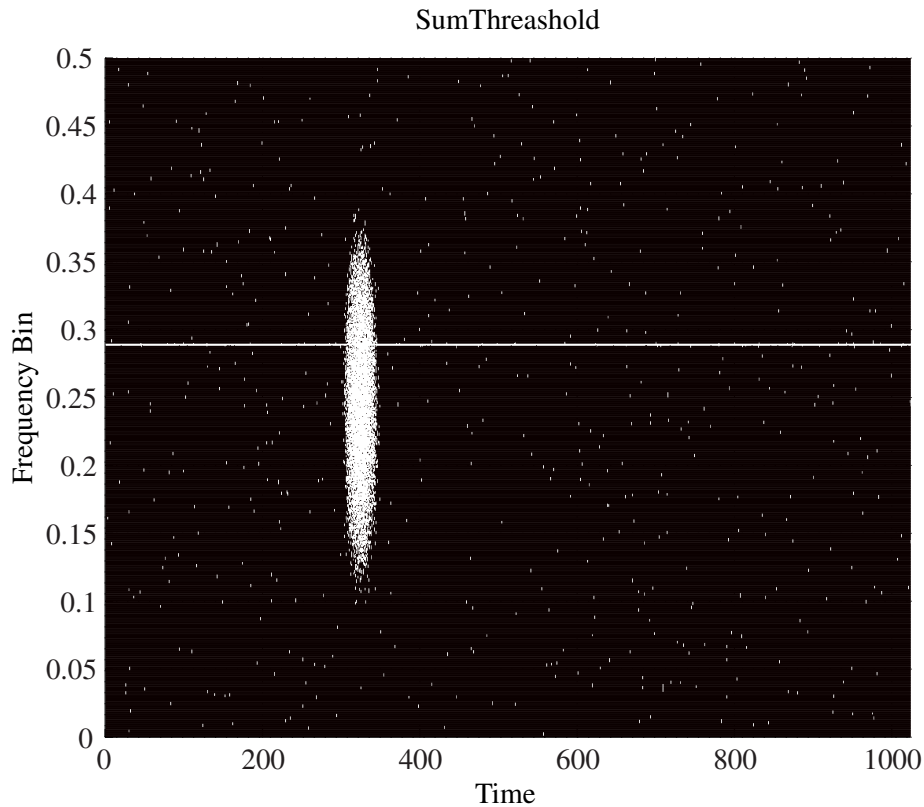


Figure 5.15: The waterfall plot of the spectrum as flagged by SumThreshold.

To more clearly see the difference between the two methods, the flag map of the VarThreshold method is subtracted for the SumThreshold flag map. The graph has 3 colours to indicate the difference between the methods, and shown in Fig. 5.16. The red indicates no difference between the graphs, the white a flag only present in SumThreshold and the black, a flag only present in the VarThreshold. It is clear to see that for the same false positive ratio for VarThreshold as for SumThreshold, the SumThreshold more accurately detects the tone as well as the edge cases of the broadband noise.

5.4 Conclusion

The computational time for both the spectral kurtosis and spectral flatness is significantly less than the Var/SumThreshold method. Var/SumThreshold does, however, allow for a much more selective flagging of RFI as a whole frequency bin for all time is not removed but only the affected bins. The thresholding method is also not blind to certain signal types such a 50% duty cycle tone, but flags all data over the threshold.

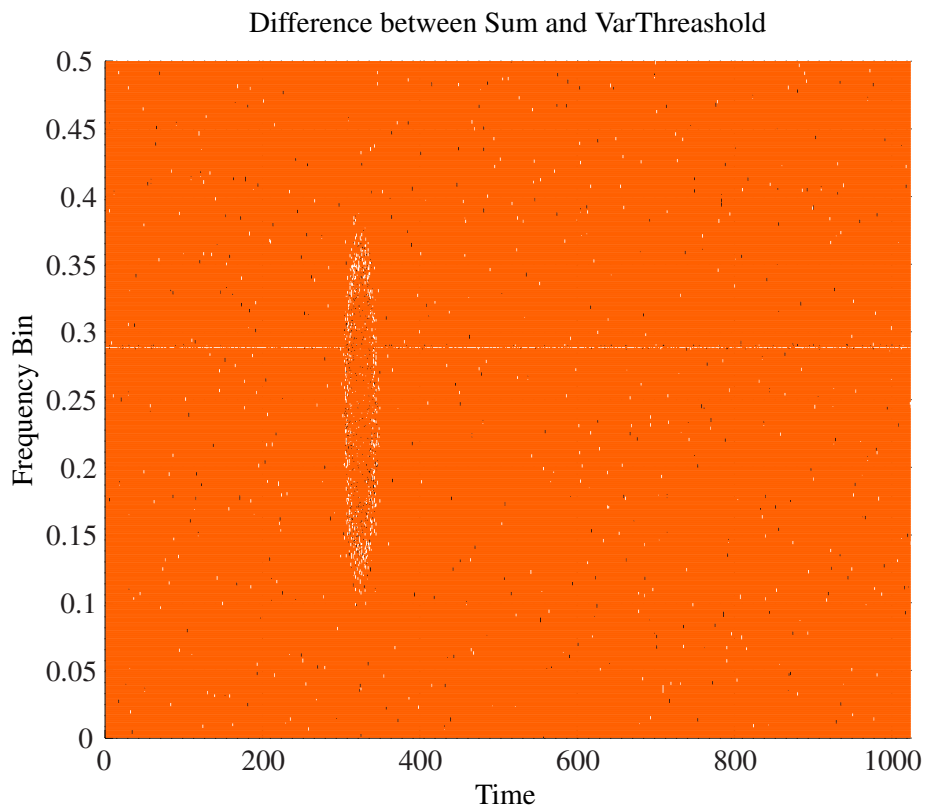


Figure 5.16: The waterfall plot of the difference between the VarThreshold and SumThreshold.

Chapter 6

Measurement Setup

This chapter will focus on the setup of each final measurement and testing of the algorithms. The Spectral Kurtosis setup will include a Simulink model that can be validated using a Simulink simulation as well as provided data. The pulsar RFI mitigation techniques will be run on PSRchive directly using pulsar data. The SumThreshold and VarThreshold methods as well as Spectral Flatness method will run on provided data using Python.

The data sets used to test spectral kurtosis, spectral flatness, VarThreshold and SumThreshold are KAT7 data provided by [34], interferometer measurements done by JP van Rensburg [35] and measurements done using a radiometer implementation.

The KAT7 data is packaged in an HDF5 binary file which contains the complex values of the spectrum of each baseline, for a number of time samples. The header of the binary file contains the following data:

- Time interval of measurement.
- Bandwidth of the observation.
- Number of frequency channels.
- Channel frequency width.

The data obtained by [35] derived from a 2-element interferometer described in that thesis. The sun was used as a test source and allowed to pass before the focus of the two 3m dishes in drifts-scan mode. The distance between these dishes was changed after each measurement to produce more baselines for the formation of the visibility function. The measurements were done at the Faculty of Engineering at Stellenbosch University. As this is an urban area, the measurement were contaminated by strong RFI sources and allows strenuous testing of the RFI mitigation techniques. The RFI environment plots shown in [35] indicate the extent of the RFI contamination on the data. The interference fringes will be used to calculate the signal-to-noise ratio of the filtered data, in order to compare the effectiveness of the algorithms.

The signal-to-noise ratio is calculated, using the raw interference fringe plot as reference. A narrow bandpass filter, with the same frequency as the interference fringes, is generated in Matlab. The RFI contaminated interference fringe data is passed through the bandpass filter to produce the clean signal. The power from the clean signal is used as the signal reference for the power-to-noise ratio calculation.

The radiometer data is obtained using the same system that is used for the interferometer data. The system is only setup as a radiometer, and the captured data is used test RFI mitigation techniques. The tests are also performed at Stellenbosch University, ensuring RFI contaminated data.

All of the offline methods are tested on desktop computer with the following specifications:

- Intel(R) Core(TM) i5-2500K CPU @ 3.30GHz
- 8GiB DDR3 Synchronous 1600 MHz
- OCZ-AGILITY3 with an ext4 partition
- Python 2.7.3
- Ubuntu 12.10
- Matlab R2012b

The run-time of the algorithms are computed in python. The starting and stopping times of the algorithms are saved, and the difference between the two values are taken as the run-time. Although the algorithms are not optimised, the timing will still indicate with some measure the computational cost of each algorithm.

6.1 Spectral Kurtosis

The spectral kurtosis is tested with three data sets: KAT7 data, interferometer data from [35] and self observed radiometer data. For both the KAT7 data and the radiometer data, the RFI is only flagged and shown in a graph. The interferometer data is processed for RFI and the flagged data is removed before post processing.

6.1.1 Radiometer Data

The spectral kurtosis is partly performed on the ROACH1 to improve the efficiency of the online technique. The block diagram of the system on the ROACH1 board is shown in Fig. 6.1.

The signal is digitised, passed to the polyphase filter and then Fourier transformed. The output of the FFT is squared to produce the power of each signal, and is then split into 2 streams. This first stream is passed to a delay, to allow for the

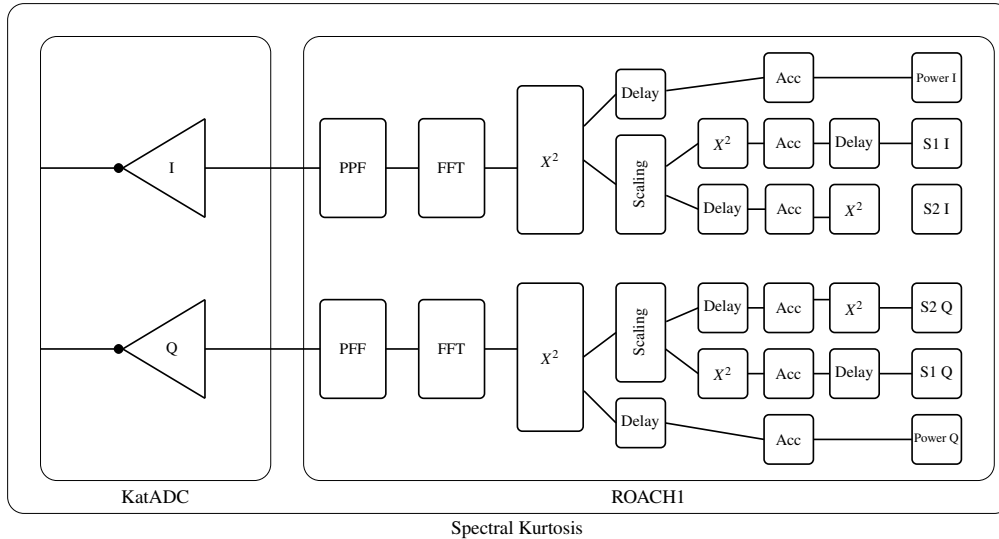


Figure 6.1: Spectral Kurtosis as implemented on the ROACH1 board set up as a radiometer.

power measurement as well as the spectral kurtosis estimate to arrive simultaneously at the computer. The delayed signal is then passed to BRAM for accumulation, then out to the PC. The second stream is down-scaled from a 37 bit fixed point number to a 15 bit fixed point number. With this downscaling, a multiplier is added to adjust the gain to the spectral kurtosis system to prevent overflowing or underflowing. The stream is then split into two streams again, one for S1 and one for S2. The S1 stream is squared again, accumulated and then delayed to allow for the S2 signal to arrive on time. The S2 signal is delayed to allow for the lost time of the S1 signal being squared. The S2 signal is then accumulated, squared and then passed to the PC.

The power of the recorded signal, as well as the S1 and S2 parameters, as described in Sec. 4.1.1, are formed on the ROACH1. A Python script is then used to communicate with the Berkeley Operating system for ReProgrammable Hardware (BORPH) server running on the ROACH1. The data is then extracted via the PowerPC. The connection between the Vertex 5 and the PowerPC is speed limited and for this reason, the recorded values must be integrated over more than a second to ensure no data loss.

Once the data is on the computer, the Python script interleaves the data received and computes the spectral kurtosis estimate. A mask is then produced to indicate all of the frequency channels that contain RFI. Since the spectral kurtosis is computed for the same data set as the integrated radiometer power data, the spectral kurtosis mask can be applied directly to the recorded values and the RFI removed. The data is then exported to a Matlab file, to be graphed using a Matlab script.

6.1.2 KAT7 Data

The data from the binary HDF5 file is read into memory via a Python script. The spectral kurtosis estimate is then computed for 1000 spectra, and is performed on two baselines as a proof of concept. The spectral kurtosis estimate is then used to determine which bins contain RFI, by flagging bins where the spectral kurtosis estimate deviates by more than a given amount. The raw spectrum as well as the spectral kurtosis estimate is written to a Matlab file. The Matlab is then used to graph the results of the spectral kurtosis estimate on the KAT7 data. The KAT-7 data contains RFI, as the data is part of a RFI scan result.

6.1.3 Interferometer Data

The interferometer data is read in via a Python script from the HDF5 file which contains the complex values of the auto correlated as well as cross correlated values of the interferometer. For the spectral kurtosis method, only the cross correlation is looked at, as the result of removing RFI can be observed in the post processing stage when forming the grating lobes.

The first step is to produce the spectral kurtosis estimate of a single measurement set. This is done in Python after the data is read in. The spectral kurtosis estimate is saved in a Matlab format to be graphed. The saved spectral kurtosis estimate is then also used as a template for the RFI that is observed. A mask is made from the spectral kurtosis estimate, where any value larger or smaller than a given deviation is flagged as RFI. This mask is then used on each sampled spectrum to remove the affected frequency bin before it is used to produce the fringes. The raw data is displayed as a waterfall plot as well as the cleaned spectrum and the fringes.

The time duration of the algorithm is extracted from the python script, and given as measure of computational cost.

6.2 Spectral Flatness

6.2.1 Radiometer Data

The power measurements done for the spectral kurtosis are also used for the spectral flatness. The power measurements are fed into Matlab from the saved radiometer data. The RFI mask is built up from the saved data to produce the cleaned spectrum.

6.2.2 KAT7 Data

The correlated data is read in via a Python script from the binary HDF5 file. The spectral flatness is then computed for the frequency spectrum, by summing the same frequency bin for a number of consecutive time samples. The standard spectral flatness is also computed by summing each spectrum for each time series. The possible corrupted frequency bins, as well as the possible corrupted time slices are

then indicated by flagging abnormalities in the spectral flatness results. The results are then also exported to a Matlab file, where it is then used to graph the results.

6.2.3 Interferometer Data

The spectral flatness is only used to flag RFI and to remove it. The interferometer data is read from the HDF5 files by a Python script. The spectral flatness is computed for each spectrum recorded in time, as well as for each frequency bin of the recorded spectrum. The spectral flatness as well as the raw spectrum is written to a Matlab file for further processing. Matlab is used to graph and display the spectral flatness as well as the raw spectrum.

The time duration of the algorithm is extracted from the python script, and given as measure of computational cost.

6.3 Pulsar RFI Mitigation

The pre-recorded pulsar data received from HartRAO[16] is used to test the algorithms of PSRchive. This included data from Pulsars J0835-4510, J0437-4715 and J0738-4042. The first basic step is to remove persistent narrowband RFI. These are bands that always contain RFI and can always be ignored as its influence is known.

The narrowband RFI mitigation is done using a script for PSRchive. The standard median algorithm looks at the total flux summed over all pulse phases. This can cause the RFI to be masked behind the total power of all of the channels, especially short bursts. In [22] a per-frequency operation is performed, where the the minimum of each channel is subtracted from the maximum. The median operation is then performed on this difference. The code for the script is given as:

```
#!/usr/bin/env psrsh
zap median exp={ $all : max-$all : min }
zap median
```

The standard *mow* function (that is part of PSRchive) performs the broadband RFI removal, but it does not utilise the MAD operation for robust statistics. A script is used to force the *mow* function to use robust statistics as well as changing the size of the window. The parameters are redefined as follows:

```
#!/usr/bin/env psrsh
zap mow robust
zap mow window=0.1
zap mow
```

The algorithms are pre-set in the software, but they do not allow for parameter changes. Using a script to define the algorithms allows for fine tuning of the parameters to suit the problem.

6.4 Var/SumThreshold

VarThreshold and SumThreshold can be run on post correlated data, as it does not use statistics to detect RFI. For this reason algorithms can be tested on both provided data from KAT7 [34] as well as data recorded using the correlator built by Jansen van Rensburg [35].

6.4.1 KAT7 Data

For the KAT7 data, the SumThreshold as well as the VarThreshold algorithm is run in both frequency as well as time. The data is read in via a Python script from a HDF5 file which contains the correlated data stored in binary format.

The final tests are done on the auto correlation and a cross correlation data sets in order to test the algorithm. The data provided explicitly contains more RFI, as the interferometer was used to capture RFI.

The data is read into Python as an $N \times M$ matrix, where one axis represents time and the other frequency. The SumThreshold and VarThreshold algorithms are run on the data with a given threshold parameter and five iterations. The threshold is firstly chosen as three times the median of the data. This is then adjusted until the best results are obtained of the given dataset. The KAT7 data is only flagged to show possible locations of RFI, which is then displayed using a waterfall plot.

6.4.2 Interferometer Data

The data contains the auto correlation of each antenna as well as the correlation of the two antennas. This cross correlation data is used to produce the fringes of the interferometer. This is then done for each baseline to produce an image of the sun at 1.4GHz. To test the SumThreshold as well as the VarThreshold method, the data is manipulated before producing the fringes. The algorithms are applied both in the time as well as frequency axis, but the uneven distribution of the power spectrum of the sun provides difficulty for algorithms to remove all RFI. The dirty channels are flagged and removed before the fringes are produced. This is done in python as the data is also stored in a HDF5 file. The results are then exported to Matlab to produce the graphs of the raw signal, flagged noise and cleaned signal.

The time duration of the algorithm is extracted from the python script, and given as measure of computational cost.

6.5 Conclusion

Computationally inexpensive algorithms of spectral flatness and spectral kurtosis allows for the implementation of these methods on both pre- and post-correlated data. The Var/SumThreshold method is, however, bound to only post-correlated data.

Chapter 7

Measurements

The final measurements are done on data that is either previously recorded, or on live data as recorded by the ROACH1 and KatADC as described in Chapter 6. The four methods that are tested include:

- Spectral Kurtosis
- Spectral Flatness
- Sum/VarThreshold
- Pulsar RFI Mitigation

The speed of each method is a factor, especially for online methods as this can cause a bottleneck in the system when dealing with high data rates. The effectiveness of the method will also be evaluated. This will include false positives as well as possible inability to flag RFI.

7.1 Spectral Kurtosis

Spectral Kurtosis is tested pre- and-post correlator. Both the interferometer data and KAT-7 data is post correlator data, where as the radiometer data is not correlated. The threshold value for each test is set manually. The threshold value is lowered until data corruption is observed. The previous value that did not cause data corruption is then used as the threshold value.

7.1.1 Radiometer Data

The spectral kurtosis algorithm is partially implemented on the ROACH1 board. The raw radiometer data as well as the $S1$ and $S2$ values are sent to the computer for analysis. The number of iterations M for this measurement is set to 20000 as the integration time has to be large enough to reduce data rates from the ROACH1 to the computer. The integrated power of the recorded spectrum is shown in Fig. 7.1

and the spectral kurtosis computed from the given $S1$ and $S2$ is shown in Fig. 7.2. The median of the spectral kurtosis is computed to determine the valid signal level with a deviation of 0.3. The spectral kurtosis of the valid signal is less than 1. The cleaned spectrum is shown in Fig. 7.3.

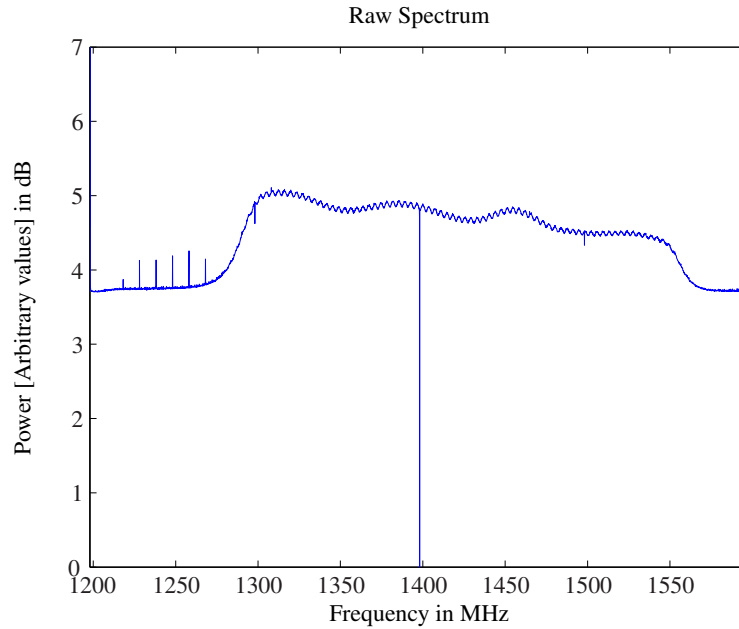


Figure 7.1: The integrated spectrum of the radiometer as recorded for the online spectral kurtosis implementation on the ROACH1.

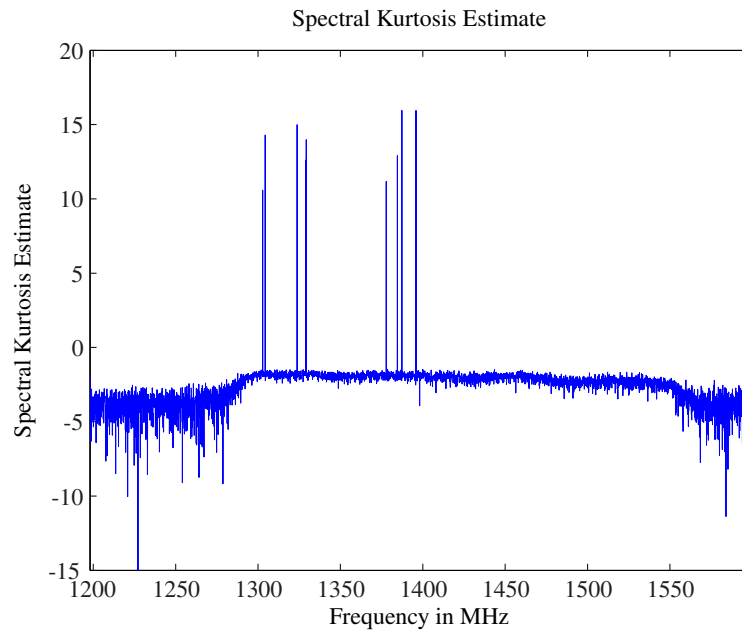


Figure 7.2: The spectral kurtosis estimate of the integrated spectrum shown in Fig. 7.1.

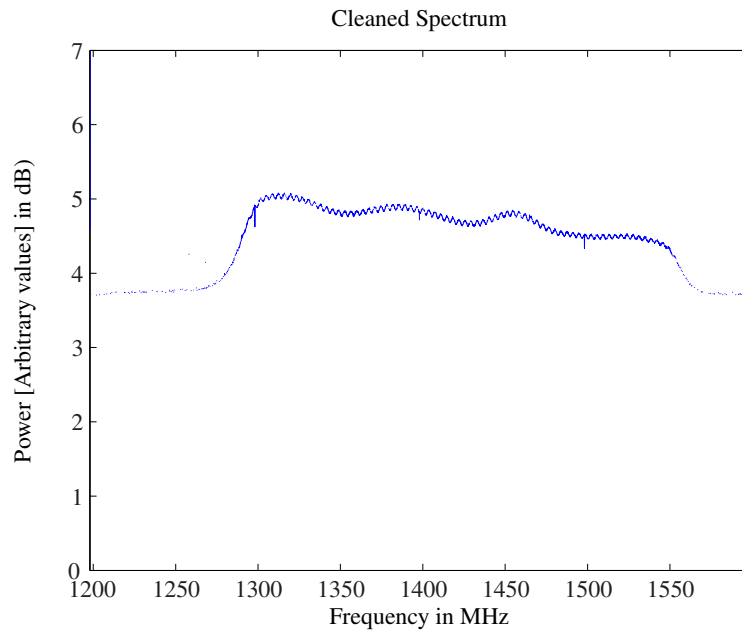


Figure 7.3: The cleaned radiometer spectrum using spectral kurtosis.

7.1.2 KAT7

The KAT7 measurement contain a number of correlation products, in this thesis correlation product 10 will be used throughout. The raw spectrum of this measurement set is shown in Fig. 7.4. The measurement has 4096 frequency bins with a bandwidth of 6.25 MHz and centre frequency at 1417 MHz. For convenience, only 4096 time sample are taken, to make the waterfall plot a 4096×4096 graph. The full 4096×4096 spectrum is read in before any algorithm is performed on it.

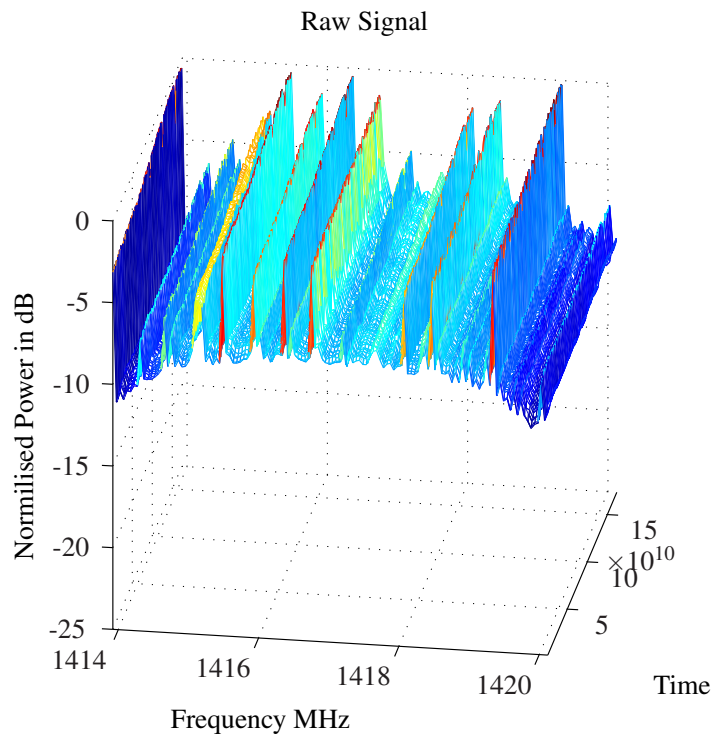


Figure 7.4: The raw KAT7 spectrum of measurement, correlation product 10[34].

Once the data is read in from the HDF5 file using python, the spectral kurtosis (using the full 4096 spectral samples) is computed. The spectral kurtosis is then saved in a Matlab file to produce the graph shown in Fig 7.5. This data is also used to produce the RFI mask for the removal of RFI in the raw data. The median of the spectral kurtosis is computed and a margin is chosen around the median. Any frequency bin of the spectral kurtosis that is more or less than the margin within the median is flagged as RFI.

The flagged bins are removed from the dataset to produce the graph shown in Fig. 7.6. The strong RFI source at 1414 MHz, as well as a weaker source at 1416 MHz still remains after filtering. The other larger sources are, however, completely removed from the cleaned spectrum. These large RFI sources can be seen in the graph containing the removed noise, shown in Fig. 7.7. The run-time is 0.7867 seconds.

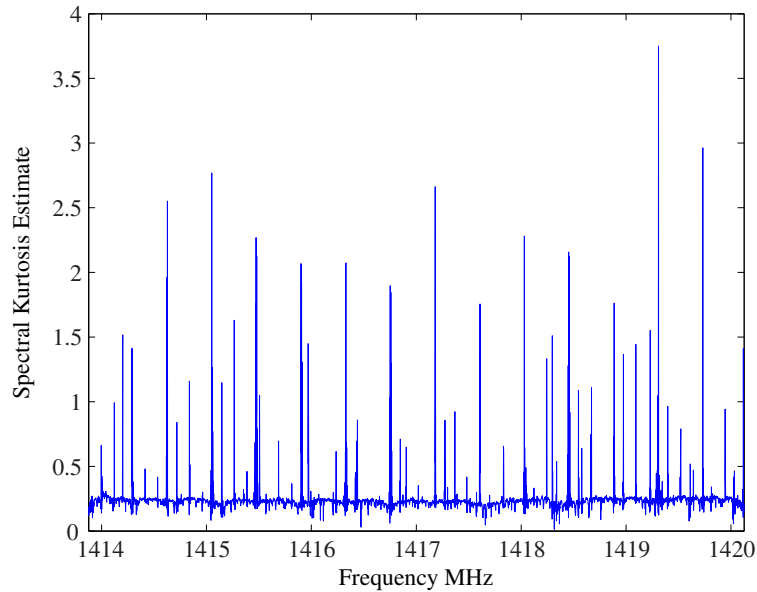


Figure 7.5: The spectral kurtosis estimate of the spectrum shown in Fig. 7.4.

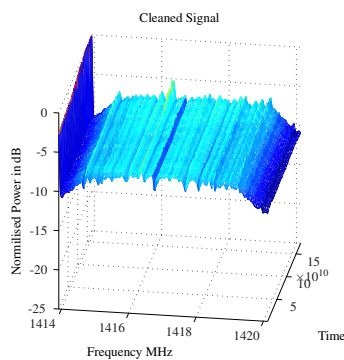


Figure 7.6: The cleaned KAT7 spectrum as filtered by spectral kurtosis.

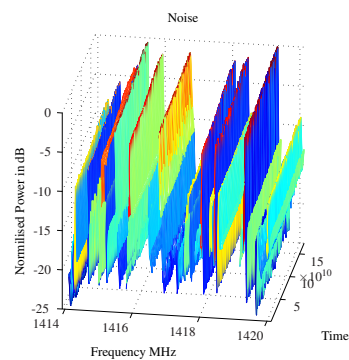


Figure 7.7: The noise of the KAT7 spectrum as filtered by spectral kurtosis.

7.1.3 Interferometer Data

Spectral kurtosis is performed on the cross correlation of an observation performed on the sun. The two antennas are placed nine meters apart and pointed in the path of the sun. The system then recorded the auto correlation of each antenna, as well as the cross correlation between the two as the sun moves past the focus of the two antennas. This is done in the effective frequency band of the system which is from 1292MHz to 1548MHz. The result of the raw data is shown in Fig 7.8. There are a number of continuous RFI sources present in the raw data, as can be seen at 1350MHz as well as broadband noise at 250 seconds.

The cross correlation data is used to create the interference fringes for the observed source. The noise in the spectrum will result in artefacts in the interference fringes of the observed source. The interference fringes produced from the

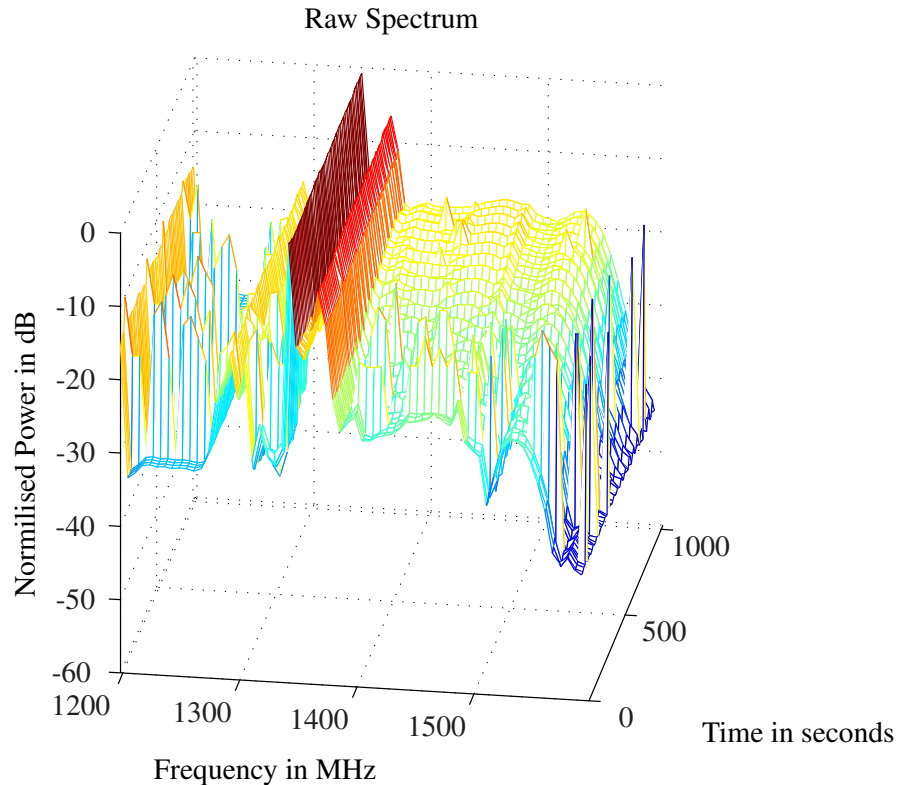


Figure 7.8: The raw interferometer spectrum as observed for a number of seconds.

above spectrum suffers from RFI as can be seen in the number of artefacts in the graph shown in Fig 7.9. The interference fringes are calculated using only the spectrum 1292MHz to 1548MHz and not the full spectrum shown, as this is the system bandwidth. The signal-to-noise ratio of the unfiltered interference fringe plot is -17.67dB . The spectral kurtosis estimate is calculated on the full spectrum for the full recorded time, and is shown in Fig 7.10 in dB. The run-time for the algorithm is 0.095217 seconds. The spectral kurtosis estimate is used to produce a mask to indicate the noise in the recorded spectrum. For the data in Fig 7.8, a threshold is set to flag RFI. This is done by taking the median of the spectral kurtosis and adding a differential limit of one. This mask is then used to filter out the bins in the spectrum above to produce the cleaned spectrum as well as a re-evaluated visibility.

The contaminated bins are simply zeroed out to produce the clean spectrum. This time-shortened spectrum can be seen in Fig 7.11. The signal-to-noise ratio of the filtered interference fringe plot is 7.27 dB. Although not visible on the averaged graph shown in Fig. 7.11, there are a number of small broadband bursts remaining in the data. This can be filtered out by removing the time series with the contaminated broadband noise. There is also uncontaminated data that is flagged as RFI and removed. In Fig 7.12, the removed frequency bins are shown. Uncontaminated data can be seen in the noise graph at the frequency bin 1500 MHz.

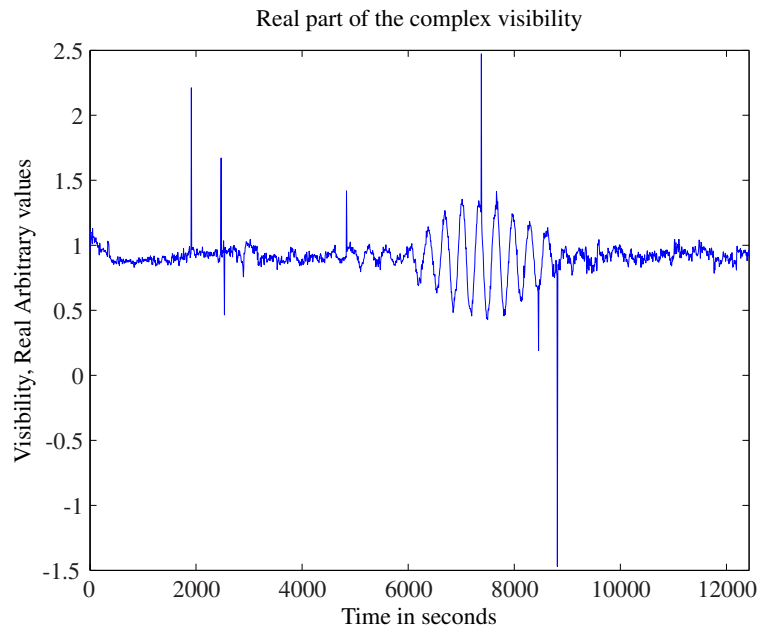


Figure 7.9: The interference fringes as produced from the recorded spectrum in Fig 7.8.

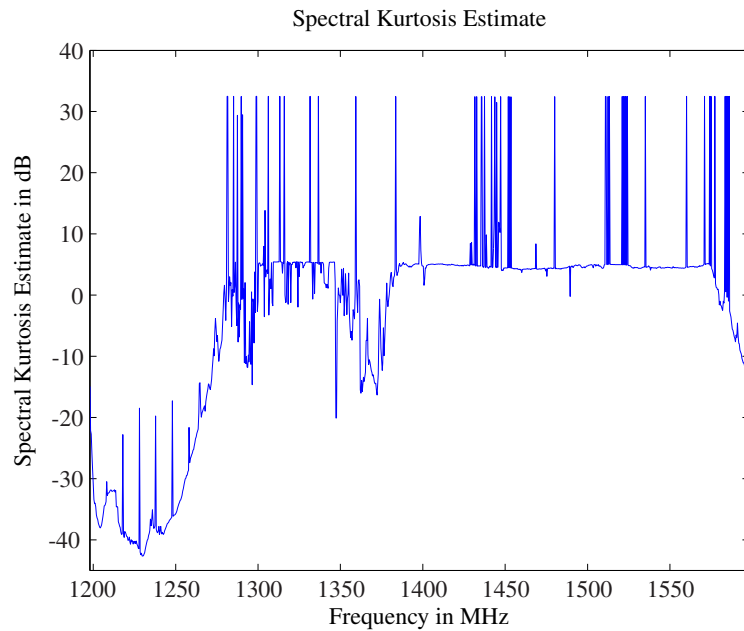


Figure 7.10: The spectral kurtosis estimate of the spectrum as seen in Fig 7.8.

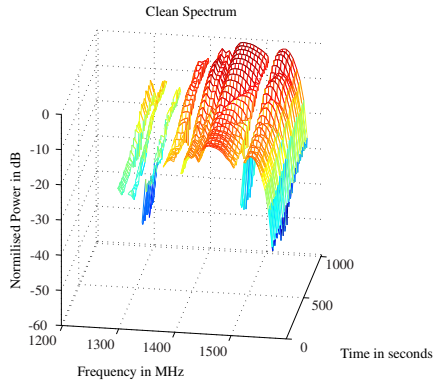


Figure 7.11: The clean interferometer spectrum as observed for a number of seconds after filtered by spectral kurtosis.

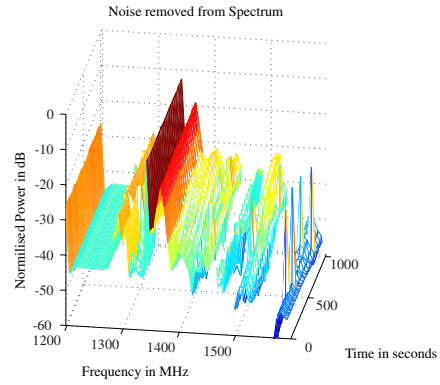


Figure 7.12: The noise of the interferometer spectrum as observed for a number of seconds after filtered by spectral kurtosis.

The interference fringes are produced again using the cleaned spectrum as shown in Fig 7.11 to produce Fig 7.13. As can be seen from this figure, all of the artefacts are removed from the visibility and more recorded time is usable from the interference fringes. The only glitch that remains is located at 4000s where the amplitude of the fringe is distorted. This can be attributed to residual RFI in the final signal.

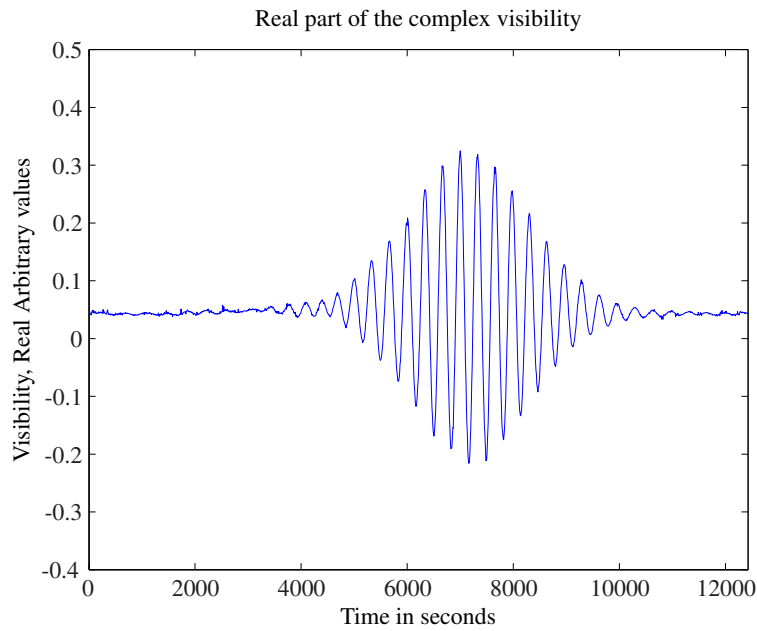


Figure 7.13: The interference fringes as produced from the cleaned spectrum in Fig 7.11.

7.2 Spectral Flatness

The spectral flatness method is used on all of the datasets described below. The threshold value for each test is set manually. The threshold value is lowered until data corruption is observed. The previous value that did not cause data corruption is then used as the threshold value.

7.2.1 Radiometer

The integrated data from the ROACH1 is sent to the computer for analysis as the FPGA cannot implement the floating point arithmetic for spectral flatness. A number of integrated spectra are saved for further processing. The raw spectrum as observed over time can be seen in Fig. 7.14. The geometric mean and arithmetic mean are computed in Matlab and the spectral flatness is produced. The result is shown in Fig. 7.15. The spectral flatness of the valid data is clearly 1 within the bandwidth of the system specifications. A number of short tones are injected over time to add RFI noise to the measurement and are visible between samples 40 and 60 as small bumps in the operating frequency band. A deviation of 0.2 less than 1 is chosen as the cut-off for RFI to produce the RFI mask. The cleaned spectrum is shown in Fig. 7.16 with the noise removed shown in Fig. 7.17. All of the added RFI is removed from the cleaned spectrum.

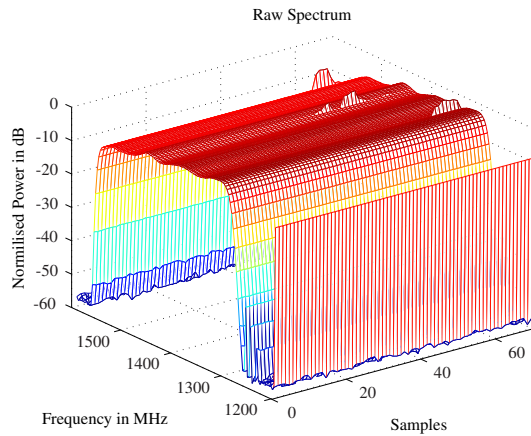


Figure 7.14: The raw radiometer spectrum used for spectral flatness.

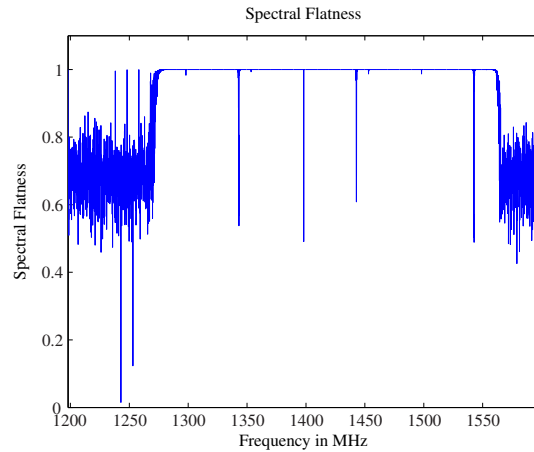


Figure 7.15: The spectral flatness as computed for Fig. 7.14.

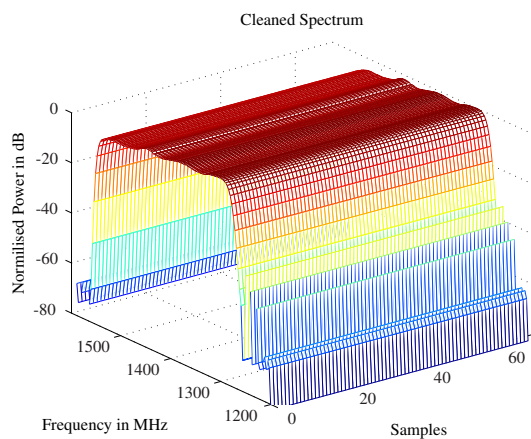


Figure 7.16: The cleaned radiometer spectrum using spectral flatness.

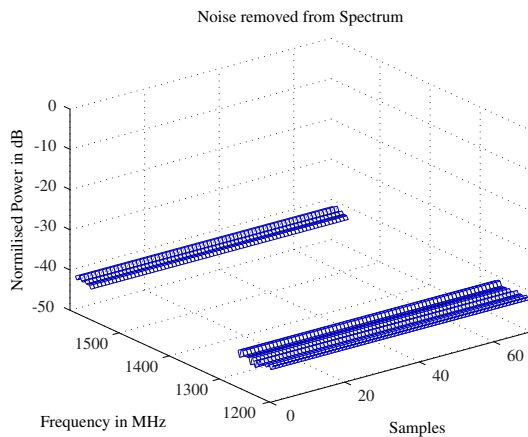


Figure 7.17: The noise in the radiometer spectrum as removed by spectral flatness.

7.2.2 KAT7 Data

The same dataset that is used for the spectral kurtosis test on the KAT7 data is also used for the spectral flatness test. The waterfall graph of the data can be seen at Fig. 7.4. The spectral flatness is again computed for the full 4096 spectra and this is shown in Fig. 7.18. The median of the spectral flatness is then computed, to determine the level of the valid signal. A threshold around the median is chosen as a valid signal and any frequency bin of the spectral flatness that does not fall within the deviation is marked as RFI.

The RFI marked bins are then used to produce a RFI mask for the dataset and are removed from the measurement. The cleaned spectrum is shown in Fig. 7.19. The strong RFI source is again present at 1414 MHz, as well as the weaker source at 1416 MHz after spectral flatness has been applied. There is also a weaker RFI source present at 1419 MHz that is not visible in the spectral kurtosis cleaned spec-

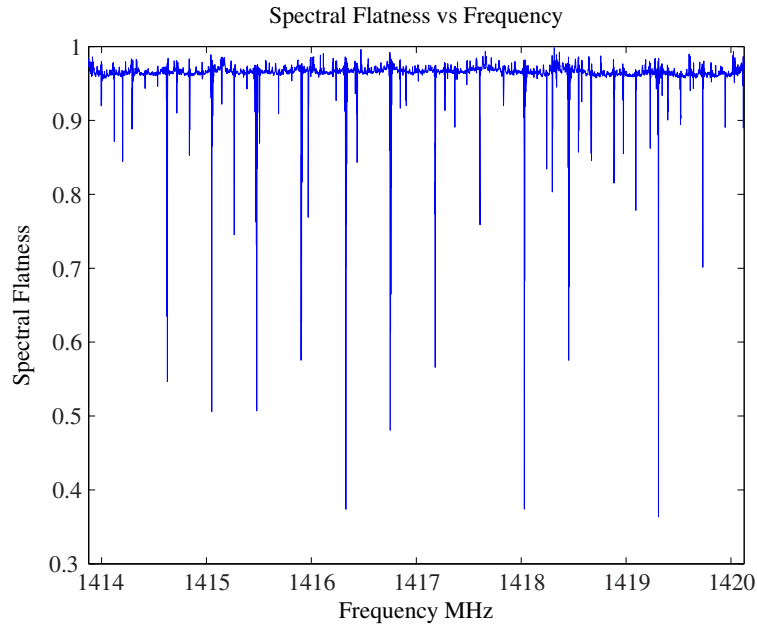


Figure 7.18: The spectral flatness measure of the KAT7 dataset shown in Fig 7.4.

trum. The frequency bins that are removed from the cleaned spectrum is shown in Fig. 7.20. The run-time is 1.2751 seconds.

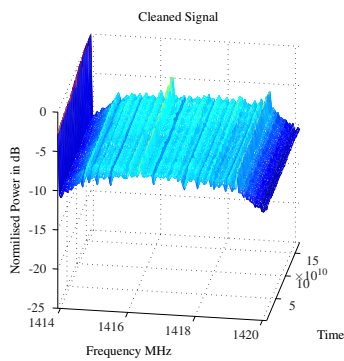


Figure 7.19: The raw spectrum Fig. 7.4 cleaned using spectral flatness.

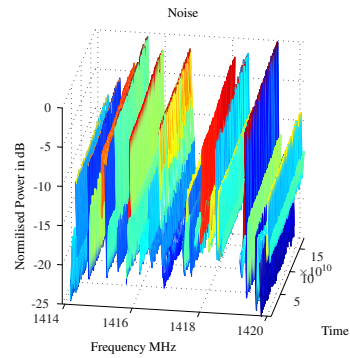


Figure 7.20: The noise from the raw spectrum Fig 7.4 when removed using spectral flatness.

7.2.3 Interferometer Data

The same dataset is used for the spectral flatness tests, that was used in the spectral kurtosis (Sec. 7.1.3), with the spectrum and resulting visibility plot shown in Fig 7.8 and 7.9. The spectral flatness for each of the frequency bins is computed for the full duration of the sun measurement. The resulting spectral flatness is then shown in Fig 7.21. The run-time for the algorithm is 0.184751 seconds. The median of the spectral flatness is then computed to use as a base for the signal content. A threshold values is set for the maximum deviation of the spectral flatness from the computed median. When the RFI mask produced from Fig 7.21 is applied to the raw data

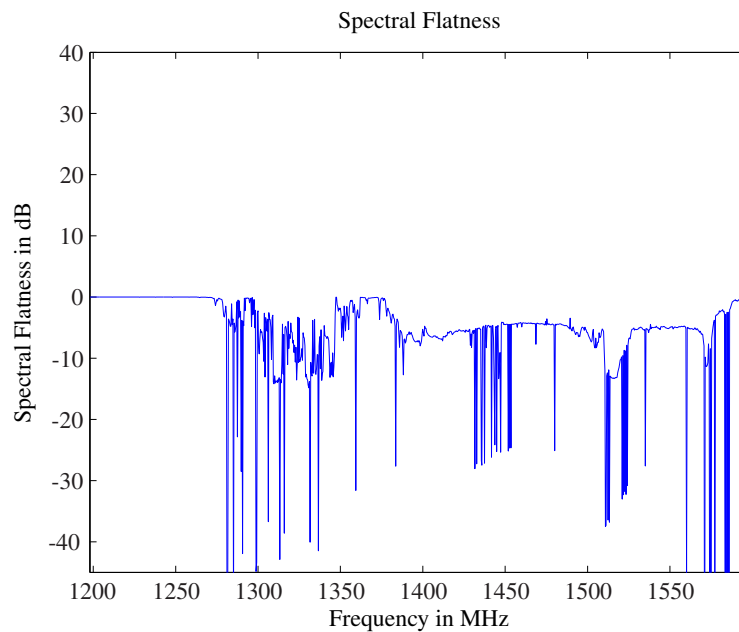


Figure 7.21: The spectral flatness for each frequency bin as computed from the spectrum shown in Fig 7.11.

shown in Fig 7.8 and the flagged bins removed, the resulting waterfall plot is shown in Fig 7.22. The amount of data that is removed, compared to spectral kurtosis, is more as the number of flagged bins indicated. This is supported by the noise plot of the spectrum shown in Fig 7.23. There are, however, still a number of broadband sources remaining in the cleaned spectrum, again masked by the averaging of the waterfall plot.

The visibility function is again generated, this time using the spectral flatness cleaned data. The resulting graph is shown in Fig 7.24. All of the glitches are removed from the final result, as well as the amplitude glitch seen in the spectral kurtosis clean visibility function (Fig. 7.13). The fringes, however, indicate minor distortions at the peaks, but this can be a result of the large amount of data that is thrown out in the cleaning process. The signal-to-noise ratio of the cleaned interference fringe plot is 5.94 dB.

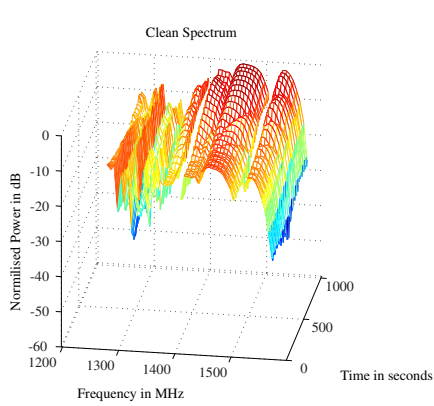


Figure 7.22: The raw spectrum Fig 7.8 cleaned using spectral flatness.

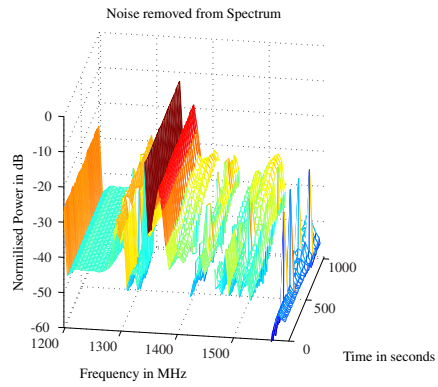


Figure 7.23: The noise from the raw spectrum Fig 7.8 when removed using spectral flatness.

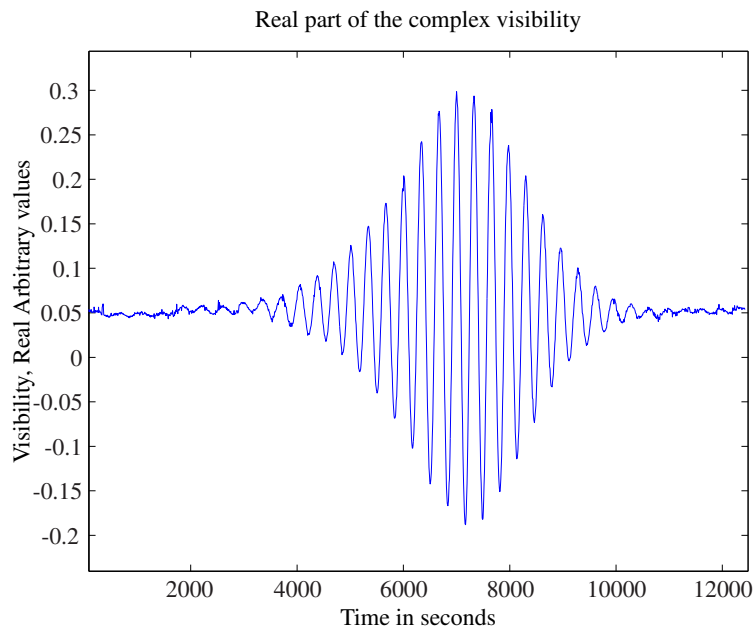


Figure 7.24: The interference fringes produced from the spectral flatness cleaned spectrum shown in Fig 7.22.

7.3 Var- and SumThreshold

The Var- and SumThreshold methods are tested on both the interferometer data and the KAT7 data as post-processing method. The χ_1 value for each test is set manually. χ_1 is lowered until data corruption is observed. The previous value that did not cause data corruption is then used as χ_1 .

7.3.1 Interferometer Data

The same measurements are used for both the Var and SumThreshold as for the spectral flatness in Sec. 7.1.3. The spectrum can be seen in Fig 7.8. The start threshold level $\chi_1 = 0.0057$, which is equals to three times the median. The number of iterations M is set to five, and the value of ρ is set to 1.5 as advised by Offeringa [4]. For the test, all of the flagged bins are removed from the final results. The clean spectra for both methods are shown in Fig. 7.25 and Fig. 7.26. The run-time for SumThreshold was 153.766261 seconds, and 174.433670 seconds for VarThreshold. All of the small strong sources are removed from the final result, which include the broadband pulses that remained after applying spectral flatness or spectral kurtosis. The weaker continuous RFI sources, however, are still visible in the cleaned spectrum. This is due to the fact that the power of the observed signal, namely the sun, does not stay constant over time. As the sun moves across the focus of the interferometer, the total power observed changes over time. Applying the threshold for the full duration of the observation, limits the threshold to the maximum peak power observed by the sun. The result is that a signal weaker than the sun's power still remains. The noise detected is shown in Fig. 7.27 and Fig. 7.28. The overall computation time of these methods are significantly longer than the spectral flatness or spectral kurtosis methods.

The visibility functions are then computed from both the VarThreshold and SumThreshold cleaned spectra. The results are shown in Fig. 7.29 and Fig. 7.30. The difference between the two methods are not clearly visible from visibility functions. The visibility plot cleaned using VarThreshold shows slightly more noise at the phase glitch (at 5000s), which is present in both plots. The signal-to-noise ratios of the interference fringe plots of VarThreshold and SumThreshold are 3.10 dB and 3.18 dB.

7.3.1.1 Parameter Check

The effect of the parameters M and ρ tested on the interferometer data. The values for M and ρ are varied and the signal-to-noise ratio is measured of the interference fringes. This is only done for the SumThreshold case, as it will have the same influence on VarThreshold. The value of ρ is varied from one to two, and the value of M is varied from one to nine. The following graphs show the signal-to-noise ratio of each test, in relation to the varied value. The effect of varying ρ is shown in Fig. 7.33 with $M = 5$. The effect of varying M is shown in Fig. 7.31, with $\rho = 1.5$.

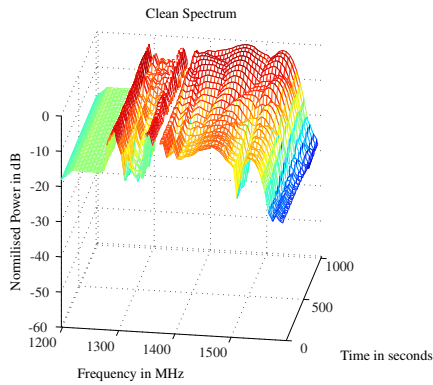


Figure 7.25: The spectrum shown in Fig 7.8 cleaned using VarThreshold.

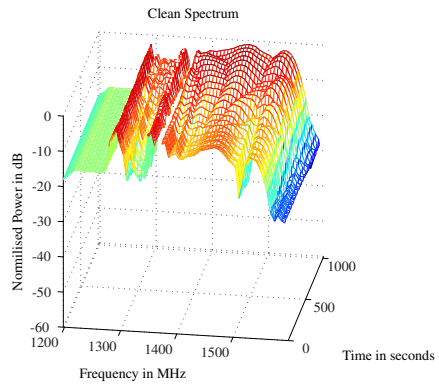


Figure 7.26: The spectrum shown in Fig 7.8 cleaned using SumThreshold.

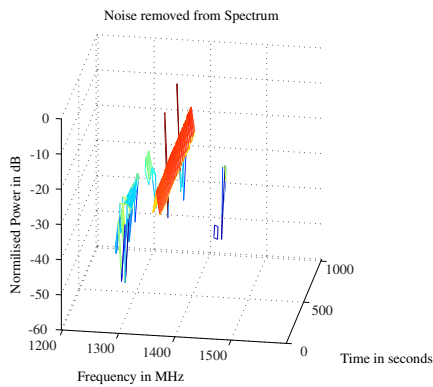


Figure 7.27: The noise in the spectrum shown in Fig 7.8 flagged using VarThreshold.

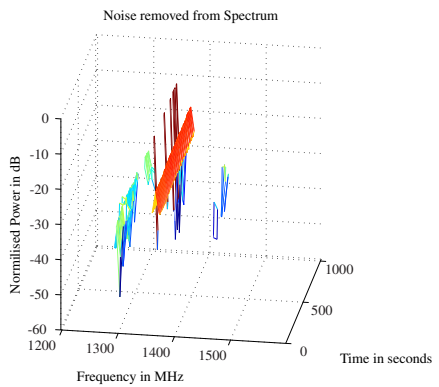


Figure 7.28: The noise in the spectrum shown in Fig 7.8 flagged using SumThreshold.

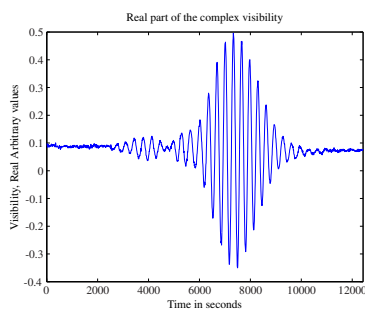


Figure 7.29: The interference fringes produced from the VarThreshold cleaned spectrum shown in Fig 7.25.

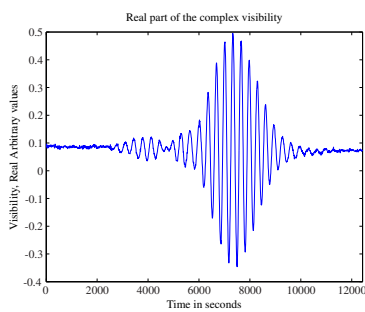


Figure 7.30: The interference fringes produced from the SumThreshold cleaned spectrum shown in Fig 7.26.

As only M effects the run-time of SumThreshold, this is shown in Fig. 7.32. At

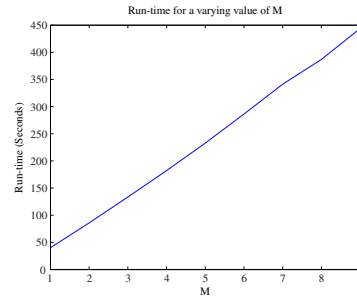
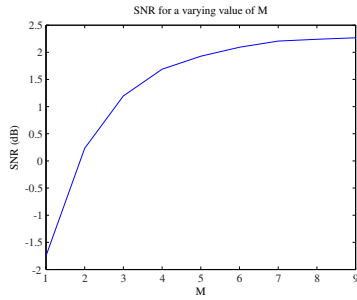


Figure 7.31: The effect on signal-to-noise ratio for varying M .

Figure 7.32: The effect on run-time for varying M .

$M = 7$ the signal becomes distorted and this in turn lowers the signal-to-noise ratio as seen in Fig. 7.31. The run-time stays constant as expected, as only the number of iterations increase. As ρ is increased, the amount of data flagged on each iteration

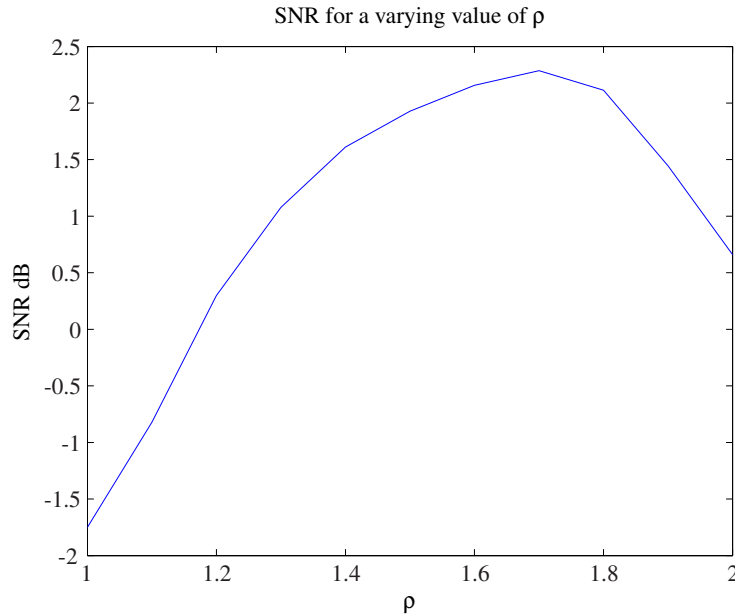


Figure 7.33: The effect on signal-to-noise ratio for varying ρ .

increases as well. The result is that valid data is being flag as RFI once ρ becomes larger than 1.6.

7.3.2 KAT7 Data

The dataset from KAT7 is used to test both the VarThreshold as well as the SumThreshold method. The χ_1 value for both is set to 0.7, with the standard $\rho = 1.5$ and $M = 5$. The data is read in via the Python script from the HDF5 file to produce the 4096×4096 matrix. The data is then iteratively run through both methods to produce the RFI map of the dataset. All of the bins that are flagged as RFI are then removed from the raw dataset. The cleaned data for both methods are shown in Fig. 7.34 and Fig. 7.35. Both cleaned spectra display a similar shape. The iterative nature of the algorithm results in a much longer computation time in comparison to the online methods. The flagged RFI bins can also be seen in Fig. 7.36 and Fig. 7.37. The run-time of VarThreshold was measured to be 1111.4 seconds, and SumThreshold 1673.6 seconds.

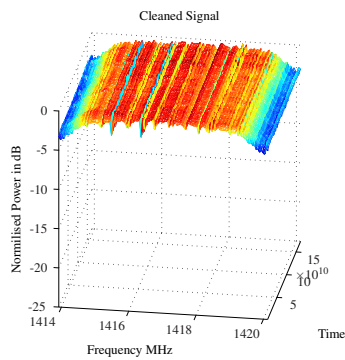


Figure 7.34: The spectrum shown in Fig 7.4 cleaned using VarThreshold.

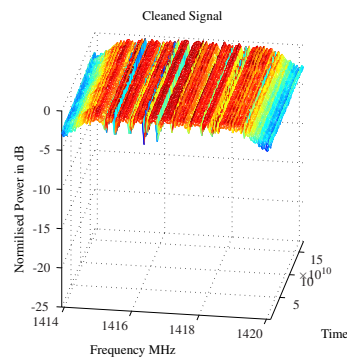


Figure 7.35: The spectrum shown in Fig 7.4 cleaned using SumThreshold.

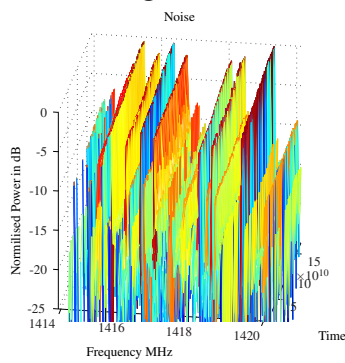


Figure 7.36: The noise in the spectrum shown in Fig 7.4 flagged using VarThreshold.

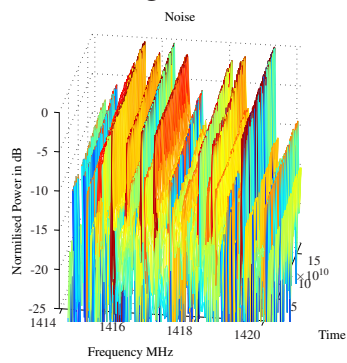


Figure 7.37: The noise in the spectrum shown in Fig 7.4 flagged using SumThreshold.

7.4 PSRchive RFI Mitigation

For the testing of the PSRchive methods, two datasets are selected that illustrate the use case of the methods. These datasets do not contain pulsar information, only RFI to illustrate the use case.

7.4.1 Narrowband Noise

The data for the narrowband noise is provided by HartRAO [16]. The noise is observed in a recording of pulsar 110904 in the 700 MHz to 760 MHz frequency band. The raw contaminated dataset is shown in Fig. 7.38. Three strong narrowband RFI sources are visible, one just below 740 MHz and two right above and right below 720 MHz. The dataset is passed through the algorithm to remove RFI. The cleaned spectrum is shown in Fig. 7.39. The three strong RFI sources are removed from the final result as well as a minor source. There are also frequency bins removed that did not indicate RFI contamination on the raw signal; however, a weaker RFI source can be hidden by the three strong sources.

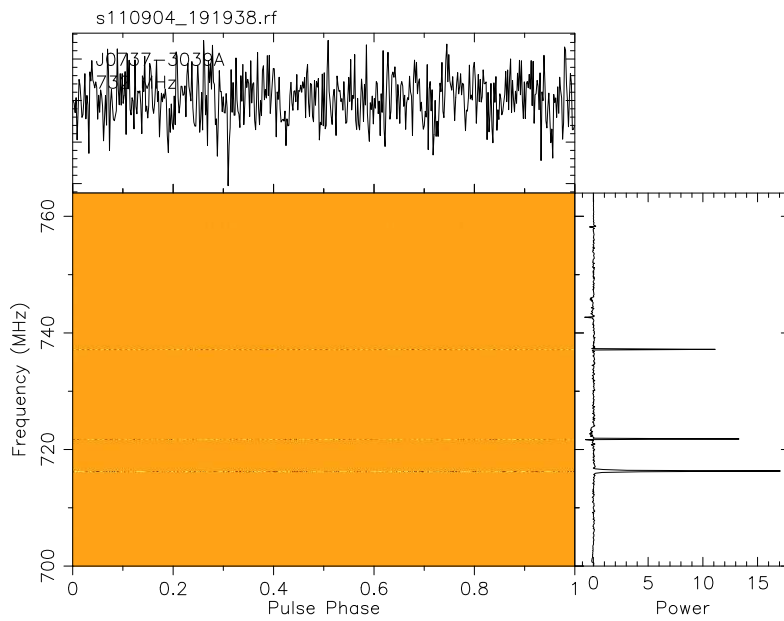


Figure 7.38: The raw data contaminated with narrowband noise as provided by HartRAO [16].

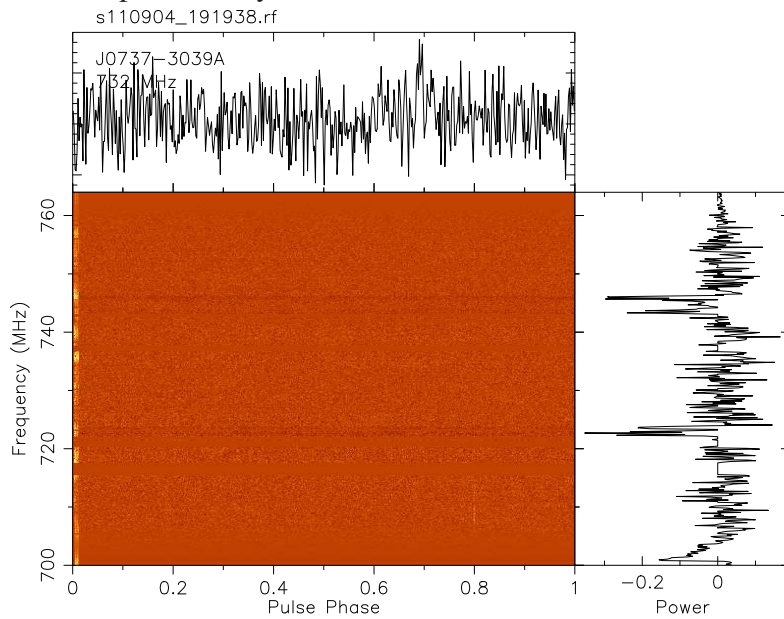


Figure 7.39: The cleaned data after it is passed through PSRchive.

7.4.2 Broadband Noise

The data recorded by [36] provided a sufficient example of broadband noise. The raw data shown in Fig. 7.40 indicates broadband noise close to phase bins 0.5 and 0.7. These are no pulsar pulses, as the raw data indicates no dispersion of the pulse over frequency. Narrowband noise is also present in the example, but will be ignored as the broadband RFI detection algorithm for timing pulsar data is only valid for broadband noise. The data is passed through PSRchive to remove the broadband noise and replace it with noise of the same power as the uncontaminated areas. The result is shown in Fig. 7.41.

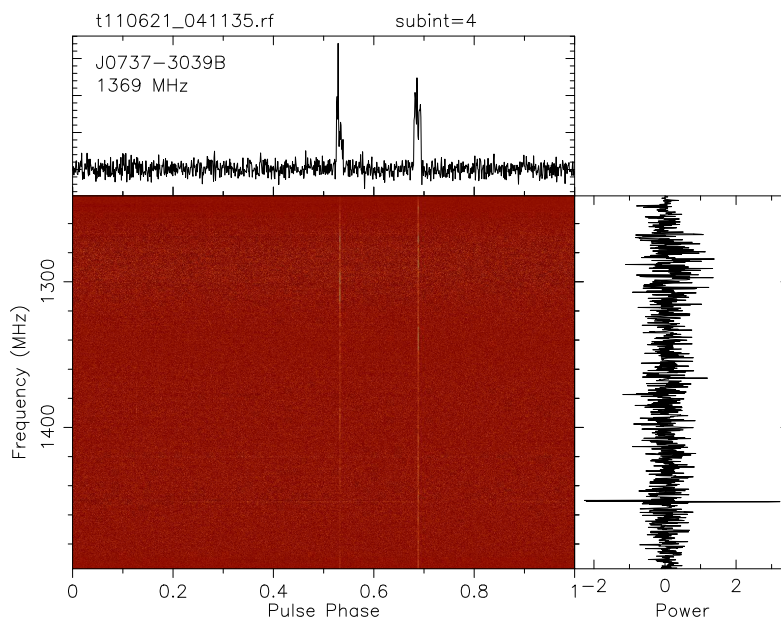


Figure 7.40: The raw data contaminated with broadband noise as provided by [36].

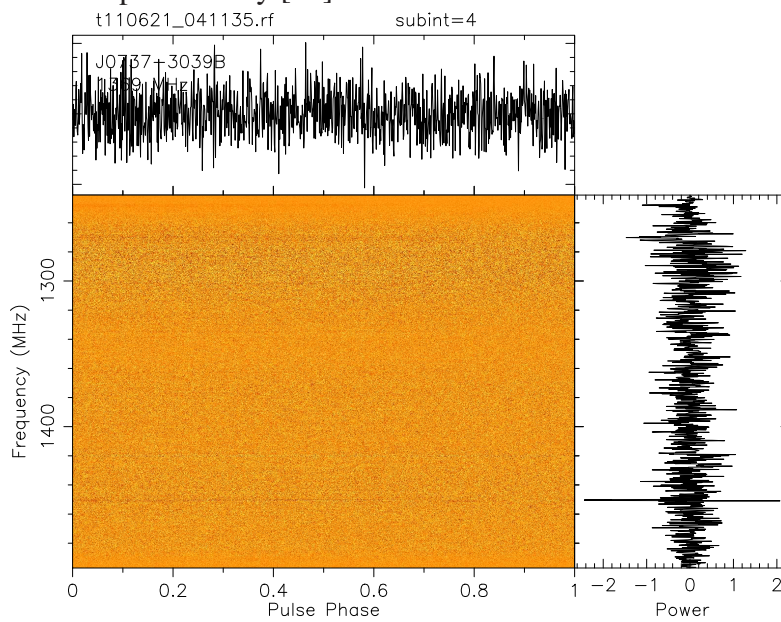


Figure 7.41: The cleaned data after it is passed through PSRchive.

7.5 Daisy-Chain

A number of the algorithms are daisy-chained, in order to test their cumulative effect. As the data provided by [35] allows for signal-to-noise measurements of the interference fringes, this dataset will be used through the daisy-chain system to validate the tests. The run-time of each algorithm will be displayed as well as the signal-to-noise measurement of the final result.

7.5.1 Spectral Kurtosis to Spectral Flatness

The raw interferometer data (Fig. 7.8) is passed through spectral kurtosis and spectral flatness to investigate the effect of daisy-chaining. The cleaned spectrum is shown in Fig. 7.42 with the resulting interference fringe plot shown in Fig. 7.43. The resulting signal-to-noise ratio is 7.27 dB, the same as using spectral kurtosis on its own.

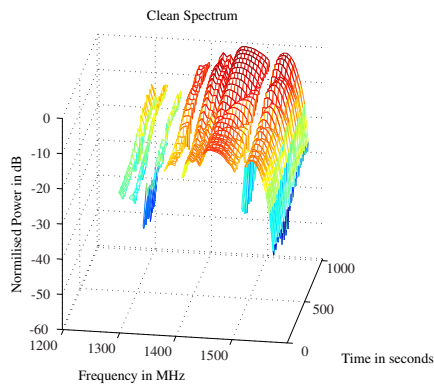


Figure 7.42: The clean interferometer spectrum as observed for a number of seconds after filtered by spectral kurtosis and spectral flatness.

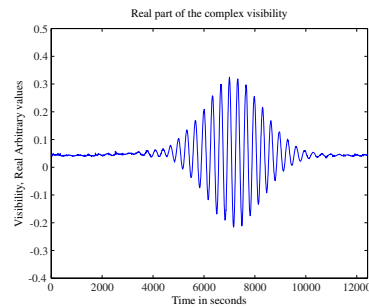


Figure 7.43: The interference fringes as produced from the cleaned spectrum in Fig 7.42.

7.5.2 Spectral Kurtosis to SumThreshold

The raw interferometer data (Fig. 7.8) is passed through spectral kurtosis and SumThreshold to investigate the effect of daisy-chaining. The cleaned spectrum is shown in Fig. 7.44 with the resulting interference fringe plot shown in Fig. 7.45. The resulting signal-to-noise ratio is 7.28 dB. This is due to the small broadband spikes mentioned, that are removed.

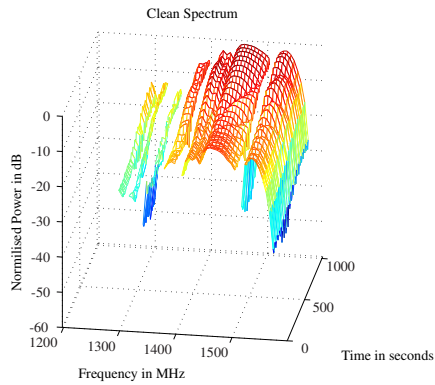


Figure 7.44: The clean interferometer spectrum as observed for a number of seconds after filtered by spectral kurtosis and SumThreshold.

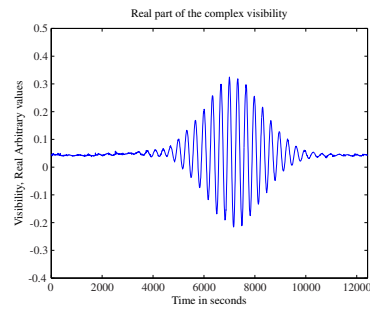


Figure 7.45: The interference fringes as produced from the cleaned spectrum in Fig 7.44.

7.5.3 Spectral Flatness to SumThreshold

The raw interferometer data (Fig. 7.8) is passed through spectral flatness and Sum-Threshold to investigate the effect of daisy-chaining. The cleaned spectrum is shown in Fig. 7.46 with the resulting interference fringe plot shown in Fig. 7.47. The resulting signal-to-noise ratio is 6.27 dB.

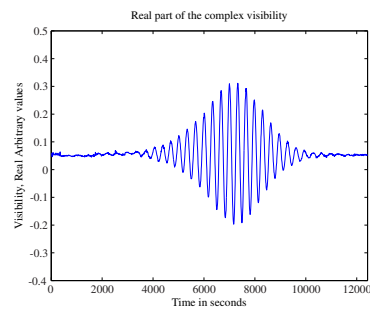
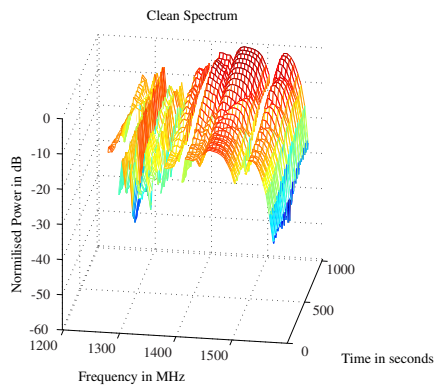


Figure 7.46: The clean interferometer spectrum as observed for a number of seconds after filtered by spectral flatness and SumThreshold.

Figure 7.47: The interference fringes as produced from the cleaned spectrum in Fig 7.46.

Algorithm	SNR (dB)	Run-Time (Seconds)
Unfiltered	-17.67	N/A
Spectral Kurtosis	7.27	0.095
Spectral Flatness	5.94	0.184
VarThreshold	3.10	174.433
SumThreshold	3.18	153.766

Table 7.1: A summary of the algorithm results using the two element interferometer data provided by JP van Renseburg [35]

7.6 Conclusion

The use of spectral kurtosis and spectral flatness in post correlated data allows for a computationally inexpensive operation to detect RFI. The values for these on Gaussian noise for the correlated data, however, does not equal the theoretical pre correlated data values, but differers for each situation. The Var/SumThreshold methods work as expected, except when the power of the observed source varies over time. The run-time results of the Var/SumThreshold methods indicate a significant increase in processing time as shown in Table. 7.1. The run-time of SumThreshold is longer than VarThreshold for the KAT-7 data, but shorter than VarThreshold for the interferometer data. The two datasets are processed differently. The difference in timing can be an effect of inefficient memory usage when manipulating the matrices.

Chapter 8

Conclusion

8.1 Summary

The broad thrust of the work done has been the investigation of RFI mitigation in radio astronomy. The implementation of these techniques relies on the digital back-end of the radio telescope. As part of this work, the design and implementation of a spectrometer on a ROACH1 (shown in Fig. 2.3) which completed for the AVN project was included in this work to permit testing of a radiometer. The AVN radio telescope is also specified to function as a pulsar observation- and timing-system, and for this reason a background study in pulsar theory has been done. This included the software that is used to de-disperse and time pulsar signals, namely Tempo2 and PSRchive [15].

The major components of the digital back-end of a modern radio telescope has been examined in Chapter 3. This included the hardware used to digitise the signal as well as process the signal to a final result. It also included the software used to manage the high data rates.

The RFI mitigation techniques can be split into two major groups, as discussed in Chapter 4; the offline methods, and the online methods. The offline methods are generally more computationally expensive, and thus are run on smaller data sets. The online methods have to be computationally inexpensive, as they are run on the raw data. The online techniques include:

- Spectral Kurtosis.
- Spectral Flatness.
- Active RFI cancelling.
- Median Absolute Deviation.

The offline methods include:

- Var/SumThreshold.

- PSRchive RFI mitigation techniques.

The majority of the tested online methods rely on statistics to determine the nature of the recorded signal. Spectral kurtosis computes the shape of the PSD of the recorded signal, whereas spectral flatness determines the power distribution of the signal. Var/SumThreshold as well as MAD makes use of thresholding to detect strong sources in the recorded signal. This is also implemented in the PSRchive RFI mitigation techniques.

The chosen methods were simulated in Chapter 5 to test the theory and most of the tests are performed in Matlab. These methods were tested on real data provided by a number of sources as indicated in Chapter 6 and 7. The final results were shown and discussed.

8.2 Tests and Measurements

Each of the techniques was tested with actual radio astronomy data. This is to ensure the validity of the test results.

8.2.1 Spectral Kurtosis

The spectral kurtosis estimate as formulated by Nita [2] is a lightweight algorithm that can be implemented alongside a full spectrometer on a standard FPGA based system such as the ROACH1. The algorithm only requires the computation of the power of the signal as well as the square of the power. This will then have to be buffered on the system before it is sent with the recorded data to a computer for processing to simplify the FPGA design.

The timing of the data, as well as the spectral kurtosis estimate, is required to be correct in order to flag the appropriate data. The special cases where spectral kurtosis breaks down will also have to be considered. The 50% duty cycle blind spot is fortunately not significant when the RFI source is a constant, known source.

This technique performs well with short RFI bursts and can easily detect a broadband pulse. If the algorithm is only implemented on the frequency bins, a number of time samples worth of frequency bins will be lost instead of removing the time samples that have been affected. The number of samples lost is equal to the value of M for spectral kurtosis estimation. A trade-off has to be made between ensuring a high signal-to-noise ratio by increasing the value of M , or reducing the number of data bins removed by reducing the value of M .

The spectral kurtosis estimate can also be implemented on post-correlator data as shown in Chapter 7. The value of the spectral kurtosis estimate can, however, differ from one for the valid signal as shown. To compensate for this, a median for the spectral kurtosis estimate can be computed and a deviance chosen to flag data as RFI or not; however, this process breaks down once the RFI contamination is large enough to completely skew the results of the median. It is also blind to a broadband

burst as the value of the spectral kurtosis estimate will be shifted for all frequencies, thus changing the computed median.

8.2.2 Spectral Flatness

Spectral flatness is mostly used for voice detection algorithms, to detect information in a noisy signal. The value of the spectral flatness is also only meant to be computed per spectra, to indicate the start of information in a time series. Applying the algorithm on each frequency bin will result in a breakdown of some of the theory. A source with constant power over time will result in a spectral flatness of one, as the power distribution over time is equal.

The value of the spectral flatness for the correlated data results in a value lower than 1. This can be useful when detecting constant tones, as the tone will not be masked by the valid celestial signal as shown in both cases for spectral flatness in Chapter 7. The spectral flatness of the radiometer power resulted in a value of one for all of the valid frequency bins. A constant tone will thus be masked by the valid signal.

The spectral flatness algorithm is not as sensitive to small power changes as spectral kurtosis, as the computation of both medians can hide the small disturbance.

The computation of the geometric mean will rely on floating point arithmetic and is difficult to implement on a FPGA based system, but easily implemented on a GPU based system.

8.2.3 Var/SumThreshold

Both the VarThreshold and SumThreshold rely on thresholding to determine whether a data set contains RFI. The algorithms pass iteratively through the data to detect stronger power sources between the valid celestial signal. Only the bins that contain the RFI are marked and removed, where as spectral kurtosis or spectral flatness removes the whole frequency bin for all time samples. This allows for a much more precise selection of the RFI.

The algorithms assume that the power of the observed source stays constant for all time. This poses a problem for the interferometer data as supplied by [35]. As the sun passes through the focus of the interferometer, the power detected increases. This results in an uneven power plot for the valid data. The threshold for the algorithms cannot be set lower than the peak of the valid data, which in turn allows weaker RFI to remain even after filtering. The value of the first threshold χ_1 must also be set by hand.

8.2.4 PSRchive Techniques

Both the techniques require user intervention in order to fully utilise the algorithms, as the length of the windows must be set. The techniques are integrated in PSRchive

which makes the application simple once PSRchive is used for pulsar timing.

If either of the two algorithm's windows are set too small, valid pulsar data will be flagged as RFI.

8.2.5 To Conclude

Although spectral kurtosis and spectral flatness require less user intervention function on new data sets than Var/SumThreshold, some user interaction is still required. The thresholds for the spectral flatness and kurtosis must be adjusted to ensure more precise RFI removal. Spectral kurtosis provides a more stable testing platform for online monitoring than spectral flatness, as the blind spots for spectral kurtosis are less likely to occur on a radio telescope.

The signal-to-noise ratio tests indicate that spectral kurtosis was able to flag more valid RFI than any of the other techniques. Spectral kurtosis also outperformed spectral flatness in run-time tests, by nearly half the time. The overall computation time for SumThreshold is extremely large compared to both spectral flatness and spectral kurtosis. The run-time could potentially be decreased using, a GPU based approach, an AVX approach on a standard x86 CPU or simply refining the algorithm. This will, however, not compensate for SumThreshold's lack in the detection of RFI with less power than the observed signal power.

8.3 Future Development

The use of spectral flatness and spectral kurtosis on a per spectrum basis, in order to flag broadband bursts can be investigated. The data can then be passed through a spectral flatness or kurtosis algorithm twice. The first iteration will review the data on a per frequency bin bases and the second iteration review the data on a per time bases. Two iterations of spectral flatness or kurtosis will still be significantly less costly in time than Var/SumThreshold as indicated by the run-time results.

An automatic level detection can be implemented for Var/SumThreshold to determine the value of χ_1 . Using the value of the median to determine χ_1 was briefly considered, but did not work with the interferometer data as the change in power resulted in valid data being thrown away.

The handling of the flagged data will have to be considered for large telescopes. If data loss is acceptable, the flagged bins can be removed as the data is captured. If data loss is not acceptable, an extra data set will have to be stored to indicate whether the bins are flagged as RFI or not. This will introduce extra storage that may not be practical, depending on the setup, although KAT-7 implements a system to indicate signal integrity.

List of References

- [1] O’SULLIVAN JOHN DAVID, DANIELS GRAHAM ROSS, P.T.M.P.O.D.I.D.J.F.: Apparatus and methods for wireless communications. 1994.
Available at: http://worldwide.espacenet.com/publicationDetails/biblio?CC=EP&NR=0599632&KC=&FT=E&locale=en_EP#
- [2] Nita, G., Gary, D., Liu, Z., Hurford, G. and White, S.: Radio Frequency Interference Excision Using Spectral Domain Statistics. *Publications of the Astronomical Society of the Pacific*, , no. 1983, pp. 805–827, 2007.
Available at: <http://www.jstor.org/stable/10.1086/520938>
- [3] Dubnov, S.: Generalization of spectral flatness measure for non-gaussian linear processes. *Signal Processing Letters, IEEE*, vol. 11, no. 8, pp. 698–701, August 2004. ISSN 1070-9908.
Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1316889>
- [4] Offringa, a.R., de Bruyn, a.G., Biehl, M., Zaroubi, S., Bernardi, G. and Pandey, V.N.: Post-correlation radio frequency interference classification methods. *Monthly Notices of the Royal Astronomical Society*, vol. 167, pp. 155–167, March 2010. ISSN 00358711.
Available at: <http://mnras.oxfordjournals.org/cgi/doi/10.1111/j.1365-2966.2010.16471.x>
- [5] NRAO: Introduction to radio astronomy. 2011.
Available at: <http://www.cv.nrao.edu/course/astr534/IntroRadioastro.html>
- [6] Wilson, T.L.: Techniques of radio astronomy. 2011. Cite arxiv:1111.1183 Comment: 47 pages, 6 Figures; to appear in Planets, Stars and Stellar Systems.
Available at: <http://arxiv.org/abs/1111.1183>
- [7] ICASA: South african table of frequency allocations. Tech. Rep., 2008.
- [8] on Frequency Allocations, P. and Spectrum Protection for Scientific Uses, Committee on Radio Frequencies, N.R.C.: *Handbook of Frequency Allocations and Spectrum Protection for Scientific Uses*. The National Academies Press, 2007. ISBN 9780309103015.
Available at: http://www.nap.edu/openbook.php?record_id=11719

- [9] Kraus, J.D., Tiuri, M. and Räisänen, A.V.: *Radio astronomy*. Cygnus-Quasar Books, 1986.
Available at: <http://books.google.com/books?id=KtVFAQAIAAJ&pgis=1>
- [10] Proakis, J. and Manolakis, D.: *Digital Signal Processing: Principles, Algorithms, and Applications*. Simon & Schuster Books For Young Readers, 1992. ISBN 9780023968150.
Available at: <http://books.google.co.za/books?id=ywgfaQAIAAJ>
- [11] Thompson, N.C.: Development of a two element interferometer. Tech. Rep., Stellenbosch University, December 2011. Final year project for BEng degree.
- [12] Clark, B.G.: Coherence in radio astronomy. In: *Synthesis Imaging in Radio Astronomy II*, vol. 180, p. 1. 1999.
Available at: <http://adsabs.harvard.edu/full/1999ASPC..180....1C>
- [13] Lorimer, D.R. and Kramer, M.: *Handbook of Pulsar Astronomy*. Cambridge University Press, 2004. ISBN 0521828236.
Available at: <http://books.google.com/books?hl=en&lr=&id=OZ8tdN6qJcsC&pgis=1>
- [14] Hewish, A., Bell, S., Pilkington, J., Scott, P. and Collins, R.: Observation of a rapidly pulsating radio source. *Nature*, vol. 217, p. 709, 1968.
Available at: <http://www.astro.ubc.ca/people/richer/a520/HewishEtal1968.pdf>
- [15] Matthew Bailes, Jagmit Sandhu, e.a.: PSRchive.
Available at: <http://psrchive.sourceforge.net/>
- [16] HartRAO: 2013.
Available at: <http://www.hartrao.ac.za/>
- [17] Stuart, A. and Ord, K.: *Kendall's Advanced Theory of Statistics: Volume 1: Distribution Theory*. No. v. 1; v. 1994 in Kendall's advanced theory of statistics. Wiley, 2009. ISBN 9780340614303.
Available at: <http://books.google.co.za/books?id=tW18thQWJQIC>
- [18] CASPER: Roach. URL <https://casper.berkeley.edu/wiki/ROACH>, 2011.
- [19] CASPER: Katadc. URL <https://casper.berkeley.edu/wiki/KatADC>, 2011.
- [20] Harvard: Leda, large-aperture experiment to detect the dark ages.
Available at: <http://www.cfa.harvard.edu/LEDA/technology.html>
- [21] NRAO: Guppi green bank ultimate pulsar processing instrument. 2011.
Available at: <https://safe.nrao.edu/wiki/bin/view/CICADA/GUPPISupportGuide>
- [22] Willem van Straten, Andrew Jameson, e.a.: PSRdada.
Available at: <http://psrdada.sourceforge.net/>

- [23] CASPER: Casper home page. URL <https://casper.berkeley.edu/>, 2009.
- [24] Peebles, P.Z.: *Probability, Random Variables and Random Signal Principles*. McGraw-Hill, 2001. ISBN 0071181814.
- [25] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: *Numerical Recipes with Source Code CD-ROM 3rd Edition: The Art of Scientific Computing*. 3rd edn. Cambridge University Press, 9 2007. ISBN 9780521884075.
Available at: <http://amazon.com/o/ASIN/0521884071/>
- [26] Dwyer, R.: Detection of non-Gaussian signals by frequency domain kurtosis estimation. *Acoustics, Speech, and Signal Processing, IEEE*, pp. 607–610, 1983.
Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1172264
- [27] Vrabie, V., Granjon, P. and Serviere, C.: Spectral kurtosis: from definition to application. *Proceedings of the 6th*, , no. Nsip, 2003.
Available at: <http://hal.archives-ouvertes.fr/hal-00021302/>
- [28] Antoni, J.: The spectral kurtosis: a useful tool for characterising non-stationary signals. *Mechanical Systems and Signal Processing*, vol. 20, no. 2, pp. 282–307, February 2006. ISSN 08883270.
Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0888327004001517>
- [29] Papoulis, A.: *Probability, Random Variables, and Stochastic Processes*. Communications and signal processing. McGraw-Hill, Incorporated, 1991. ISBN 9780070484771.
Available at: <http://books.google.co.za/books?id=4IwQAQAIAAJ>
- [30] Demorest, P. and Os, S.: Pulsar data analysis with PSRCHIVE. *Astronomical Research and Technology*, vol. 9, no. 3, pp. 237–256, 2012. arXiv:1205.6276v1.
Available at: <http://arxiv.org/pdf/1205.6276v1.pdf>
- [31] Baburaj, N.: Development of MAD based spectral domain RFI filtering techniques. , no. April, 2013.
Available at: <http://ncralib1.ncra.tifr.res.in:8080/jspui/handle/2301/591>
- [32] Briggs, F.H. and Kocz, J.: Overview of Technical Approaches to RFI Mitigation. *Radio Science*, p. 11, February 2005. 0502310.
Available at: <http://arxiv.org/abs/astro-ph/0502310>
- [33] Kumar, S.: Direct sequence spread spectrum. 2009.
Available at: http://www.mathworks.com/matlabcentral/fileexchange/24491-direct-sequence-spread-spectrum/content/Direct_sequence_spread_Spectrum.m
- [34] SKA-South-Africa: Kat7 data. 2013. Dataset 1356555246.
Available at: <http://www.ska.ac.za/contactus/index.php>

- [35] Jansen Van Rensburg, J.: *The design of a two-element correlation interferometer operating at L-band*. Master's thesis, Stellenbosch University, December 2012.
- [36] et al., M.B.A.P.R.M.M.K.A.L.M.M.: Parkes observations for project p455 semester 2011aprs. June 2011.
Available at: <https://data.csiro.au/dap/landingpage?pid=csiro:P455-2011APRS&v=2&d=true>

Appendices

Appendix A

Spectral Variability of a Windowed Time Domain Gaussian Signal

The derivation of the spectral variability is done in [2] as follows.

Equation 4.1.7 is firstly split up in to the following form.

$$\hat{P}_k = A_k^2 + B_k^2 \quad (\text{A.0.1})$$

where

$$A_k = \sqrt{\frac{2}{W_{ss}}} \sum_{n=0}^{N-1} w_n x_n \cos\left(\frac{2\pi kn}{N}\right) \quad (\text{A.0.2})$$

$$B_k = \sqrt{\frac{2}{W_{ss}}} \sum_{n=0}^{N-1} w_n x_n \sin\left(\frac{2\pi kn}{N}\right) \quad (\text{A.0.3})$$

As seen in [24, p89], a linear combination of a zero mean Gaussian signal will also produce a zero mean output. The result is that equations (A.0.2) and (A.0.3) are also zero mean with a computable variance. This is shown in [17, p350] using equation (A.0.4).

$$\sigma_{g(x_1, \dots, x_n)}^2 = \sum_{i,j=0}^{N-1} \frac{\partial g}{\partial m_i} \frac{\partial g}{\partial m_j} \text{Cov}(x_i, x_j) \quad (\text{A.0.4})$$

$\sigma_{g(x_1, \dots, x_n)}^2$: The variance estimate.

$\frac{\partial g}{\partial m_i}$: Partial derivative that is evaluated at the mean value $m_i = \langle x_i \rangle$.

$\text{Cov}(x_i, x_j)$: The covariance of the random Gaussian variables.

As the variables are Gaussian and independent, the covariance will evaluate to zero for all values of i and j , except where $i = j$. The covariance will result in the variance σ_x^2 at $i = j$. The answer for equation (A.0.4) can now be computed for (A.0.2) and (A.0.3) as follows:

$$\begin{aligned}
 \sigma_{A_k}^2 &= \sum_{n=0}^{N-1} \left(\frac{\partial A}{\partial \langle x_n \rangle} \sigma_x \right)^2 \\
 &= \left(\sqrt{\frac{2\sigma_x^2}{N \sum_{n=0}^{N-1} w_n^2}} \right)^2 \text{sum}_{n=0}^{N-1} \left(w_n \cos \left(\frac{2\pi kn}{N} \right) \right)^2 \\
 &= \frac{2\sigma_x^2}{N \sum_{n=0}^{N-1} w_n^2} \sum_{n=0}^{N-1} \left(w_n^2 \frac{1 + \cos \left(\frac{4\pi kn}{N} \right)}{2} \right) \\
 &= \frac{\sigma_x^2}{N \sum_{n=0}^{N-1} w_n^2} \sum_{n=0}^{N-1} \left(w_n^2 + w_n^2 \cos \left(\frac{4\pi kn}{N} \right) \right) \\
 &= \left[1 + \Re(W_{2k}) \right] \frac{\sigma_x^2}{N}
 \end{aligned} \tag{A.0.5}$$

Here,

$$\Re(W_{2k}) = \sum_{n=0}^{N-1} \frac{w_n^2 \cos \left(\frac{4\pi kn}{N} \right)}{\sum_{n=0}^{N-1} w_n^2} \tag{A.0.6}$$

which is the real part of the normalised Fourier transform of the windowing function

$$W_k = \frac{1}{\sum_{n=0}^{N-1} w_n^2} \sum_{n=0}^{N-1} w_n^2 e^{-2\pi i kn/N} \tag{A.0.7}$$

The same hold for B_k .

$$\sigma_{B_k}^2 = \left[1 - \Re(W_{2k}) \right] \frac{\sigma_x^2}{N} \tag{A.0.8}$$

The values of $W_N = W_0 = 1$ for all windowing functions, as this is part of the definition. This results in a simple answer for DC and Nyquist frequencies $\sigma_{A_0}^2 = \sigma_{B_0}^2 = \frac{2\sigma_x^2}{N}$. We also already know that A_k and B_k are zero mean random variables, as per definition. This implies that $\langle A_k^2 \rangle = \sigma_{A_k}^2$ and $\langle B_k^2 \rangle = \sigma_{B_k}^2$, and can now be used to compute the expected mean of the PSD estimate $\mu_k \equiv \langle \hat{P}_k \rangle$ as shown in (A.0.9).

$$\begin{aligned}
 \mu_k = \langle \hat{P}_k \rangle &= \langle A_k^2 \rangle + \langle B_k^2 \rangle = \frac{2\sigma_x^2}{N} \\
 &\text{for } k = 0, 1, \dots, \frac{N}{2}
 \end{aligned} \tag{A.0.9}$$

As expected, the power is shown to be equally distributed across all channels of the PSD estimate. The power is however not equally distributed between the real and imaginary part, as can be seen in equations (A.0.5) and (A.0.8), but is influenced by the Fourier transform as well as the windowing function. It can also

be seen from equation (A.0.3) that for $k = 0$ and $k = \frac{N}{2}$, $B_0 = B_{\frac{N}{2}} = 0$. This means that the probability distribution function for the PSD can be computed using only A_k as a standard Gaussian distribution.

$$p(A_k) = \frac{1}{\sqrt{2\pi\sigma_{A_k}^2}} e^{\left(-\frac{A_k^2}{2\sigma_{A_k}^2}\right)} \quad (\text{A.0.10})$$

Substituting \hat{P}_k for A_k in (A.0.10) results in the following.

$$p(\hat{P}_k) = \frac{1}{\sqrt{2\pi\sigma_{A_k}^2 \hat{P}_k}} e^{\left(-\frac{\hat{P}_k}{2\sigma_{A_k}^2}\right)} \quad (\text{A.0.11})$$

The n th-order moment about the origin of the PSD estimate can be computed using (A.0.11) as follows.

$$\begin{aligned} \mu'_k(n) = \langle \hat{P}_k^n \rangle &= 1 \times 3 \dots \times (2n-1) \sigma_{A_k}^{2n} \\ &\text{for } k = 0, \frac{N}{2} \end{aligned} \quad (\text{A.0.12})$$

This follows that the variance of the PSD estimate can be computed at $k = 0, \frac{N}{2}$.

$$\sigma_{\frac{N}{2}}^2 = \sigma_0^2 = \mu'_0(2) - \mu'_0(1)^2 = \frac{8\sigma_x^4}{N^2} \quad (\text{A.0.13})$$

Having computed both σ_k^2 and μ_k^2 the value of V_k (See (2.6.3)) for $k = 0, \frac{N}{2}$ can now be computed using (2.6.3).

$$V_{\frac{N}{2}}^2 = V_0^2 = 2 \quad (\text{A.0.14})$$

To compute the spectral variability for all other frequencies, the PSD estimate for both A_k and B_k will have to be computed. This is again done using the Taylor expansion in [17, p350].

$$\text{Cov}(f, g) = \sum_{i,j=0}^{N-1} \frac{\partial f}{\partial m_i} \frac{\partial g}{\partial m_j} \text{Cov}(x_i, x_j) \quad (\text{A.0.15})$$

As in equation A.0.4, the covariance of A_k and B_k will be zero for $j \neq i$ and where $i = j$ the result is $\text{Cov}(x_i, x_j) = \sigma_x^2$. Replacing (A.0.2) and (A.0.3) in (A.0.15)

results in the following answer.

$$\begin{aligned}
 \text{Cov}(A_k, B_k) &= \sum_{n=0}^{N-1} \frac{\partial A_k}{\partial \langle x_n \rangle} \frac{\partial B_k}{\partial \langle x_n \rangle} \sigma_x^2 \\
 &= \frac{2}{N \sum_{n=0}^{N-1} w_n^2} \sum_{n=0}^{N-1} w_n^2 \cos\left(\frac{2\pi kn}{N}\right) \sin\left(\frac{2\pi kn}{N}\right) \sigma_x^2 \\
 &= \frac{\sigma_x^2}{N \sum_{n=0}^{N-1} w_n^2} \sum_{n=0}^{N-1} w_n^2 \sin\left(\frac{4\pi kn}{N}\right) \\
 &= \Im(W_{2k}) \frac{\sigma_x^2}{N}
 \end{aligned} \tag{A.0.16}$$

where

$$\Im(W_{2k}) = \frac{\sum_{n=0}^{N-1} w_n^2 \sin\left(\frac{4\pi kn}{N}\right)}{\sum_{n=0}^{N-1} w_n^2} \tag{A.0.17}$$

This forms the imaginary part of the Fourier transform of the square windowing function and is computed at frequencies $f_{2k} = 2f_k$. For a symmetrical window, the result of $\Im W_{2k} = 0$ as can be seen from the (A.0.17). This result implies that A_k and B_k is statistically independent, but in Sec. 4.1.1 it was already mentioned that any asymmetry in the windowing function will result in a correlation between A_k and B_k . To find the result for the cases where $k \neq 0, \frac{N}{2}$, the Gaussian-Bivariate function is used as shown in [24, p440].

$$p(A_k, B_k) = \frac{1}{2\pi\sigma_{A_k}\sigma_{B_k}\sqrt{1-\rho_k^2}} \times \exp\left[\frac{1}{1-\rho_k^2}\left(-\frac{A_k^2}{2\sigma_{A_k}^2} - \frac{B_k^2}{2\sigma_{B_k}^2} - \rho_k \frac{A_k B_k}{\sigma_{A_k}\sigma_{B_k}}\right)\right] \tag{A.0.18}$$

where

$$\rho_k = \frac{\sigma_{A_k B_k}^2}{\sigma_{A_k}\sigma_{B_k}} \tag{A.0.19}$$

To simplify the evaluation of (A.0.18), the moment-generating function is used.

$$M(t) = \langle e^{(A_k^2 + B_k^2)} \rangle = \frac{1}{\sqrt{1 - 2(\sigma_{A_k}^2 + \sigma_{B_k}^2)t + 4\sigma_{A_k}^2\sigma_{B_k}^2(1 - \rho_k^2)t^2}} \tag{A.0.20}$$

The moments can then simply be calculated using the following equation.

$$\mu_k'(n) = \frac{d^n}{dt^n} \Big|_{t=0} M(t) \tag{A.0.21}$$

With the above (A.0.21), the variance of the PSD can be computed for all values of $k \neq 0, \frac{N}{2}$.

$$\sigma_k^2 = 2(\sigma_{A_k}^4 + \sigma_{B_k}^4 + 2\sigma_{A_k B_k}^2) \tag{A.0.22}$$

It can however also be shown that (A.0.22) holds for $k = 0, \frac{N}{2}$, as the following is already proven.

$$\begin{aligned}\sigma_{B_0} &= \sigma_{B_{\frac{N}{2}}} = 0 \\ \sigma_{A_0 B_0} &= \sigma_{A_{\frac{N}{2}} B_{\frac{N}{2}}} = 0 \\ \sigma_{A_0}^2 &= \sigma_{A_{\frac{N}{2}}}^2 = \frac{2\sigma_x^2}{N}\end{aligned}$$

With the results of (A.0.22) and (A.0.9), the answer for (2.6.3) can now be computed for $k = 0, 1, \dots, \frac{N}{2}$.

$$V_k^2 = \frac{2(\sigma_{A_k}^4 + \sigma_{B_k}^4 + 2\sigma_{A_k B_k}^2)}{(\sigma_{A_k}^2 + \sigma_{B_k}^2)^2} \quad (\text{A.0.23})$$

Equation A.0.23 can then be simplified using (A.0.5), (A.0.8) and (A.0.16) to the following simple answer.

$$\begin{aligned}V_k^2 &= \frac{2([1 + \Re(W_{2k})]^2 \frac{\sigma_x^4}{N^2} + [1 - \Re(W_{2k})]^2 \frac{\sigma_x^4}{N^2} + 2\frac{\sigma_x^4}{N^2} \Im(W_{2k}))}{\frac{\sigma_x^4}{N^2} ([1 + \Re(W_{2k}) + [1 - \Re(W_{2k})]])^2} \\ &= 1 + |W_{2k}|^2\end{aligned} \quad (\text{A.0.24})$$

It is also shown that $W_0 = W_n = 1$, which in turn corresponds with the results obtained in (A.0.14). This implies that (A.0.24) holds for all values of $k = 0, 1, \dots, \frac{N}{2}$.

Appendix B

Variance derivation of the Spectral Kurtosis estimator

The following derivation is done in [2] and starts by rewriting (4.1.24).

$$\hat{V}_k^2 = \frac{M}{M-1} \left(\frac{m'_2}{m'^2_1} - 1 \right) \quad (\text{B.0.1})$$

where

$$m'_1 = \frac{S_1}{M} \quad (\text{B.0.2})$$

$$m'_2 = \frac{S_2}{M} \quad (\text{B.0.3})$$

\hat{V}_k^2 : Spectral Kurtosis estimator.

m'_n : The nth moment about the origin.

M : The number of spectral samples.

(A.0.4) can then be used to compute the variance of \hat{V}_k^2 .

$$\text{Var}(\hat{V}_k^2) = \left(\frac{M}{M-1} \right)^2 \text{Var} \left(\frac{m'_2}{m'^2_1} \right) \quad (\text{B.0.4})$$

This simplifies the problem to only computing the variance of the ratio $\frac{m'_2}{m'^2_1}$. The variance can be computed using the equation on [17, p351].

$$\begin{aligned} \text{Var} \left(\frac{m'_2}{m'^2_1} \right) &= \frac{\text{Var}(\mu'_2)}{\mu'^4_1} + \frac{\mu'^2_2 \text{Var}(m'^2_1)}{\mu'^6_1} - \frac{2\mu'_2 \text{Cov}(\mu'_2, \mu'^2_1)}{\mu'^5_1} \\ &= \frac{\text{Var}(m'_2)}{\mu'^4_1} + \frac{4\mu'^2_2 \text{Var}(\mu'_1)}{\mu'^6_1} - \frac{4\mu'_2 \text{Cov}(\mu'_2, \mu'_1)}{\mu'^5_1} \end{aligned} \quad (\text{B.0.5})$$

Again from [17], the equation can be simplified.

$$\text{Var}\left(\frac{m'_2}{m'_1}\right) = \frac{1}{M} \left\{ \frac{\mu'_4 - \mu'^2_2}{\mu'^4_1} - \frac{4\mu'_2\mu'_3}{\mu'^5_1} + \frac{4\mu'_1}{\mu'^6_1} \right\} \quad (\text{B.0.6})$$

If $k = 0$, $\frac{N}{2}$ (B.0.6) can be simplified even further to.

$$\text{Var}\left(\frac{m'_2}{m'_1}\right) = \frac{24}{M} \quad (\text{B.0.7})$$

For $k \neq 0$, $\frac{N}{2}$ (B.0.6) can be rewritten in terms of the windowing function.

$$\text{Var}\left(\frac{m'_2}{m'_1}\right) = \frac{4}{M} (1 + 5|W_{2k}|^2 - |W_{2k}|^4 + |W_{2k}|^6) \quad (\text{B.0.8})$$

Equation (B.0.8) however also holds for $k = 0$, $\frac{N}{2}$ as $|W_0| = |W_N| = 1$.

The variance of the spectral kurtosis estimator can now be written as.

$$\text{Var}(\hat{V}_k^2) = \frac{4M}{(M-1)^2} (1 + 5|W_{2k}|^2 - |W_{2k}|^4 + |W_{2k}|^6) \quad (\text{B.0.9})$$