Data Challenges In Pulsar Searches

Elmarie van Heerden

Lady Margaret Hall College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2017

Yes, everything else is worthless when compared with
the infinite value of knowing Christ Jesus my Lord.

Philippians 3:8

This thesis is dedicated
to my parents,
Bernard and Amanda.
Thank you for loving me unconditionally.

Acknowledgements

I would like to extend my greatest appreciation for the support and guidance that I received from my supervisors, Prof. Steve Roberts and Dr. Aris Karastergiou. They were patient when I was unsure, they were calm when I panicked and they were always willing to listen even at the most unreasonable of times. Prof. Roberts, thank you for frequently reminding me of the bigger picture and for helping me keep focus. Dr. Karastergiou, thank you for spending countless hours with me combing through my code and for bearing with my lack of database structure and weird file naming conventions.

I would like to thank the Commonwealth Scholarship Commission of the U.K. for giving me the opportunity to pursue my post-graduate studies at Oxford and for supporting me financially. Without their generous support I would not have been able to do a DPhil in the U.K..

To my cousin Belinda, thank you for feeding me, for checking up on me and for helping me to settle into life in the U.K..

To Esté-Marié and Shiyang, thank you for always believing in me, for always checking up on me even though we are separated by continents and for genuinely caring about my well-being. I am blessed and honoured to have you as friends. I cannot wait for the next time we see each other and for all the years and adventures to come.

To my dear friends, Justin (Dragon), Diane (Frenchie), Ali, Jonny and Marisa, our bonds of friendship have been forged through hysterical laughter, green scarves, broken body parts, sharing of good food and wine, late nights in the

lab, countless hours of talking about the future and the adventures we plan to go on, warm hugs, sexy titles and many Barefoot coffee runs. Writing a DPhil is as much an emotional journey as an intellectual one; I thank you for listening to my woes, sharing in my excitement and for enriching my life day in and day out over the last three years.

To all those who prayed for me not to lose heart, to stay humble and to remind me that I have a bigger purpose, I thank you. This thesis is a testimony that God answers prayers and of His greatness.

To my brothers, Ian and Jacques, who shaped me in my younger years and who taught me to be tough and to stand my ground, I thank you. My success is your success.

I dedicate this thesis to my parents, Bernard and Amanda van Heerden, whose unfailing encouragement, daily hour long phone calls, prayers, trust and love enabled me to write a doctoral thesis which must be among life's rarest privileges. From my parents, I have learned how to love and to be loved.

Abstract

Technological advances coupled with a decline in digital storage costs have resulted in a profusion of data being created, collected and consumed. These data give rise to new challenges and opportunities in many disciplines ranging from science and engineering to biology and finance.

An example of a future project in radio astronomy that promises both Big Data and Big Discoveries is the Square Kilometre Array (SKA) radio telescope project. Astrophysicists are confident that the Big Data amassed by the SKA will not only answer fundamental questions regarding the Universe but also contain big discoveries not yet postulated. The transformational potential of the SKA and its ensuing data and algorithmic challenges, in particular for the discovery and study of pulsars, drive the research of this thesis.

Discovering all pulsars beaming towards Earth is one of the key science goals of the SKA. However, in addition to low signal strengths, searching for pulsars is extremely difficult due to the intrinsic weakness of their signals, propagation effects and the presence of anthropogenic interferences. Numerous techniques have been developed to overcome some of these difficulties and to assist in the quest to find more pulsars. However, despite the success of these techniques, the number of pulsars discovered in recent surveys (Swiggum et al. 2014, Lazarus et al. 2015) has fallen well short of the number predicted by pulsar population synthesis models (Lorimer 2011). This shortfall in pulsar detections can be attributed to radio frequency interference (RFI), red noise and scintillation (Lazarus et al. 2015).

For this thesis, and in order to investigate and quantify these claims, I first developed a new technique to simulate pulsar search data that contain different types of RFI and varying noise baselines (i.e. red noise). This surrogate modelling technique was then used in a framework that I developed to inexpensively explore the sensitivity of pulsar search pipelines for different noise and RFI settings. The results from this framework highlight the necessity to develop algorithms that are able to identify and remove non-stationary variations from the data before RFI excision and searching is performed in order to limit false positive detections.

To address the shortcomings identified with the framework which assessed the performance of existing pulsar search pipelines, I developed a new real-time algorithm for excising RFI while simultaneously normalising the variability in time and frequency inherent to pulsar observations. Processing synthetic data with the algorithm resulted in an expansion of the noise/pulsar spin period parameter space for which we are able to successfully detect pulsars. Furthermore, the algorithm is shown to reduce the number of false positive detections.

In conclusion, the insights gained from the work presented in this thesis and the improvements achieved will contribute to the development of a new real-time pulsar search pipeline adept at dealing with the challenges posed by the SKA.

Statement of Originality

I declare that no part of this thesis has been, or is being, submitted for any qualification other than the degree of Doctor of Philosophy at the University of Oxford.

This thesis is the result of my own work unless otherwise stated.

Chapters of this work have been published in part as a paper in the journal below. This project is fully my own. I wrote the relevant code for generating pulsar search data and conducted the data analysis. I am the main author of the manuscript. Prof. Roberts and Dr. Karastergiou contributed through their supervision and advice.

Chapter 5, 6 and part of Chapter 8:

A framework for assessing the performance of pulsar search pipelines, E. van Heerden, A. Karastergiou and S. J. Roberts. Monthly Notices of the Royal Astronomical Society, May 2017, volume: 467 (2), pages: 1661–1677.

Elmarie van Heerden, August 2017

Contents

List of Figures

List of Tables

Chapter 1

Big discoveries from Big Data

1.1 Preamble

The Era of Big Data is under way. The explosion and profusion of available data in a wide range of application domains give rise to new challenges and opportunities in a plethora of disciplines ranging from science and engineering to biology and business. Large data sets of information are indisputably being amassed as a result of our social, mobile, and digital world. According to the International Data Corporation (IDC), in 2011, the overall created and copied data volume in the world was 1.8 Zettabytes ($\approx 10^{21}$ bytes), which has increased by nearly nine times within five years (Gantz & Reinsel 2011). In the future, describing the extent of data with terms like Petabyte (PB), Exabyte (EB), and Zettabyte (ZB) will be the norm (Dobre & Xhafa 2014).

There are various domains that generate Big Data which include but, are not limited to:

- *Data from scientific measurements and experiments* - A five hour observation with the Low-Frequency Array (LOFAR) amounts to 1.2 TB of raw data. The input data rate of the streaming processing section of the LOFAR central system is 25 000 TB/day (de Vos et al. 2009).

- *Computational biology* - GenBank is a nucleotide sequence database maintained by the U.S. National Bio-Technology Innovation Centre. Data in this database doubles

every 10 months. By August 2009, GenBand had more than 250 billion bases from 150 000 different organisms (Bryant 2011).

- *Social networking* - Facebook reported that it was processing 2.5 billion pieces of content (links, comments, etc.), 2.7 billion 'Like' actions and 300 million photo uploads *per day* (Constine 2012).

- *Retail* - In 2012, Walmart reported that it was generating more than 2.5 PB of data relating to more than one million customer transactions *every hour* (Open Data Center Alliance 2012).

- *Advertising* - Akamai analyses 75 million events per day for its target advertisement (Zikopoulos et al. 2011).

The above examples demonstrate the increase in Big Data applications where data collection have grown beyond the ability of current software tools to capture, manage, and process in near real-time. The most fundamental challenge for Big Data applications is how to take advantage of the unprecedented scale of data in order to acquire further insights and knowledge of the Universe, improve the efficiencies of enterprises as well as the quality of human lives (Leskovec et al. 2014). In many situations, the knowledge extraction process needs to be very efficient and close to real-time mainly because the storage of all observed data is neither physically nor economically feasible.

Big Data are simply not denoted by *volume* alone, but it is also characterised by being generated on a continuous basis, seeking to be exhaustive, fine-grained in scope, and flexible and scalable in its production. The sheer *volume* of the data on its own, is of course a major challenge, and it is the one characteristic most easily recognised. However, there are other characteristics: *variety*, *velocity* and *veracity* (Laney 2001). *Variety* of data refers to the heterogeneity of data types, representation, and semantic interpretation. *Velocity* of data denotes both the rate at which data arrive at the collection point and the time frame in which the data must be acted upon. *Veracity* of data refers to the accuracy of data which

is influenced by low signal to noise ratios, low-fidelity signals providing biased estimates of desired quantities and incomplete data that complicate or hinders the extraction of information from the data.

The defining quality of these four '*V*'s is that they prohibit a simple scaling of existing approaches to the acquisition, management, analysis and interpretation of data, but instead demand new approaches to deal with data. The key to new approaches in Big Data management, is the utilisation of intelligent algorithms on sophisticated computing architectures.

In the next section, I highlight the key technical challenges that must be addressed in order to exploit the full potential of Big Data.

1.2 Challenges

Traditionally, data analysis techniques have been designed to extract insights from scarce, static, clean and poorly relational data sets, scientifically sampled and adhering to strict assumptions (such as independence, stationarity, and normality), and generated and analysed with a specific question in mind (Miller 2010). The challenge with analysing Big Data is coping with the abundance, exhaustivity and variety, timeliness and dynamism, messiness and uncertainty and high relationality of Big Data as well as the fact that much of the data being generated, have no specific question in mind or are a by-product of another activity (Kitchin 2014).

While the potential benefits of Big Data are real and significant, there remain many technical challenges that must be addressed before its full potential will be realised.

1.2.1 Heterogeneity

Big Data's complexity is a result of many aspects, including complex heterogeneous data types, complex intrinsic semantic associations in data, and complex relationship networks amongst data. It is a great challenge to effectively describe semantic features in data sets

and to build association models to bridge various heterogeneous data sources. In general, algorithms perform well with homogeneous data but are poor at processing and understanding nuances associated with real data. It is therefore necessary that data should be carefully structured as a first step in data analysis.

An associated challenge in data analysis is to ensure that the correct metadata, that describes the recorded data, is automatically generated. For example, in scientific experiments, considerable detail regarding specific experimental conditions and procedures may be required in order to interpret the results correctly. Hence, metadata acquisition systems can minimise the human burden in recording metadata. Recording the detail of data at its origin is senseless unless it can be interpreted and channelled concurrently with the data through the data analysis pipeline. For example, a processing error at one step can render subsequent analysis useless. With suitable provenance, it might be easy to identify all subsequent processing that is dependent and affected by this step. It is therefore important that data analysis systems should include data provenance in their pipelines.

1.2.2 Inconsistency and incompleteness

Data increasingly include information obtained from diverse sources with varying reliability. Sparse, unreliable and incomplete data are endemic, and must be managed.

Sparsity is normally a complication of data dimensionality issues, where data in a high-dimensional space (such as more than 1,000 dimensions) do not show clear trends or distributions (Wu et al. 2014). For most machine learning and data mining algorithms, high-dimensional sparse data, significantly deteriorate the reliability of the models derived from the data.

Various levels of certainty are another data reality where each data field is not necessarily deterministic any more, but subject to some random/error distributions (Wu et al. 2014). This reality is mainly linked to domain specific applications with inaccurate data readings and collections. For uncertain data, the major challenge is that each datum is no longer

merely represented as a single value but as a distribution with a mean value plus a variance to indicate expected errors. Existing data mining algorithms can therefore not be applied to Big Data directly.

Missing values are another problem in data analysis which is caused by mechanisms such as the malfunctioning of a sensor node, or some systematic designs to intentionally skip some values (e.g. dropping some sensor node readings to save power for transmission) (Wu et al. 2014). While most modern data mining algorithms have built-in mechanisms to deal with missing values (such as ignoring data fields with missing values), data imputation is an established research field that seeks to impute missing values to produce improved models (compared to those built from original data).

Incomplete and erroneous data are likely to remain even after error correction protocols have been applied. It is therefore imperative that these challenges should be addressed during the data analysis stage, despite the fact that it is not easily mitigated. Recent attempts to manage and query probabilistic and conflicting data sets are proving to bear fruits (Jagadish et al. 2014).

1.2.3 Scale

Managing large and rapidly increasing volumes of data has been a challenging issue for many decades. In the past, this challenge was mitigated by faster processors, following Moore's Law. However, there is currently a fundamental shift under way: data volumes are increasing faster than CPU speeds and other computational resources (Jagadish et al. 2014).

Due to power constraints, clock speeds have largely stalled and processors are being built with increasing numbers of cores. In short, one has to deal with parallelism within a single node. Unfortunately, parallel data processing techniques that were applied in the past for processing data across nodes do not directly apply to intranode parallelism, since the architecture is very different. For example, there are many more hardware resources such

as processor caches and processor memory channels that are shared across cores in a single node (Sanchita & Anindita 2016). Compounding this challenge, memory bandwidth also lags behind, meaning that CPUs typically receive data at rates, orders of magnitude, slower than they are capable of processing. No clear direction has emerged to deal with these key issues, which represent some of the greatest challenges in data-intensive computing.

In order to utilise the computing power of new complex computer systems and to advance our scientific understanding, there exists a need to take a fresh look at key algorithms underpinning data analytics. Specialist (parallel) research software developers collaborating with computer scientists and mathematicians are needed to develop novel algorithms and/or improved implementations of existing methods to exploit the latest multi-core processor architectures. New algorithms need to be developed and heterogeneous architectures should be considered as solutions to the increasing disparity between the ability to process data (in the CPU or accelerator) and the ability to access data (from the memory or disk).

Another dramatic shift under way is the move towards cloud computing, which aggregates multiple disparate workloads with varying performance goals into very large clusters. This level of sharing of resources on expensive and large clusters, stresses existing grid and cluster computing techniques, and requires new ways of determining how to run and perform data processing in order to meet the goals of each workload cost-effectively, and to deal with system failures, which occur more frequently when operating on larger and larger systems.

1.2.4 Timeliness

As data increase in volume, there is a need for real-time techniques to process and analyse Big Data. For example, if a fraudulent credit card transaction is suspected, it should ideally be flagged before the transaction is completed - potentially preventing the transaction from taking place at all. Obviously, a full analysis of a user's purchase history is unlikely to be achievable in real-time. Rather, partial results need to be available in advance such that

only incremental computation with the new data is required to resolve the matter faster. The fundamental challenge is to provide interactive response times to complex queries at scale over high-volume event streams.

Much of Big Data can be filtered and compressed by orders of magnitude making it economically viable to store. A challenge with filtering Big Data is defining on-line filters in such a way that they do not compromise inference about the underlying activity of interest by discarding useful information.

Another emergence is to find elements in a very large data set that meet a specified criterion. In the course of data analysis, this type of search is likely to occur repeatedly. Scanning the entire data set to find suitable elements is obviously impractical. Instead, index structures can be created in advance to find qualifying elements faster. For example, consider a traffic management system with information regarding thousands of vehicles and local hot spots on roadways. The system may need to predict potential congestion points along a route chosen by a user, and suggest alternatives. Doing so requires evaluating multiple spatial proximity queries working with the trajectories of moving objects. Consequently, there is a need to devise new index structures to support a wide variety of such criteria.

1.2.5 Privacy and data ownership

The privacy of data is another huge concern, and one that increases in the context of Big Data. For electronic health records, there are strict laws governing which data can be revealed in different contexts. For other data, regulations, particularly in the U.S., are less forceful. However, there is great public fear regarding the inappropriate use of personal data, particularly through the linking of data from multiple sources. Managing privacy effectively is both a technical and a sociological problem, which should be addressed jointly from both perspectives to realise the promises of Big Data.

Another issue is that today, many online services require individuals to share private

information (for instance Facebook applications), but beyond record-level access control, people do not understand what it means to share data, how the shared data can be linked, and how to give users fine-grained control over this sharing in an intuitive, but effective way. In addition, real data are not static but get larger and change over time. None of the prevailing techniques result in any beneficial information being released in this scenario.

1.2.6 Distilling meaning from data

For Big Data to fully reach its potential, it is necessary to not only consider the scale of the processing system, but also the interpretability of the end product from the perspective of humans. It is important to ensure that the end users - humans - can properly 'absorb' the information released by the analysis of Big Data and are not swamped by it. For example, ranking and recommendation algorithms can help to identify the most interesting data for a user, taking into account his/her preferences. However, when these techniques are being used for scientific discovery and exploration, special care must be taken not to imprison end users in a 'filter bubble', restricting them to data which they are familiar with. It should be noted that many interesting discoveries came from detecting and researching outliers (Jagadish et al. 2014).

1.3 Frontiers

Science is believed to have entered a fourth paradigm based on the growing availability of Big Data and new analytics (Hey et al. 2009). It is chronologically preceded by experimental science, theoretical science and computational science (Hey et al. 2009). Big Data coupled with new data analytics are disruptive innovations which challenge established epistemologies across the sciences, social sciences and humanities. Specifically, these innovations propose a radically different way to make sense of science, culture, history, economics and society by being data-driven rather than knowledge-driven. In other words,

Big Data analytics enable an entirely new epistemological approach for making sense of the world; rather than testing a theory by analysing relevant data, new data analytics seek to gain insights 'born from the data' (Kitchin 2014). Increasingly, data are collected not with the aim of solely testing human-generated hypotheses but for the possibility of testing hypotheses that have not yet been envisioned at the time of collection (Agarwal & Dhar 2014). This new approach to thinking, challenges the accepted way of interrogating the world and synthesising knowledge.

An example of a future project that promises both Big Data and big discoveries is the Square Kilometre Array (SKA) radio telescope. The SKA will monitor the sky in unprecedented detail and map it hundreds of times faster than any existing radio telescope with data rates of 4 Tbits/s (Dewdney et al. 2009). It is expected that, once operational, the SKA will collect in excess of an Exabyte of data per day (Kitchin 2013). Astrophysicists are confident that these Big Data will not just answer fundamental questions about the Universe but contain Big Discoveries not yet postulated. In this thesis, I will investigate the data and algorithmic challenges posed by the SKA in the context of searching for pulsars.

There are significant new questions and opportunities created by the availability of Big Data and major advancements in machine intelligence. Decisions that were previously based on guesswork, or on painstakingly handcrafted models of reality, can now be made using data-driven mathematical models. The Big Data revolution that is under way, not only has far-reaching consequences as to how knowledge is produced, business conducted and governance enacted (Walker 2014) but it is also opening the door to discovering the unknown unknowns.

1.4 Conclusion

Technological advances coupled with a decline in digital storage costs have resulted in an abundance of data being created, collected and consumed. These data sets are poised

to revolutionise the way we live, work and think. However, as I have highlighted in this chapter, there are a number of challenges that must still be addressed in order to exploit the full potential of Big Data. Addressing these challenges will require the scaling out of global digital infrastructures and scaling up of standard data analysis and mining techniques.

Radio astronomy and its ensuing Big Data challenges drive the research of this thesis. Specifically, the SKA project as both a producer of Big Data and a consumer of big data analytics. In the next chapter, I explore the technical challenges associated with and the scientific possibilities created by the SKA project.

Chapter 2

The Square Kilometre Array

2.1 Introduction

The basic principles of observational astrophysics dictate that weaker sources of signals require larger photon collecting areas, similarly the observation of more detail in the sky requires a telescope of larger diameter. Interferometry is a technique that allows for the construction of a telescope with an enormous diameter (on the scales of thousands of kilometres) by connecting a large number of small telescopes with one another. The signals from these large number of receiving elements need to be combined to produce the desired data products, which are then analysed to address specific scientific problems.

The genesis of an interferometric array with a square kilometre of collecting area (SKA) at wavelengths between millimetres and tens of metres has been discussed since the early 1990s (Wilkinson 1991). However, it was not until 2004 (Carilli & Rawlings 2004) that a major international effort was initiated with the sole purpose to define and quantify the impact that such a next generation radio mega-facility would have on the biggest questions in modern astronomy (Carilli 2014). The effort resulted in the identification of key science projects (KSPs) which would lead to great scientific advances in astrophysics and drive the design of a telescope that would lead to such advances (Carilli & Rawlings 2004, Dewdney et al. 2009). Since then, the conceptualisation, design and construction of the SKA, the most ambitious radio astronomy project of the 21st century, took shape. Upon com-

pletion, it is expected to be the largest radio interferometer in the world. Consequently, it will provide unique and essential information in astronomy complimentary to other multi-wavelength campaigns.

In the next section, I provide a summary of the primary KSPs. A section on the technical characteristics of the SKA is followed by a section on the pathfinder telescopes which, through commissioning, assisted in guiding the design of the SKA and continue to influence both hardware and software data processing considerations. In the subsequent section, I outline how the scientific challenges of the SKA are translated into analytical challenges. In conclusion, I give a summary of the technical advances of the SKA which promises to revolutionise radio astronomy.

2.2 Key science projects

Each KSP addresses an unanswered question in fundamental physics or astrophysics and is either scientifically unique to the SKA or for which the SKA may play a key role. The KSPs include (Cordes et al. 2010):

- *Probing the dark ages and the epoch of reionisation* - Images of neutral hydrogen, captured by the SKA, will be used for probing both the transition of the intergalactic medium from neutral to ionised state and the formation of large-scale structure.

- *The origin and evolution of cosmic magnetism* - The SKA will track the evolution of magnetic fields in galaxies and clusters of galaxies over a large fraction of cosmic time by measuring the Faraday rotation towards large numbers of background sources.

- *The cradle of life* - The SKA will probe a key regime in the planetary formation process by observing and monitoring changes associated with the centimetre-wavelength thermal radiation from pebbles in the inner regions of nearby proto-planetary disks

during planet formation.

- *Galaxy evolution, cosmology and dark energy* - The ultimate goal of the SKA is to complete a 'billion galaxy' spectroscopic survey, studying the 21-cm hyperfine transition of neutral atomic hydrogen (HI), which will provide a comprehensive sample for inferences on galaxy evolution and dark energy.

- *Exploration of the unknown: the dynamic radio sky* - The focus of the dynamic radio sky is on the time domain, which is ripe for exploration. In addition to known classes of radio transients, the SKA will explore the radio sky for objects such as orphan gamma ray burst afterglows, radio supernovae, tidally disrupted stars, flare stars, exoplanets, magnetars, and transmissions from extraterrestrial civilisations.

- *Strong field tests of gravity using pulsars and black holes* - The SKA will be commissioned to survey the Galaxy with the objective to find all pulsars beaming towards Earth and thereafter time them with high precision in order to probe fundamental physics. The timing of newly discovered pulsars along with known pulsars will aid the detection and measurement of gravitational waves from binary black holes. In addition, the timing of relativistic binaries should yield the orbital elements to a high precision and, depending on the system, selected post-Keplerian parameters. Post-Keplerian parameters allow for the determination of individual stellar masses to high precision and may also constrain theories of gravity in the strong-field regime. I will focus on this KSP in my thesis and the work presented here will assist the SKA to discover all pulsars beaming towards Earth.

2.3 Key technical characteristics

The SKA will consist of three different receptor types and configurations: low-frequency array (SKA 1-low), mid-frequency array (SKA 1-mid) and high-frequency array (SKA-

high) (Dewdney et al. 2009). Australia will host the SKA low-frequency array. The core of the SKA mid-frequency array will be constructed in South Africa and be connected to numerous additional radio dishes spread over long distances expanding the project into other African countries (Taylor 2012).

The layout of the SKA will have a spiral shape, which allows for many different baselines and angles between antennas resulting in very high-resolution images. The spiral configuration was chosen after a detailed study (Dewdney et al. 2013) found it to be the best trade off between image resolution and cost.

The SKA is to be deployed in three phases: Phase one comprises the construction of the low- and mid-frequency arrays; and phase two, the completion and extension of the constructed arrays. It is planned that phase one should commence before the end of this decade. Phase three, which consists of the construction of a high-frequency array, will commence once the required technology has been developed and the engineering design completed, hopefully soon after 2025 (Cordes et al. 2010).

The two telescopes comprising the first phase of the SKA will be made up of approximately 200 parabolic dishes (see Figure 2.1) sampling frequencies above 350 MHz (mid-frequency array) and approximately 130 000 dipole antenna elements sampling the frequencies below 350 MHz (low-frequency array). At each element of the interferometer, the data will be digitised at rates of order 1 Gsamples/s, generating broadband data streams that are combined digitally downstream, in two different ways using correlators and beamformers.

The SKA will use beamformer technology to simultaneously generate 500 beams using the low-frequency array, and 1500 beams using the mid-frequency array. Each of these beamformers will carry information at a rate of 2.5 Gbits/s, leading to a total data bandwidth of 1.3 Tbits/s and 4 Tbits/s for the low and mid-frequency arrays (Dewdney et al. 2009).

The infrastructure required to support these high data rates includes real-time data transmission, processing and analysis as well as the capacity to publish the results in a timely

Figure 2.1: This MeerKAT antenna is an example of a parabolic dish which will be used in the mid-frequency array. Credit: Photowise

fashion for interpretation by the global astrophysics community. Storing the data and processing off-line is both extremely expensive and restrictive for science. It is restrictive because it leaves no opportunity for a rapid follow-up by other instruments in response to events detected in the data, which are astrophysically speaking most interesting. The processing that is required to detect the interesting signals should therefore be carried out in real-time. In this regard, I present a new real-time pre-processing technique for pulsar search data in this thesis.

The necessary algorithms required for pre-processing the data obtained and searching for periodic and quasi-periodic signals embedded in noise, require extreme optimization. Carefully designed high performance computing hardware and optimized algorithms will allow for unprecedented surveys in real-time and within tight power budgets. Communi-

cations infrastructures will range from intra-chip and inter-chip with optical fibre to the correlator and on to a high-performance computer, to trans-oceanographic facilities where the immense amount of data will be available to scientists the world over.

The performance improvements of the SKA over existing telescopes can be assessed with the following three factors: resolution, sensitivity and survey speed. SKA 1-low will, in addition to having 1.2 times greater resolution, be eight times more sensitive and one hundred and thirty five times faster than LOFAR, the best telescope at low-frequencies. SKA 1-mid will be approximately five times more sensitive, sixty times faster and it will have four times greater resolution than the Jansky Very Large Array in the USA, which is the best telescope at mid-frequencies (Dewdney et al. 2014).

The observational advances of the SKA, coupled with its large instantaneous fields of view are destined to accelerate the pace of discovery and improve our understanding of the Universe as well as the laws of fundamental physics. Ultimately, the SKA is poised to make big discoveries from Big Data. However, the data deluge that will be pouring in from the SKA, the sheer volume, velocity and variety, poses huge challenges for the project.

Radio astronomers stand to benefit immensely from this Big Data, provided they are able to make sense of it. The transformational potential of Big Data hinges largely on new streaming techniques, capable of coping with huge data volumes in real-time, which can fuel discovery and innovation within astrophysics and the radio astronomical sciences.

In the next section I list some of the pathfinder telescopes for the SKA project.

2.4 Pathfinder telescopes

The design of the SKA is based on science requirements, experiences from the building of pathfinder and precursor telescopes that provide design options, and technology capability considerations. Pathfinder and precursor telescopes have been spawned to pursue many of the KSPs in advance of the full SKA.

Telescopes such as LOFAR in Europe and the Murchison Wide-Field Array in Western Australia are pathfinders for SKA 1-low. These telescopes are currently performing deep cosmological observations, at low-frequencies (< 200 MHz), providing the first limits on the reionisation HI 21-cm signal.

At mid-frequencies (∼1 GHz), telescopes such as the Australian Square Kilometre Array Pathfinder (ASKAP) and the Karoo Array Telescope (MeerKAT) in South Africa are pathfinders for SKA 1-mid. ASKAP is a thirty six dish telescope located at the Murchison Radio-astronomy Observatory in Western Australia (CSIRO 2017). It is equipped with innovative phased array feed receivers which provide multi-pixel images of the sky, allowing it to survey large areas of the sky quickly (Beck 2010). ASKAP is already conducting ground breaking research which makes it an important technology demonstrator for the SKA. MeerKAT is currently being built and will provide an important testing ground for both science and techniques appropriate to SKA 1-mid (Carilli 2014).

2.5 Key analytical challenges

There are several key analytical challenges that the SKA has to overcome in order for it to be the premier instrument for radio astronomy. These challenges include, but are not limited to: digital signal processing, calibration and image formation, non-imaging processing, system sensitivity and scalability of solutions.

Digital signal processing - The challenge associated with beamformers and correlators is the high input data rates which require bespoke hardware and software. These bespoke solutions should also adhere to strict power consumption constraints because the SKA project has a limited power budget (Dewdney et al. 2009).

Calibration and image formation - Calibration issues, for wide-field image formation by the SKA, focus on adequate characterisation and correction of direction dependent calibration effects. In order to yield robust wide-field images, careful design is necessary to

ensure that only a small number of system-wide parameters require calibration (Cordes et al. 2010).

Non-imaging processing - Finding pulsars and transients are key non-imaging SKA science requirements. In subsequent sections, I go into great detail on how pulsars and transients are searched for and I identify the challenges that still remain in finding pulsars. Thereafter, I address some of the challenges identified.

System sensitivity - The SKA will achieve increased system sensitivity compared to existing radio arrays due to a larger aperture, lower temperature and larger input bandwidth. A large input bandwidth with a high number of spectral bands means the SKA will produce a lot of data. How to clean these bands from radio frequency interference (RFI) and system induced noise, pose a threat to the sensitivity of the telescope. This is also a challenge that I address in this thesis.

Scalability of solutions - The SKA has processing requirements that greatly exceed those of existing arrays because of the larger number of antennas, distances and bandwidths. The algorithms that are currently used on LOFAR and other pathfinder telescopes for finding pulsars need to be scalable to SKA data sizes. Scalability is another issue that I consider in this thesis.

2.6 Conclusion

The promises of the SKA to revolutionise radio astronomy is threefold: (1) The high sensitivity, large field of view and multiplexing capabilities of the array will enable complete surveys over a significant fraction of the sky; (2) Surveys (both line and continuum) offer a natural platform for synoptic operation to cover the sky repeatedly; and (3) Computational and algorithmic advances will enable commensal observations to achieve multiple scientific goals (Carilli 2014).

The transformational potential of the SKA, in particular for the discovery and study of

pulsars, prompted the work that I present in this thesis. It also stimulated the design of an algorithm for the real-time pre-processing of Big Data in pulsar searching. In the next chapter, I will focus on pulsars and the difficulties associated with the discovery of new pulsars.

Chapter 3

Pulsar phenomenology

3.1 Introduction

In this chapter, I outline the different classes of pulsars, the propagation effects that their pulsed emissions undergo when traversing the interstellar medium (ISM) and the methods involved in searching for new pulsars. In the second section of this chapter, I explore the current strategies to search for pulsars inside and outside our Galaxy. Thereafter, I highlight the latest computational developments which prompted speed ups in current pulsar search pipelines. Lastly, I describe the existing software packages developed to find new pulsars.

3.2 Pulsars

The year 2017 marks the 50th anniversary of the serendipitous discovery of the first pulsating neutron star (pulsar) by Bell and Hewish (Hewish et al. 1968). Since then, physicists, astrophysicists and astronomers have collaborated to figure out what gives rise to these objects. Although there are many questions remaining, particularly with regard to the emission mechanism (Lorimer & Kramer 2005), the basic model has long been established *viz*: Pulsars are rapidly rotating, highly magnetised neutron stars formed during the supernova explosions of massive ($\sim 5-10\ M_{\odot}$) stars that emit a beam of electromagnetic radiation (see Figure 3.1). The radiation, which is observable if the Earth is in the field of view of

the emission beam, combined with the rotation of the pulsar around its spin axis, results in precisely periodic pulsed emissions. The emissions are broadband. Over the decades after their discovery, the number of phenomenological features of these objects have increased constantly, naturally leading to a more diversified classification.

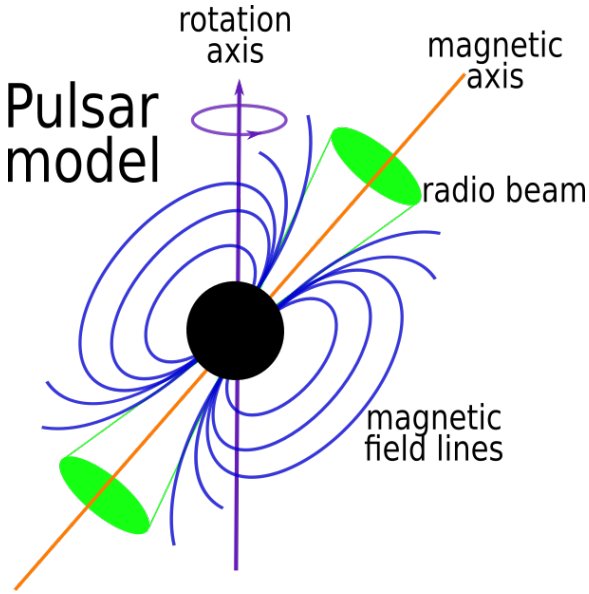


Figure 3.1: The lighthouse model for the rotating neutron star and its magnetosphere.

In particular, two main classes of pulsars are recognised (Lorimer & Kramer 2005). The first class is that of the so-called 'normal' pulsars, relatively young objects aged $\sim 10^7$ yr with strong magnetic fields of $\sim 10^{12}$ G and spin periods P of the order one second which are observed to increase at rates of $\dot{P}$ of typically 10^{-15} s/s. The second class, the so-called 'millisecond pulsars' (MSPs), are older objects with typical ages of $\sim 10^9$ yr and weaker magnetic fields of $\sim 10^8$ G, which have spin periods primarily in the range 1.5 ms and 30 ms and rates of slowdown $\leq 10^{-9}$ s/s.

A very important difference between normal pulsars and MSPs is the presence of an orbiting companion. Orbital companions are much more commonly observed around MSPs (~ 80 % of the observed sample) than around normal pulsars (≤ 1 %) (Lorimer & Kramer 2005). Pulsars can therefore be further classified as either isolated or as binary pulsars. In the context of searching, it is important to distinguish between isolated and binary pulsars

because the algorithmic approaches for finding these systems differ.

Before exploring the algorithmic approaches designed to search for pulsars, it is imperative to first understand the propagation effects that the pulsed emissions of pulsars undergo. Thus, the following section details the pulse propagation from origin to acquisition.

3.3 Pulsar observables

A pulsar emits radio pulses from a radiation beam along its magnetic axis, which in general, is misaligned with the pulsar's axis of rotation (see Figure 3.1). For detectable pulsars the radiation beam crosses the observer's line of sight. Rotation periods are very stable, owing to the fast rotation and large moment of inertia of pulsars (Lorimer & Kramer 2005). Figure 3.2 illustrates schematically how periodic radio pulses emitted by pulsars are affected as they propagate towards Earth.

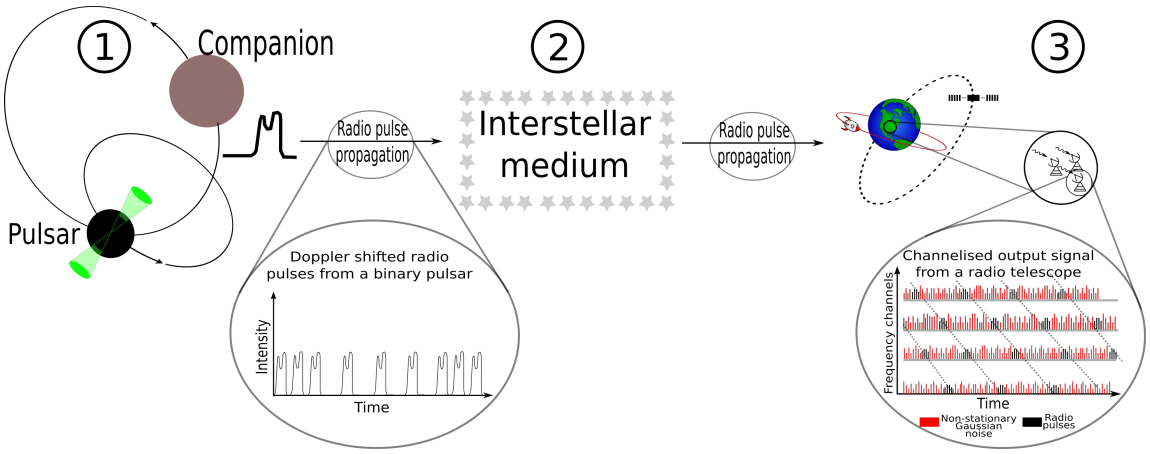


Figure 3.2: Schematic illustration of the propagation path of periodic radio pulses emitted by a pulsar in a binary system, see text for details. Figure taken from van Heerden et al. (2014).

Radio pulses from pulsars in binary systems typically exhibit a rapidly varying Doppler shift in their rotational period (see Figure 3.2.1), caused by the acceleration of the pulsar around its companion. In other words, the observed period P_{obs} of a pulsar in a binary orbit

is time dependent

$$P_{\rm obs} = P_{\rm int}(1 + V(t)/c), \tag{3.1}$$

where $P_{\rm int}$ is the intrinsic period of the pulsar, c is the speed of light and $V(t)$ is its radial velocity along the line of sight as a function of time. To be efficient, these Doppler shifts need to be accounted for in the so-called acceleration searches (see Section 3.5.3). Given that the orbital parameters of systems that are being searched for are by definition unknown, there are significant computations involved in this process.

Furthermore, radio pulses from pulsars, at frequencies relevant to the SKA (∼100 MHz to ∼3 GHz), interact with the free electrons in the ISM (Figure 3.2.2), resulting in frequency dispersion and scattering.

Frequency dispersion is the phenomenon in which the arrival time of the high frequency components of a broadband radio pulse precedes, the arrival time of its lower frequency counterparts. The interpretation for this observed phenomenon is the frequency dependence of the group velocity of radio waves as they propagate through the ionised components of the ISM. The delay in pulse arrival times, Δt measured in seconds, between a high frequency pulse $\nu_{\rm hi}$ and a low frequency pulse $\nu_{\rm lo}$ is given (Lorimer & Kramer 2005) by

$$\Delta t = 4.15 \times 10^6 \text{ ms } \times (\nu_{\rm lo}^{-2} - \nu_{\rm hi}^{-2}) \times \text{DM}, \tag{3.2}$$

where the frequencies $\nu_{\rm lo}$ and $\nu_{\rm hi}$ are both in MHz and the dispersion measure (DM) (cm^{-3} pc) is the integrated column density of free electrons along the line of sight:

$$\text{DM} = \int_0^d n_{\rm e}\, dl. \tag{3.3}$$

In this equation, d represents the distance from the Earth to the pulsar (pc) and $n_{\rm e}$ represents the free electron density (cm^{-3}). From Equation 3.3 it is obvious that a measurement of a delay across a finite bandwidth yields the DM. Pulsars at large distances away from

the Earth typically have higher column densities and therefore larger DMs compared to those pulsars closer to the Earth. Given that the distance from the Earth or the DM of an undiscovered pulsar is unbeknownst during the search process means that the DM becomes an additional parameter that needs to be searched for during the search process.

Scattering is a propagation effect induced by irregularities in the electron density distribution in the ISM. These regions in the ISM with a larger electron density have structure, which scatters radio waves causing them to take multiple paths to the observer. Due to scattering, some rays will have longer path lengths and will therefore be slightly delayed compared to those rays which travel straight. This delay results in pulses becoming smeared, showing an exponentially decaying tail, and therefore makes detection more difficult. For a thin screen the duration associated with an exponentially decaying tail is called the scattering time constant, τ_s. The scattering time constant is related to the observation frequency ν and the distance to the pulsar d as given by

$$\tau_s \propto \nu^{-\alpha} d^2, \tag{3.4}$$

where α is positive and observed to be variable with a typical value of four. Equation 3.4 is a model that is used to parametrise scattering. From Equation 3.4 it is clear that the scattering time is smaller for higher frequencies than for lower frequencies. Therefore, the best way to deal with scattering is to observe at higher frequencies.

When these dispersed and scattered radio pulses reach the Earth, they are recorded together with signals from satellites, aeroplanes, other terrestrial sources as well as instrumentation noise. Such interferences will show up as spurious signals in the recorded channelised data (Figure 3.2.3) during the search for periodic pulses from new pulsars, which may result in slow or abrupt baseline drifts of the noise.

Terrestrial as well as extraterrestrial sources of radio frequency interference (RFI) have a profound impact on the sensitivity of a pulsar search. If a particular observation contain-

ing pulses from a pulsar is affected by RFI, the pulsar signal may be occluded by the RFI rendering it undetectable as an emittance from a possible candidate. Therefore, most pulsar search pipelines contain a multitude of RFI excision methods each targeting a different terrestrial or extra-terrestrial signal (Lorimer & Kramer 2005).

The propagation effects described above result in observational phenomena that must be corrected in order to improve the detection of emissions from pulsars. The next section details different search strategies employed to find undiscovered pulsars.

3.4 Search strategies

Search strategies that underpin pulsar surveys are governed by the type of objects that are searched for or phenomena that scientists wish to study. Strategies differ when searching for young pulsars or MSPs, or when studying globular clusters, the inner Galaxy or the Magellanic Clouds. Different strategies are employed to establish whether or not an unidentified X-ray or gamma-ray source is in fact a pulsar.

Pulsars are concentrated strongly along the Galactic plane, as illustrated by Figure 3.3, which is consistent with the standard picture of the birth of neutron stars in the core collapse supernova explosions of massive stars and young age (Lorimer & Kramer 2005). Therefore, if the objective is to find predominantly young pulsars, surveys should search along the Galactic plane where young pulsars are likely to be found near their place of birth. Due to severe propagation and sky background effects on the sensitivity at low frequencies (< 1 GHz), most Galactic plane surveys are carried out in the 1-2 GHz band (Lorimer & Kramer 2005).

A good strategy, in terms of high detection rates, when searching for MSPs, is to target globular clusters. The main reason for this strategy is the high stellar density in globular clusters relative to the rest of the Galaxy. As a result, low-mass X-ray binaries are almost ten times more abundant in clusters than in the Galactic disk. An advantage of searching

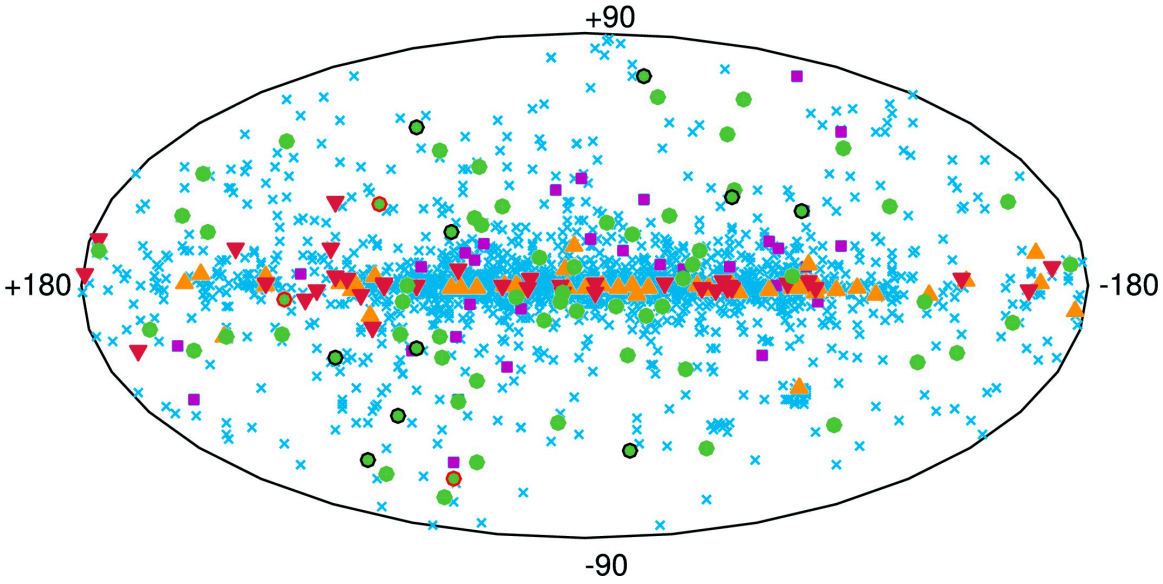


Figure 3.3: Hammer-Aitoff projection showing the distribution of pulsars in Galactic Coordinates. The colours are representative of the following: Orange triangles - radio and γ-loud young pulsars; red triangles - radio faint and γ-loud young pulsars; green filled circles - radio and γ-loud MSPs; green circles with black borders - black-widow MSP systems; green circles with red border - redback MSP systems; purple squares - radio pulsars recently discovered in the direction of *Fermi*-LAT point sources; blue crosses - other radio pulsars. Figure taken from Grenier & Harding (2015)

for pulsars in a globular cluster is that once the DM of a pulsar in known in a globular cluster, the DM parameter space for subsequent searches is essentially fixed. This allows computation power to be invested in acceleration searches for short-period binary systems (Lorimer & Kramer 2005).

Searching intermediate latitudes, in the range $5^\circ < |b| < 15^\circ$ away from the Galactic plane, typically result in the discovery of older pulsars, as it takes some time for the pulsars to reach this position. Interstellar propagation effects are less pronounced away from the Galactic plane. Consequently, the optimal survey frequency is below a gigahertz which takes advantage of the fact that pulsars are brightest (higher flux densities) at low frequencies. Higher flux densities require shorter integration times which mean that the effects of binary acceleration are far less problematic than for deep searches of the Galactic plane at higher frequencies.

More recently, pulsar search strategies have targeted X-ray or gamma-ray point sources.

Specifically, gamma-ray sources discovered with the *Fermi* Large Area Telescope (LAT) (Ray et al. 2012, Abdo et al. 2013) that do not have strong associations with known gamma-ray emitting source classes but have pulsar-like spectra and variability characteristics, have been targeted. The radio follow-up of these sources identified with the Fermi-LAT telescope requires only one pointing because its gamma-ray error boxes are typically close to the beam size of a 64 m telescope at 1.4 GHz. Hence, these campaigns have been hugely successful because long integration times can be afforded to reach very low brightness limits.

Finding pulsars in the Galactic centre is extremely useful in probing the Galactic centre and its conditions. Only one pulsar has been found near Sagittarius A* (which is a radio source related to the black hole at the centre of the Galaxy) referred to as the Galactic centre Magnetar (Kennea et al. 2013). The dearth of pulsars in the Galactic centre is in stark contrast to the widely held belief that the Galactic centre is a site of past and present star formation with a large population of massive stars (Mezger et al. 1999). The reason for the difficulties of finding pulsars around the Sagittarius A* could be ascribed to the large amount of ISM scattering expected for Galactic centre pulsars (Cordes & Lazio 1997). A possible strategy is to search at higher frequencies where scattering is less severe, but in this scenario the limiting factor is the flux densities (brightness) of pulsars which decreases with observation frequency. Therefore a highly sensitive instrument such as the SKA will be important to make up for the reduced flux density received from these Galactic centre pulsars (Keane et al. 2014, Eatough et al. 2015).

In the past, extragalactic survey strategies have focused on the Small and Large Magellanic Clouds which resulted in a number of discoveries (McConnell et al. 1991, Crawford et al. 2001, Ridley et al. 2013). Surveying the Magellanic Clouds for radio pulsars is difficult because of the large surface area to be covered compared to the typical collecting area of existing radio telescopes and their huge distances from Earth which require long integrations. This means that only the most luminous pulsars are detectable. Thus, long

integrations, using a telescope with a large collecting area, are needed to detect pulsars outside our Galaxy.

3.5 Discovering new pulsars

3.5.1 Pipeline for a standard pulsar search

Known pulsars with weak radio emissions have flux densities measured at 1.4 GHz varying between 20 μJy and 5 Jy (1 Jy $\equiv$ 10^{-26} W m^{-2} Hz^{-1}), but undiscovered pulsars may have even weaker emissions. Thus, searching for an undiscovered pulsar is analogous to finding a needle in a haystack except that the haystack is noise and the needle is the faint periodic signal from a pulsar, presupposing the presence of a pulsar in the observation. In addition to low signal strengths, radio pulses produced by pulsars are intrinsically difficult to detect due to their narrow duty cycles, dispersion effects and the presence of RFI in addition to non-stationary Gaussian noise. Note that the effects of dispersion can also be positive in distinguishing pulsars from RFI.

Numerous techniques have been developed to overcome some of the difficulties highlighted above and to assist in the quest to find more pulsars. These techniques are combined to form a standard pulsar search pipeline. The typical pipeline used when searching for new pulsars consists of nine stages as depicted in Figure 3.4.

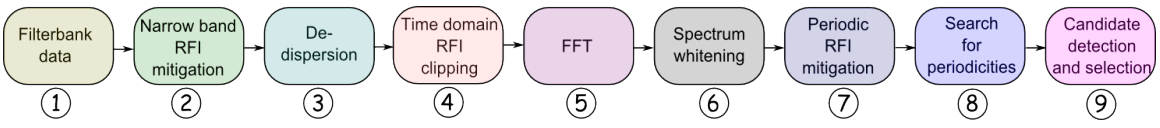


Figure 3.4: Schematic illustration of a typical pulsar search pipeline. See Sections 3.5.1.1 to 3.5.1.9 for details regarding each of the building blocks 1-9 depicted in the pipeline.

3.5.1.1 Data recording

The pipeline starts with splitting the finite bandwidth of the data collected by a radio telescope into a number of channels ($\sim$ 1024 channels), typically using a filterbank or a correlator (Backer et al. 1990). A filterbank is an array of bandpass filters that separates the input signal into multiple narrow frequency channels. The output of the filterbank is detected and the power in each channel, as a function of time, is recorded to disk (Figure 3.4.1). The digitisation level of the recorded data is typically 1, 2, 8 or 16-bit depending on the instrument. Some telescope backends compute a running average bandpass and removes this from the data before digitisation to ensure that the digitisation levels are always in the optimal regime. A three dimensional presentation of what the output of the filterbank looks like can be seen in Figure 3.5. Note that this description is for a typical observing setup, since not all telescope backends at all observatories work exactly as previously described.

Traditionally, filterbank data were recorded to disk or tape for off-line processing weeks or months later. However, with the increase in scope and sensitivity of future surveys it will become infeasible to store the raw data for off-line processing due to capacity and input/output constraints. Hence, the need for a paradigm shift from off-line to real-time processing of survey data. An example of a pipeline that includes real-time pulsar and fast transient search methods is the pipeline used in the latest SUrvey for Pulsars and Extragalactic Radio Bursts I (SUPERB)(Keane et al. 2017).

3.5.1.2 Narrowband RFI mitigation

Each filterbank file is examined for the presence of sporadic bursts of interference, more commonly referred to as narrowband RFI signals, which are excised by replacing the affected samples. In the PALFA survey the statistically identified samples were replaced with constant values chosen to match the median bandpass (Figure 3.4.2) (Lazarus et al. 2015), whereas in the High Time Resolution Universe (HTRU) high-latitude survey at Parkes (Keith et al. 2010) the bad time-samples are replaced with random bytes selected from

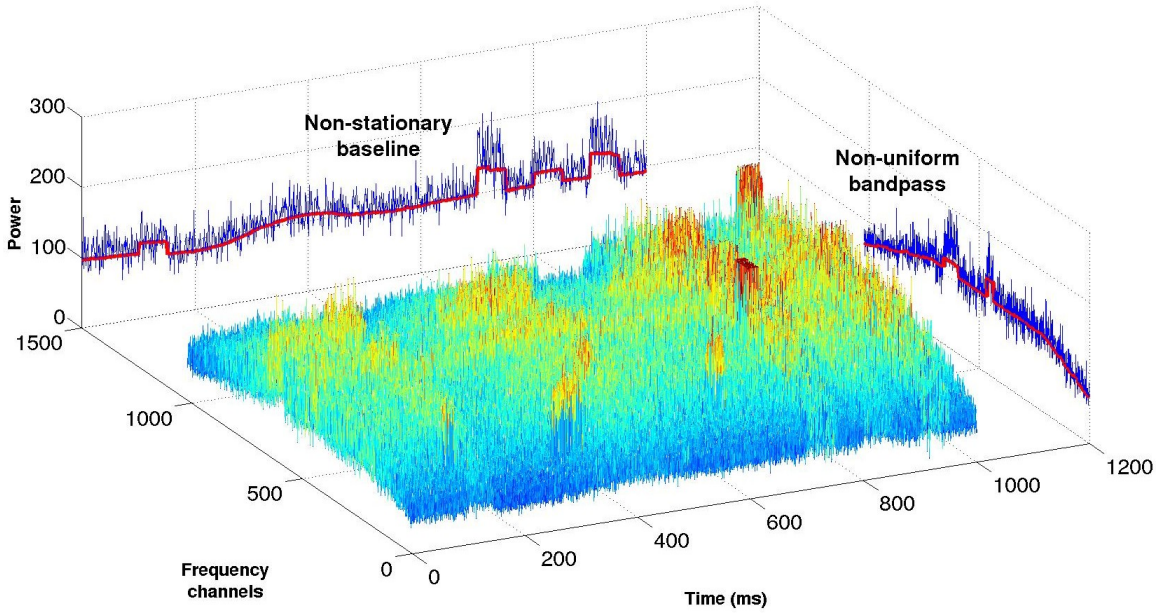


Figure 3.5: 3D plot of an 8-bit mock filterbank file with a non-stationary baseline, non-uniform bandpass and RFI.

a small window behind the sample.

3.5.1.3 Dedispersion

The corrected filterbank files are then dedispersed for a number of trial DM values to compensate for the dispersion induced by the ISM (Figure 3.4.3) (Lorimer & Kramer 2005). DM trial values typically range from 0 pc cm $^{-3}$ up to a few thousand, while some surveys push for DM values of $\sim$ 10000 pc cm $^{-3}$ to ensure sensitivity for highly-dispersed, potentially extragalactic sources (Lazarus et al. 2015).

Real-time processing entails block-wise dedispersion for the purposes of rapid reporting of Fast Radio Burst (FRB) detections, which constitute a byproduct of pulsar searches. The time duration of each block depends on the DM search and the observing frequency, and is likely to be in the order of a few seconds for typical searches. RFI cleaning must take place prior to dedispersion. Thereafter, all frequency information is lost and likewise the opportunity to detect and mitigate RFI in the time-frequency plane. The relevant timescale on which any RFI excision technique needs to operate is therefore more likely to be related

to the FRB detection buffer size, rather than the full integration required for periodicity searches.

3.5.1.4 Time domain RFI clipping

Weak broadband RFI can quite easily go unnoticed by the narrow band RFI mitigation procedure described above. Fortunately, since most sources of RFI originate close to Earth, they are not dispersed and are thus detectable in the zero-DM time series. Consequently, most pulsar search pipelines include algorithms to remove such weak broadband signals by combing over the zero-DM time series and identifying all the samples that are significantly larger than the surrounding data samples Lorimer & Kramer (2005). Various approaches exist to replace the affected samples, one such method replaces the spectra corresponding to the bad time intervals by the local median band-pass Lazarus et al. (2015). The zero-DM time series is used to identify and mitigate broadband RFI (Figure 3.4.4) that went undetected by the narrow band RFI excision process.

3.5.1.5 From time domain to frequency domain

After mitigating broadband RFI, the Fast Fourier Transform (FFT) of each single dedispersed noisy time series is computed (Figure 3.4.5).

3.5.1.6 Spectrum whitening

The power spectrum is whitened (Figure 3.4.6) in order for the response to be as uniform as possible, i.e. mitigating the frequency dependence associated with non-stationary Gaussian noise. Spectrum whitening typically entails subtracting a running median and normalising the local root mean square (RMS) of the power spectrum such that it has a zero mean and unit RMS. This approach to spectrum whitening is just one of many approaches to deal with the non-uniform response of the power spectrum. A whitened power spectrum is preferred, because estimating the significance level of any signal present is easier. Different

techniques have been implemented to whiten the spectrum and these will be described in more detail in Chapter 6.

3.5.1.7 Periodic RFI mitigation

The next stage of the pipeline, Figure 3.4.7, entails the identification of periodic RFI. Known periodic signals which are present all or most of the time, such as power lines carrying alternating current and communication systems such as airport radar systems, are flagged with their harmonics and their bandwidths determined. These interferences are mitigated by creating a spectral mask. This mask consists of a list of all the Fourier bins affected and which should be ignored in all subsequent processing.

3.5.1.8 Search for periodicities

In general, radio pulses from pulsars have narrow duty cycles which, in the Fourier domain, result in the power to be distributed between the fundamental frequency and a number of harmonics (van Heerden et al. 2014). Therefore, to take full advantage of the power contained in the harmonics, the whitened spectrum is harmonically summed by adding the higher harmonics to the fundamentals, a technique known as 'harmonic summing'. The original power spectrum as well as the composite spectra formed by summing 2, 4, 8 and 16 harmonics (Lyne & Graham-Smith 2012) are each searched for periodicities (Figure 3.4.8) (Cordes et al. 2006). The values in these power spectra which are larger than a predefined threshold signal-to-noise ratio (SNR), are stored as pulsar candidates. The best candidates from each trial DM are saved.

3.5.1.9 Candidate identification and follow-up

After all the time series have been processed, a list of pulsar candidates is compiled. This list is pruned by post-processing procedures (Figure 3.4.9) ranging from sifting and folding to sophisticated machine learning candidate selection (Lyon et al. 2016). The most

promising pulsar candidates are saved for future observation and follow-up (Cordes et al. 2006).

3.5.2 Time domain search algorithms

Standard pulsar searches use the FFT to find new pulsars. However, two time domain search techniques exist. The first technique is called the Fast Folding Algorithm (Staelin 1969) which predates the FFT for finding pulsars. It involves folding an observational time series for a range of trial periods and examining the resulting folded profiles.

The second technique is called the single pulse search technique (Cordes & McLaughlin 2003). This is a time domain technique aimed at detecting bright individual pulses from pulsars and other radio transients such as Rotating Radio Transients and FRBs through the analysis of dedispersed time series. The single pulse search involves the identification of significant outlying data points in a given time series by applying a matched filter which is able to identify pulses of varying widths.

3.5.3 Binary search algorithms

As indicated before, there are algorithmic differences in searching for binary pulsars compared to the search for isolated pulsars. The algorithms developed for the search of pulsars in a binary system are explained in this section.

Standard pulsar searches use Fourier techniques to search for *a-priori* unknown periodic signals and usually assume that an apparent pulse period remains constant throughout the observation. For searches with observation times $T_{\rm obs}$ much greater that a few minutes, this assumption is only valid for isolated pulsars or those in binary systems where the orbital period $P_{\rm orb}$ is comparatively much longer.

Pulsars in highly relativistic binary systems show periodic changes in their pulse frequency due to the Doppler effect which results in a spreading of the signal power over a number of frequency bins in the Fourier domain, leading to a reduction in the SNR. A

number of time and frequency domain techniques have been developed to recover the loss in sensitivity due to Doppler smearing. These techniques can be grouped according to the orbital period of the binary system $P_{\rm orb}$ relative to the observation time $T_{\rm obs}$.

The first group of techniques were developed to find pulsars for which $T_{\rm obs} \leq P_{\rm orb}/10$. These techniques, the so-called 'acceleration searches', assume that the pulsars have constant orbital accelerations. Three acceleration searches are known from literature: the stack searches (Wood et al. 1991), the time domain re-sampling (Middleditch & Kristian 1984) and the coherence recovery (Ransom 2001). Stack searches is a fast but incoherent method that divides the dedispersed time series into many short segments and stack resulting spectra with linear offsets corresponding to different frequency drift rates, i.e. different acceleration trials. Time domain re-sampling coherently demodulates with typically one parameter, constant acceleration, but it can use more parameters for example the acceleration derivative. This operation is applied before doing a frequency domain periodicity search. Lastly, coherence recovery involves taking an FFT of the time series to generate the power spectrum with the pulse smeared over a number of spectral bins. The smeared power is collected into one frequency bin by using a filter that is the complex conjugate of the smearing function.

The second group comprises two techniques which were developed to find pulsars for which $T_{\rm obs} > P_{\rm orb}$. The first technique makes use of the Hough transform whereby the dynamic spectra are searched for faint sinusoidal tracks (Aulbert 2007). The second technique, known as the phase modulation search, detects periodic side-bands in power spectra created by orbital phase modulation of a binary pulsar signal. Both the orbital and pulsar periods can be found using this method (Jouteux et al. 2002, Ransom 2001).

The last group was developed to find pulsars for which $T_{\rm obs} \geq P_{\rm orb}/2$. Fully coherent demodulation, better known as the matched filtering technique, re-samples the time series with a search over three to five Keplerian orbital parameters.

Table 3.1: A breakdown of the the time associated with each of the steps in the pulsar search process expressed as a percentage of time.

Step	% Time
1. RFI mitigation	1 %
2. Dedispersion of the raw data	10 %
3. FFT search for isolated pulsars	10 %
4. Acceleration search for binary pulsars	68 %
5. Single-pulse search	7 %
6. Sifting of candidates	1 %
7. Folding of candidates	3 %

3.5.4 Computational challenges and advances in pulsar searching

A breakdown of the time spent during each of the steps in a pulsar search, expressed as a percentage, is summarised in Table 3.1.

Acceleration searching is typically the dominant process in pulsar searches ($\sim$ 68 %), followed by dedispersion ($\sim$ 10 %). Processing a single $\sim$ 150 s pointing takes $12-32$ hours on one CPU, depending on the number of DM and orbital acceleration trials searched for. A full survey can consist of $\sim$ 65000 pointings or more, hence 2-3 CPU centuries are required to search for pulsars in these data sets (Ransom 2008, Eatough 2011).

Graphical processing units (GPUs) have been responsible for most of the performance gains in pulsar search pipelines over the last five years (Armour et al. 2011, De & Gupta 2016, Adámek et al. 2016). New implementations of the pulsar search pipeline on GPUs that exploit high-performance computing techniques have shown to provide up to a factor 8 improvement in execution time and up to a factor 6 decrease in power compared to results obtained by traditional multi-core CPUs (Sclocco et al. 2015). A recent implementation of the Fourier domain acceleration search (FDAS) algorithm on GPUs by Dimoudi & Armour (2015) has shown to not only provide better than real-time performance for the SKA requirements but also allows for two beams of the SKA to be processed on one NVIDIA TitanX GPU card. These accelerated computational speeds are crucial in reducing energy

consumption as well as operational costs of the SKA facilities, which is an important factor considering the overall project budget.

3.6 Pulsar search software

The pulsar search pipeline described in Section 3.5.1 is available in a number of pulsar search software packages: SIGPROC developed by Lorimer (Lorimer 2001), PRESTO developed by Ransom (Ransom 2011), PEASOUP developed by Barr (Barr 2013) and PULSARHUNTER developed by Keith (Keith 2007). The two most frequently used packages are SIGPROC and PRESTO, both of which are freely available and well tested. Together they have been responsible for the discovery of a large fraction of the pulsars known today and are representative of typical pulsar surveys. Interested readers are referred to Cordes et al. (2006), Keith et al. (2010), Rane et al. (2016), Stovall et al. (2014) and Lazarus et al. (2015) for a comprehensive description of how these two pipelines are typically used in real pulsar surveys.

There are a couple of differences between SIGPROC and PRESTO. Firstly, PRESTO performs acceleration searches in the Fourier domain (Ransom 2001). SIGPROC on the other hand does time-domain re-sampling to carry out acceleration searches. Secondly, SIGPROC uses SNR as a metric to identify peaks in the normalised power spectrum whereas PRESTO uses the Gaussian significance (adjusted for the number of trials searched) of the peaks as a metric under a white noise assumption. Hereinafter, the terms SNR and Gaussian significance shall be collectively referred to as *detection significance*.

3.7 Conclusion

The objective with pulsar searches is to discover as many pulsars as possible. Searching for pulsars, through periodic pulses emitted by them, is difficult for a whole range of reasons. Several methods have been developed over time to mitigate the propagation effects which

hamper pulsar discovery. In the context of the SKA it is necessary to take stock of these methods in terms of their limitations, scalability and sensitivity before embarking on the development of a new pulsar search pipeline. Exploring and identifying the limitations of the various methods used in current pulsar search pipelines is the topic of the next chapter. Identifying the limitations, will contribute towards the development of a new search pipeline to enhance the pulsar discovery process.

Chapter 4

Pulsar detection: Big data, big challenges

4.1 Introduction

In this chapter, which concludes the introduction to this thesis, I identify, based on a literature review, the limitations of existing pulsar search pipelines. In addition, I explore the challenges posed for pulsar detection in the data rich era of the SKA. The limitations outlined will serve as a guide for further analysis and algorithmic development for improved pulsar search pipelines in this thesis.

4.2 Problem statement

One of the main motivations for the enormous computing effort engendered by the SKA and this work lies with the amazing properties of pulsars. The study of pulsars provides a wealth of information regarding neutron star physics, the interstellar medium and stellar evolution (Orginisation 2015). Aggregated data of all known pulsars provide insight into the pulsar evolutionary process. The SKA will enable the trajectory tracking of pulsars over time with enormous fidelity. The study of these trajectories and associated proper motions is extremely valuable for constraining the origin of pulsar velocities, which are associated with the pre-supernova orbital velocities of their binary progenitors and with asymmetries

in the supernova explosion (Kramer & Stappers 2015). The clock-like properties of pulsars also allow for sensitive measurements of their orbital dynamics which are used to probe the physics of binary evolution and test the predictions of General Relativity (Antoniadis 2014). The continued discovery of pulsars is paramount to the understanding of the radio pulsar population as well as the expansion of research in the aforementioned areas.

In addition to the propagation effects described in Chapter 3 there are several phenomena that hamper pulsar discovery. These phenomena are identified and described in the following sections along with the real-time requirements posed by the SKA.

4.2.1 Discrepancies between predicted and actual discoveries

The known pulsar population is ever increasing with various surveys running at radio telescopes around the world: e.g., the High Time Resolution Universe (HTRU) high-latitude survey at Parkes (Keith et al. 2010, Champion et al. 2016), the Northern HTRU pulsar survey at Effelsberg (Barr et al. 2013), the Pulsar survey Arecibo L-band Feed Array (Cordes et al. 2006), the Green Bank Telescope drift scan survey (Boyles et al. 2013), the Green Bank Northern Celestial Cap (GBNCC) survey (Stovall et al. 2014), the Arecibo all-sky 327 MHz drift pulsar survey (Deneva et al. 2013), the LOFAR Pulsar Pilot Survey survey (Coenen et al. 2014), the Survey for Pulsars and Extragalactic Radio Bursts (SUPERB) at Parkes (Keane et al. 2017), and Fermi-directed MSP surveys (Ray et al. 2012).

Pulsar population synthesis models (Lorimer 2011), based on surveys and the known pulsar population, are used to predict the number of pulsars expected to be discovered in future surveys (Lorimer et al. 2006, Bates et al. 2014). These techniques are also used to estimate the number of potentially detectable (i.e. those that are beaming towards Earth as well as being luminous enough) normal pulsars and millisecond pulsars (MSPs) in the Galaxy.

The number of pulsars discovered in recent surveys (Swiggum et al. 2014, Lazarus et al. 2015) has fallen well short of the number as predicted by the aforementioned estimation

techniques. It was predicted that the Arecibo PALFA Precursor survey (Swiggum et al. 2014) should have detected 490^{+160}_{-115} normal pulsars and 12^{+70}_{-5} millisecond pulsars (MSPs) by the beginning of 2014, but it managed to detect only 283 normal pulsars and 31 MSPs. On completion, the full PALFA survey was expected to have detected 1000^{+330}_{-230} normal pulsars and 30^{+200}_{-20} MSPs. However, close to completion it has only managed to detect ~ 443 normal pulsars and ~ 40 MSPs respectively (Lazarus et al. 2015). It is worth noting that the largest discrepancy between predictions and detections is for normal pulsars, i.e. pulsars with long periods. Furthermore, it is estimated that there are many tens of thousands of detectable normal pulsars and several thousand detectable MSPs in the Galactic disk alone (Lorimer et al. 2006, Swiggum et al. 2014), yet to date only some ~2613 pulsars have been discovered (Manchester et al. 2005). The shortfall in pulsar detections can inter alia be attributed to RFI, red noise and scintillation (Lorimer 2011). These effects are not addressed in current population synthesis models (Levin et al. 2013). Note that it is important to acknowledge that there could be overestimates in the population models and that the shortfall in pulsar detections is smaller than anticipated.

4.2.2 Limitations of current pulsar search pipelines

It is clear from Chapter 3 that the FFT is one of the most efficient and widely used techniques in pulsar searching. However, despite the success of this algorithm in many large-scale pulsar search campaigns, it is not without limitations. In a recent study by Lazarus et al. (2015), synthetic pulsars with various periods and pulse widths were injected into actual PALFA survey (Cordes et al. 2006) data with the aim to assess the effect of RFI and red noise[1] on the survey sensitivity. The study found that there is a significant degradation in sensitivity of between 10 % and a factor of 2 for pulsars with spin periods between 0.1 s and 2 s and dispersion measure (DM) > 150 pc cm^{-3}, due to red noise induced by RFI,

[1]Red noise is a type of frequency dependent noise with a power spectral density inversely proportional to f^2, which means it has more energy at lower frequencies.

receiver gain fluctuations and opacity variations of the atmosphere. Additionally, a population synthesis analysis, based on an empirical survey sensitivity, found that 35 ± 3 % of pulsars, with predominantly long periods, are missed compared to expectations which are based on the theoretical sensitivity curves as derived from the radiometer equation. With these results it is clear that FFT-based search pipelines are vulnerable to the presence of frequency dependent noise and RFI.

The environment for pulsar searching is continuously changing due to the ever-increasing number of RFI sources near most existing telescopes. Furthermore, the large number of bits being recorded in modern surveys has exacerbated the red noise issues and wideband instruments are also more susceptible to bandpass variations. These factors are limiting the achievable sensitivity of current and future observations. Consequently, pulsar search software needs to constantly evolve in order to adapt to these dynamic environments.

4.2.3 Radio frequency interference

The influence of RFI on radio astronomy measurements ranges from total disruption due to saturation of the receiver to very subtle distortions of the data, which not only significantly degrades the sensitivity of radio observations but also yields large numbers of false positives in pulsar searches (Eatough et al. 2009, Lazarus et al. 2015).

Several methods have been developed to reduce the effects of RFI by flagging or excising input data, such as identifying and clipping spikes either in the time-domain or in the spectral-domain based on a threshold parameter (Fridman 2000, 2001, Fridman & Baan 2001, Nita et al. 2007, Fridman 2008, Offringa et al. 2010, 2012), removing portions of the fluctuating spectrum (Winkel et al. 2007), removing broadband specific RFI (Pen et al. 2009, Eatough et al. 2009) and using dual-station observations to identify common sources of RFI (Bhat et al. 2005). The quality of results from these methods depend on the fine tuning of parameters for specific telescopes and observation types. Tools also exist to manually flag time and frequency channels with significant RFI signals (McMullin et al.

2007). These methods do not work on a streaming basis but rather on integrated data. Consequently, most RFI excision methods used in pulsar search pipelines are algorithms incapable of dealing with the real-time demands of the SKA.

4.2.4 Real-time requirements of the SKA

Discovering new pulsars with the SKA requires high-performance signal detection and computing algorithms. The expected data rates require that substantial processing is performed in real-time, necessitating high-performance computing at Petascale to Exascale levels, as storing the observed data is nearly infeasible. Specifically, one of the principle technological challenges of the mid-frequency range SKA is the identification and development of data processing methods that will reach the required sensitivities in time-domain observations (Cordes et al. 2010).

In contrast, current pulsar search processes require a disproportionate amount of off-line processing time compared to the actual observation time to identify pulsar candidates. Consequently, existing methods are incapable of handling Big Data in real-time. Additionally, these searches yield a large number of false positive detections which adds to the computational burden of sifting procedures to prune the list of follow up observations (Lyon et al. 2016).

Extracting information from large amounts of data naturally requires an effective data analysis and prediction platform to achieve fast response and real-time classification. Real-time candidate selections methods are currently being developed to reduce the high number of false positive detections whilst simultaneously increasing the number of positive detections (Lazarus et al. 2015, Lyon et al. 2016).

4.3 Objective of the study

Processing Big Data in real-time is crucial for future discoveries in astrophysics. This will require considerable advances in hardware, software, data analysis and time series modelling to ensure that the SKA is the premier instrument for pulsar surveys.

The objective with this study is to establish the limitations of current pulsar search pipelines in order to develop algorithms to improve pulsar searching whilst meeting the real-time processing requirements of the SKA. Specifically, this thesis aims to address the following questions related to pulsar searching:

1. Can the shortfall between the predicted and discovered number of pulsars be attributed to the very algorithms employed to find them?

2. Why are FFT-based search pipelines vulnerable to the presence of frequency dependent noise and RFI when various RFI mitigation methods and spectrum whitening routines exist in pulsar search pipelines to prevent this vulnerability?

3. Are the existing spectrum whitening methods modelling the frequency dependent noise correctly?

4. To what extent does a non-stationary noise baseline induced by opacity variations of the atmosphere and system temperature variability contribute to the poor sensitivity of pulsar search pipelines to long period pulsars?

5. How can RFI mitigation be improved whilst meeting the real-time requirements of the SKA?

The first three questions aim to understand algorithmically what the existing RFI mitigation and spectrum whitening methods do and how their presence/absence affects the sensitivity and false positive detections of pulsars search pipelines. Sensitivity, as a statistical measure, refers to the number of detected pulsars, whereas false positive detections measure the number of detections which eventually result in actual pulsars detected.

Current pulsar search pipelines assume that the noise of radio observations is stationary and do not account for the variability associated with receiver gain fluctuations and opacity variations of the atmosphere. The fourth question aims to explore the effect of non-stationary noise on the ability of pulsar search pipelines to detect pulsars.

The last question, aims to address mass data processing with a new real-time RFI mitigation algorithm. Without algorithms capable of extracting information from streaming data in real-time, the discovery of many new pulsars with the SKA will not materialise.

In developing a new real-time pulsar search pipeline adept at dealing with the challenges posed by the SKA, it is important that the shortcomings of exiting methods used in pulsar search pipelines are well understood as this would provide the necessary insights to overcome them. Overcoming these shortcomings will ensure that the new pulsar search pipeline is optimal from the start in terms of detection accuracy and minimal waste of resources.

4.4 Research methodology

The research methodology followed in this thesis is a mixed method approach between quantitative and problem solving methods. A literature review of the constituent parts of typical software pipelines, used for the detection of pulsars, is followed by a proposed framework to quantify the performance of these pipelines. A shortcoming of existing pipelines, identified within the framework, is addressed with a logical inductive designed algorithm capable of pre-processing radio observations in real-time. The effectiveness of this algorithm is empirically tested on synthetic as well as pseudo-real data.

4.5 Outline of the study

This study is outlined in the following structure:

In Chapter 5, an abridged pedagogic introduction to stochastic processes is given,

followed by a description of power-law noise which is a specific type of non-stationary stochastic process. A new method for the discrete simulation of power-law noise is presented. This method is used to simulate synthetic pulsar search data, which contain non-stationary Gaussian noise baselines (i.e. power-law noise) and RFI. These search data are used in Chapter 6 to asses current pulsar search pipelines and to benchmark a new real-time pre-processing algorithm.

The aim of the framework, presented in Chapter 6, is to identify areas in current pulsar search pipelines that can be improved, with respect to how the presence of non-stationary Gaussian noise and RFI affects the sensitivity and false positive ratios.

In Chapter 7, a new real-time streaming technique for the removal of targeted types of RFI from pulsar search data is described.

In Chapter 8, the results obtained from processing the synthetic search data with the framework and the RFI excision algorithm is presented. The efficacy of the RFI excision algorithm is also demonstrated with pseudo-real data.

In Chapter 9, the results presented in Chapter 8 are assessed. Avenues for further inquiry, inspired by the insights gained from the research, is given in the last section of this chapter.

In Chapter 10, a summary of the work is given and it concludes with a broad outlook to the future of radio astronomy in the era of the SKA.

4.6 Scope and limitations of the study

This study focus on pulsar search pipelines which utilise frequency domain search techniques. Synthetic search data are used to quantitatively assess the effectiveness of these existing pipelines as a function of the spin period of pulsars. The metrics used for the assessment are limited to the number of false positive detections per true positive detected pulsar and the sensitivity (also called the true positive rate) of detections. The former met-

ric captures the number of false detections made for every true detection, whereas the latter metric measures the number of actual pulsars detected.

The study neither addresses the effects of dedispersion and pulse duty-cycle on the sensitivity of pulsar search pipelines, nor does it quantify the ability of time domain search techniques to find pulsars. Furthermore, no sophisticated machine learning or sifting algorithms are used to aid the search process. Lastly, the study is limited to searching for normal pulsars and not MSPs or binary pulsars.

The pseudo-real data used to demonstrate the effectiveness of the RFI excision algorithm developed in this thesis are limited to radio observations acquired by the Arecibo radio telescope.

4.7 Conclusion

In this chapter, I have introduced possible limitations of current pulsar search pipelines and how to measure them. I then described the objectives of this study which are to quantify and address the limitations identified in pulsar search pipelines. I also described the proposed research methodology and the outline of the study as well as the scope and limitations thereof.

In the next chapter, I describe the characteristics of non-stationary Gaussian noise processes. Thereafter, a new method for simulating discrete pulsar search data which contain non-stationary Gaussian noise and RFI is presented.

Chapter 5

Simulating Pulsar Search Data

5.1 Introduction

Computer simulations are considered the 'third pillar' of science, complementing theory and experiment. Simulating the desired aspects of a system or phenomenon and performing experiments and virtual tests on it, is fast and inexpensive. A direct consequence of computer simulations is accelerated development through methodical troubleshooting, contributing to more effective and efficient systems design in general.

This chapter presents a method to address a missing tool for simulating pulsar search data that contain different types of RFI and varying noise baselines. The surrogate modelling technique is used in subsequent chapters to inexpensively explore the sensitivity of pulsar search pipelines for different noise and RFI settings as well as to benchmark a new RFI mitigation technique.

In Section 5.2, an introduction to stochastic processes with specific emphasis on the differences between stationary and non-stationary processes is given. The characteristics of frequency dependent noise and Gaussian processes (GPs), which are types of stochastic process, are presented in Sections 5.3 and 5.4 respectively. The different types of RFI that affect pulsar surveys are presented in Section 5.5. The new method for the discrete simulation of frequency dependent noise is presented in Section 5.6. The innovative attribute of the method is that it generates the noise by sampling a Gaussian process (GP). Lastly,

the chapter concludes with the software package that I developed to generate pulsar search data along with instructions on how to use it.

5.2 Stochastic processes

A random variable, $z(\varsigma)$, can be defined from a random event, ς, by assigning values z_i to each possible outcome of the event. A stochastic process, $z(t,\varsigma)$, can be defined as a function of both the event and time, by assigning to each outcome of a random event, ς, a function in time, $z_1(t)$, chosen from a set of functions, $z_i(t)$. This set of functions, $z_i(t)$, is called the ensemble of the random process and may contain infinitely many $z_i(t)$ which can be functions of many independent variables.

If the cumulative probability distribution function of z is given by $F()$, then we can define the general nth order, time-varying, joint distribution function, $F(z_1,\ldots,z_n;t_1,\ldots,t_n)$, for the random variables $z(t_1),\ldots,z(t_n)$. The nth order density function of the process $Z(t)$ is then given by

$$f(z_1,\ldots,z_n;t_1,\ldots,t_n) = \frac{\partial^n F(z_1,\ldots,z_n;t_1,\ldots,t_n)}{\partial z_1 \partial z_2 \ldots \partial z_n}. \tag{5.1}$$

For Gaussian processes, the first and second order properties of the process completely define the process distribution. These properties are defined via the following equations:

$$\mu_z(t) \triangleq \mathbb{E}\{z(t)\} = \int_{-\infty}^{\infty} z f(z,t) dz, \tag{5.2}$$

where μ_z is the mean value of the process and the notation $\mathbb{E}\{\ \}$ refers to the expected value operation taken over the ensemble of processes $\{z(t,\varsigma)\}$.

The statistical variance of the process is given by

$$\sigma^2(t) \triangleq \mathbb{E}\{[z(t)-\mu(t)]^2\} = \int_{-\infty}^{\infty} (z-\mu_z)^2 f(z,t) dz. \tag{5.3}$$

The two-time autocorrelation of the process is given by

$$R_z(t_1,t_2) \triangleq \mathbb{E}\{[z(t_1,\varsigma)z(t_2,\varsigma)]\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(z_1-\mu_{z_1})(z_2-\mu_{z_2})f(z_1,z_2;t_1,t_2)dz_1dz_2. \quad (5.4)$$

In general, the moments of a stochastic process defined in Equations 5.2 and 5.3 vary with time. Hence, such processes are considered non-stationary. A stationary process is defined as one whose density function in invariant to time shifts and thus independent of the times $t_1, t_2, \ldots, t_n$. Such a process is called strict-sense stationary. A wide-sense stationary process is one whose first and second order properties only are independent of time, that is, $\mu_z(t) = \mu_z$, and $R_z(t_1,t_2) = R_z(t_1 - t_2)$.

Examples of stationary and non-stationary stochastic processes include noise processes. These processes are believed to be responsible for the reduced sensitivity of pulsar search pipelines as described in Chapter 4. The next section introduces the characteristics of non-stationary noise processes along with the terminology used in literature to refer to these processes.

5.3 Frequency dependent noise processes

Noise is an extremely important concept since a lot of the technical challenges in radio astronomy come from trying to extract a desired signal from a background of unwanted noise.

The 'purest' form of noise comes from a totally random process. Randomness corresponds with unpredictability: knowing one part of a signal conveys nothing about the future of the signal, i.e. the process has no memory. This kind of minimum-information noise is called white noise, by analogy with white light which is a uniform mixture of all the different possible colours. In the frequency domain white noise has equal power in every unit of bandwidth (i.e. per one Hertz).

This analogy naturally extends to other 'colours'. Just as a piece of coloured glass can be used to transform white light into, say, a deep red hue, natural phenomena like the interstellar medium or the temperature variability of the sky act like filters that alter the balance of frequency components so that the noise is no longer 'white' but has some other quality. In this case, the underlying process is still noise, but it is a little more predictable than white noise, because certain frequencies will be more prominent than others.

A coloured noise process is a one-dimensional stochastic process with a power spectral density (PSD) function that follows a power-law of the form

$$P_x(f) = P_0\left(\frac{f}{f_0}\right)^{\kappa}, \tag{5.5}$$

where f is the temporal frequency, P_0 and f_0 are normalising constants, and κ is the spectral index (Mandelbrot & Van Ness 1968). Typically, the spectral index, κ, lies within the range $[-3, 1]$ (Agnew 1992). The processes within this range are subdivided into 'fractional Brownian motion' with $-3 < \kappa < -1$ and 'fractional white noise' with $-1 < \kappa < 1$ (Mandelbrot 1979, 1983). Special cases within this stochastic model occur at the integer values:

- $\kappa = 0$ White noise.

  A stochastic process is considered white if and only if it is strict-sense stationary and independent at all points. As a consequence, a white noise process has a flat spectrum $\propto f^0$ (constant). Integrating the power spectrum from a finite frequency down to zero frequency (finite bandwidth) results in a finite power at the low frequency end of the spectrum. However, the integral from some finite frequency towards infinity diverges, i.e. there is an infinite amount of power at the hight frequencies. Since white noise converges at low frequencies but diverges at high frequencies, the expected value of the noise converges when averaged over longer time intervals, but the instantaneous value of the noise is undefined.

- $\kappa = -1$ Flicker/pink noise.

  Flicker noise has a power spectrum that varies as f^{-1}, so it is exactly halfway between white noise and red noise. Flicker noise is divergent when integrated to either zero frequency or infinite frequency. It does not have a well-defined mean (i.e. non-stationary) and can give rise to fluctuations coherent over very long scales (Press 1978).

- $\kappa = -2$ Brownian/red noise.

  Red noise is the integral of white noise. Integration in time results in a factor of $1/f$ in the Fourier transform, which is a factor of $1/f^2$ in its square, the power spectrum. Notice, this spectrum converges when integrated from some constant to infinity, but diverges when integrated down to zero frequency. This is just a statement of the fact that if integrated over longer and longer time scales, the value of the Brownian noise wanders farther away from its initial value, i.e. this noise has no well-defined expected value over long time scales (i.e. it is non-stationary).

Another characterisation of noise processes is by means of their autocorrelation functions (ACFs). An important guide to the persistence in a time series is given by the series of quantities referred to as the autocorrelation coefficients, which measure the correlation between successive observations of the same time series. The set of autocorrelation coefficients, arranged as a function of separation in time is the ACF. An estimate of the autocorrelation coefficient $r_{xx}(j)$ at lag j of an N-length sequence x is given by

$$r_{xx}(j) = \frac{\sum_{i=1}^{N-j} (x_i - \bar{x})(x_{i+j} - \bar{x})}{\sum_{i=1}^{N} (x_i - \bar{x})^2}, \tag{5.6}$$

where $\bar{x}$ is the empirical mean.

There are no correlations between samples of a white noise process at different time instances, i.e. the ACF of a white noise process is an impulse function at lag $L = 0$ with all other lags equal or close to zero. If $\kappa < 0$ in Equation 5.5 then $P_x(f)$ goes to infinity as the frequency, f, approaches 0, i.e. low frequency divergent noise. Stochastic processes with PSDs of this form exhibit long memory, i.e. coloured noise processes. Long-memory processes have autocorrelations that persist for a long time, i.e. the ACF is maximum for lag $L = 0$ and decreases gradually for increasing lag values (Beran et al. 2013).

In summary, a non-stationary noise process is characterised by a PSD function that follows a power-law and an ACF that is non-zero for a large number of lags. It is evident from this characterisation why the terms coloured noise, power-law noise and long-memory processes are used interchangeably in literature. Henceforth, I will use the term *frequency dependent noise* when I refer to a noise process that is non-stationary in time and whose PSD function follows a power-law.

In radio astronomy, several data sets have provided evidence for the presence of frequency dependent noise e.g. Cordes (1980) and Lazarus et al. (2015). However, no investigation has been conducted to study the effect of frequency dependent noise, with different correlation lengths, on the ability of pulsar search pipelines to detect pulsars with different periods. Furthermore, there is no rigorous study on the efficacy of spectral whitening techniques to mitigate noise with different correlation lengths in the literature. In order to study this phenomenon, GPs are introduced in the next section, which will be used to simulate frequency dependent noise with different correlation lengths.

5.4 Gaussian Processes

A GP is a type of stochastic process used to describe a distribution over functions (Rasmussen 2006). In this work a GP is the mathematical model used to generate synthetic time series with correlated samples, i.e. frequency dependent noise.

A GP is completely specified by its mean and covariance functions. The mean function $m(x)$ and covariance function $k(x, x')$ of a real process $g(x)$ are defined as:

$$m(x) = \mathbb{E}\{g(x)\}, \tag{5.7}$$

$$k(x, x') = \mathbb{E}\{(g(x) - m(x))(g(x') - m(x'))\}. \tag{5.8}$$

Given Equations 5.7 and 5.8 a GP can be written as:

$$g(x) \sim GP(m(x), k(x, x')). \tag{5.9}$$

In Equation 5.9, the random variables represent the value of a function $g(x)$ at location x. In this work, the GPs are defined over time, i.e. the index set of stochastic variables is time.

A GP allows for the correlation between any pair of outputs in a synthesised time series to be specified by means of the covariance function $k(x, x')$. The covariance function used for generating frequency dependent noise is the squared exponential (SE) kernel function $k_{SE}(x, x')$. The interested reader is referred to Rasmussen (2006) and Roberts et al. (2013) for a comprehensive discussion on this and other kernel functions.

The SE kernel is defined as:

$$k_{SE}(x, x') = h^2 \exp\left[-\left(\frac{x - x'}{\lambda}\right)^2\right], \tag{5.10}$$

where h^2 is an output-scale amplitude which determines the average distance of the function away from its mean and λ is a length (or "memory") scale. In general these free parameters are called hyperparameters. The SE kernel function gives smooth variations with a typical time-scale of λ.

Samples drawn from $g(x)$, of Equation 5.9, yield a time series whose samples are correlated over length scales specified by the value of λ in Equation 5.10. In Section 5.6, I will

show how this time series is used to generate pulsar search data. The different types of RFI that compromise the quality of radio observations are presented next.

5.5 Radio frequency interference

5.5.1 Characterisation

Electromagnetic radiation with frequencies between circa 10 kHz and 100 GHz is referred to as radio frequency. Radio frequency interference (RFI) in the context of a pulsar survey is any signal or disturbance emitted from a terrestrial source that corrupts the measurements of data obtained. The spatial and temporal variability of RFI make it difficult to identify and to mitigate. If RFI is not dealt with then spurious trends may occur in the data collected, thereby decreasing the signal to noise ratio (SNR) and making it more difficult, or impossible, to detect new pulsars.

The mean flux densities of pulsars have a strong inverse dependence with observing frequency Lorimer & Kramer (2005), which means that observations at lower frequencies are desired because pulsars are brighter at these frequencies. However, pulsar surveys are more often than not conducted in the L-band (the 1 to 2 GHz range of the radio spectrum), more specifically the frequency range 1.2 GHz to 1.7 GHz. The reason for observing at the L-band is twofold: firstly, because the International Telecommunications Union have allocated the frequency bands 1.4 GHz to 1.427 GHz and 1.610 GHz to 1.613 GHz for the dedicated use by radio astronomy due to the HI line; secondly, because of the negligible effect of pulsar scattering at these frequencies. Unfortunately, the frequency range 1.2 GHz to 1.7 GHz happens to overlap with frequencies that have been earmarked for other applications such as satellite navigation, telecommunication, aircraft surveillance, amateur radio and digital audio broadcasting (Regulations 2008). Most of the aforementioned RFI sources severely decrease the sensitivity of surveys conducted in parts of the L-band.

Spectrum occupancy in the L-band, depicted in Figure 5.1, is dominated by RFI mainly

from satellites. The colours in Figure 5.1 represent interference from different satellites: red - Afristar, yellow - Thuraya, blue - Inmarsat, cyan - Satellite Radio, grey - IRIDIUM, green - {Galileo, Beidou, GPS, GLONASS} and grey - {Fengun, Meteosat}.

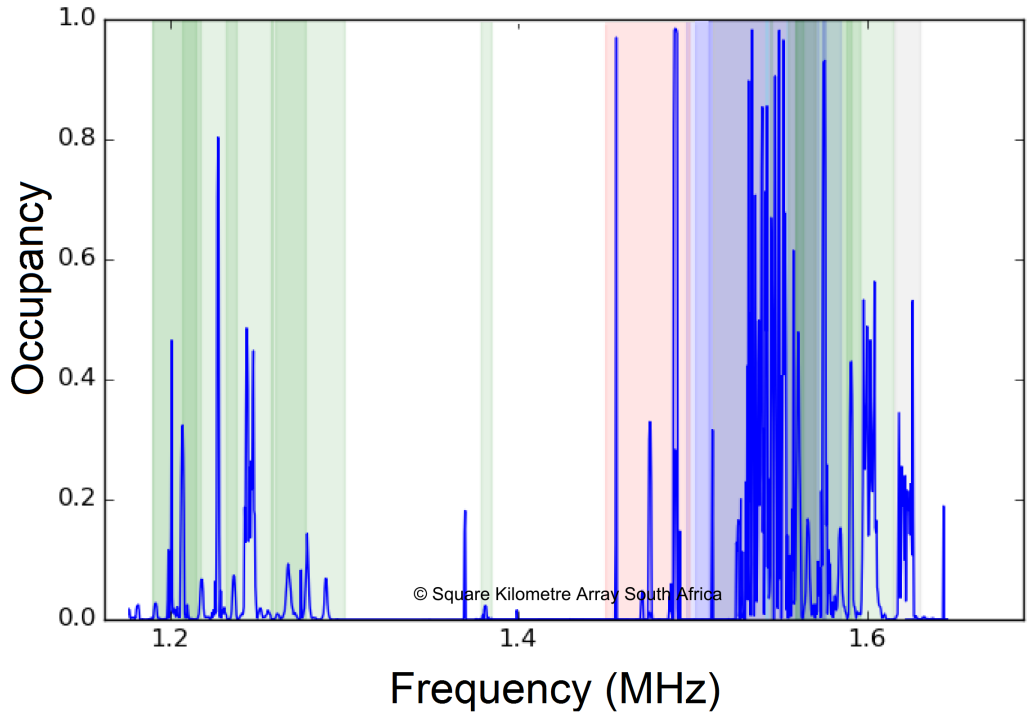


Figure 5.1: This graph is a qualitative view of the typical spectrum occupancy in the L-band taken by the KAT-7 system. The data depicted in blue on the graph is the correlation of the horizontal polarization from one antenna with the vertical polarization from a second antenna, which gives a good representation of anthropogenic RFI. The y-axis is RFI occupancy and the x-axis is the frequency in MHz. If the RFI occupancy is 0.5 (50 %) means that half the data was flagged by the KAT-7 RFI algorithm that only looks at one second of frequency data at a time. The background colours indicate interference from Satellites. *Source: Square Kilometre Array South Africa*

5.5.2 Types of RFI

RFI sources are typically much brighter than astronomical sources and can be separated into approximately five categories, see Figure 5.2 for a schematic example:

(i) Persistent in time, narrow in frequency RFI sources including television, mobile, GPS

and communication satellite constellations (OrbComm, Irridium) (Bhat et al. 2005).

(ii) Periodic in time, narrow in frequency RFI sources including radar (Niamsuwan et al. 2005).

(iii) Periodic in time, broad in frequency RFI sources including high-voltage power cables, and electric fences.

(iv) Aperiodic in time, broad in frequency RFI sources including lightning and aeronautic communication.

(v) Impulsive RFI sources, either in time or frequency, such as air traffic communication and low Earth orbit (LEO) satellites.

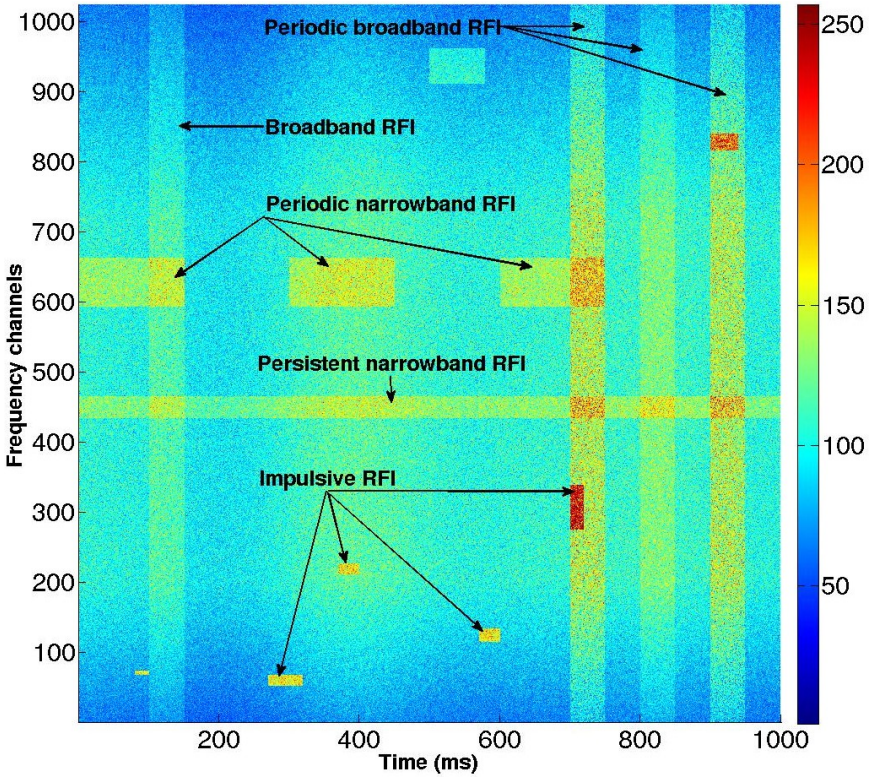


Figure 5.2: 2D plot of an 8-bit mock filterbank file with a non-stationary baseline and non-uniform bandpass which contains the five categories of RFI.

These categories are not strict, as RFI sources cover a wide amount of the time-frequency parameter space of possible signal structures. Furthermore, a RFI source can be cate-

gorised into multiple categories depending on the time and frequency resolution of the observational instrumentation. In fast radio burst observations, a balance between time and frequency resolution is sought due to DM considerations. As an example, short-period radar signals will show time-variable structure based on the signal encoding. Conversely, in the observation of stable, weak sources, spectra are integrated for long time periods as no variability is expected. The same radar signal will appear as a bright, stable event as the time resolution has effectively averaged out the time-variable signal.

The software that I developed for generating synthetic pulsar search data allows for the five different categories of RFI to be injected into the data. The next section describes the characteristic of pulsar search data and presents the method for generating it.

5.6 Pulsar search data

5.6.1 Characterisation

In order to determine how phenomena like RFI and non-stationary noise can hinder pulsar detections it is necessary to first understand the nature of the acquired data. The data that are searched, from pulsar observations, are time series of total power in many frequency channels typically referred to as filterbank data. The number of frequency channels, the temporal resolution and the dynamic range (i.e. 1-bit, 8-bit or 16-bit) of the data are unique to each survey.

Filterbank files contain quantised power values computed by superimposing a number of single Nyquist power measurements (see Equation 5.11), where the number of measurements that are superimposed depends on the channel bandwidths and the sampling rate. The power measurement of a single Nyquist sample is computed from the real and imaginary parts of the raw voltages associated with either linear or circular polarised electromagnetic waves observed by radio telescopes. The power measurement of a single Nyquist sample

is given as:

$$\text{Power} = X_{real}^2 + X_{imag}^2 + Y_{real}^2 + Y_{imag}^2, \tag{5.11}$$

where X and Y are either the horizontal and vertical components of linear polarisation or the left and right handed components of circular polarisation.

The power values found in filterbank files comprise both signal and noise. The noise levels in the filterbank files are 'related to' the overall system temperature which is affected by RFI, the sky temperature and the receiver temperature. During an observation various sources of RFI with different brightness levels are encountered so the duration and magnitude of the non-stationarity associated with each of these phenomena differ greatly.

Software currently available for generating synthetic pulsar search data allows only for the generation of additive white Gaussian noise. Hence, the need for software to emulate wandering noise baselines induced by phenomena like the fluctuating nature of the sky temperature (Lorimer & Kramer 2005, Nice et al. 1995) and time/frequency varying RFI, in a controlled fashion.

5.6.2 Simulation of synthetic pulsar search data

Time series with correlated samples (i.e. frequency dependent noise) can be generated by drawing samples from a GP as described in Section 5.4, which is equivalent to passing additive white Gaussian noise through a shaping filter. The shaping filter is a dynamic filter, usually a low pass filter. The SE kernel defined in Equation 5.10 which is used to construct the covariance function of the GP acts as the dynamic filter in the context of GPs. The response of the frequency dependent noise can be varied by adjusting the parameters of the shaping filter, i.e. the hyperparameters of the SE kernel defined in Equation 5.10.

A time series of N samples generated with a GP requires a covariance matrix of size $N \times N$ and a vector of size N to be stored in memory, which is impractical for very large N. Consequently, GPs are computationally expensive and requires significant amounts

of memory. To circumvent the necessity to store these $N^2 + N$ samples in memory, I rewrote the GP as a state-space model. Instead of computing the matrix-vector product of the covariance matrix and a vector containing samples drawn from a standard normal, I constructed a low-pass filter (see Equation 5.12) based on the SE kernel defined in Equation 5.10 and convolved that with random samples drawn from a Gaussian distribution with zero mean and unit variance ($\mathcal{N}(0,1)$) (see Equation 5.13). In Equation 5.12, $\epsilon = 1 \times 10^{-5}$, t is the sampling interval and N the number of samples in the observation. Consequently, the convolution yields a vector, $\mathbf{w}$ with samples correlated over length scales defined by λ and magnitudes proportional to h (see Equation 5.14).

$$\mathbf{u} := h^2 \exp\left[-\left(\frac{t}{\lambda}\right)^2\right] \forall\, t \in \mathbb{R} \quad \text{such that} \quad \mathbf{u} > \epsilon \tag{5.12}$$

$$\mathbf{v} := a_1, a_2, ..., a_N \sim \mathcal{N}(0,1) \tag{5.13}$$

$$\mathbf{w} = \mathbf{u} * \mathbf{v} \tag{5.14}$$

To generate N data points which are correlated over long length scales (i.e. $\lambda \gg$) require N random samples drawn from a standard normal distribution ($\mathcal{N}(0,1)$) to be convolved with a finely sampled low-pass filter which is compact on a large interval. However, convolving two large vectors is computationally very expensive. To circumvent this challenge, data points are generated with the required correlation length by convolving a fraction of the random samples drawn from a standard normal distribution with a coarsely sampled low-pass filter and then interpolating between the resultant points to produce a time series with the desired number of points.

The vector $\mathbf{w}$ generated by convolving the low-pass filter with samples drawn from a standard normal distribution is not always positive. However, in order to simulate a non-stationary process the mean and variance should vary with time. One way to generate a non-stationary process is by drawing samples from normal distributions whose means and variances are set equal to $\mathbf{w}$ and $\sqrt{\mathbf{w}}$ for each sample of $\mathbf{w}$ if and only if $\mathbf{w}$ is non-negative.

Therefore, a new vector $\mathbf{b}$ is defined as:

$$\mathbf{b} = \mathbf{w} - \min(\mathbf{w}), \tag{5.15}$$

such that the offset is equal to zero and all the values are non-negative.

The mean vector $\mathbf{b}$ is used to generate samples for the vectors X_{real}, X_{imag}, Y_{real}, Y_{imag}:

$$X_{\text{real}}, X_{\text{imag}}, Y_{\text{real}}, Y_{\text{imag}} \overset{i.i.d.}{\sim} \mathcal{N}\left(\mathbf{b}, \sqrt{\mathbf{b}}\right), \tag{5.16}$$

such that the power for each sample in the synthetic filterbank file can be computed using Equation 5.11. Note that computing the power in this way is equivalent to drawing a sample from a χ^2 distribution with 4 degrees of freedom.

A large value for the length scale variable λ of the low-pass filter results in a slowly wandering baseline as depicted in Figure 5.3a. As the value of λ deceases the baseline wandering becomes more capricious as depicted in Figure 5.3b to Figure 5.3e.

The next section gives a brief overview of the software developed for the purpose of simulating pulsar search data along with instructions on how to use it.

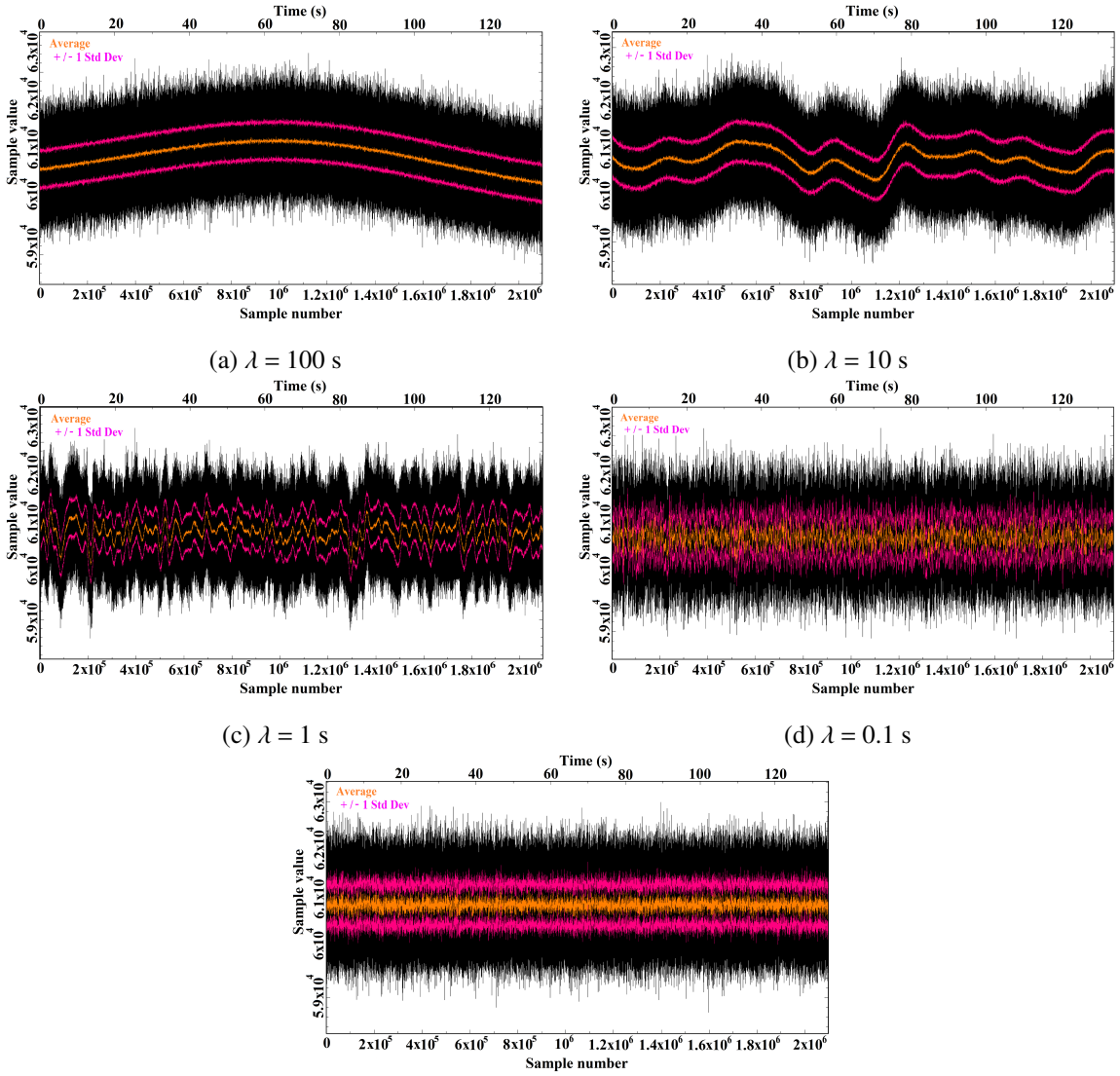
5.7 Ersatz: synthetic file generation software

The Python code developed for simulating pulsar search data is called `Ersatz`. It is publicly available and can be downloaded from: `https://github.com/EllieVanH/FilterbankFileGeneration`.

An example of how to call the `Ersatz` function is:

```
» python ersatz.py --tobs 20.0 --tsamp 64 --fch1 1536
--foff -0.62890625 --nchans 512 --nbits 8 --statioanry "No"
--noiseInput FakeNoiseParameters.txt --bandPass
```

A summary of the inputs for `Ersatz` is give in Table 5.1. The noise characteristics,



(a) $\lambda = 100$ s

(b) $\lambda = 10$ s

(c) $\lambda = 1$ s

(d) $\lambda = 0.1$ s

(e) $\lambda = 0.01$ s

Figure 5.3: Examples of dedispersed time series that correspond to the five correlation lengths used to simulate non-stationary noise processes. Black represents the actual signal in the filterbank file, orange the mean and pink the standard deviation (1σ) of the dedispersed time series. The correlation length decreases from (a) 100 s to (e) 0.01 s.

RFI and bandpass shape are passed to `Ersatz` via a standard text file, an example of which can be seen in Figure 5.4a. A description of the identifiers used in the text file is given in Figure 5.4b. If any of the RFI phenomena should be ignored then their associated 'Occurrences' parameter should be set to zero.

The dynamic range of the data depends on the number of bits (`nbits`) specified to

Table 5.1: A summary of the inputs for `Ersatz` for generating pulsar search data.

Input	Description
`--tobs`	Total observation time in seconds (def=10).
`--tsamp`	Sampling interval in microseconds (def=1000).
`--fch1`	Highest frequency of observation in MHz (def=1550.0).
`--foff`	Channel bandwidth in MHz (def=-0.078125 MHz).
`--nchans`	Number of output channels (def=16).
`--nbits`	Number of bits (8, 16, 32) (def=8).
`--stationary`	Generate stationary noise (Yes/No) (def=Yes).
`--noiseInput`	Name of the file which contains the noise and RFI specifications.
`--bandPass`	Set this flag if the bandpass should have the shape specified in the noiseInput text file.

represent each datum. For the simulated data, one standard deviation of the noise σ is set to have a value one tenth of the dynamic range. The mean of the data is placed at 4σ, which leaves 6σ above for the positive noise and signal. To control the σ of the simulated data, the value of the 'Baseline Amplitude' parameter in the text file that is passed to `Ersatz` should be set accordingly (see Figure 5.4b).

The output of `Ersatz` is a binary `SIGPROC` filterbank file with a header containing all the metadata passed to `Ersatz` as inputs. This file type can be processed by most pulsar search software. Examples of pulsar search data produced by `Ersatz` can be seen in Figures 3.5 and 5.2.

5.8 Conclusion

A frequency dependent process is characterised by a given distribution of the power per unit frequency along the available frequency bandwidth. Frequency dependent stochastic processes are prevalent in many disciplines including engineering, finance and astrophysics. Hence, the terms *power-law processes*, *coloured processes* and *long memory processes* are used interchangeably in the literature to refer to stochastic processes that are non-stationary.

In this chapter, I presented a new method for simulating pulsar search data with non-

```
**************************************
Baseline Lambda 2
Baseline Amplitude 4
Broadband Occurrences 0
Broadband t_start 0
Broadband t_end 0
Broadband Magnitude 0
Narrowband Occurrences 0
Narrowband F_start 0
Narrowband F_end 0
Narrowband t_start 0
Narrowband t_end 0
Narrowband Magnitude 0
Periodic Broadband Occurrences 0
Periodic Broadband Period 0
Periodic Broadband Duty cycle 0
Periodic Broadband t_start 0
Periodic Broadband t_end 0
Periodic Broadband Magnitude 0
Periodic Narrowband Occurrences 0
Periodic Narrowband Period 0
Periodic Narrowband Duty cycle 0
Periodic Narrowband F_start 0
Periodic Narrowband F_end 0
Periodic Narrowband t_start 0
Periodic Narrowband t_end 0
Periodic Narrowband Magnitude 0
BandPass ramp-up 0.3
BandPass ramp-down 0.2
BandPass Amplitude 0.5
**************************************
```

(a)

Identifier	Description
Baseline	Precedes the variables associated with the non-stationary baseline.
Broadband	Precedes the variables associated with impulse RFI.
Narrowband	Precedes the variables associated with narrowband RFI.
Bandpass	Precedes the variables associated with giving the bandpass a shape.
Baseline Lambda	Non-stationary baseline correlation length in seconds.
Baseline Amplitude	Amplitude of the non-stationary noise (0< value ≤ 6).
Occurrences	Number of RFI instances.
t_start	The start time of the RFI instances (s)
t_end	The end time of the RFI instances (s)
Magnitude	The magnitude of the noise i.t.o. the number of standard deviations from the expected value (0< value ≤6)
F_start	First frequency affected by RFI artefact (MHz)
F_end	Ending frequency affected by RFI artefact (MHz)
Period	Period of the periodic noise (s)
Duty cycle	Specify how long the RFI should be 'on' as a percentage of its period (0<value<99).
Ramp-up	Specify the protion of highest frequency channels to be modulated (0<value<1).
Ramp-down	Specify the portion of lowest frequency channels to be modulated (0<value<1).
Amplitude	Specifies the height of the slope (0<value<1).

(b)

Figure 5.4: (a) An example of a text file which contains specifics regarding the noise characteristics, RFI and bandpass shape to be simulated with `Ersatz`. (b) Description of the identifiers in the text file.

stationary noise baselines. The non-stationarity is induced by frequency-dependent receiver gain fluctuations, opacity variations of the atmosphere, sky and system temperature variability and RFI (Lazarus et al. 2015). It is poorly understood how the aforementioned phenomena affect the ability of pulsar search pipelines to detect pulsars. Hence, in the next chapter `Ersatz`, the synthetic file generation software I described here, is used to produce

pulsar search data with varying correlation lengths to study the effect of these phenomena on the ability of pulsar search pipelines to detect pulsars. Additionally, these simulated data will allow the investigation of the efficacy of existing spectrum whitening techniques to mitigate frequency dependent noise with different correlation lengths.

Chapter 6

Framework for performance assessment of pulsar search pipelines

6.1 Introduction

This chapter is based on an article (van Heerden et al. 2016) that has been published in *Monthly Notices of the Royal Astronomical Society*.

It has been postulated, as I have highlighted in Chapter 3, that frequency dependent noise and RFI are responsible for the reduced sensitivity of pulsar search pipelines to normal pulsars, i.e. pulsars with long periods. Therefore, in this chapter I present a framework for explicitly assessing the effect of these two phenomena on the signal to noise ratio, the number of false positives detected per true positive and the sensitivity of pulsar search pipelines. The results and discussion of this work are presented in Chapter 8 and Chapter 9 of this thesis respectively.

In summary, the aims of this assessment are:

(i) to quantify the effect that non-stationary Gaussian noise and RFI has on the performance of pulsar search pipelines;

(ii) to examine the effectiveness of the current spectrum whitening methods available in pulsar search software suites;

(iii) to determine if detrending the data with a moving average filter before searching for

pulsars is effective;

(iv) to examine the effectiveness of the current RFI detection and mitigation methods available in pulsar search software suites.

(v) to investigate the reduction in sensitivity as a function of both the correlation length of the non-stationary noise and the pulse period.

In Section 6.2, I mathematically describe the different spectrum whitening algorithms available in the pulsar search software packages `SIGPROC` and `PRESTO`. The effectiveness of these spectrum whiting methods is to be investigated as part of the framework for assessing pulsar search pipelines. In Sections 6.3.1 to 6.3.5, I introduce the experimental framework I used for generating and processing filterbank files which contain pulsars embedded in non-stationary Gaussian noise amidst RFI.

It is worth noting that this study differs from the Lazarus et al. (2015) study in that the aim is to quantify the sensitivity of different pulsar search pipelines as a function of noise correlation length and pulsar spin period, whereas the latter aimed at quantifying the Arecibo PALFA survey's sensitivity as a function of DM and pulsar spin period.

6.2 Spectrum whitening

6.2.1 Motivation for spectrum whitening

Spectrum whitening is an operation which attempts to make the spectrum of a signal more uniform, i.e. similar to a white noise spectrum. In the case of pulsar searching, well-behaved white noise is sought after, because it simplifies any attempt at estimating the significance levels of any signal present in the data and consequently makes detection easier. Hence, it is standard practice to whiten the power spectral density by suppressing frequency dependant noise, in particular red noise, so that the spectrum is as uniform as possible.

The spectrum whitening techniques implemented in SIGPROC and PRESTO are similar in that they aim to normalise the spectrum. However, the way in which these techniques algorithmically operate in normalising the spectrum is quite different. In the subsequent two sections, I will mathematically describe the different whitening options available in SIGPROC and PRESTO.

6.2.2 Spectrum whitening in SIGPROC

In SIGPROC there are three spectrum whitening options for the SEEK function which searches dedispersed time series for pulsars. In all three approaches the spectrum is divided into blocks of max{128, (NumberOfSpectralDataPoints/400000)} Fourier bins. The simulation parameters of this analysis resulted in the number of Fourier bins per block to be consistently 128 and thus for all subsequent explanations I assume that the spectrum is divided into blocks of 128 Fourier bins as depicted in Figure 6.1.

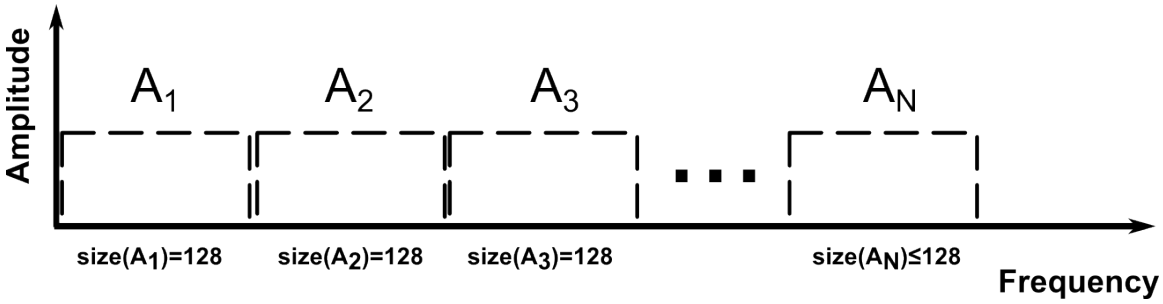


Figure 6.1: Amplitude spectrum partitioning for the whitening algorithm implemented in SIGPROC, see text for details.

The algorithms implemented for the three options in SIGPROC are:

- Option 1:

The default whitening algorithm executed when the function SEEK is called computes the mean and corrected sample standard deviation, $s = \sqrt{1/(N-1)\sum_{i=1}^{N}(x_i - \bar{x})^2}$, with iterative 3σ spike rejection for each of the blocks A_1,...,A_N in Figure 6.1. The iterative 3σ spike rejection is done so as to try to avoid overestimating the baseline noise. The mean is subtracted from each Fourier bin in the block, whereafter the bins are scaled by the corrected

sample standard deviation of that particular block. The algorithmic steps are detailed in Algorithm 1.

Algorithm 1 SIGPROC: `default`

1: **for** $i = 1, ..., N$ **do**
2: sum = 0, sumsq = 0, $\mu_i = 0$, $\text{rms}_i = 0$
3: numRejected = 0, numRejectedOld = 0, counter = 0
4: **for** $j = 1, ..., 128$ in A_i **do**
5: sum = sum + $A_i[j]$
6: sumsq = sumsq + $A_i[j] * A_i[j]$
7: **end for**
8: s = sum
9: ss = sumsq
10: rms_i = sqrt((sumsq-sum*sum/(128-numRejected))/(128-numRejected-1.0))
11: μ_i = sum/(128-numRejected)
12: sum = s
13: sumsq = ss
14: numRejected=0
15: **for** $j = 1, ..., 128$ in A_i **do**
16: **if** (abs($A_i[j] - \mu_i$)>3*rms_i) **then**
17: sum = sum - $A_i[j]$
18: sumsq = sumsq - $A_i[j] * A_i[j]$
19: numRejected = numRejected -1
20: **end if**
21: **end for**
22: **if** ((numRejected == numRejectedOld) or (counter > 4)) **then**
23: $A_i^{new} = (A_i^{old} - \mu_i)/\text{rms}_i$
24: **else**
25: counter = counter + 1
26: numRejectedOld = numRejected
27: Go to line 10
28: **end if**
29: **end for**

- Option 2:

The whitening algorithm executed when the function `SEEK` is called with the flag `-submn` computes the mean and rms for each of the blocks A_1,...,A_N in Figure 6.1. The mean is subtracted from each Fourier bin in the block, whereafter the bins are scaled by the rms of that particular block. The algorithmic steps are detailed in Algorithm 2.

Algorithm 2 SIGPROC: `submn`

for $i = 1, ..., N$ **do**
    $\mu_i = \text{mean}(A_i)$
    $s_i = \text{rms}(A_i)$
    $A_i^{\text{new}} = (A_i^{\text{old}} - \mu_i)/s_i$
end for

- Option 3:

The whitening algorithm executed when the function `SEEK` is called with the flag `-submjk` computes the mean and corrected sample standard deviation with iterative 3σ spike rejection (see Algorithm 1 for details on iterative spike rejection when calculating the mean and standard deviation) for each of the blocks A_1,...,A_N in Figure 6.1. Thereafter, the gradients of the mean and corrected sample standard deviation between adjacent blocks of 128 Fourier bins are computed. For each Fourier bin $j = 1, ..., 128$ in a block, the mean is subtracted and the result scaled with the corrected sample standard deviation, whereafter the mean and corrected sample standard deviation is updated with the calculated gradients for that particular block. The algorithmic steps are detailed in Algorithm 3.

Algorithm 3 SIGPROC: `submjk`

for $i = 1, ..., N$ **do**
    $\mu_i = \text{mean}(A_i)$
    $\mu_{i+1} = \text{mean}(A_{i+1})$
    s_i = corrected standard deviation(A_i)
    s_{i+1} = corrected standard deviation(A_{i+1})
    $\text{slope}_{\text{mean}_i} = (\mu_{i+1} - \mu_i)/128$
    $\text{slope}_{s_i} = (s_{i+1} - s_i)/128$
    for $j = 1, ..., 128$ in A_i **do**
        $A_i^{\text{new}}[j] = (A_i^{\text{old}}[j] - \mu_i)/s_i$
        Update: $\mu_i = \mu_i + \text{slope}_{\text{mean}_i}$
        Update: $s_i = s_i + \text{slope}_{s_i}$
    end for
end for

6.2.3 Spectrum whitening in PRESTO

In PRESTO there is only one spectrum whitening technique implemented to suppress frequency dependent noise (Ransom et al. 2002). The median power level is measured in blocks across Fourier bins and then multiplied by log 2 to convert the median value to an equivalent mean level assuming that the powers are distributed exponentially. Thereafter, the measured mean power values (variable P in Algorithm 4) are used to compute the slope between two adjacent Fourier bins, which in turn is used to normalise the complex Fourier amplitudes (variable A in Algorithm 4).

The number of Fourier frequency bins per block starts with 6 and increases logarithmically to 200, see Figure 6.2. Thus, for frequencies where there is little coloured noise, the number of bins used per block is 200. The algorithmic steps for the spectrum whitening technique implemented in PRESTO are detailed in Algorithm 4.

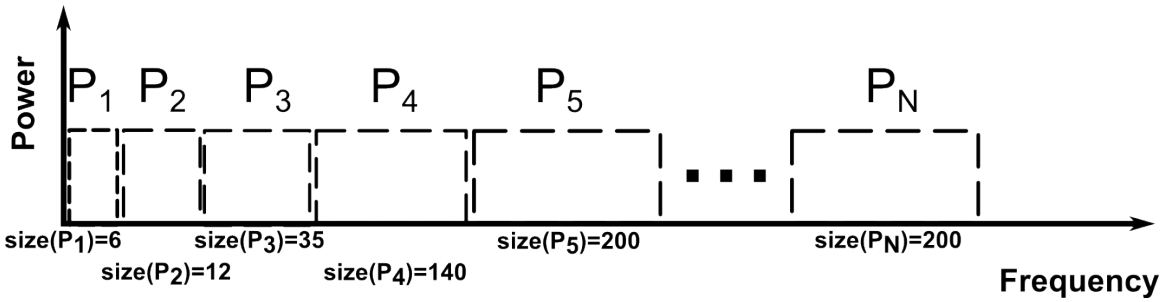


Figure 6.2: Power spectrum partitioning for the whitening algorithm implemented in PRESTO, see text for details.

Irrespective of the red noise suppression algorithm being applied, PRESTO by default normalises the power spectrum using median blocks before performing harmonic summing and searching.

6.3 Framework for Pulsar Search Pipeline Analysis

In this section, I introduce the framework that was developed to generate and process non-stationary noise files with RFI. This framework allows for the understanding of how non-

Algorithm 4 `PRESTO`: `default`

for $i = 1, ..., N$ **do**
  $\mu_i = \text{median}(\text{P}_i)/\log 2$
  $\mu_{i+1} = \text{median}(\text{P}_{i+1})/\log 2$
  $\text{slope}_i = (\mu_{i+1} - \mu_i)/(\text{size}(\text{P}_i) + \text{size}(\text{P}_{i+1}))$
  $\text{lineoffset} = \frac{1}{2}(\text{size}(\text{P}_i) + \text{size}(\text{P}_{i+1}))$
  for $j = 1, ..., \text{size}(\text{P}_i)$ **do**
    Update: $\text{lineval} = \mu_i + \text{slope}_i \times (\text{lineoffset} - j)$
    Update: $\text{scaleval} = 1/\sqrt{\text{lineval}}$
    Update: $\mathbf{Re}(\text{A})_i^{\text{new}}[j] = \mathbf{Re}(\text{A})_i^{\text{old}}[j] \times \text{scaleval}$
    Update: $\mathbf{Im}(\text{A})_i^{\text{new}}[j] = \mathbf{Im}(\text{A})_i^{\text{old}}[j] \times \text{scaleval}$
  end for
end for

stationary noise processes with different correlation lengths can impede the detection of pulsars with specific periods. Additionally, it contributes to the understanding of how RFI can pass undetected through current pulsar search pipelines and the consequences of not mitigating these spurious sources of interference.

The generation part of this framework includes the synthetic observation parameters (see Section 6.3.1), the choice of RFI injected into a subset of the files (see Section 6.3.2), the experimental design specifics for each experiment (see Section 6.3.3) and the periods of the pulsars injected for this analysis (see Section 6.3.4).

The processing part of this framework includes the different configurations of `SIGPROC` and `PRESTO` that were analysed (see Section 6.3.5).

The synthetic filterbank files processed in this thesis were generated with the software `Ersatz` described in Chapter 5. Pulsars were injected into these files with the function `inject_pulsar` available in Dr. Mike Keith's version of `SIGPROC`[1].

6.3.1 Simulated observation parameters

The observation parameters that were used for generating the synthetic filterbank files were chosen to match the Arecibo PALFA survey (Lazarus et al. 2015) parameters and are sum-

[1] https://github.com/SixByNine/sigproc

Table 6.1: Simulated observation parameters

Parameter	Value
t_{obs}	300 s
t_{samp}	64 μs
n_{bits}	8
n_{nchans}	512
f_{low}	1214 MHz
f_{high}	1536 MHz
Bandwidth, Δf	322 MHz
Channel Bandwidth, Δf_{chan}	628.91 kHz

marised in Table 8.1.

6.3.2 RFI injection

For the experiments aimed at investigating the effects of RFI on the performance of pulsar search pipelines I injected the same RFI into all the filterbank files, see Figure 6.3. The choice of injected RFI was obtained by studying spectrum occupancy data from the KAT-7 radio telescope over several months (see Figure 5.1) and cross-correlating that with the L-band spectrum allocation as determined by the International Telecommunication Union (ITU) (Regulations 2008). The RFI injected includes:

(i) broadband periodic RFI with a period of 0.02 s and a duty cycle of 50 %,

(ii) narrowband periodic RFI with a period of 12 s and a duty cycle of 25 % affecting the frequency channels 1.266 GHz to 1.276 GHz,

(iii) several instances of narrowband RFI with random durations affecting the frequency channels identified from the RFI characterisation plot in Figure 5.1.

Various routines exist in current pulsar search pipelines for excising bright RFI, but the effect of weak and unknown sources of RFI is important and not perfectly understood. Therefore, the magnitudes of the various instances of injected RFI were deliberately chosen

to be within one sigma of the baseline for a single sample such that the effect of weak RFI on pulsar search pipelines could be investigated. The percentage of samples affected by RFI is 12 % for each filterbank file.

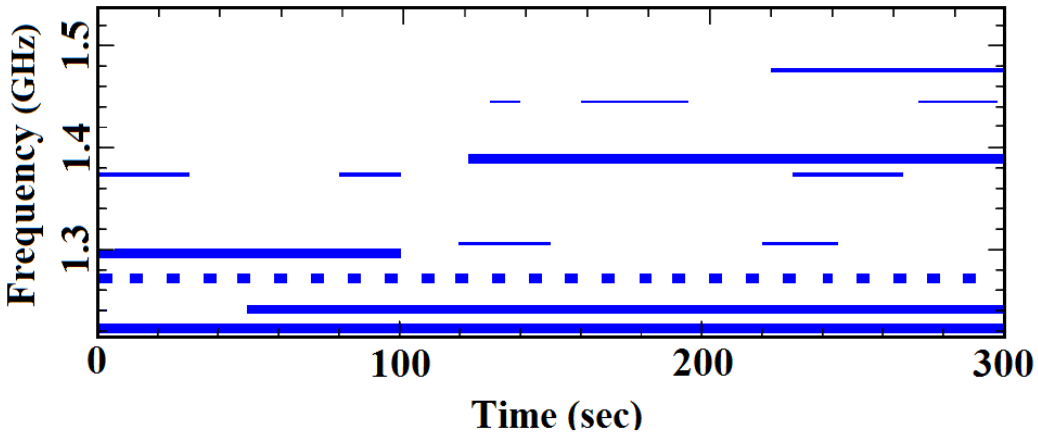


Figure 6.3: The RFI injected into each filterbank file.

6.3.3 Experiments

The experiments that I designed for analysing various configurations of `SIGPROC` and `PRESTO` are summarised in Table 6.2.

The design of experiments 1 to 4 is such that each one emulates a blind pulsar survey. Each experiment comprises one hundred simulated pointings with a subset of these containing a pulsar. The differences between the experiments are the type of noise processes simulated and whether RFI is present of not.

Experiment 1 comprises one hundred files with stationary Gaussian noise. Fifteen of the hundred files contain an injected pulsar (see Table 6.3) and the remainder are without. The results from experiments 2 to 4 will be benchmarked against the results of experiment 1 because of its idealised noise process and lack of RFI.

Experiment 2 comprises one hundred files with non-stationary Gaussian noise. Fifteen of the aforementioned files contain an injected pulsar that is unique (see Table 6.3) and the remainder of the files are without a pulsar. The non-stationary Gaussian noise processes

were generated according to the procedure described in Section 5.6.2. Note that every noise process is unique in that each one is defined by a different non-stationary vector, **g**, and the additive Gaussian noise has zero mean and variance proportional to the square root of the non-stationary vector (see Equation 5.16). The length scales, λ, for the non-stationary variability of the noise baselines range from 10^{-2} s to 10^{2} s in factors of 10, i.e. twenty files were generated for each of the values of λ. Consequently, each file exhibits a unique variation because of the stochastic nature of the generation process despite having the same correlation length.

The correlation lengths can be chosen to represent any timescale that could be considered to have an effect on a survey's sensitivity to periodic pulsars, and could result from instrumental variability, to environmental effects and RFI. The power spectrum of a non-stationary noise baseline with a given correlation length will contain more power in the frequencies that correspond to that length. For this reason, I chose values that sampled a broad range of length scales relevant to the pulse periods searched, as indicated above. Comparing the results from experiment 2 with the results of experiment 1 enables the quantification of the effect that non-stationary Gaussian noise has on the performance of pulsar search pipelines. Moreover, experiment 2 allows for the determination of the effectiveness of the spectrum whitening techniques described in Section 6.2 and whether or not detrending the data with a moving average filter before searching for pulsars is effective.

Experiment 3 is identical to experiment 1 except for the addition of RFI (see Section 6.3.2). The experimental design of experiment 3 serves to investigate the ramifications when weak RFI (see Section 8.3.5) passes undetected through a pulsar search pipeline. Furthermore, it serves to investigate the efficacy of RFI detection and mitigation algorithms currently employed.

Table 6.2: Summary of the experiments conducted

Experiment	# Files	# Pulsars	Noise	h	λ (s)
1. Stationary	100	15	Stationary Gaussian	-	-
2. Non-stationary	100	15	Non-stationary Gaussian	0.1, 0.2, 0.3, 0.4	0.01, 0.1, 1.0, 10.0, 100.0
3. Stationary +RFI	100	15	Stationary Gaussian	-	-
4. Non-stationary +RFI	100	15	Non-stationary Gaussian	0.1, 0.2, 0.3, 0.4	0.01, 0.1, 1.0, 10.0, 100.0
5. All pulse periods per λ	75	75	Non-stationary Gaussian	0.1, 0.2, 0.3, 0.4	0.01, 0.1, 1.0, 10.0, 100.0
6. All pulse periods per λ + RFI	75	75	Non-stationary Gaussian	0.1, 0.2, 0.3, 0.4	0.01, 0.1, 1.0, 10.0, 100.0

Experiment 4 is identical to experiment 2 except for the addition of RFI (see Section 6.3.2). Comparing the results from experiment 4 to the results of experiments 1, 2 and 3 respectively serves to quantify the combined effect that non-stationary Gaussian noise and RFI have on the performance of pulsar search pipelines and to deduce which phenomenon has the greatest impact on said performance.

Experiment 5 comprises seventy five files in total, fifteen files per correlation length λ (see Table 6.2). Each one of the pulsars in Table 6.3 was separately injected into the fifteen files with the same correlation length and this was repeated for all five values of λ.

Experiment 6 is identical to experiment 5 except for the addition of RFI (see Section 6.3.2). The experimental design of experiments 5 and 6 serves to investigate the reduction in sensitivity of pulsar search pipelines as a function of both the correlation length of the non-stationary noise and the pulse period of a pulsar.

Lastly, experiments 2, 4, 5 and 6 are repeated for four different values of the magnitude parameter h defined in Equation 5.12, namely $h = 0.1, 0.2, 0.3, 0.4$.

6.3.4 Pulsar properties

The properties of the fifteen pulsars that I randomly injected into the synthetic filterbank files were taken in part from the Arecibo sensitivity study (Lazarus et al. 2015) and are summarised in Table 6.3.

A pulse with the profile of pulsar PSR B0833-45 at 1.4 GHz obtained from the EPN-database (Lorimer et al. 1998) with a duty cycle of 12 % was injected into all the files.

6.3.5 Pulsar search pipeline configurations

All of the files that I generated for the experiments described in Table 6.2 were processed by both `SIGPROC` and `PRESTO`, i.e. twelve different configurations of `SIGPROC` (see Table 6.4) and eight different configurations of `PRESTO` (see Table 6.5) were used to search every

Table 6.3: Synthetic pulsar properties

Parameter	Value			
Period(ms)	1.102	2.218	5.218	10.870
	18.505	61.965	126.175	286.555
	533.320	850.158	1657.496	2643.410
	3927.013	5580.899	9964.532	
Amplitude	All pulsars are detectable with a detection significance of ~ 12 in SIGPROC in the presence of stationary Gaussian noise when processed with pipeline H in SIGPROC.			
Duty cycle	12 % (fixed)			
Dispersion measure	68			

single synthetic file. The aim with this analysis is not to investigate sensitivity as a function of DM so all the files were dedispersed at the correct DM.

SIGPROC by default removes the baseline of a dedispersed time series by linearly detrending the time series unless the flag `-nobaseline` is set when the `dedisperse` function is called. In addition to the option available in SIGPROC for detrending the baseline, a 10 s moving average filter was implemented as a second option for normalising the baseline of a dedispersed time series. The red noise mitigation techniques applied in the different SIGPROC pipelines are described in Section 6.2.2.

To process all the files with SIGPROC the following functions and their associated flags were called:

(a) function `dedisperse` with the flags `-d`, `-o` and with/without `-nobaseline`,

(b) function `seek` (number of summed harmonics is 16) with the flags `-z` and `-submn` or `-submd` or `-submjk`,

(c) function `best` with flag `-s8`.

The function `best` in SIGPROC produces a '.lis' file which was searched for possible candidates based on the SNR of the peaks.

Table 6.4: The twelve SIGPROC pipeline configurations used to process all the files in this analysis.

Pipeline	Baseline	Red-noise mitigation
A	Removed	default
B	Removed	submn
C	Removed	submjk
D	Removed	-
E	Intact	default
F	Intact	submn
G	Intact	submjk
H	Intact	-
I	Moving average filter	default
J	Moving average filter	submn
K	Moving average filter	submjk
L	Moving average filter	-

The RFI mask configuration option in Table 6.5 refers to the RFI mask computed in PRESTO when the rfifind function is called. In this analysis a RFI mask was computed and applied to each synthetic filterbank file at integration intervals of 8 s. An integration time of 8 s was chosen to resemble typical real-time processing intervals. For example, a pulse with DM=500 will have a $\sim$ 1 s dispersion delay across a 300 MHz band at 1.4 GHz. Therefore, integration time are typically in the order of a few seconds. The default values for the time and frequency rejection thresholds in the rfifind function were selected. A moving average filter of 10 s was also implemented as a processing step in the PRESTO pipelines. Lastly, details of the red noise mitigation technique in PRESTO can be found in Section 6.2.3.

To process all the files with PRESTO the following functions and their associated flags were called:

(a) function prepdata with the flags -dm, -o and with/without the flag -mask

(b) function realfft,

(c) function zapbirds with the flags -zap and -zapfile,

Table 6.5: The eight PRESTO pipeline configurations used to process all the files in this analysis.

Pipeline	RFI mask	Moving average filter	Red-noise mitigation
A	X	X	X
B	X	X	✓
C	X	✓	X
D	X	✓	✓
E	✓	X	X
F	✓	X	✓
G	✓	✓	X
H	✓	✓	✓

(d) function `accelsearch` with the flags `-sigma 1.0`, `-flo 0.1`, `-zmax 0` (acceleration searching was turned off by setting the flag `-zmax 0`) and `-numharm 16` (i.e. the number of summed harmonics is 16).

The `accelsearch` function in PRESTO produces an ACCEL file which was searched for possible candidates based on the Gaussian significance of the peaks under the assumption of pure white noise.

6.4 Conclusion

In this chapter, I give a comprehensive description of the algorithms available in the pulsar search software packages SIGPROC and PRESTO for spectrum whitening. The simulation parameters that I chose for generating synthetic filterbank files with RFI and pulsars along with a justification for their selection are also given. The experimental setup required for addressing the objectives set forth in the introduction of this chapter are described in detail. Lastly, this chapter contains a summary of the functions inherent to SIGPROC and PRESTO that were assessed with the framework. The results and implications of this analysis are presented in Chapter 8.

Chapter 7

Radio Frequency Interference Mitigation

7.1 Introduction

The work presented in this chapter is being prepared for publication in *Monthly Notices of the Royal Astronomical Society*.

Surveys for pulsars account for a significant portion of observing time at radio observatories. A limiting factor in these surveys is the effect of anthropogenic RFI (Lazarus et al. 2015, van Heerden et al. 2016) which affects the usable observation time and bandwidth. During pulsar search surveys, RFI signals yield large numbers of false positives, overwhelming the number of detections requiring significant further processing (Eatough et al. 2009, Lazarus et al. 2015, van Heerden et al. 2016). Even in RFI-quiet zones around many radio observatories, there is a significant number of RFI sources which, due to their power and the sensitivity of radio telescopes, are detected by the radio receivers.

As most radio sources are terrestrial and located near dense populations, the number, intensity, and type of RFI sources varies as a function of observatory location, beam size and pointing, and time of day. A RFI source detected by a receiver can vary significantly in apparent amplitude (primarily due to the beam pointing direction relative to the source). The apparent amplitude can range from being sufficiently strong and saturating the analogue or digital front-end of the telescope, to being barely detectable after sufficient integration in

time or frequency has raised the signal above the system noise. These factors have lead to RFI mitigation methods being developed for specific science cases, instruments, and telescopes.

In order to mitigate the effects of RFI sources in the recorded signal, there are two standard approaches: flagging and excision. Flagging leaves the original signal unaltered but records the location (in time and frequency) of likely RFI. This method is typically used when data are iteratively reprocessed, such as in interferometric self-calibration cycles. This allows for the 'RFI mask' to be updated when additional calibration reveals lower-level RFI. The cost of flagging is the additional memory to store the mask, and the computational overhead in how the mask is applied to the data. In excision, the RFI signal is replaced with some nominal signal, representative of uncontaminated data in order to preserve the overall signal statistics. Excision is typically used in real-time, automated pipelines in which the signal is only processed once. These methods are implemented to be computationally and memory efficient but at the cost of being an irreversible operation. This is the approach I take for developing the RFI mitigation method presented here.

A challenge to develop a general RFI mitigation method is to account for the time and frequency variations (see Figure 3.5) in any analogue receiver front-end (Fridman 2000). Typically the analogue electronics are stabilised and well-modelled, but there will always be some variation. Furthermore, the overall gain can vary as a function of the RFI environment. When there is significant RFI present, the power input into the receiver can drive the system to a non-linear response region of the electronics causing broadband variation or saturation even if the RFI source is narrowband. Bright, narrow-band RFI sources will leak into adjacent spectral channels due to the dynamic range limitations of the receiver front-end electronics. Additionally, system temperature varies as a function of the sky temperature, i.e. where the telescope is pointed at on the sky also influences its overall temperature. This gain variation is also known as non-stationary noise baseline variation (see Figure 3.5). This variation has been shown to reduce pulsar surveys' sensitivity to

long-period pulsars (Lazarus et al. 2015, van Heerden et al. 2016).

As I listed in Chapter 4, various methods have been developed to excise RFI from radio observations. These methods are limited for they require fine tuning of their parameters for specific telescopes and observation types, and they perform off-line processing of recorded data that are integrated in time. However, modern radio observations require robust, automatic, real-time procedures for processing data on a streaming basis in order to maximise the sensitivity and scientific output of these instruments.

I have developed a method to address what I see as a missing tool for real-time RFI mitigation for time-domain search and timing observations. Namely, I have produced a computationally-efficient, generic method which operates on streaming channelised data that is adaptable to time-variant RFI environments, while maintaining the sensitivity to detect dispersed pulses and long-period pulsars. The synthetic data created for assessing pulsar search pipelines in Chapter 6 are used to benchmark the efficacy of the RFI excision algorithm presented here. These synthetic data along with the metrics described in Chapter 8 for benchmarking the efficacy of any RFI excision algorithm should be used as standard practice when comparing the performance of existing and new RFI excision algorithms.

A fundamental requirement of any generic RFI mitigation method is adaptability to evolving time and frequency structure due to instrumentational response variations and the broad-range of RFI signals. Further, as RFI mitigation is applied in real-time, the adaptability of the method can only rely on previous-in-time data as a leading indicator to predict the system variation. My method uses a leading, boxcar-window average, the time-scale of which is adaptively determined to learn the instrumentational variation. This allows an adaptive thresholding filter to accurately excise RFI under the assumption that the instrumentational variation is smooth in time and frequency, and that variation is on a longer time scale than the resolution of the data.

In Offringa et al. (2010) the authors list several considerations when designing a new

RFI mitigation strategy. I have appended the original list with two additional considerations. I will address these considerations in the context of the method in Section 9.2:

(i) The true-/false-positive ratio of the RFI classification, i.e Type I vs. Type II error rates.

(ii) The speed of the algorithm, i.e. can the algorithm run in real-time?

(iii) Detection or recovery, i.e. whether detection or flagging of contaminated areas is sufficient. In certain situations, it might be necessary to recover a partially contaminated signal, i.e. to subtract the RFI from the data.

(iv) The effect of the RFI mitigation on the noise, i.e. does the mitigation strategy enhance pulsar detection sensitivity whilst causing minimal damage to astronomical signal or does it introduce new spurious trends.

(v) The ability of the algorithm to process streaming data, i.e. streaming data versus integrated samples. The study by Winkel et al. (2007) not only found that RFI signal variations occur on the order of less than 100 ms but that integration times can severely affect the successfulness of an RFI detection strategy.

(vi) The ability of the algorithm to correct for a time-variant gain and bandpass model.

The algorithms underpinning the RFI excision method are described in the remainder of this Chapter. The application of the method to synthetic and real data is presented in Section 8.4.

7.2 Methodology

The RFI excision and data normalisation algorithm developed in this chapter builds on the work done by Karastergiou et al. (2015). RFI cleaning and data normalisation must happen prior to dedispersion because the dispersion transform integrates total power over

Table 7.1: Nomenclatures used for explaining the functioning of the RFI excision algorithm

Term	Description
Filterbank file	A file containing power samples as a function of frequency and time from a radio observation. See Figure 3.5 for an example.
`_filterLength`	Length of the boxcar-window used to compute the moving average of the noise baseline. Can either be specified by the user or automatically determined from the data.
`_crFactor`	User specified scaling parameter used for computing `_thresholdC`.
`_thresholdC`	Threshold used for identifying bright samples.
`_srFactor`	User specified scaling parameter used for computing `_thresholdS`.
`_thresholdS`	Threshold used for identifying bright spectra.
`_irFactor`	User specified scaling parameter used for computing `_thresholdI`.
`_thresholdI`	Threshold used for identifying bright channels.

frequency, thereby removing all frequency resolved information. Thus, the algorithm presented here operates on channelised data and comprises six modules, each of which is described subsequently. A description of the nomenclatures used in this section is summarised in Table 7.1.

7.2.1 Algorithm overview

The RFI mitigation algorithm takes as input a filterbank file and outputs four files which contain the time-frequency pairs of the four types of RFI identified as well as an output filterbank file normalised and excised of RFI. The pseudo code for the RFI mitigation algorithm is given in Algorithm 5.

The algorithm is summarised in the following steps:

(1) The data are input via a streaming interface or read in blocks from a file. Real-time processing entails receiving the data in a block-wise fashion, where the time duration of each block depends on the maximum DM searched and the observing frequency,

and is likely to be of the order of a few seconds. The number of blocks and the time duration for each can be specified by the user with the flag (`-dataChunksInSec`).

(2) The pseudo code given in Algorithm 6 (see Section 7.2.2) is used to determine the ideal window length of the moving average filter to be used by Algorithms 7−9. Alternatively, the window length can be specified by the user with the flag `-filterLength`.

(3) The bandpass is learned using the pseudo code given in Algorithm 7 (see Section 7.2.3). Thereafter, the bandpass is adjusted to coincide with the moving average of the data.

(4) The power in each frequency channel is compared to the learned bandpass at that frequency. If it exceeds a threshold `_thresholdC` it is replaced with the last known good value for that frequency channel (see pseudo code given in Algorithm 8). The threshold is a function of the root mean square (RMS) moving average and a scaling value `_crFactor`.

(5) Moving averages of both the mean and RMS are computed which in turn are used to update the thresholding values and to adjust the offset of the bandpass and the Last Good Spectrum (LGS).

(6) If the average value of the incoming spectrum deviates from the mean running average, due to broadband RFI, then the whole spectrum is replaced by the known LGS (see pseudo code given in Algorithm 9). The threshold `_thresholdS` that governs the clipping is a function of the RMS moving average divided by the square root of the number of channels and a scaling value `_srFactor`.

(7) The first spectrum for which neither the whole spectrum nor any individual samples are compromised is saved as the `_lastGoodSpectrum`.

(8) If the number of consecutive spectra replaced equals the size of the `_filterLength`, the algorithm deems the learned bandpass no longer relevant to the data and it goes

into learning mode again, i.e. the bandpass, RMS and moving average buffers are reset.

(9) The bandpass is subtracted from each spectrum, whereafter it is scaled to maintain a RMS equal to one in the post-filtered data (see pseudo code given in Algorithm 10).

(10) Post normalisation, low-level RFI is identified but not replaced, by integrating each individual channel in time and checking if it surpasses the threshold `_thresholdI` (see pseudo code given in Algorithm 11).

(11) Finally, the cleaned data are rescaled to the original number of bits and then written to a new filterbank file and the time-frequency pairs of identified RFI is saved to file.

Algorithm 5 RFI Clipper

```
1: N_TB = # fixed time blocks in observation
2: N_SB = # samples per channel in fixed time block
3: for i = 1,...,N_TB do
4:     Read data from file
5:     if i = 1 then
6:         Algorithm 5: Determine filter length
7:     end if
8:     for j = 1,...,N_SB do
9:         Algorithm 6: Learn bandpass
10:        Algorithm 7: Identify and clip bright channels
11:        Algorithm 8: Identify and clip bright spectra
12:        Algorithm 9: Normalise data
13:    end for
14:    Algorithm 10: Channel integrator for weak RFI
15:    Write cleaned and normalised data to file
16:    Save RFI1, RFI2, RFI3, RFI4 to separate files
17: end for
```

7.2.2 Determining the ideal filter window length

A digital filter is a mathematical algorithm that operates on a digital input signal to produce a digital output signal for the purpose of achieving a filtering objective. The digital filtering

operation for a finite impulse response filter (FIR) is defined as:

$$y(k) = \sum_{n=0}^{L-1} h(n)x(k-n) \tag{7.1}$$

where $h(n)$, $n = 0, 1, \ldots, L-1$, are the coefficients of the filter, and $x(k)$ and $y(k)$, respectively, the input and the output of the filter (Ifeachor & Jervis 2002). For a given filter, both the number and values of the coefficients are unique to it and determine the filter's characteristics.

A common filtering objective is to remove or reduce noise from a wanted signal. In pulsar searching the objective is to remove the non-stationary noise present in each channel. However, the correlation length of the non-stationary baseline is unknown and therefore the ideal filter length for removing the non-stationarity is also unknown. A rough estimate of the ideal filter length can be obtained by computing the auto-correlation function (ACF) (see Equation 5.6) of the data (Beran et al. 2013).

It can be shown that if $x_1, \ldots, x_N$ in Equation 5.6 are independent and identically distributed random variables, i.e. white Gaussian noise, that the expected value of $r_{xx}(j)$ is:

$$\mathrm{E}(r_{xx}(j)) = 0 \tag{7.2}$$

and the variance of $r_{xx}(j)$ is

$$\mathrm{Var}(r_{xx}(j)) \simeq {}^{1}\!/_{N}. \tag{7.3}$$

The 95 % confidence limits for the ACF lie at $\pm {}^{2}\!/\sqrt{N}$ (Chatfield 2016).

The lag for which the ACF of a time series first drops below the upper 95 % significance level (see Figure 7.1) is a good estimate for the most appropriate window length of a filter capable of removing correlated noise from data (Chatfield 2016, Beran et al. 2013).

The power spectral density of a function $x(k)$ and the auto-correlation of $x(k)$ form a Fourier transform pair via the Wiener-Khinchin theorem (Ifeachor & Jervis 2002). Thus,

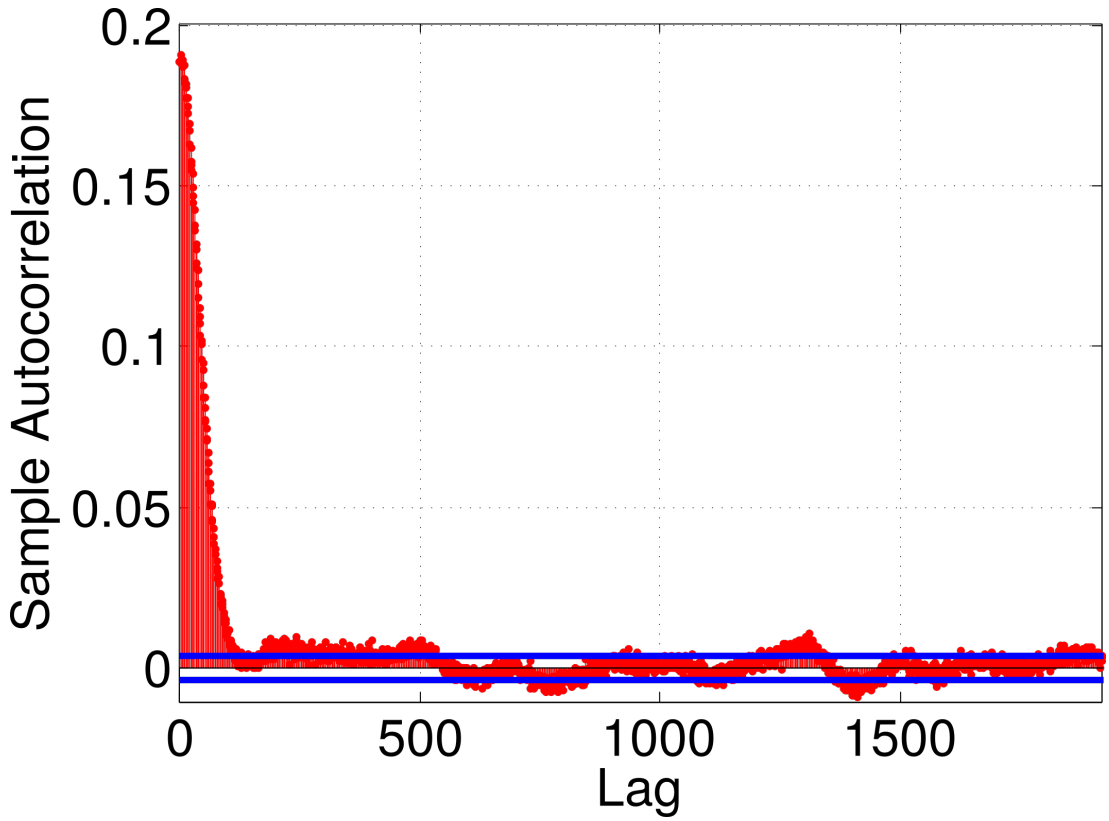


Figure 7.1: Autocorrelation function of a non-stationary time series with the upper and lower 95 % significance levels depicted in blue. A good estimate for the most appropriate window length of a filter capable of removing correlated noise from this data, is the lag at which the ACF of a time series first drops below the upper 95 % significance level.

the auto-correlation can be computed relatively fast by taking the inverse Fourier transform of the power spectral density.

The steps for determining the ideal filter length are summarised in Algorithm 6. The first data block is read, integrated across frequency and then normalised to have zero mean. Thereafter, the ACF of the time series is computed by means of the Wiener-Khinchin theorem. The filter length is set equal to the lag at which the first ACF coefficient drops below the upper 95 % significance level.

Algorithm 6 Ideal filter length

1: N_{SB} = # samples per channel in fixed time block
2: Read **data** from file
3: Integrate across frequency $\xrightarrow{\text{yield}}$ single time series
4: Normalise time series to zero mean
5: Compute: ACF by means of the FFT and IFFT
6: Compute: 95 % confidence limits
7: **for** $j = 1,...,N_{SB} - 1$ **do**
8: **if** $r_j < 95$ % upper confidence level **then**
9: `_filterLength` = j
10: break
11: **end if**
12: **end for**

7.2.3 Bandpass learning (Algorithm 6)

The bandpass of radio receivers cannot be assumed to be stable for reasons listed in Section 7.1. Therefore, considering that the RFI identification part of this algorithm is based on the amplitude of incoming data relative to the bandpass (see Section 7.2.4−7.2.5) it is not only necessary to learn the bandpass (i.e. integrating successive spectra until the model of the bandpass has converged to a stable state) but also to place it at the level of the moving average of the data (`_meanRunAve`). Additionally, using the learned bandpass to normalise the data in frequency and over time enhances detection sensitivity (see Section 8.4).

The bandpass is learned over multiple phases and learning stops once it converges otherwise it continues until the end of the observation. A schematic of the learning process is depicted in Figure 7.2 where each phase, $\text{LP}_0, \ldots, \text{LP}_M$, has a duration equivalent to the `_filterLength` (see Section 7.2.2). During the initial learning phase, LP_0, the bandpass is learned as a sequence of n flat spectral segments, based on the average total power per segment, also averaged over the time samples in LP_0. Learning the bandpass in n flat segments initially safeguards the learning process from being corrupted by spurious bright RFI. From LP_1 onwards, the bandpass is learned per channel. Note that the bandpass is only learned from spectra that do not violate the spectrum threshold (see Section 7.2.5).

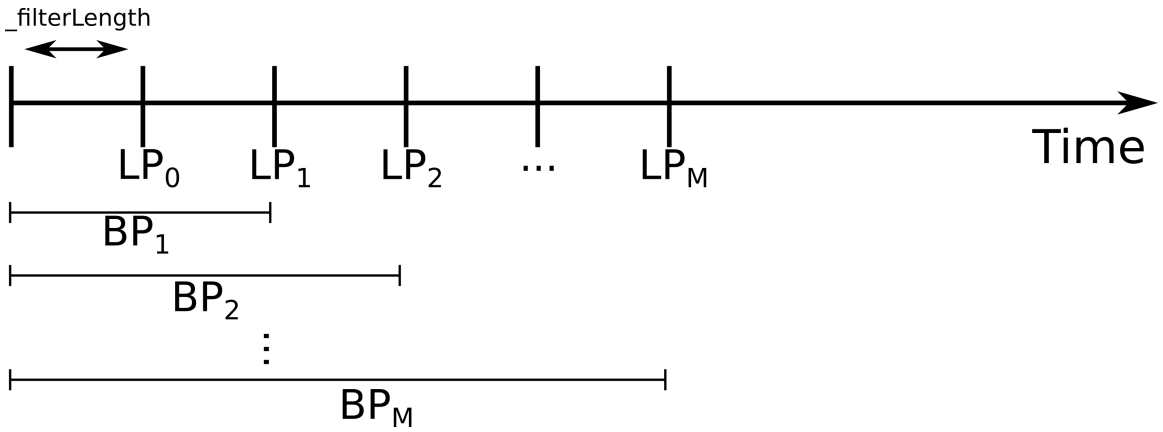


Figure 7.2: Schematic to show the various phases across which the bandpass is learned. Each learning phase, $\mathrm{LP}_0, \ldots, \mathrm{LP}_M$, has a duration equivalent to the filter length. Convergence of the bandpass is determined after each learning phase, starting at the end of LP_2, where bandpass BP_i is compared to bandpass BP_{i-1}. The bandpass is considered to have converged and learning stops when the normalised difference between BP_i and BP_{i-1} falls below a threshold (`_learningThreshold`).

Convergence is determined after each learning phase, starting at the end of LP_2, where bandpass BP_i is compared to bandpass BP_{i-1}. The bandpass is considered to have converged and learning stops when the normalised difference between BP_i and BP_{i-1} falls below a threshold (`_learningThreshold`). The normalised difference between BP_i and BP_{i-1} is computed as:

$$Q(\mathrm{BP}_i, \mathrm{BP}_{i-1}) \equiv \frac{\| \mathrm{BP}_i(\cdot) - \mathrm{BP}_{i-1}(\cdot) \|_2}{\| \mathrm{BP}_i(\cdot) \|_2 + \| \mathrm{BP}_{i-1}(\cdot) \|_2}, \tag{7.4}$$

where $\| \cdot \|_2$ is the Euclidean norm and $Q(\mathrm{BP}_i, \mathrm{BP}_{i-1})$ varies between zero (for perfect agreement) and one (for disagreement) (Perlin & Bustamante 2016).

At each time step during the learning process and after convergence the bandpass is placed at the moving average of the data (`_meanRunAve`) which is computed over the `_filterLength`. A summary of the learning and update procedures for the bandpass is given by Algorithm 7.

Algorithm 7 Learning the bandpass

```
1: if learning==true then
2:     Learn: BP_i
3:     Update: BP_i offset
4:     if LP_i==_filterLength then
5:         Compute: Q(BP_i, BP_{i-1})
6:         if Q(BP_i, BP_{i-1})<_learningThreshold then
7:             learning=false
8:             _dataBandPass = BP_i
9:         else
10:            BP_{i-1} = BP_i
11:        end if
12:    end if
13: else
14:    Update: _dataBandPass offset
15: end if
```

7.2.4 Channel thresholding (Algorithm 7)

Since RFI increases the measured absolute amplitude of a signal (Offringa et al. 2010), thresholding is an effective method and one that is most commonly used to identify RFI in the time-frequency plane (Bhat et al. 2005, Ransom et al. 2004). In this section I describe the channel-wise thresholding approach that is used to identify localised bright RFI in streaming data.

The threshold `_thresholdC` used to identify RFI is set proportional to the running average of the RMS of the spectrum (`_rmsRunAve`) where the constant of proportionality, `_crFactor`, has to be specified a priori. For each incoming spectrum the channel-wise difference between the total intensity and the learned `_dataBandPass` is computed and if the difference exceeds the `_thresholdC` that particular channel's value is replaced (see Algorithm 8).

The value used to replace the affected data sample comes from the `_lastGoodSpectrum` that was saved plus a sample drawn from a random normal distribution, $\mathcal{N}(0,1)$, scaled by the moving average of the spectrum RMS (`_rmsRunAve`). Replacing the affected samples with random samples scaled by the spectrum RMS will cause minimal distortion to

Algorithm 8 Channel thresholding

1: N_{SB} = # samples per channel in fixed time block
2: **for** j = 1,...,N_{SB} **do**
3: Compute: `_rmsRunAve`
4: Compute: `_thresholdC = _crFactor*_rmsRunAve`
5: Update: `_lastGoodSpectrum` offset
6: **for** c = 1,...,`nchans` **do**
7: **if** `data[j,c] - _dataBandPass[c]>_thresholdC` **then**
8: `data[j,c] = _lastGoodSpectrum[c]` + $\mathcal{N}(0,1)$ `*_rmsRunAve`
9: Save: position of affected sample in `RFI1`
10: **else**
11: Compute: `spectrumSum`
12: Update: `++goodChannels`
13: **end if**
14: **end for**
15: `spectrumSum=spectrumSum/goodChannels`
16: **end for**

the overall statistics of the data, i.e. the dedispersed time series will exhibit neither jumps nor flat segments. If the flag `-staticReplace` is passed to the algorithm the affected samples are replaced with only the `_lastGoodSpectrum`.

Similarly to the `_dataBandPass`, the `_lastGoodSpectrum` is placed at the level of a shorter moving average of the data (`_meanRunAveShort`), computed from the last twenty spectra. The rationale for keeping track of two means is that the global moving average (_meanRunAve) is primarily used for updating the `_dataBandPass` offset and for normalising the data, whereas a moving average computed over fewer samples is used to place the `_lastGoodSpectrum` as close as possible to the level of the data.

The time-frequency pairs of the samples which were identified to be affected by RFI are saved in the vector `RFI1`.

Lastly, the intensities from all the channels that do not contain RFI are kept track of and used to compute the mean of the current spectrum, which in turn is used to update both moving averages.

7.2.5 Spectrum thresholding (Algorithm 8)

Thresholding spectra in time is useful for identifying RFI that is broadband and transient in nature. For this, the algorithm integrates over all the channels for every observed spectrum, followed by the computation of its statistics (in time), and applies thresholding (see Algorithm 9).

Algorithm 9 Spectrum Thresholding

1: N_{SB} = # samples per channel in fixed time block
2: **for** j = 1,...,N_{SB} **do**
3: Update: `_meanRunAve`
4: Compute: `_thresholdS` = `_srFactor` $*$ `_rmsRunAve`/ $\sqrt{\texttt{nchans}}$
5: **if** (`spectrumSum` − `_meanRunAve` > `_thresholdS`) **or**
(`goodChannels` > `_fracExpGoodChans` $*$ `nchans`) **then**
6: **for** c = 1,...,`nchans` **do**
8: `data[j,c]` = `_lastGoodSpectrum[c]` + $\mathcal{N}(0,1)$ $*$ `_rmsRunAve`
9: Save: position of affected sample in RFI2
10: **end for**
11: **end if**
12: **end for**

The threshold `_thresholdS` used to identify broadband RFI is set proportional to the running average of the spectrum RMS (`_rmsRunAve`) divided by the square root of the number of channels in the spectrum (`nchans`) where the constant of proportionality, called the `_srFactor`, has to be specified a priori. For each incoming spectrum the difference between the spectrum mean and the moving average `_meanRunAve` is computed and if the difference exceeds the `_thresholdS` then the whole spectrum is replaced (see Algorithm 9).

The values and manner in which the affected samples are replaced is identical to the procedure described in Section 7.2.4.

The time-frequency pairs of the samples which were identified to be affected by RFI are saved in the vector `RFI2`.

7.2.6 Data normalisation (Algorithm 9)

To optimize the detection rate of pulsar signals, all bright RFI should be removed and the underlying noise baseline should ideally be constant in both time and frequency. The latter is achieved by subtracting the learned `_dataBandPass` from the data such that it has zero mean and dividing by the spectrum RMS such that it has a RMS of one (see Algorithm 10).

The algorithm can also be set to remove the mean of every spectrum, effectively removing signals at 0 DM (Eatough et al. 2009) by calling the algorithm with the flag `-ZeroDM`.

Values which, after normalisation, surpass the threshold `_crFactor` are set to zero and their time-frequency pairs are saved in the vector `RFI3`.

Algorithm 10 Data normalisation

```
1:  N_SB = # samples per channel in fixed time block
2:  for j = 1,...,N_SB do
3:      for c = 1,...,nchans do
4:          data[j,c] = data[j,c] - _dataBandPass[c]
5:          Compute: spectrumRMS
6:      end for
7:      Compute: spectrumSum=spectrumSum/nchans
8:      Compute: spectrumRMS
9:      for c = 1,...,nchans do
10:         if _zeroDM==true then
11:             data[j,c] = data[j,c] - spectrumSum
12:         end if
13:         data[j,c] = data[j,c]/spectrumRMS
14:         if |data[j,c]| > _crFactor then
15:             data[j,c] = 0.0
16:             Save: position of affected sample in RFI3
17:         end if
18:         Update: _chanIntegrator[c] = _chanIntegrator[c] + data[c]
19:     end for
20: end for
```

As briefly listed in Section 7.2.3, `_dataBandPass` is adjusted to coincide with the moving average of the data (`_runMeanAve`). The moving average was chosen as it is a computationally cheap low-pass filter choice for estimating the underlying noise baseline; however, it comes at a cost of sidelobe ringing (see Figure 7.3). The impulse response

and transfer function of an $L-$sample moving average filter is given by Equation 7.5 and Equation 7.6 respectively.

$$h(n) = \begin{cases} \frac{1}{L}, & n = 0\,1, 2, \ldots, L-1 \\ 0, & \text{otherwise.} \end{cases} \tag{7.5}$$

$$H_{low}(f) = {}^{1}/_{L} \frac{1 - \exp^{-j2\pi Lf/f_s}}{1 - \exp^{-j2\pi f/f_s}} \tag{7.6}$$

Note that by subtracting the moving average from the data the operation is no longer acting as a low-pass filter but instead as a high-pass filter. The transfer function of subtracting the moving average from the data is then $1 - H_{low}(f)$ as given in Equation 7.7.

$$H_{high}(f) = 1 - {}^{1}/_{L} \frac{1 - \exp^{-j2\pi Lf/f_s}}{1 - \exp^{-j2\pi f/f_s}} \tag{7.7}$$

The magnitude of $H_{high}(f)$ is plotted in Figure 7.3 as a function of frequency for $L = 0.01$ s (blue), 0.1 s (green), 1.0 s (red), and 10.0 s (magenta). It is evident that the transfer function has a high-pass characteristic. The plot in Figure 7.3 will aid with the interpretation of the results presented in Section 8.4.3.2.

Note that the shorter the filter, the more frequencies get completely eliminated, i.e. when searching for slow rotating pulsars a short moving average filter should be avoided as it will attenuate the fundamental frequency and all its harmonics rendering it undetectable. Conversely, the longer the filter the more frequencies pass unattenuated.

7.2.7 Channel integrator (Algorithm 10)

The thresholding Algorithms 8−9 previously described are amplitude-based and therefore likely to be insensitive to persistent low-level RFI, i.e. interference with intensities below the critical level (Offringa et al. 2013). Low-level narrowband RFI can be detected by integrating successive spectra in time to yield an averaged spectrum. The presence of low-

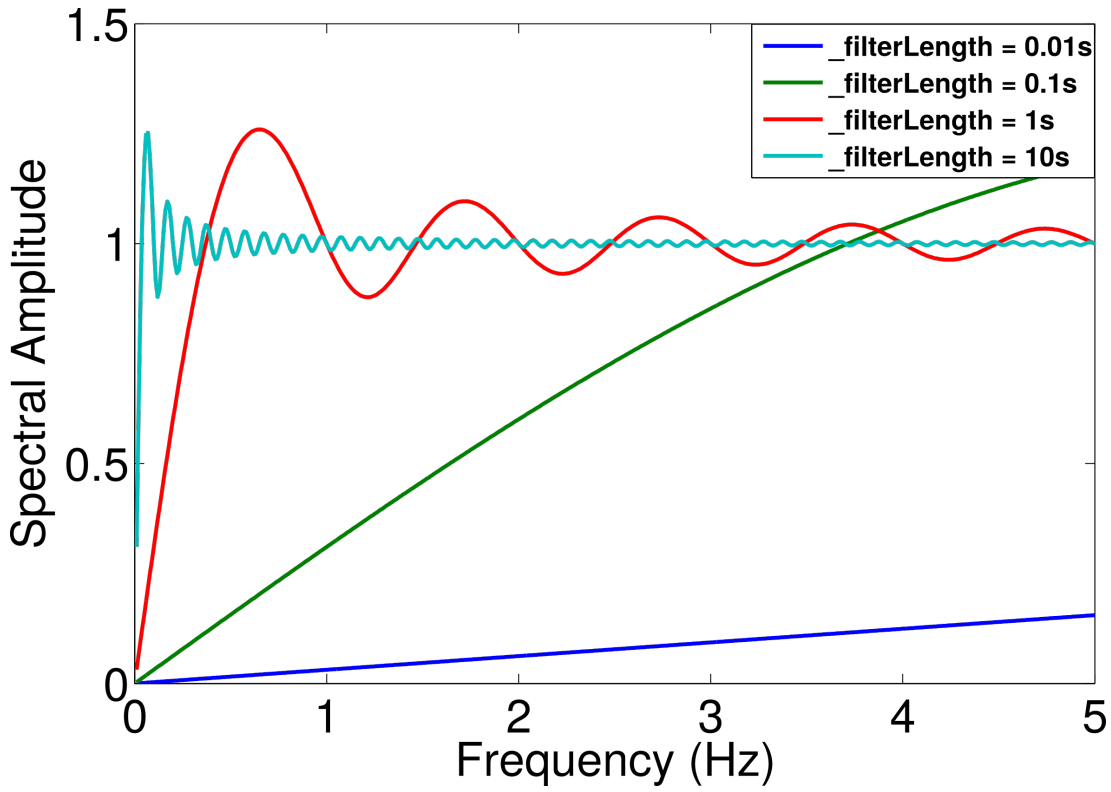


Figure 7.3: The transfer function of the operation of subtracting the moving average from the data.

level narrowband RFI will result in the affected spectral channels to be statistically different from the other integrated channels. This approach is known as the CumSum method and is a well known analysis method used to detect changes in distribution parameters (Offringa et al. 2010).

At this stage in the algorithm the data have been cleaned of bright RFI and normalised such that each channel has mean zero and RMS of one. Thus, if the averaged spectrum in a particular channel (`_chanIntegrator`) exceeds the threshold `_thresholdI` this is most likely due to low-level persistent narrow-band RFI. The threshold `_thresholdI` is proportional to one over the square root of the number of channels, assuming Gaussian statistics, where the constant of proportionality, called the `_irFactor`, has to be specified a priori. The time-frequency pairs of all the samples in the channels that were identified by this threshold are flagged and saved in the vector `RFI4` (see Algorithm 11). Saving the

time-frequency pairs facilitates the creation of an RFI mask.

A drawback of this method is that the start time of the RFI during the segment being integrated is not well known, therefore good samples will be incorrectly flagged but equally RFI affected samples may leak through if the duration is not sufficient to build up statistical significance.

Algorithm 11 Channel integrator

1: N_{TB} = # fixed time blocks in observation
2: N_{SB} = # samples per channel in fixed time block
3: **for** $i = 1, \ldots, N_{TB}$ **do**
4: Compute: `_thresholdI` = `_irFactor`/ $\sqrt{\texttt{nchans}}$
5: **for** c = 1,...,`nchans` **do**
6: Compute:`_chanIntegrator[c]` = `_chanIntegrator[c]`/N_{SB}
7: **if** `_chanIntegrator[c]` > `_thresholdI` **then**
8: Save: position of affected sample in RFI4
9: **end if**
10: **end for**
11: **end for**

7.3 Conclusion

In this chapter, I described in detail a new algorithm for excising RFI while simultaneously normalising both the variability in time and frequency inherent to pulsar observations. The algorithm is able to process streaming data which is ideal considering that the time variability of most interferences demands analysis on short time scales.

The efficacy of the RFI excision method and its ability to fulfil all the considerations listed in Section 7.1 are demonstrated in Chapter 8..

Chapter 8

Results: From Synthetic to Real

8.1 Introduction

In this chapter, I present the findings from the methodology and experiments described in Chapter 6 and Chapter 7. It begins by listing the heuristics used for parsing the results to decide whether or not a pulsar has been detected. In Section 8.3, I present the performance assessment results of existing pulsar search pipelines. In Section 8.4, I present the results of processing the same files from the framework with the RFI mitigation algorithm along with the results from processing pseudo-real data. The assessment and a discussion on the findings presented in this chapter can be found in Chapter 9.

8.2 Heuristics

The heuristics used for quantifying the results are:

(a) a signal is considered a candidate if its detection significance is greater than the default detection threshold in `SIGPROC` (SNR of eight) or `PRESTO` (Gaussian significance greater than two under white noise assumptions).

(b) injected pulsars are considered discovered when (a) holds true AND if the difference between the periods of the discovered and injected signals are less than an *error* value, i.e.: $|\text{Period}_{discovered} - \text{Period}_{injected}| \leq error$, where

error = 0.02 ms if period$\leq$ 10 ms

error = 0.20 ms if 10 ms<period$\leq$ 100 ms

error = 2.00 ms if 100 ms<period$\leq$ 1000 ms

error = 20.0 ms if 1000 ms<period$\leq$ 10000 ms.

The injected and discovered periods of the pulsars never exactly match as there are various processes involved with generating these synthetic data as well as searching them. Hence, different error margins which allow for the injected and discovered periods to differ between $\sim$ 0.2 % to 2 % of the pulse period were chosen.

(c) the discoveries from (b) are validated by visual inspection of their folded profiles produced by folding the inverse Fourier transform of their whitened spectra at the detected periods. At this stage a detection is rejected if the folded profile does not resemble a real pulsar,

(d) in determining whether a pulsar was detected or not the non-fundamental harmonics were not considered,

(e) harmonically related candidates are removed and,

(f) only candidates with 1 ms $\leq$ period $\leq$ 10 s are considered.

For the direct `SIGPROC` and `PRESTO` comparisons the exact same files were searched by both routines.

The sensitivity of a pipeline refers to the ability of the pipeline to detect the randomly injected pulsars expressed as a percentage. The number of false positives detected per true positive is the total number of false positive candidates detected across all files in an experiment divided by the number of true positives detected.

Note that all the results presented here should be interpreted as per dispersion measure, i.e. for a real survey these results should be multiplied by the number of dedispersion trails.

8.3 Performance assessment of pulsar search pipelines

8.3.1 Introduction

The results for the performance assessment of pulsar search pipelines are organised according to the aims set forth in Section 6.1, which for readability are reiterated here:

(i) to quantify the effect that non-stationary Gaussian noise and RFI has on the performance of pulsar search pipelines;

(ii) to examine the effectiveness of the current spectrum whitening methods available in pulsar search software suites;

(iii) to determine if detrending the data with a moving average filter before searching for pulsars is effective;

(iv) to examine the effectiveness of the current RFI detection and mitigation methods available in pulsar search software suites;

(v) to investigate the reduction in sensitivity as a function of both the correlation length of the non-stationary noise and the pulse period.

8.3.2 Non-stationary Gaussian noise and RFI

The results of processing the synthetic files from the emulated blind surveys (see experiments 1-4 in Table 6.2) with the default pulsar search pipelines of `SIGPROC` and `PRESTO` are shown in Figure 8.1. Note that the metric used in this section to express the performance of each pipeline is the number of false positives detected for every true positive detected across all 100 files for each experiment.

The number of false positives per true positive detected by `SIGPROC` increases approximately proportionally with a linear increase in the amplitude of the non-stationary noise

(see Figure 8.1a). This trend is also visible in Figure 8.1b when RFI is injected. Hence, the default SIGPROC pipeline is very sensitive to non-stationary noise.

The number of false positives detected per true positive by PRESTO is unaffected by the type and amplitude of the noise process present in the files, i.e. the number of false positives detected per true positive is almost constant irrespective of the amplitude of the non-stationary noise (see Figure 8.1a). However, the number of false positives detected per true positive by PRESTO is slightly higher when weak RFI is present (see Section 8.3.5) compared to when no RFI is injected.

The sensitivity of SIGPROC and PRESTO can be seen in Figure 8.1a to decrease by at least 20 % and 7 % respectively in the presence of non-stationary noise compared to the stationary noise case. The 20 % and 7 % losses recorded in sensitivity were averaged over all the pulse periods; however, the long period pulsars were much more affected. Note that the amplitudes of the injected pulsars were chosen such that they are detectable at a SNR of ~ 12 in the presence of white noise when processed with pipeline H in SIGPROC (see Table 6.3); however, the addition of any (significant) amount of non-stationary noise rendered the pulsars undetectable. Consequently, there is no correlation visible between sensitivity loss and the non-stationary noise amplitude.

There is no correlation between the loss in sensitivity of SIGPROC and the amplitude of the non-stationary noise when weak RFI is present. Interestingly, the presence of weak RFI leads to an increase in the sensitivity of SIGPROC for the stationary noise case with RFI compared to the stationary noise files without RFI. Similarly, there is an increase in sensitivity for the non-stationary 4 case when RFI is present compared to the no RFI case.

Comparing the sensitivity of PRESTO in Figure 8.1a to the sensitivity in Figure 8.1b it becomes apparent that PRESTO is not sensitive to weak RFI. The highest sensitivity attainable with PRESTO for files containing weak RFI and non-stationary noise is 73 %; furthermore, a direct comparison reveals that PRESTO's sensitivity is on average 11 % better than SIGPROC's sensitivity.

8.3.3 Spectrum whitening methods

The power versus log-frequency plot in Figure 8.2 shows the power spectrum density of two non-stationary Gaussian noise processes with correlation lengths $\lambda = 1$ s (red) and $\lambda = 100$ s (blue) as well as for a stationary Gaussian noise process (black). It is evident from Figure 8.2 that the power spectrum density of a non-stationary process diverges from the desired flat power spectrum density of a stationary white noise process as the correlation length of the non-stationary process shortens.

With Figure 8.2 in mind, four spectrum whitening methods (see Section 6.2) were assessed and the results can be seen in Figure 8.3. The spectrum whitening techniques in both `SIGPROC` and `PRESTO` reduce the number of false positives detected per true positive significantly compared to the case when no spectrum whitening is applied. In the presence of non-stationary noise the sensitivity of `SIGPROC` improved slightly from 53 % to 60 % when the default spectrum whitening method was applied but the other methods had no effect on sensitivity (see Figure 8.3a). The sensitivity of `PRESTO` is unchanged when the spectrum whitening method is applied compared to no spectrum whitening.

8.3.4 De-trending the data before processing

De-trending the baseline in `SIGPROC` with either a 10 s moving average filter or the built-in de-trending method led to an increase in the number of false positives detected per true positive compared to when the baseline was left intact (see Figure 8.4a). However, the 10 s moving average filter did improve the sensitivity by 6 %.

De-trending the baseline in `PRESTO` with a 10 s moving average filter reduced the number of false positives detected per true positive and increased `PRESTO's` sensitivity with 7 % (see Figure 8.4b).

These results hint at the improved sensitivity attainable when the file contains both a baseline with long correlations and a slowly pulsating pulsar, i.e. removing the baseline significantly improves the sensitivity of detecting slow pulsars. This fact is highlighted

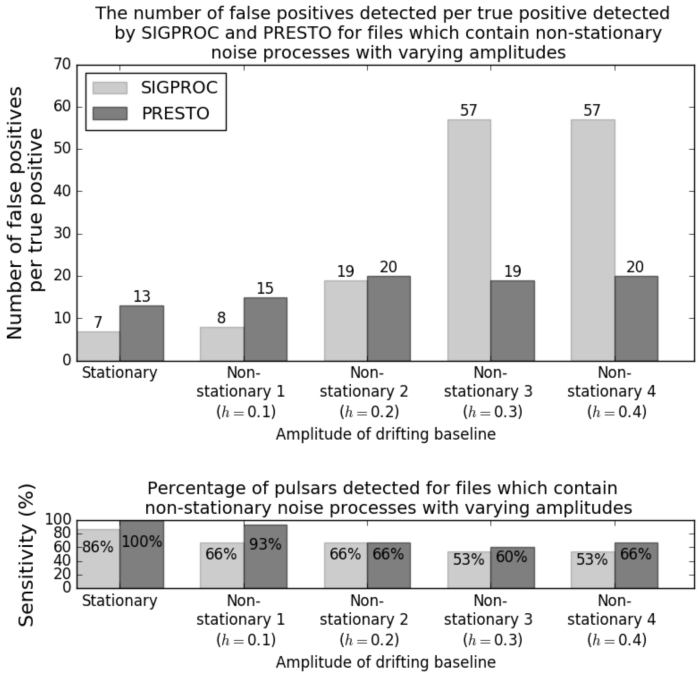
with the postcard plots in sections Section 8.3.7 and Section 8.3.8, described later in the chapter.

8.3.5 RFI detection and mitigation methods

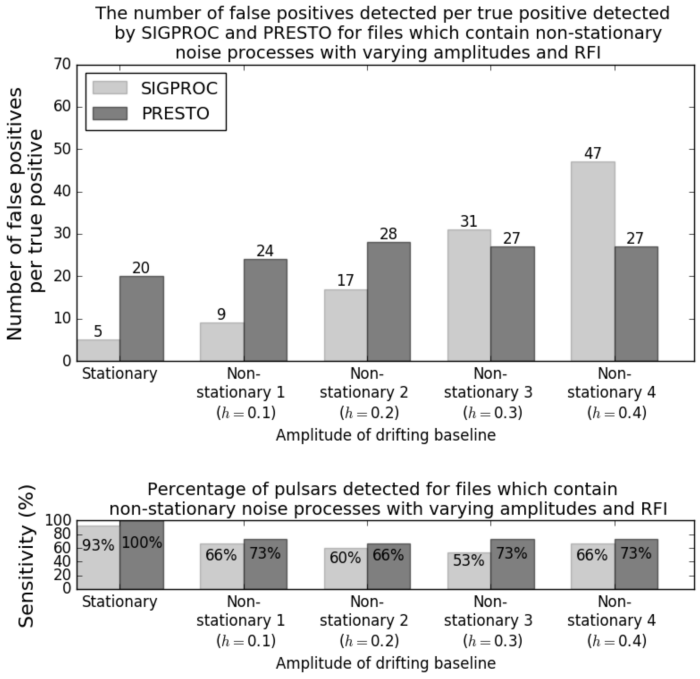
RFI masks created with PRESTO's `rfifind` function, for the same file, are depicted in Figure 8.5. The masks differ with respect to the integration times used to create them. The plots in Figure 8.5 I show that the RFI I injected, although visible, is weak compared to the amplitude of the non-stationary baseline.

The default integration length of 30 s is most successful at detecting the actual injected RFI (see Figure 8.5c). The two masks created with shorter integration times mostly flagged the maxima of the non-stationary baseline as opposed to the actual injected RFI (see Figure 8.5a and Figure 8.5b).

RFI was injected such that 12 % of all the samples in the data are affected. The 2 s mask in Figure 8.5a found 6.9 % of the 2 s intervals to be affected by RFI, the 8 s mask in Figure 8.5b found 6.9 % of the 8 s intervals to be affected by RFI and the 30 s mask found that 16.9 % of the 30 s intervals are affected by RFI.



(a) No RFI



(b) RFI

Figure 8.1: The performance of SIGPROC and PRESTO for processing files which contain either stationary noise or non-stationary noise with varying amplitudes (i.e. different values for h): (a) without RFI (see experiments 1 and 2 in Table 6.2); (b) with RFI (see experiments 3 and 4 in Table 6.2). (SIGPROC pipeline: default baseline subtraction → default red-noise removal. PRESTO pipeline: RFI mask → baseline not subtracted → default red-noise removal.)

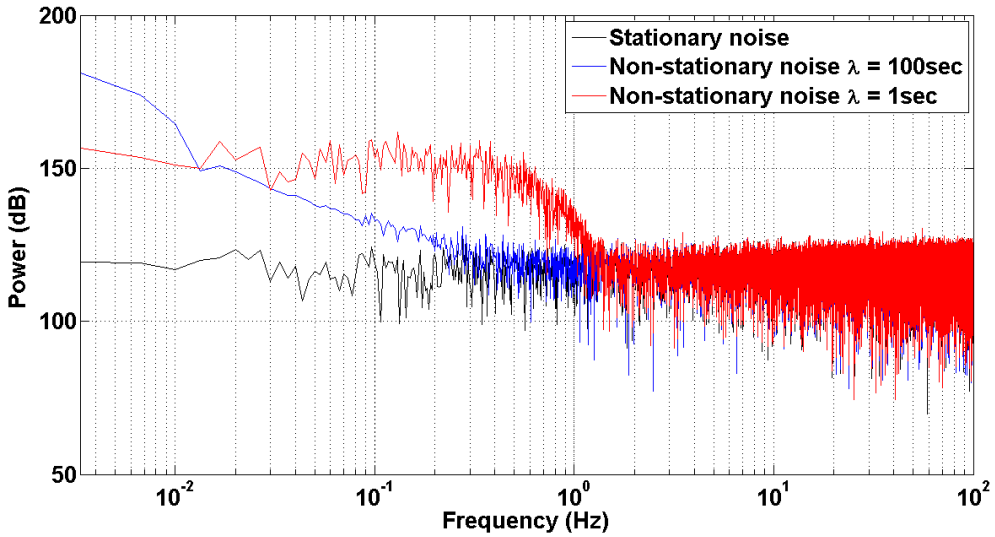
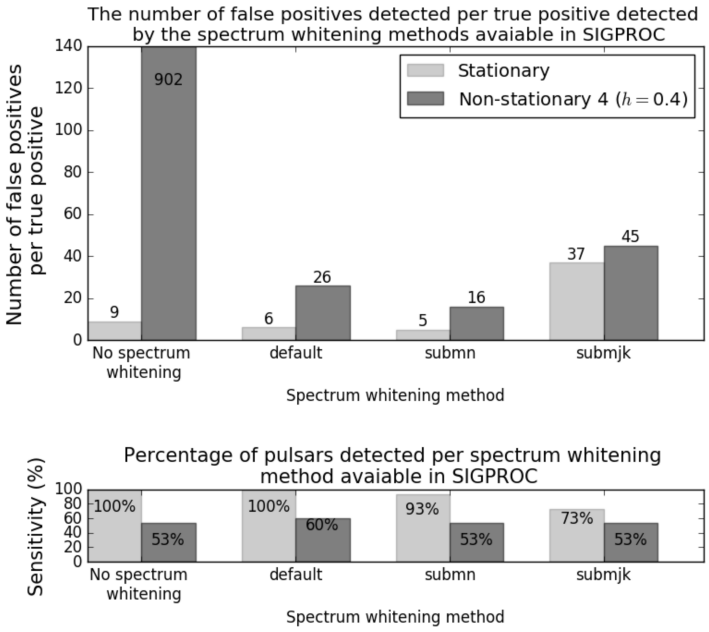
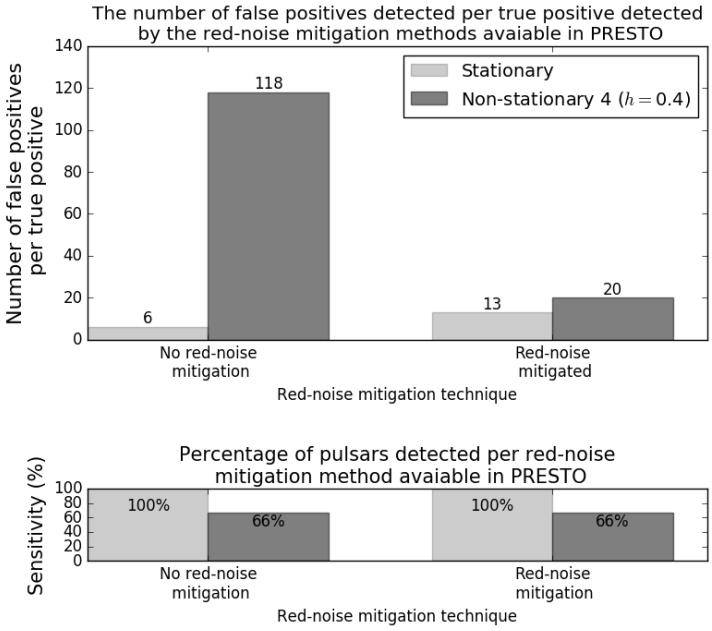


Figure 8.2: In this power spectrum density plot, the frequency dependence of different stochastic processes is depicted. The power spectrum density of a stationary noise process is depicted in black and it is clear from the flat response that this process has no systematic frequency dependence. Conversely, two non-stationary noise processes are depicted in red and blue which clearly exhibit frequency dependence in their power spectra. The value of the parameter λ of the low-pass filter changes the frequency dependence of a non-stationary process, which can be seen in this figure.

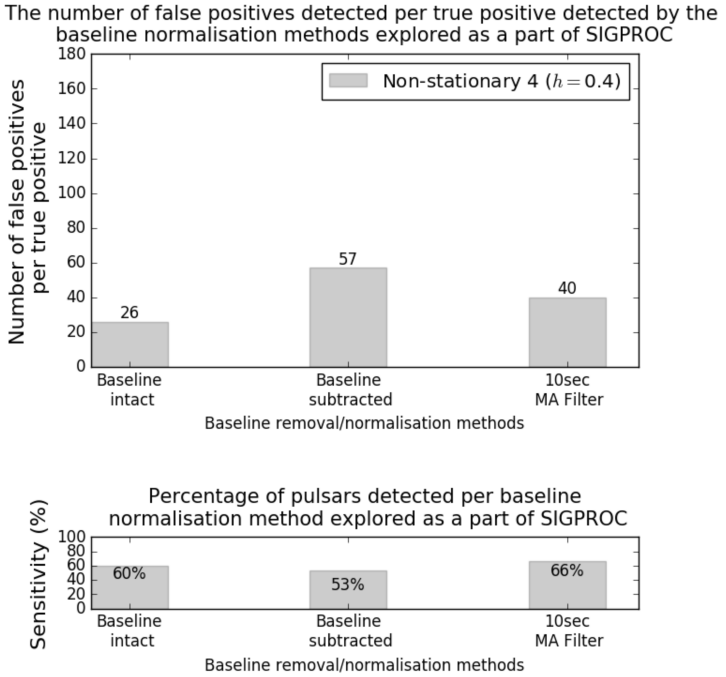


(a) SIGPROC

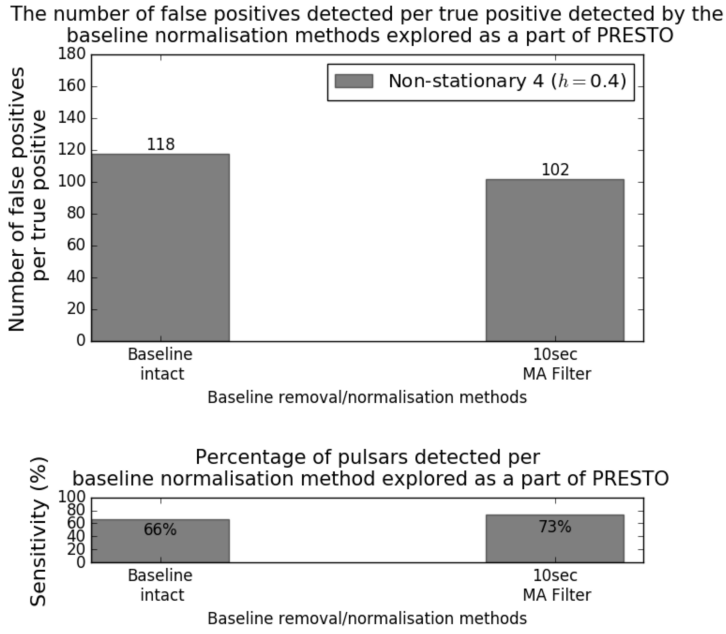


(b) PRESTO

Figure 8.3: The performance of the red-noise mitigation methods available in (a) SIGPROC and (b) PRESTO for processing files which contain either stationary noise (see experiment 1 in Table 6.2) or non-stationary noise (see experiment 2 in Table 6.2). No RFI were present in the files analysed. (SIGPROC pipeline: baseline intact → red-noise removal methods. PRESTO pipeline: RFI mask → baseline not subtracted → red-noise removal method.)

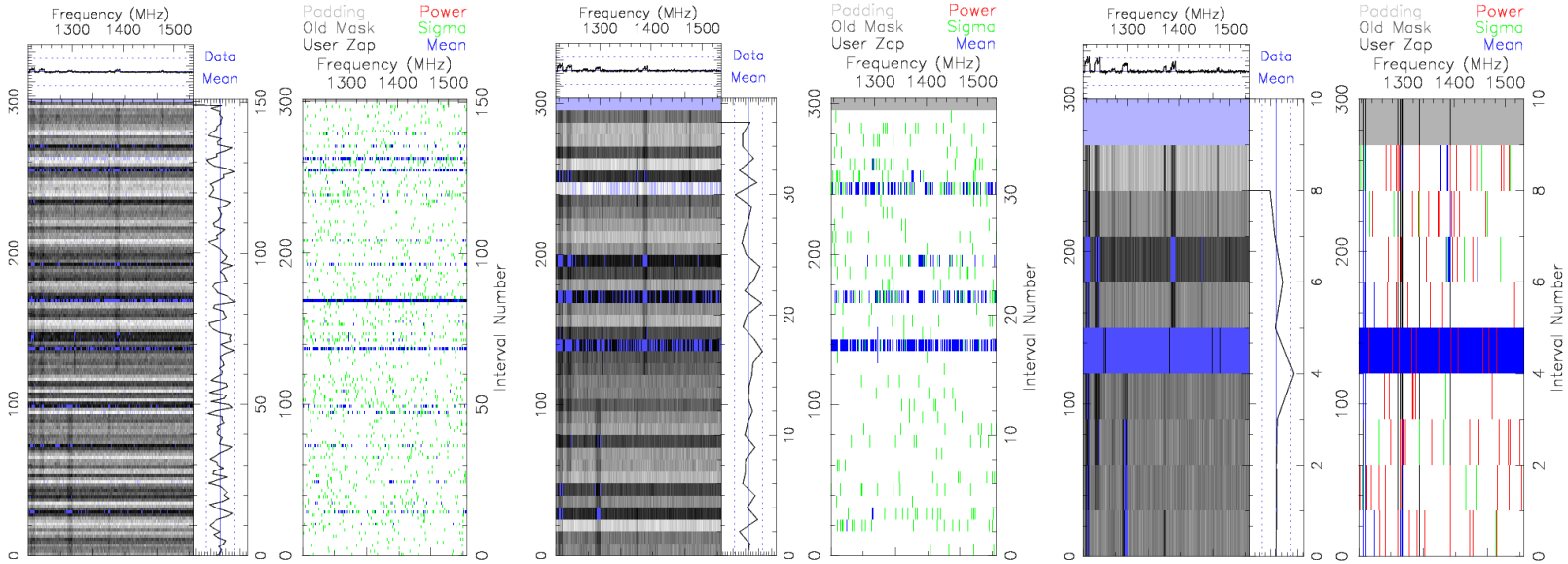


(a) SIGPROC



(b) PRESTO

Figure 8.4: The performance of the time-domain baseline normalisation methods available in (a) SIGPROC and (b) PRESTO for processing files which contain non-stationary noise (see experiment 2 in Table 6.2). No RFI was present in the files analysed. (SIGPROC pipeline: three time-domain baseline normalisation methods → default red-noise removal. PRESTO pipeline: No RFI mask → baseline normalisation method → no red-noise removal.)



(a) 2 s integration (b) 8 s integration (c) 30 s integration

Figure 8.5: RFI masks created with PRESTO's `rfifind` function. The plots are for the same file but created with different integration times. The `-timesig` threshold was set to three and the `-freqsig` threshold was set to eight for the `rfifind` function.

The focus of this section is on the real-time detection and mitigation of RFI, hence the decision to investigate the effectiveness of a RFI mask integrated over a few seconds when applied to the synthetic filterbank files. The results of which can be seen in Figure 8.6.

The RFI detection and masking routine available in PRESTO have on average little to no effect on both the sensitivity and the number of false positives detected per true positive except for the 'non-stationary 3 with RFI' case where the sensitivity was increased from 53 % to 73 % when the mask was applied. Greater insight on the effect of applying the RFI detection and mitigation routine in PRESTO to files containing weak RFI can be gained by comparing the results of pipelines A-D to those of pipelines E-H in Figure 8.8.

8.3.6 Variation of detection with signal significance

Pulsars, embedded in different noise processes, were searched for by the default search pipelines of SIGPROC and PRESTO. The detection significance for each detected pulsar is shown in Figure 8.7; where the colours indicate:

(a) Green square: an injected pulsar was detected (the detection significance is printed in the square),

(b) Orange square: an injected pulsar is amongst the detected signals but is not considered a candidate because its detection significance is not above the default threshold value,

(c) Red square: signifies that an injected pulsar was missed,

(d) Grey square: signifies that an injected pulsar was detected but is not considered a candidate because it has an abnormally high detection significance,

(e) White square: is the detection significance of pulsars embedded in stationary Gaussian noise and the pipelines used to search for them applied no spectrum whitening.

Each coloured box in Figure 8.7 shows a single instantiation of a pulsar/non-stationary baseline pair that was searched by both SIGPROC and PRESTO, whereas each white box

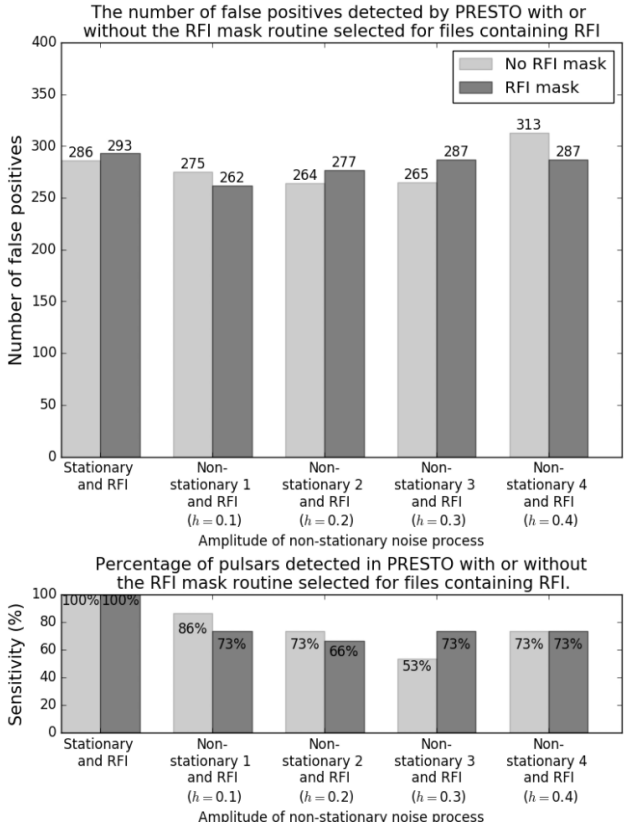
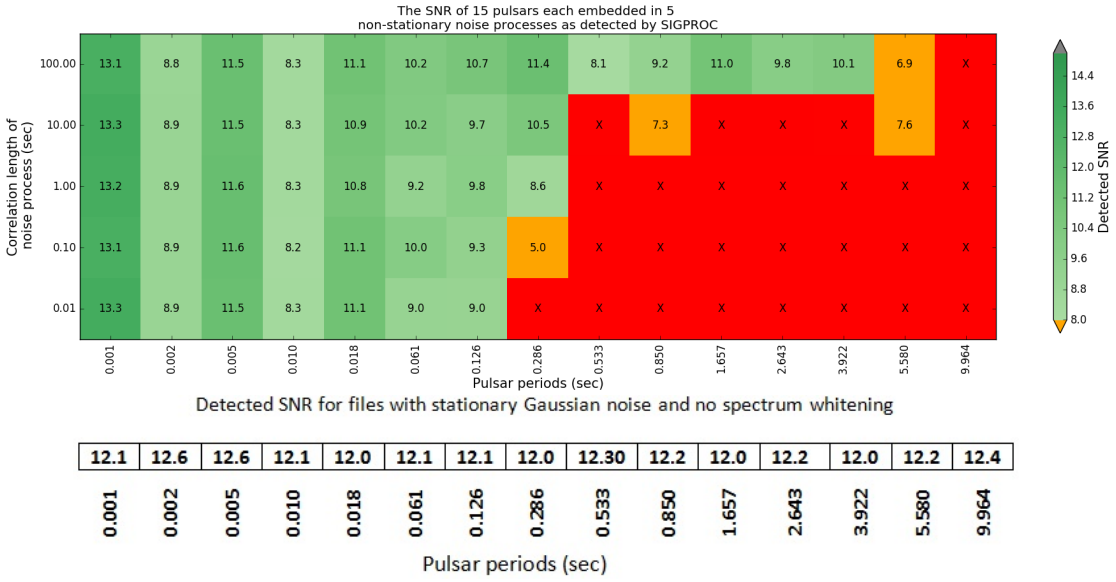


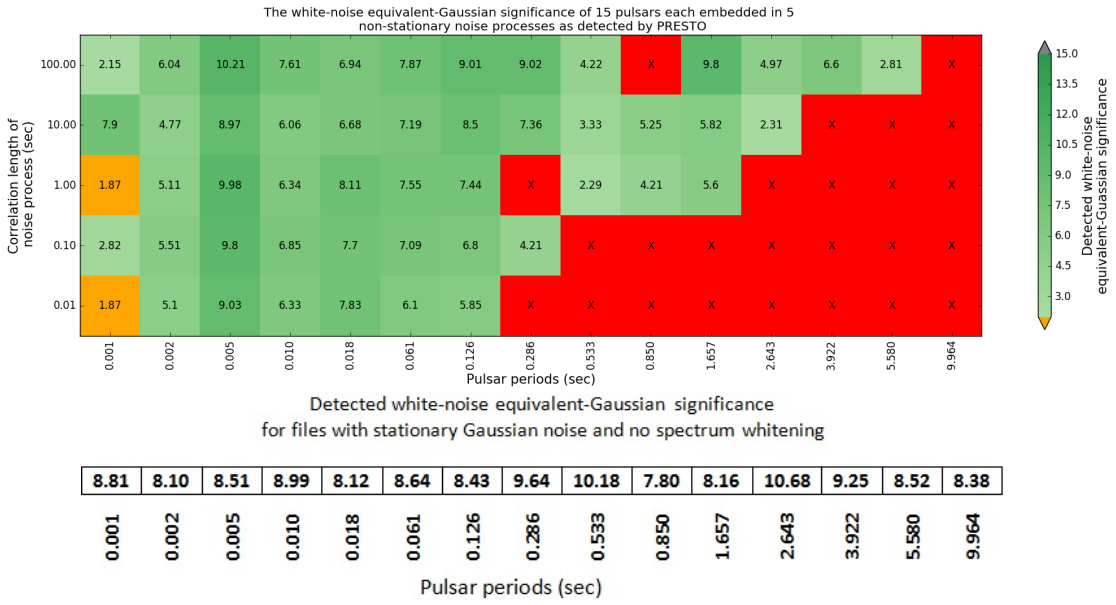
Figure 8.6: The efficacy of the RFI detection and masking routine in PRESTO when processing files which contain either non-stationary noise (see experiment 2 in Table 6.2) or non-stationary noise with weak RFI (see experiment 4 in Table 6.2). (PRESTO pipeline: RFI Mask Yes/No → baseline intact → default red-noise removal.)

shows a pulsar/stationary baseline pair that was searched with the pipelines in both SIGPROC and PRESTO for which no spectrum whitening was applied. These values can be used to assess the change in SNR when non-stationary noise is present in the data and spectrum whitening is applied in the search process.

The left-hand side of both colour matrices in Figure 8.7 is populated with detections, whereas the right hand sides are predominantly populated with misses. Hence, long-period pulsars embedded in non-stationary noise processes across a range of correlation lengths are missed by both the default SIGPROC and PRESTO pulsar search pipelines.



(a) SIGPROC



(b) PRESTO

Figure 8.7: The detection significance (values in black) at which 15 pulsars with different periods were detected (green squares) in files containing non-stationary noise with a relative amplitude of $h = 0.4$ and five different correlation lengths is depicted as traffic light plots (see experiment 5 in Table 6.2). The detection significance at which these pulsars were detected in files with stationary Gaussian noise is depicted in the panel below the traffic light plots. Each coloured box represents a single instantiation of a pulsar/non-stationary baseline pair that was searched by both SIGPROC and PRESTO. The red squares represent missed pulsars and the orange squares represent detected pulsars with detection significances below the default threshold levels. The results (traffic light plots) for files processed with the default pipelines in (a) SIGPROC and (b) PRESTO.

The detection significance at which pulsars with periods greater than 50 ms are detected decreases as the correlation length of the non-stationary noise shortens, whereas the detection significance of fast-period pulsars is unaffected by the correlation length of the non-stationary noise.

The results in Figure 8.7 portray single-trials for near-threshold signals which are very sensitive to the noise realisation used. To help understand the average and variance associated with the detection significance of these signals, I injected a pulsar with a period of 0.126 s in an ensemble of 20 noise realisations each with the same correlation length of 1 s and amplitude $h = 0.4$ (see Equation 5.12). This additional experiment showed that the average SNR at which the pulsar was detected in `SIGPROC` is 9 with a standard deviation of 1.45 compared to the SNR of 12.1 at which the pulsar is detected when embedded in stationary Gaussian noise (see bottom panel of Figure 8.7a). Similarly, the average Gaussian significance of the detected pulsar in `PRESTO` is 6.621 with a standard deviation of 1.14 compared to the stationary Gaussian noise case of 8.43 (see bottom panel of Figure 8.7b). Consequently, the results for this particular combination of period and correlation length that are plotted in Figure 8.7, Figure 8.8, Figure 8.9 and Figure 8.10 would show very little variability had there been more realisations of the same experiments. Multiple repetitions of this experiment over all combinations is extremely time costly and has not been attempted here. I have, however, computed similar standard deviations for other combinations of period and length scale (e.g periods of 5 s, 0.01 s and 0.002 s, and λs of 1 s, 0.01 s and 100 s), using small numbers of realisations (5 to 20). I find that the standard deviation in the experiments with no injected RFI remains similar to the measurements above, whereas the cases with RFI show increased variance, with measured standard deviations of between 2 and 3. Although this will have an effect on a case by case basis, the overall statistical picture can be interpreted.

It is evident from Figure 8.7a and Figure 8.7b that the default search pipeline in `PRESTO` is better at finding pulsars of various periods embedded in different non-stationary noise

processes compared to SIGPROC.

8.3.7 Sensitivity postcard plots of all the pipelines used to process files with PRESTO

The sensitivity plots for all the search pipelines explored in PRESTO (see Table 6.5) are depicted in Figure 8.8.

None of the pipelines in PRESTO is able to detect all the pulsars embedded in the different noise processes. Moreover, most of the pipelines miss the long-period pulsars. The addition of weak RFI, in general, does not alter PRESTO's ability to find pulsars.

Pipelines A, C, E and G in Figure 8.8 contain detections with Gaussian significances well in excess of the expected maximum Gaussian significance. I do not consider these outliers as true detections. However, do note that these pipelines all have one thing in common and that is they do not whiten the spectrum.

The only difference between pipelines A to D and E to H is the application of the RFI masking routine in PRESTO. From the results it appears that the RFI routine attenuates the Gaussian significance of short period pulsars below the detection threshold both in the presence and absence of RFI.

From these plots it is evident that running a moving average filter to normalise the time-domain data results in improved sensitivity, for example compare pipeline E with G and pipeline F with H. Note that when the moving average filter is applied in conjunction with the red-noise suppression method (see pipeline H in Figure 8.8) then more long-period pulsars embedded in non-stationary noise processes with long correlation lengths are detected compared to when only the moving average filter is applied (see pipeline F in Figure 8.8).

The pulsar search pipeline D in PRESTO (No RFI mask $\rightarrow$ MA filter $\rightarrow$ red noise mitigation) yields the best results amongst all the set-ups both in the presence and absence of RFI.

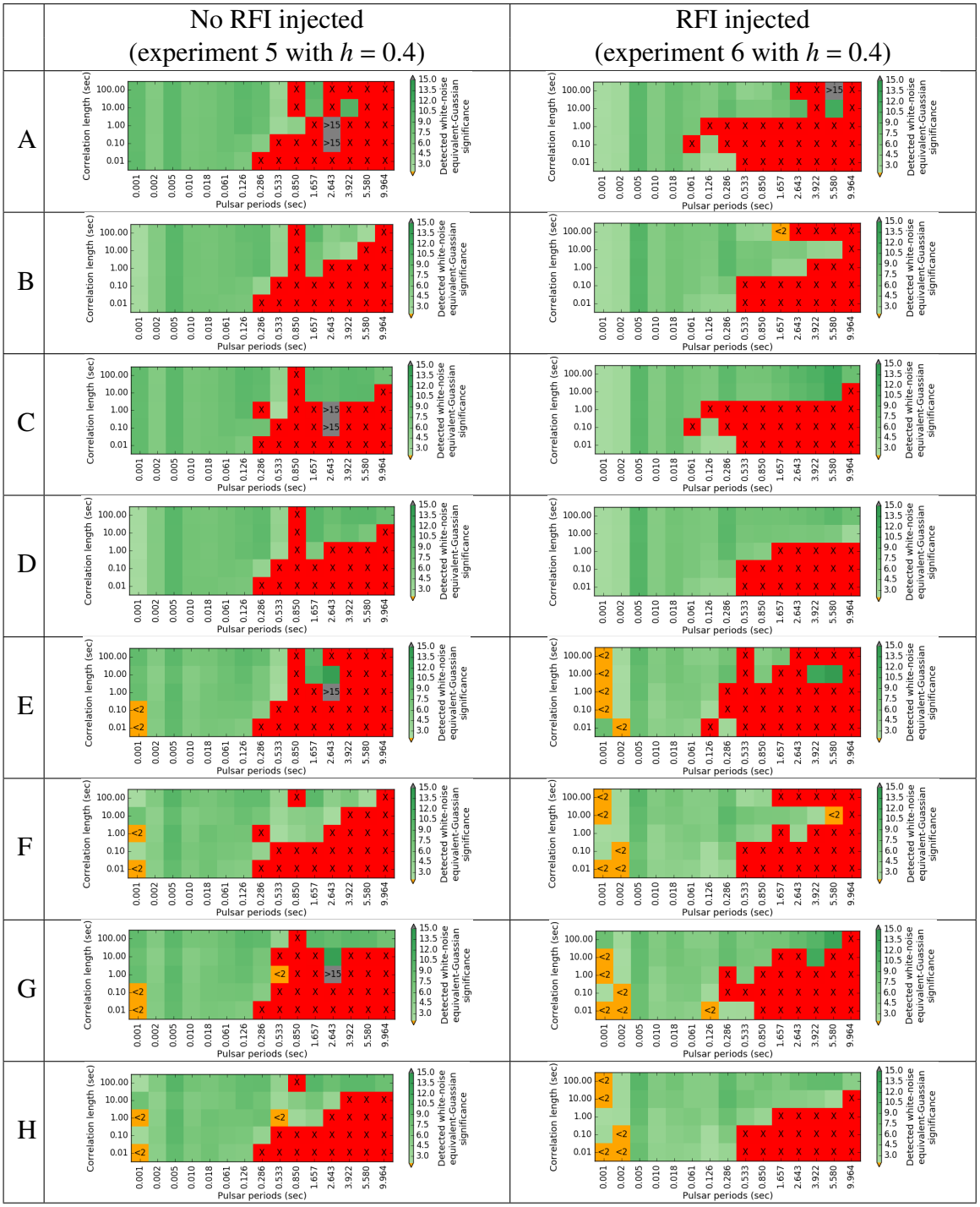


Figure 8.8: The Gaussian significance at which pulsars were detected (green squares) after files containing them (see experiments 5 and 6 in Table 6.2) were processed by eight different pipelines in PRESTO. The red squares represent missed pulsars, the orange squares represent detected pulsars with Gaussian significances below the default threshold level of 2 and the grey squares represent detected pulsars with Guassian significances above the average maximum Gaussian significance of 15.

8.3.8 Sensitivity postcard plots of all the pipelines used to process files with SIGPROC

The sensitivity plots for all the search pipelines explored in SIGPROC (see Table 6.4) are depicted in Figure 8.9 and Figure 8.10.

Note that the pulsar with period 0.002218 s is detected below the detection threshold (see orange squares in Figures 8.9 and 8.10) by almost all of the pipeline configurations in SIGPROC when RFI is present despite the other millisecond pulsars being detected. This pulsar is missed due to the increased variance of the SNR associated with the presence of RFI as explored and explained in section Section 8.3.6.

It is apparent from pipelines D and H in Figure 8.9 and pipeline L in Figure 8.10 that not normalising the spectrum results in a lot of pulsars being missed. Furthermore, mostly the long-period pulsars are regularly missed irrespective of the pipeline used in SIGPROC.

Overall PRESTO's performance across all the pipelines is more consistent when compared to the pipelines in SIGPROC.

8.4 Performance assessment of RFI mitigation

8.4.1 Introduction

In this section, I present the functionality of the RFI mitigation algorithm that was described in Chapter 7 along with the results of processing the files that were generated for the performance assessment of exiting pulsar search pipelines with the algorithm. Lastly, the efficacy of the RFI mitigation algorithm is demonstrated by processing pseudo real data.

8.4.2 Effect of RFI on the ACF

A synthetic filterbank file which contains non-stationary Gaussian noise and a non-uniform bandpass is plotted in Figure 8.11a. The ACF, corresponding to the time series of the

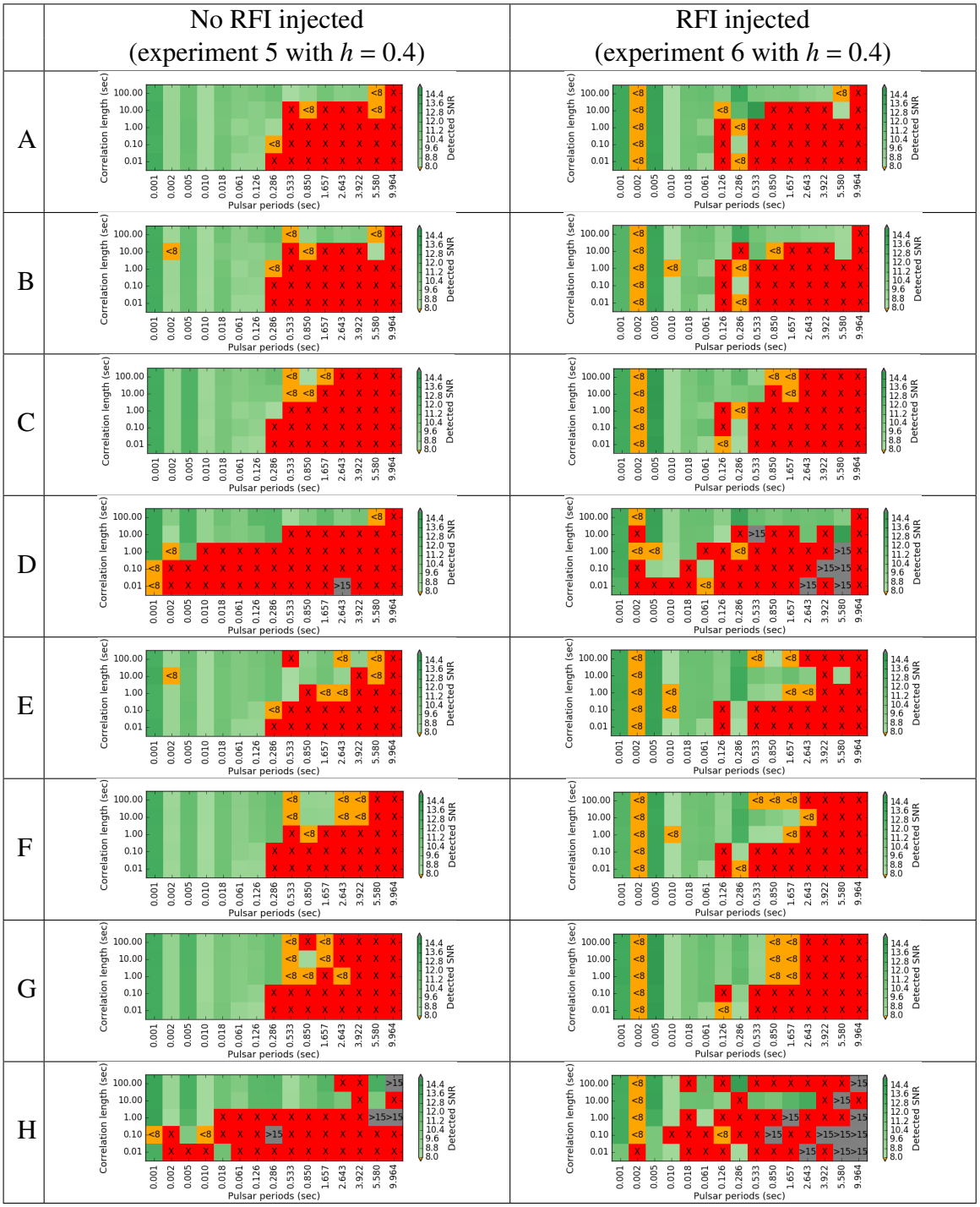


Figure 8.9: The SNR at which pulsars were detected after files containing them (see experiments 5 and 6 in Table 6.2) were processed by eight different pipelines in SIGPROC. The red squares represent missed pulsars, the orange squares represent detected pulsars with SNRs below the default threshold level of 8 and the grey squares represent detected pulsars with SNRs above the average maximum SNR of 15.

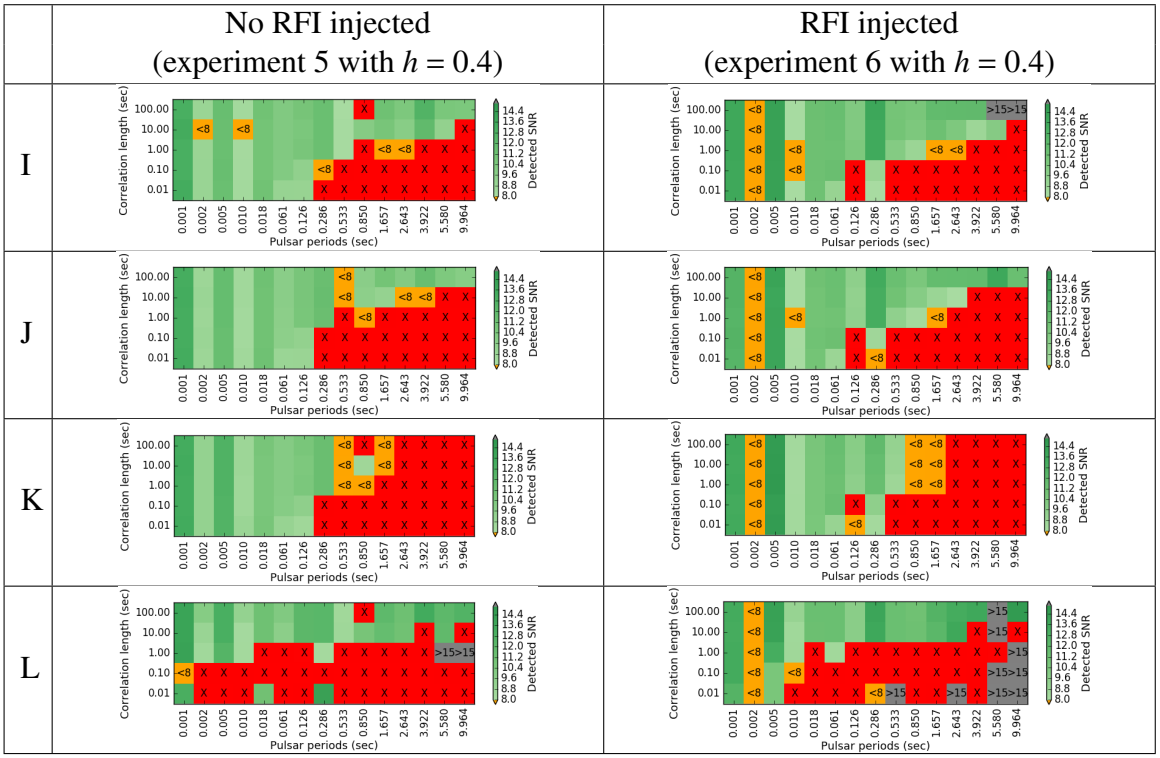


Figure 8.10: The SNR at which pulsars were detected after files containing them (see experiments 5 and 6 in Table 6.2) were processed by four additional pipelines in SIGPROC. The red squares represent missed pulsars, the orange squares represent detected pulsars with SNRs below the default threshold level of 8 and the grey squares represent detected pulsars with SNRs above the average maximum SNR of 15.

total power integrated across frequency of the first 5 s of the filterbank file, is plotted in Figure 8.11b. In this ideal case, a good estimate for both the correlation length of the non-stationary noise and the filter window length is the lag for which the ACF crosses the upper 95 % confidence interval which, for this data, is 1800 ms.

The same synthetic filterbank file plotted in Figure 8.11a was modified by injecting RFI into it, which resulted in the filterbank file plotted in Figure 8.11c. The ACF of the time series of the total power integrated across frequency for the first 5 s corresponding to this file is plotted in Figure 8.11d. The lag at which the ACF crosses the 95 % upper confidence bound for the first time is 400 ms.

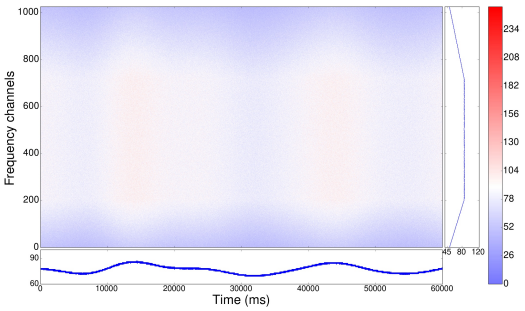
Comparing Figure 8.11b to Figure 8.11d it is evident that RFI significantly alters the shape of the ACF and the lag of the first upper 95 % crossing of the ACF.

8.4.3 Application to synthetic data

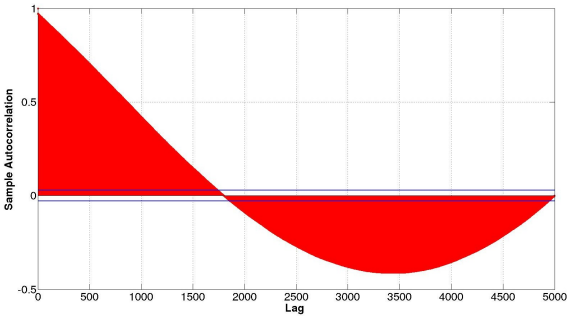
In order to illustrate the functionality and to test the efficacy of the algorithm described in Section 7.2, I first apply it to simulated data.

8.4.3.1 RFI excision algorithm functionality

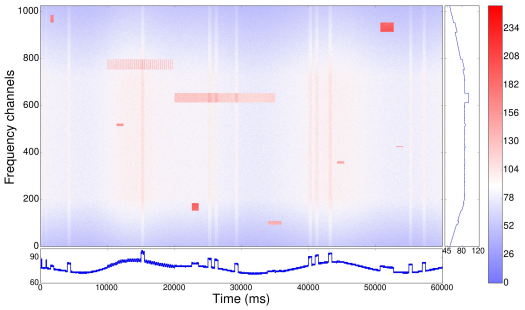
To illustrate the functionality of the various configurations of the algorithm, I simulated a synthetic filterbank file with observational parameters given in Table 8.1 with the software `Ersatz`. Several instances of RFI with varying intensities and durations were randomly injected into the file and the noise was simulated to be Gaussian and non-stationary with a correlation length of 5 s. A time-frequency plot and power spectrum density (PSD) plot of the synthetic file are shown in Figure 8.12A. The upper bound of the original PSD is traced to serve as a comparison in subsequent plots. Additionally, the PSD upper bound of a file containing only stationary Gaussian noise is also plotted as this is the idealised noise process assumed by all further processing steps in pulsar search pipelines. Notice, the PSD of a file with stationary Gaussian noise has a flat upper bound with no inflection points



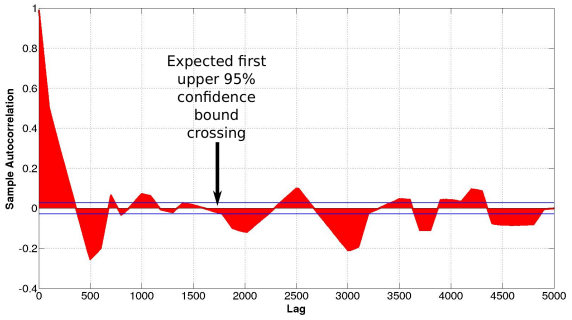
(a) Filterbank file with non-stationary baseline and without RFI.



(b) Autocorrelation of the first 5 s of the data without RFI.



(c) Filterbank file with non-stationary baseline plus RFI.



(d) Autocorrelation of the first 5 s of the data with RFI.

Figure 8.11: Failure of the ACF to determine the correct correlation length of the underlying noise process when RFI is present in the data can be seen in these plots. Left: Time-frequency plots of synthetic filterbank data without RFI (a) and with RFI (c). The lower panel shows the total power integrated across frequency and the right-hand panel shows the bandpass averaged over time. Right: ACFs corresponding to the filterbank data shown in the left hand plots with the 95 % confidence bounds associated with the variance of a white Gaussian noise process plotted as blue solid lines. (b) The ideal case: the ACF associated with the data which only contain non-stationary noise is a good proxy for determining the correlation length of the underlying noise process. (d) The worst case: the presence of RFI corrupts the ACF such that it is no longer a good proxy for estimating the correlation length of the underlying noise process.

whereas the PSD for the file with noise and RFI has about four inflection points. The aim is to process the file such that the upper bound of the PSD is uniform with no inflection points because then data are stationary, which in turn results in fewer false positive detections and increased sensitivity.

The results of processing the synthetic file with the configurations listed in Table 8.2 are shown in Figures 8.12 and 8.13.

It is clear from Figure 8.12B that filtering, normalising and rescaling of the synthetic file

Table 8.1: Observation parameters used to simulate the file used to illustrate the functionality of the RFI excision algorithm.

Parameter	Value
t_{obs}	60 s
t_{samp}	1000 μs
n_{bits}	8
n_{nchans}	1024
f_{low}	1024 MHz
f_{high}	1536 MHz
Bandwidth, Δf	512 MHz
Channel Bandwidth, Δf_{chan}	500 kHz

Table 8.2: The six configurations used to illustration the functionality of the complete RFI excision algorithm.

	Filter	Channel thresholding	Spectrum thresholding	Data imputation
A	-	-	-	-
B	✓	-	-	Noisy
C	✓	✓	-	Noisy
D	✓	-	✓	Noisy
E	✓	✓	✓	Noisy
F	✓	✓	✓	Static

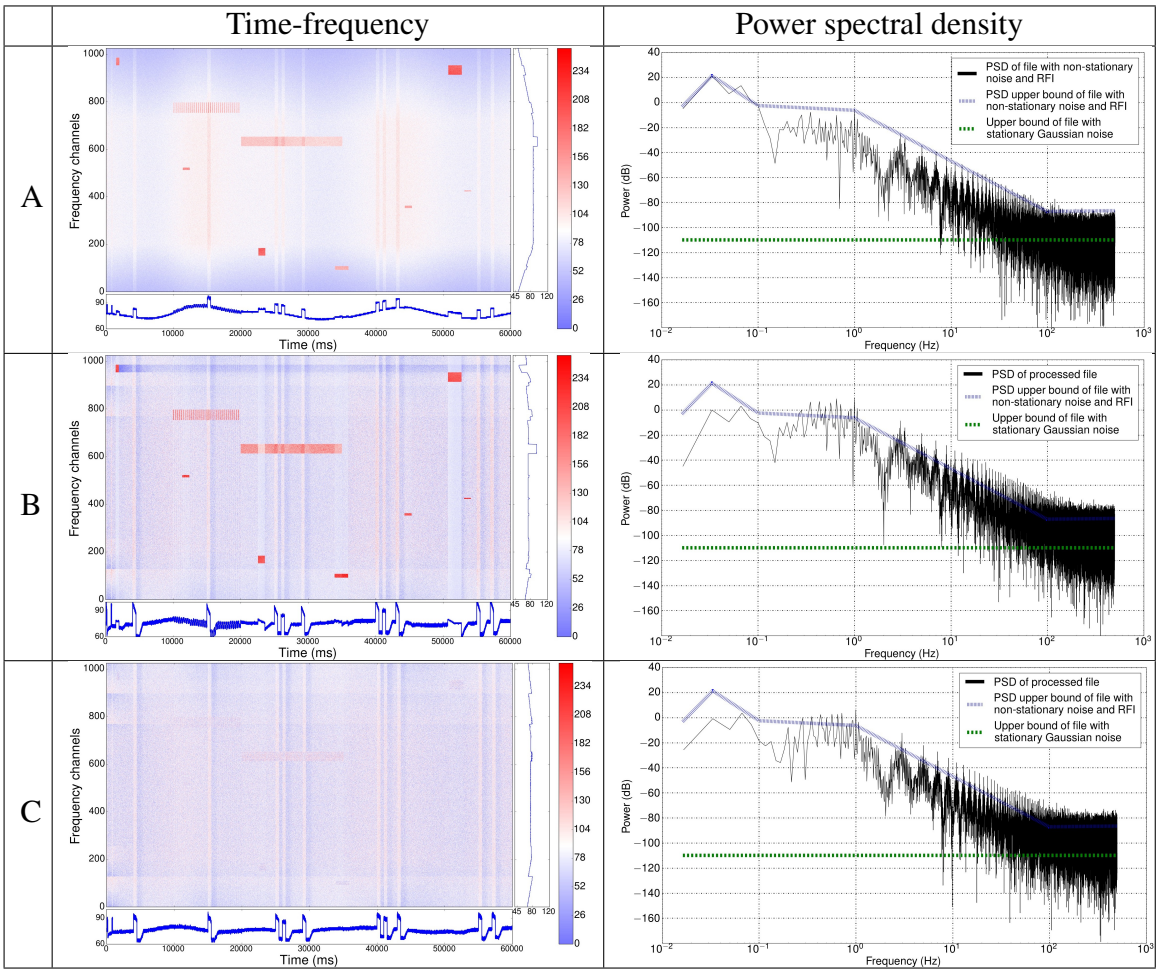


Figure 8.12: The effect of processing a filterbank file which contains non-stationary noise, a non-uniform bandpass and RFI with different configurations (see Table 8.2) of the RFI excision algorithm developed in this thesis. The plots on the left show the dynamic spectra. The plots on the right are the corresponding power spectral density plots of the time series of the total power integrated across frequency of the processed files.

merely changes the offset of the PSD but not its shape. Similarly, filtering and thresholding bright impulsive RFI by setting the scaling variable `_crFactor` does not improve the PSD (see Figure 8.12C); however, this configuration does result in a more uniform bandpass compared to the case where no thresholding was applied (see Figures 8.12B and 8.12C).

The result of filtering and thresholding broadband RFI by setting the scaling variable `_srFactor` is depicted in Figure 8.12D. This configuration improves the PSD by shifting the inflection point at 10^2 Hz to 20 Hz, which means that a larger number of frequencies in the power spectrum are at the same power level.

The effect of filtering in conjunction with setting the scaling variables `_crFactor` and `_srFactor` results in Figures 8.13E and 8.13F. In both cases the inflection point that was initially at 10^2 Hz has shifted to 10^1 Hz increasing the number of frequencies that are at the same power level. However, the difference between configurations E and F is the manner in which the samples that were identified to be affected by RFI are replaced. In configuration E the affected samples were replaced with the `_lastGoodSpectrum` plus samples drawn from a random normal distribution scaled by the moving average of the spectrum RMS (`_rmsRunAve`), i.e. noisy samples. Conversely, for configuration F the affected sample were replaced with only the `_lastGoodSpectrum`, i.e. static samples.

8.4.3.2 Searching for pulsars in synthetic data

The test data used in this section are the same data generated for experiments 5 and 6 of the framework for assessing pulsar search pipelines described in Section 6.3.3. For readability, I give a summary of the data: the data set comprises 150 files, 75 with RFI and 75 without. The simulated observation parameters are given in Table 8.3. Additionally, each file contains an injected pulsar and non-stationary Gaussian noise with a specific correlation length. The pulse periods explored range from 1.1 ms to 9.964 s and correlation lengths explored range from 10 ms to 100 s.

To benchmark the efficacy of the algorithm each of the synthetic files was processed with different configurations of the RFI excision algorithm (see Table 8.4) and then searched with the pulsar search software `PRESTO` (Ransom 2011).

To search the files with `PRESTO`, the following functions and their associated flags were called:

(a) function `prepdata` with the flags `-dm`, `-o`

(b) function `realfft`,

(c) function `zapbirds` with the flags `-zap` and `-zapfile`,

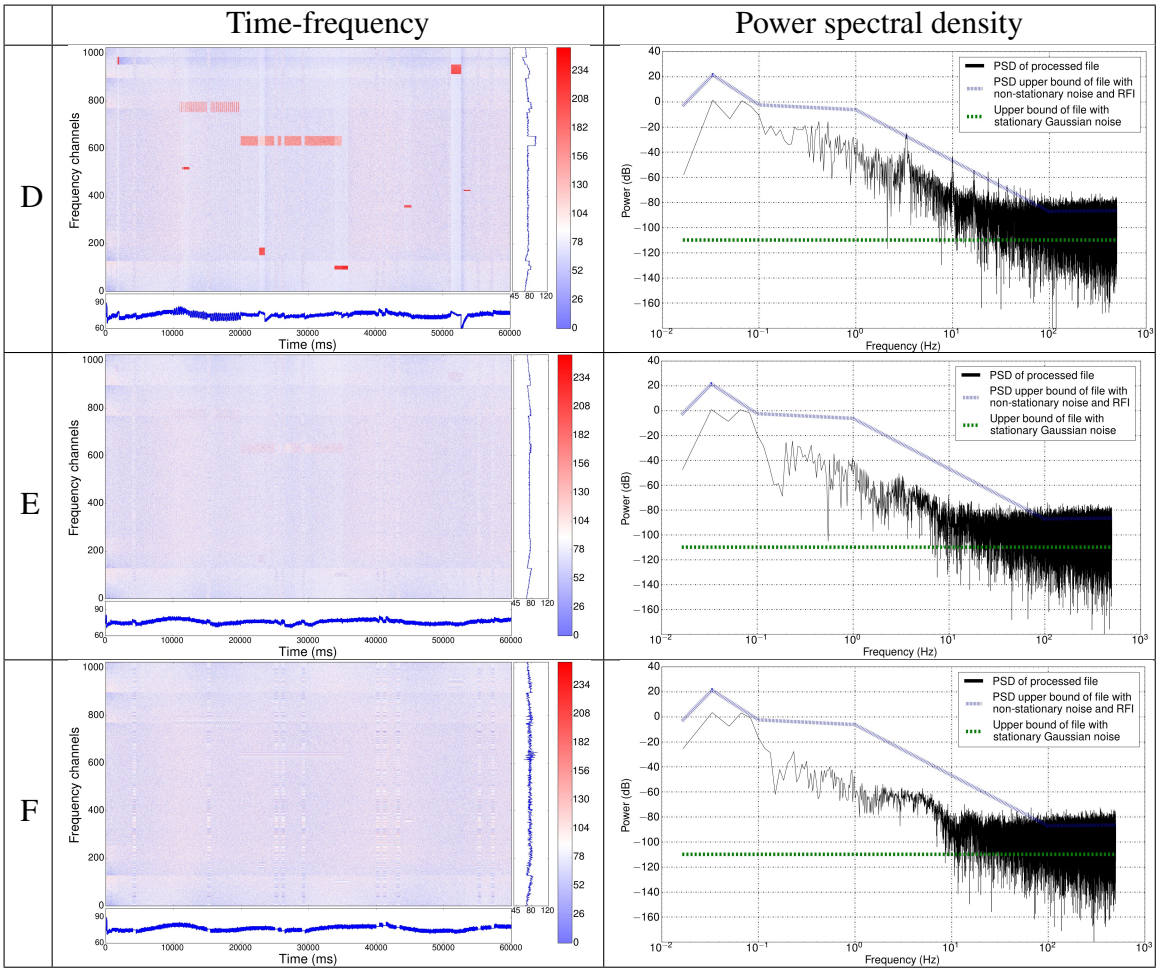


Figure 8.13: The effect of processing a filterbank file which contain non-stationary noise, a non-uniform bandpass and RFI with different configurations (see Table 8.2) of the RFI excision algorithm developed in this thesis. The plots on the left show the dynamic spectra. The plots on the right are the corresponding power spectral density plots of the time series of the total power integrated across frequency of the processed files.

Table 8.3: Simulated observation parameters for the files which contain pulsars and non-stationary Gaussian noise.

Parameter	Value
t_{obs}	300 s
t_{samp}	64 us
n_{bits}	8
n_{nchans}	512
f_{low}	1214 MHz
f_{high}	1536 MHz
Bandwidth, Δf	322 MHz
Channel Bandwidth, Δf_{chan}	628.91 kHz

Table 8.4: Eight configurations of the algorithm used to process all the synthetic files in this analysis.

	Filter length	Channel thresholding	Spectrum thresholding	Data imputation
A	-	-	-	-
B	Dynamic	✓	✓	Noisy
C	0.01 s	✓	✓	Noisy
D	0.10 s	✓	✓	Noisy
E	1.00 s	✓	✓	Noisy
F	10.0 s	✓	✓	Noisy
G	ZeroDM	✓	✓	Noisy
H	1.00 s	✓	✓	Static

(d) function `accelsearch` with the flags `-sigma 1.0`, `-flo 0.1`, `-zmax 0` (acceleration searching was turned off by setting the flag `-zmax 0`) and `-numharm 16` (i.e. the number of summed harmonics is 16).

The `accelsearch` function in PRESTO produces an ACCEL file which was searched for the injected pulsar.

The same heuristics (see Section 8.2) and colour codes (Section 8.3.6) are used to plot the results in Figures 8.14 and 8.15.

The search software PRESTO is able to remove frequency dependent noise. The plots in the left column of Figures 8.14 and 8.15 have had no frequency dependent noise removed, whereas the plots in the right column have had frequency dependent noise removed. Furthermore, the results for the filterbank files without RFI are plotted in Figure 8.14 and with RFI are plotted in Figure 8.15.

For configuration A (see Table 8.4) the files were searched with PRESTO, but neither the RFI excision algorithm nor the RFI clipper internal to PRESTO were used. In configuration B the filter length was determined by the ACF of the first 12 s of data. The results in Figure 8.14 for configuration B show an improvement compared to that of A. However, the same cannot be said when RFI is present in the data. In Figure 8.15 it is clear that setting the filter length equal to the lag for which the ACF of the time series first drops below the upper 95 % significance level results in fewer pulsars detected. These results confirm what was shown in Section 8.4.2 that the presence of RFI alters the ACF such that it is no longer a good estimate of the filter length and should therefore be used with caution.

Configurations C, D, E and F explore the effect that different filter lengths have on the detectability of pulsars embedded in different noise processes. The results in Figures 8.14 and 8.15 for these configurations should be interpreted with the transfer function of the high-pass filter depicted in Figure 7.3 in mind. It is evident that longer filter lengths are more conducive to finding long period pulsars irrespective of the presence of RFI. The short periods pulsars are unaffected by the filter length as these were consistently detected for all

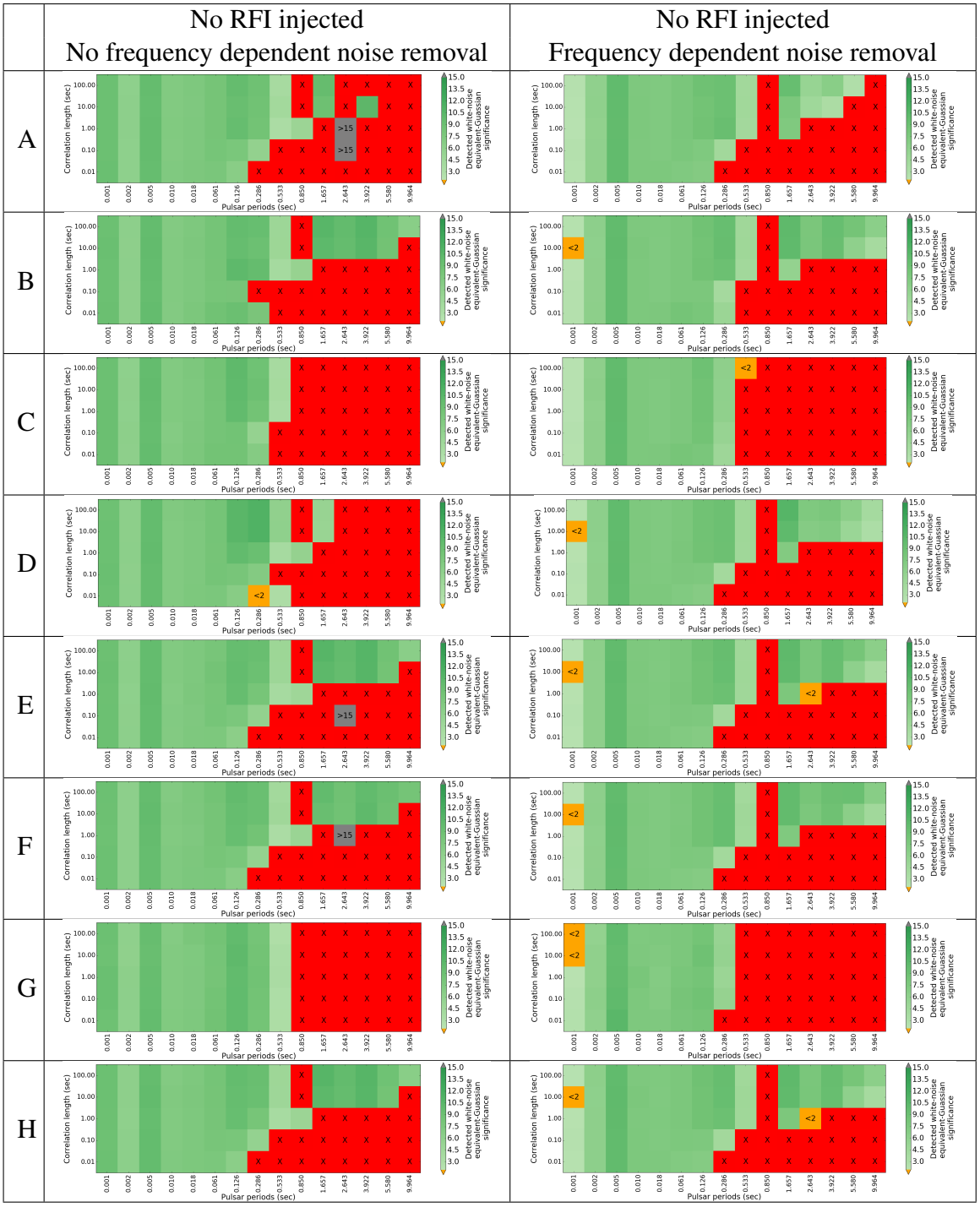


Figure 8.14: The Gaussian significance at which pulsars were detected (green squares) after files containing them and non-stationary noise were processed by eight different configurations of the RFI excision algorithm and then searhced by PRESTO. The red squares represent missed pulsars, the orange squares represent detected pulsars with Gaussian significances below the default threshold level of 2 and the grey squares represent detected pulsars with Guassian significances above the average maximum Gaussian significance of 15.

four configurations.

In configuration G the data were normalised by subtracting the mean of every spectrum, effectively removing signals at 0 DM. This is equivalent to having a filter with a length of exactly one sample. Consequently, this process results in approximately all frequencies < 3.5 Hz to be lost. The results in Figures 8.14 and 8.15 confirm this as all the pulsars with spin frequencies < 1.5 Hz have gone undetected. Note that normal pulsars with pulse widths a few percent of their pulse period usually have most of their power in spectral harmonics at higher frequencies, thus pulsars with spin frequencies on the cusp are still detectable because of the power in their harmonics. I refer the interested reader to Eatough et al. (2009) for a detailed explanation of the zero-DM filtering method and its dependence on DM, observing frequency and observation bandwidth.

The results of configuration H should be compared to those of E in both Figures 8.14 and 8.15 as the filtering processes for these two configurations were identical except for the values used to replace the affect samples. The number of detected pulsars is almost indistinguishable for the two configurations suggesting that replacing the affected samples with static values versus noisy values does not make a difference to the detection sensitivity. However, in Section 8.4.3.3 I will look at how data imputation affects the number of false positive detections per true detection.

Lastly, comparing the overall results it is clear that processing the files with the RFI excision algorithm in conjunction with removing frequency dependent noise yields the best results.

8.4.3.3 Data imputation

The results from configurations A, E and H are expressed here with two different metrics. The first is the number of false positives detected for every true positive detected. The second is the sensitivity of the configuration which is the number of pulsars detected out of the possible seventy five as a percentage.

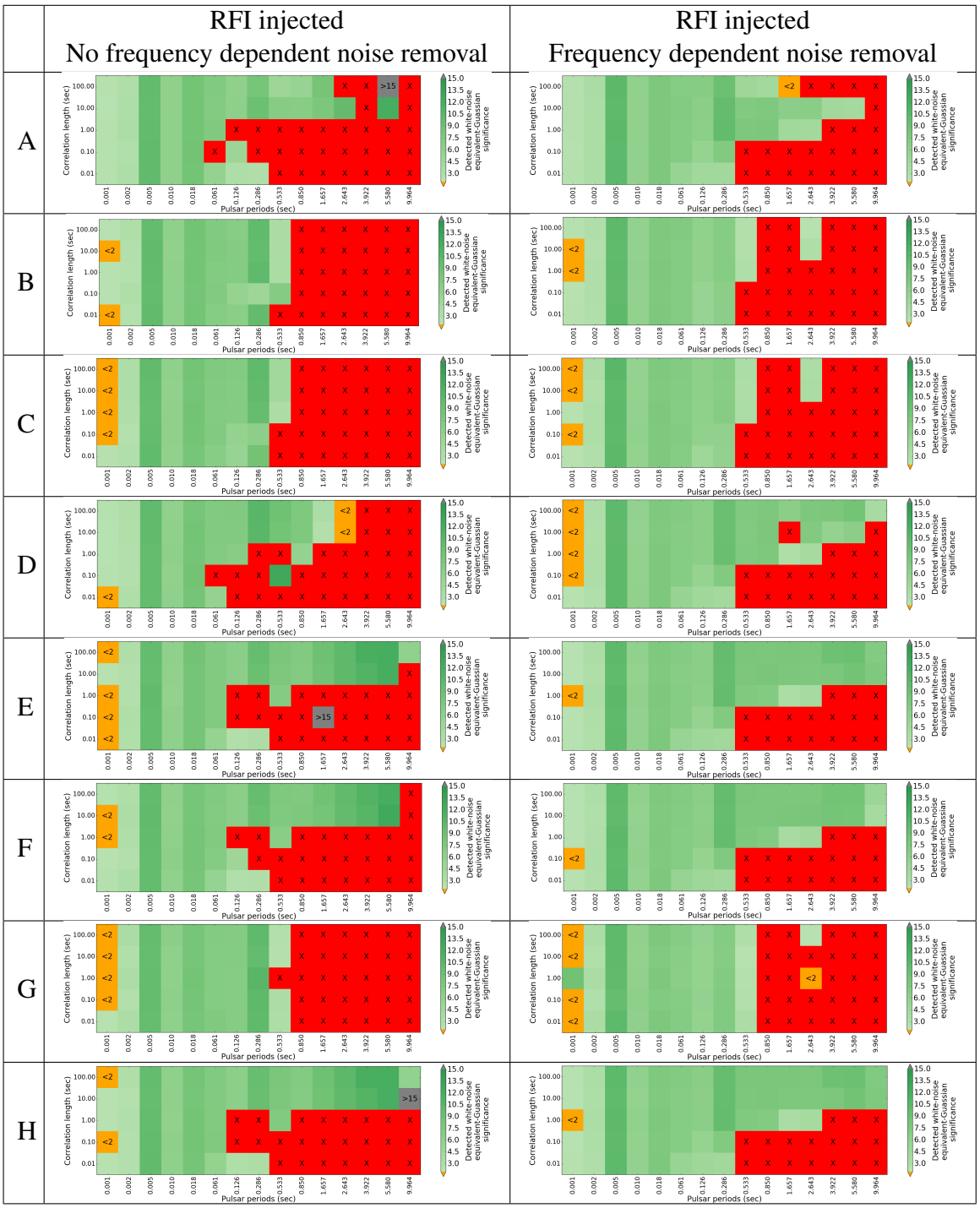


Figure 8.15: The Gaussian significance at which pulsars were detected (green squares) after files containing them, non-stationary noise and RFI were processed by eight different configrations of the RFI excision algorithm and then searhced by PRESTO. The red squares represent missed pulsars, the orange squares represent detected pulsars with Gaussian significances below the default threshold level of 2 and the grey squares represent detected pulsars with Guassian significances above the average maximum Gaussian significance of 15.

Table 8.5: Particulars of the Arecibo observation.

Parameter	Value
t_{obs}	30 s
t_{samp}	128 μs
n_{bits}	16
n_{nchans}	4096
f_{low}	1214 MHz
f_{high}	1177 MHz
Bandwidth, Δf	448 MHz
Channel Bandwidth, Δf_{chan}	109.375 kHz

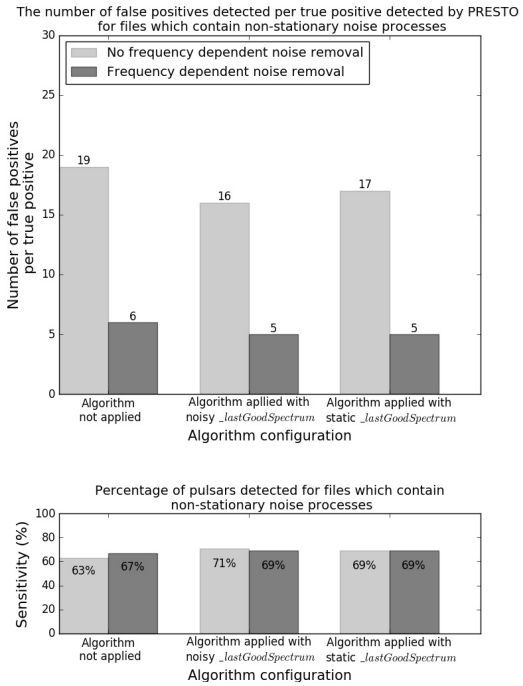
The results in Figure 8.16 reveal that in all instances the number of false positives per true positive detected were reduced when the files were processed with the RFI excision algorithm. Likewise, the sensitivity increased for all the configurations and search settings.

The best results in terms of the number of false positives per true positive detection, sensitivity and computational efficiency were achieved for both files with and without RFI when the affected samples were replaced by static values of the `_lastGoodSpectrum` and the frequency dependent noise was removed during the search process. A possible explanation for this result is that the static values of the `_lastGoodSpectrum` do not alter the overall statistics of the data, whereas noisy values of the `_lastGoodSpectrum` do.

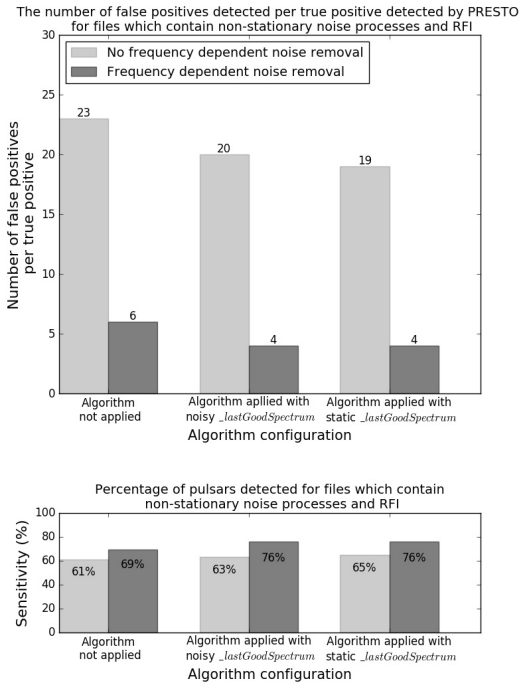
8.4.4 Application to real data with synthetic pulsar

To assess the performance of the RFI excision algorithm on pseudo real data I injected a synthetic pulsar signal with a period of 0.533 s and DM of 68 cm^{-3} pc into a filterbank file containing no known pulsar, obtained using the Alfa L-band receiver of Arecibo at full bandwidth. The particulars of the observation are summarised in Table 8.5. I attempt to recover the period of the input signal by processing the file with both the RFI excision algorithm presented here and the `rfifind` function available in `PRESTO`.

Different configurations of the `rfifind` function available in `PRESTO` were used to



(a) Results of processing filterbank files which do not contain RFI.



(b) Results of processing filterbank files which contain RFI.

Figure 8.16: The performance of processing the synthetic filterbank files (a) which contain no RFI and (b) which contain RFI with configurations A, E and H and then searching them with two configurations of PRESTO.

process the Arecibo fullband observation. The variable `-time`, which is the number of seconds to integrate for statistics and FFT calculations, was set to 1 s, 2 s and 5 s. The variable `-timesig` which is the ±sigma cut-off to reject time-domain chunks, indicated on the horizontal axes of the sub-plots in Figure 8.17, was set to 5, 10 and 15. The variable `-freqsig` which is the ±sigma cut-off to reject freq-domain chunks, indicated on the vertical axes of the sub-plots in Figure 8.17, was set to 4, 8 and 12.

Likewise, different configurations of the RFI excision algorithm were used to process the Arecibo fullband observation. The variable `-filterLength` was set to 1 s, 2 s and 5 s. The variable `-crFactor`, indicated on the horizontal axes of the sub-plots in Figure 8.18, was set to 5, 10 and 15. The variable `-srFactor`, indicated on the vertical axes of the sub-plots in Figure 8.18, was set to 4, 8 and 12.

The heuristics used for deciding if the pulsar was detected or not are the same as those listed in Section 8.2 with one exception and that is I consider the non-fundamental harmonics of the pulse period as a detection.

The detection of the pulsar is indicated with a green square in Figures 8.17 and 8.18. The intensity of the colour green is indicative of the Gaussian significance at which the pulsar was detected under the assumption of pure white noise. If the pulsar was missed it is indicated with a red square. Blue squares indicate that the pulsar was detected at one of its non-fundamental harmonics. The number that appears in the centre of each block is the number of false candidates that were identified for that particular file and configuration combination.

From the results plotted in Figures 8.17 and 8.18 it is evident that the RFI excision algorithm outperforms the function `rfifind`. Thus, the RFI excision algorithm described in this work requires less fine tuning of its parameters to effectively and consistently remove RFI and achieve high detection accuracy compared to the `rfifind` function available in `PRESTO`.

The RFI excision algorithm reduces the number of false detections to such an extent

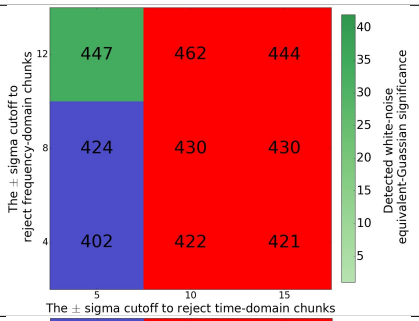
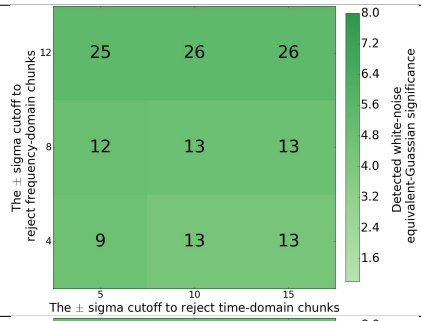
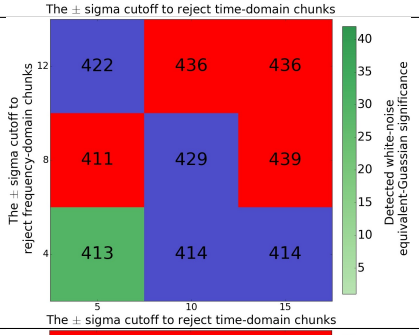
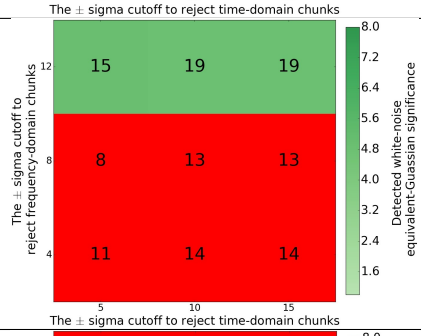
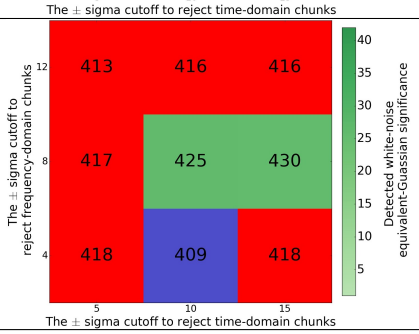
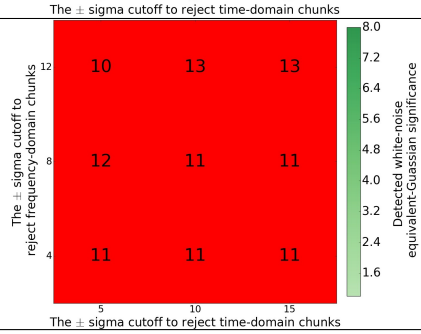
Integration time	PRESTO:rfifind No frequency dependent noise removal	PRESTO:rfifind Frequency dependent noise removal
1 s		
2 s		
5 s		

Figure 8.17: The green squares in this plot indicate that the pulsar which was injected into the Arecibo observation was detected by PRESTO after the data were processed with different configurations of the `rfifind` function avaiable in PRESTO. The intensity of the green squares are representative of the Gaussian significance at which the pulsar was detected. The red squares indicate that the pulsar was missed and the blue squares indicate that pulsar was detected at a non-fundamental harmonic. The number that appears in the centre of each block is the number of false candidates that were identified for that particular file and configuration combination.

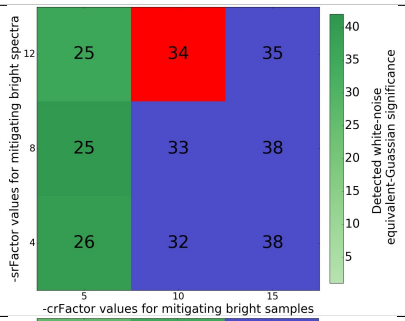
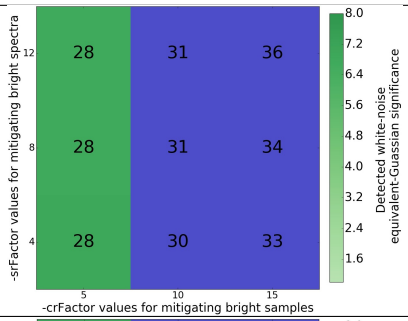
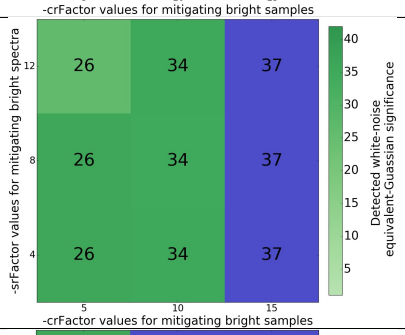
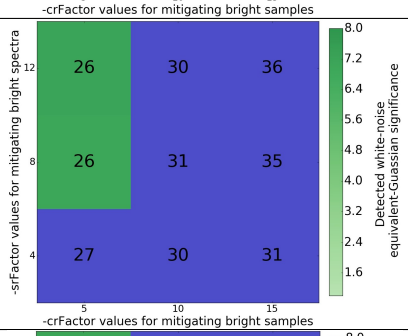
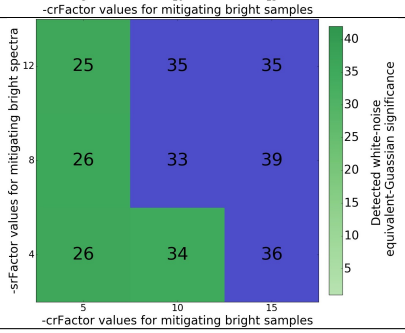
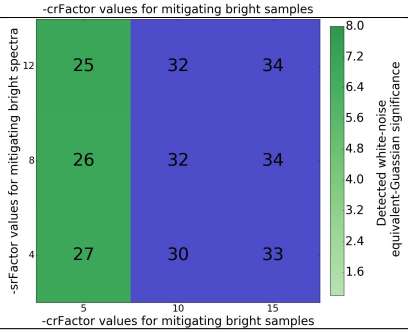
Filter length	RFI excision algorithm No frequency dependent noise removal	RFI excision algorithm Frequency dependent noise removal
1 s		
2 s		
5 s		

Figure 8.18: The results shown in these plots indicate whether or not the pulsar that was injected into the Arecibo observsation was detected by PRESTO after the data were processed with different configurations of the RFI excision algorithm. A description of what the colours represent can be found in the caption of Figure 8.17.

that the frequency dependent noise removal methods have no effect on the number of false detections. This is evident from the comparable number of reported false detections for the left and right columns of Figure 8.18. However, the files processed with `rfifind` for which no frequency dependent noise removal was applied resulted in a large number of false candidates. Thus, the RFI excision algorithm is as effective as the frequency dependent noise removal method in `PRESTO` at curtailing the number of false detections.

Considering the configurations for which the pulsar was detected at its fundamental frequency (green squares in Figures 8.17 and 8.18) it can be inferred that filter lengths/integration times of 1 s to 2 s are sufficient to process an observation. This is ideal, for shorter filter lengths require less memory and are computationally faster.

8.5 Conclusion

In this chapter, I presented the results from the performance analysis of existing pulsar search pipelines. One of the shortcomings identified by this analysis prompted the development of a RFI excision algorithm that is able to identify and remove both non-stationary variations and RFI from the data. The efficacy of this algorithm was demonstrated in this chapter by processing both synthetic and real data.

In the next chapter, I assess and discuss the implications of the results presented here.

Chapter 9

Discussions

9.1 Performance assessment of pulsar search pipelines

With the advent of instruments like the SKA, real-time processing will become essential. Therefore, it is crucial that the pipeline employed for this processing is optimal from the start. The purpose of the performance assessment of pulsar search pipelines was to investigate what improvements to these pipelines are necessary before embarking on the development of a new real-time processing pipeline that is adept at dealing with the demands posed by this new era of pulsar astronomy.

This analysis demonstrated that non-stationary Gaussian noise processes with different correlation lengths lead to an increase in the number of false detections per true pulsar detection because of the static threshold applied in the power spectrum to distinguish between possible pulsar candidates and noise, i.e. non-stationary Gaussian noise is partly to blame for the so called 'crisis' in candidate selection (Lyon et al. 2016). In order to reduce the high number of false positives, `SIGPROC` as well as `PRESTO` employ spectrum whitening methods. The analysis has revealed how these methods decrease the number of false positives per true positive at the cost of a loss in sensitivity and detection significance to long-period pulsars.

The spectrum whitening techniques assessed in this analysis suppress the power in the lower frequencies to conform to the power levels of the higher frequencies. Consequently,

the spectral power of real signals from slowly rotating pulsars is attenuated along with the noise. This analysis serves as evidence that there is room for improvement in the effectiveness of the current spectrum whitening methods. Instead of forcing the spectrum to be uniform in the lower frequencies, the solution should rather be to accurately model the noise both in the spectral and in the time domain. In fact I have shown that applying a 10 s moving average filter in the time domain, which is common in many pulsar search instruments, resulted in a greater number of detections of long period pulsars. This simple filtering result suggests that a more advanced real-time filter should be employed in order to increase surveys' sensitivity to long period pulsars. The algorithm described in Section 7 is exactly that, an advancement on this simple 10 s filter.

As I indicated earlier, the results presented in Figure 8.7, Figure 8.8, Figure 8.9 and Figure 8.10, are based on single realisations of period-length scale combinations. However, I have sampled the standard deviation of the significance of these detections for several cases, and I conclude that although the picture may change for different realisations and different initial S/N values of the injected pulsars, the areas in the plots which are most affected remain the same.

In this analysis I dedispersed all the files at the same DM as the injected DM = 68 pc cm^{-3}. However, a subset of the files were dedispersed at four additional DM values, namely 0, 20, 150 and 300 pc cm^{-3}. Dedispersing the files at these four additional DMs allowed us to confirm that the number of false positives detected by the pulsar search pipelines for the files containing both non-stationary noise and RFI is greatest when the filterbank files are not dedispersed and decreases as one moves away from 0 DM. However, the number of false positives detected by the pulsar search pipelines is very similar for the five DMs used to dedisperse the data. Consequently, the number of false positives detected for files containing only non-stationary noise is similar irrespective of the DM used to dedisperse the data.

In this analysis it was demonstrated that the RFI detection algorithm in PRESTO is very

sensitive to the interplay between integration length over which the statistics of the filterbank files are computed and the rejection thresholds both in time and frequency of said statistics. For off-line processing this interplay can be fine tuned so that most RFI at different brightness levels can be detected and masked. However, for the real-time detection of RFI this exploration of parameter space is not always possible because of the time constraint as well as the dynamic nature of the RFI environment.

There are a multitude of modules each placed strategically throughout current pulsar search pipelines for detecting different sources of RFI. Most of these RFI detection algorithms are largely amplitude-based and are therefore very sensitive to non-stationary baselines. Consequently, data which contain no RFI but which have a number of peaks induced by correlations in the noise component are flagged as RFI. This analysis demonstrated that by flagging and replacing blocks of non-stationary data which contain no RFI or weak RFI may result in short period pulsars being attenuated below the detection threshold.

In conclusion, there is a need for algorithms that can simultaneously normalise a non-stationary baseline and excise RFI signals superposed on said baseline without compromising the data that is not affected.

9.2 Performance assessment of RFI excision algorithm

The results from the performance assessment analysis of existing pulsar search pipelines highlight the necessity for algorithms that are able to identify and remove both non-stationary variations and RFI from the data before searching is performed in order to limit false positive detections. To address this shortcoming I developed a RFI excision algorithm for simultaneously normalising non-stationary noise baselines and mitigating RFI.

The algorithm that I developed was used to process the files generated for experiments 5 and 6 as described in Section 6.3.3 before being searched by the pulsar search software `PRESTO`. Processing the data with the algorithm marginally improved the sensitivity for

long period pulsars embedded in non-stationary noise with long correlation lengths. Furthermore, the results for the files which were processed with the algorithm compared to those that were not processed before being searched, had a lower number of false positive detections for each true detection, validating that the algorithm does not introduce artefacts by replacing the affected samples with the last good spectrum. The results obtained from processing real data with the algorithm demonstrate that, in addition to excising bright RFI, the algorithm normalises both the bandpass and the non-stationary noise baseline inherent to radio observations.

The true-/false-positive ratio of RFI classification depends in part on the true nature of the RFI environment and in part on the choice of threshold values. If the intensity of the RFI is low and the threshold values are set generously then both the true-/false-positive ratios will be low and RFI will leak through. Conversely, if the threshold values are set too conservatively then both the true-/false-positive ratios of RFI classification will be high resulting in healthy data being replaced. Ideally, both these scenarios should be avoided as both compromise the fidelity of the data. What is desired is a high true-positive ratio and a low false-positive ratio of RFI classification which can be achieved by choosing optimal threshold values for the `_crFactor` and `_srFactor` parameters in the RFI excision algorithm. For example setting `_crFactor` = 10 and `_srFactor` = 4 dictate that all samples with signal-to-noise ratios (SNRs) ≥ 10 and spectra for which all the channels in the spectrum have SNRs ≥ 4 should be replaced. Alternatively, these specific threshold values (`_crFactor` = 10 and `_srFactor` = 4) say that if the number of channels n with SNRs ≥ 9 (essentially any value below `_crFactor`) are greater or equal to m, where $m = (=\frac{\texttt{srFactor}*\sqrt{\texttt{nchans}}}{SNR=9})^2$, the whole spectrum will be replaced. The variable `nchans` is the number of channels in the spectrum. Consequently, these two parameters are coupled because setting the one influences the other. Further, the values at which these two thresholds, `_crFactor` and `_srFactor`, are set encode our expectation/knowledge about the type, the intensity and the number of channels affected by the RFI environment.

Considering the speed of any algorithm which forms part of a pulsar processing pipeline is crucial for the real-time restrictions that current and future surveys demand. The running time of the RFI excision algorithm scales linearly with the number of input samples and the space requirements of the algorithm scales with the number of samples times the filter length. Thus, choosing a `_filterLength` $\ll N$ where N is the number of samples in the observation will result in the algorithm processing radio observations in a fraction of real-time.

This algorithm flags all the samples believed to be affected by RFI and saves the time-frequency pairs associated with those samples according to the type of RFI it is identified as. Furthermore, it recovers the contaminated samples by replacing them with values saved in `_lastGoodSpectrum`. Consequently, this algorithm not only discovers samples affected by RFI but it also recovers them.

Two approaches for replacing RFI affected samples were considered and implemented in this algorithm (see Section 7.2.4). A thorough analysis revealed that the optimal approach is the static replacement of RFI affected samples. By replacing the RFI affected samples with a static version of the `_lastGoodSpectrum` caused minimal damage to the astronomical signal of interest by yielding the best sensitivity of the two approaches. Moreover, the static approach produced the lowest number of false positives per true positive detected. Great care was taken to place the `_lastGoodSpectrum` at the level of the data such that the dedispersed time series did not exhibit any jumps or spurious offsets. Note that samples identified as RFI type 4 are only flagged but not replaced as replacing them after the data have been normalised and scaled has been found to alter the statistics. Ultimately, an advancement upon the algorithm developed in this thesis can be made by finding a way to replace the channels that contain low-level persistent RFI without introducing offsets or artefacts in the baseline of the normalised data.

The time variability of most RFI demands analysis on short time scales. This algorithm was designed to process streaming channelised data such that it can adapt instantly to the

time and frequency evolution of the measured signal during a pulsar observation.

The ability of the algorithm to correct a non-uniform bandpass and baseline was achieved by learning the bandpass over a number of consecutive phases until it converged to a stable state and thereafter adjusting its offset based on the moving average of the data. The results presented in Section 8.4.3.2 revealed that a good filter length for correcting the non-stationary noise baseline whilst maintaining sensitivity to long-period pulsars is 1 s.

The algorithm developed in this thesis adheres to all the consideration listed in Section 7.1 and can thus be considered as a *bona fide* RFI mitigation method.

9.3 Avenues for further inquiry

Following from the results of this thesis there are a number of avenues, related to pulsar searching, that warrant further inquiry which include: excising weak narrowband RFI from pulsar search data, developing a new peak detection algorithm and exploring alternative methods to extract unbiased features from pulsar profiles to aid candidate selection methods in classifying pulsars, RFI and noise correctly. The effectiveness of these methods can be benchmarked by applying them to the synthetic files created for the sensitivity analysis and then comparing their effectiveness to the results obtained in this thesis.

Regarding weak narrowband RFI, the RFI excision algorithm presented in this work can be improved upon by finding alternative ways to replace samples affected by low level persistent narrowband RFI without altering the statistics of the data. Low level persistent RFI is only detected by integrating the already normalised data in time. Thus, replacing the affected samples is not straightforward and, if done incorrectly, results in the data not preserving their lowest order moments.

Another avenue to explore is the development of a new peak detection method that is insensitive to slowly changing and locally monotonic functions, i.e. frequency dependent noise, which follow a power law in the frequency domain. The wavelet transform is one

such method that can be used to perform pattern matching of peaks in the frequency domain without needing the baseline to be removed or normalised. Exploring various types of wavelets and parameter settings for the optimal peak detection of pulsar candidates in the frequency domain is not trivial and requires a lot of analysis.

Lastly, alternative methods to extract unbiased features from folded pulsar profiles to aid supervised learning algorithms in classifying pulsars, RFI and noise correctly is an exciting avenue ripe for exploration. Currently, features used for classifying pulsars, noise and RFI are carefully crafted and knowledge-driven instead of tapping into the fourth paradigm of science where the data dictate what the important features are. Techniques such as variational auto-encoders can be used to extract unbiased data-driven features from folded pulsar profiles, which are sure to improve pulsar classification and limit the number of false candidates.

The sheer volume and rate of astrophysical data has, to date, prohibited the use of techniques such as deep neural networks, but the impressive performance of such methods on image analysis tasks, for example, indicated that they may be a valuable future methodology.

Chapter 10

Conclusions and Outlook

10.1 Conclusions

The two main drivers behind the work presented in this thesis are the SKA with its ensuing Big Data challenges and the low pulsar detection rates of recent pulsar surveys.

The number of pulsars discovered in recent surveys (Swiggum et al. 2014, Lazarus et al. 2015) has fallen far short of the number predicted by pulsar population synthesis models. These shortfalls are attributed to frequency dependent noise, RFI and scintillation.

In this thesis, I set out to corroborate these findings and to further investigate if the shortfall between the predicted and discovered number of pulsars can be attributed to the very algorithms employed to find them. Furthermore, I tried to ascertain why these search pipelines are vulnerable to the presence of frequency dependent noise and RFI when various RFI mitigation methods and spectrum whitening routines exist in pulsar search pipelines to prevent this vulnerability.

In order to achieve these objectives, I first developed a new method for simulating pulsar search data that contain different types of RFI and frequency dependent noise. The novelty of the method is that it is a state-space representation of a Gaussian Process, which is computationally and memory-wise more efficient than a traditional Gaussian Process. This surrogate modelling technique was then used in a framework that I developed to inexpensively assess the performance of existing pulsar search pipelines for different noise and

RFI settings. The findings of this analysis accords with the Lazarus et al. (2015) PALFA sensitivity analysis that frequency dependent noise and weak RFI leads to an increase in the number of false positives and lower sensitivity for long period pulsars. These two effects have resulted in overestimates of survey yields. Furthermore, it revealed that the severe degradation of the detection significance is partly due to frequency dependent noise and partly due to the attenuating nature of the spectrum whitening algorithms implemented in pulsar search software. Both these effects serve as an explanation for why so many detectable long period normal pulsars are missed by pulsar search pipelines. In conclusion, the results highlighted the necessity to develop algorithms that are able to identify and remove non-stationary variations from the data before RFI excision and searching is performed in order to limit false positive detections.

To address the shortcoming identified with the framework that assessed the performance of existing pulsar search pipelines, I developed a new real-time algorithm for excising RFI while simultaneously normalising the variability in time and frequency inherent to pulsar observations. The algorithm works on streaming data, which is ideal considering that the time variability of most interferences demands analysis on short time scales. Furthermore, the algorithm scales almost linearly with the number of input samples. I demonstrated the efficacy of the algorithm by processing and searching synthetic and pseudo-real data, which revealed that by applying the algorithm before searching for pulsars with conventional pulsar search software, expands the parameter space for which we are able to successfully detect pulsars.

In conclusion, the findings that I presented in this thesis and the new methods developed edge us a bit closer to a pulsar search pipeline that will enable the SKA to detect all pulsars beaming towards Earth and successfully conduct the planned experiments in pulsar science.

10.2 Outlook

The techniques developed in this thesis address some of the challenges associated with Big Data analysis such as inconsistency and incompleteness of data, timeliness, and scalability. However, these techniques slot into the bigger SKA project, which still has a lot of challenges looming. Thus, considerable advances in hardware, software, data analysis and time series modelling are still required to ensure that the SKA project comes to fruition.

Only once all these challenges are overcome will astrophysicists be able to distil meaning from SKA observations and do ground breaking research, which include addressing all the open questions that prompted the conceptualisation, design and development of the SKA.

Future advancements in machine intelligence and Big Data from the SKA will enable radio astronomy to advance into the fourth paradigm of science. This will allow astrophysicists to interrogate the Universe in a radically different way by using data-driven mathematical models instead of knowledge-driven models. This way of synthesising knowledge will allow us to go beyond the known and allow us to explore, learn and discover the unknown unknowns.

Beyond the SKA lies a future where the interplay of machine learning and astrophysics allows us to peer into the Universe with a resolution hitherto unknown.

Bibliography

Abdo, A., Ajello, M., Allafort, A., Baldini, L., Ballet, J., Barbiellini, G., Baring, M., Bastieri, D., Belfiore, A., Bellazzini, R. et al. (2013), 'The second fermi large area telescope catalog of gamma-ray pulsars', *The Astrophysical Journal Supplement Series* **208**(2), 17.

Adámek, K., Novotnỳ, J. & Armour, W. (2016), 'A polyphase filter for many-core architectures', *Astronomy and Computing* **16**, 1–16.

Agarwal, R. & Dhar, V. (2014), 'Big data, data science, and analytics: The opportunity and challenge for IS research'.

Agnew, D. C. (1992), 'The time-domain behavior of power-law noises', *Geophys. Res. Lett* **19**(4), 333–336.

Antoniadis, J. (2014), *Multi-wavelength studies of pulsars and their companions*, Springer.

Armour, W., Karastergiou, A., Giles, M., Williams, C., Magro, A., Zagkouris, K., Roberts, S., Salvini, S., Dulwich, F. & Mort, B. (2011), 'A GPU-based survey for millisecond radio transients using ARTEMIS', *arXiv preprint arXiv:1111.6399* .

Aulbert, C. (2007), 'Finding binary millisecond pulsars with the Hough transform', *arXiv preprint astro-ph/0701097* .

Backer, D., Clifton, T., Wertheimer, D. & Kulkarni, S. (1990), 'A digital signal processor for pulsar research', *Astronomy and Astrophysics* **232**, 292–300.

Barr, E. (2013), 'Peasoup'.

Barr, E. D., Champion, D. J., Kramer, M., Eatough, R. P., Freire, P. C., Karuppusamy, R., Lee, K., Verbiest, J. P., Bassa, C. G., Lyne, A. G. et al. (2013), 'The northern high time resolution universe pulsar survey–i. setup and initial discoveries', *Monthly Notices of the Royal Astronomical Society* **435**(3), 2234–2245.

Bates, S., Lorimer, D., Rane, A. & Swiggum, J. (2014), 'PsrPopPy: an open-source package for pulsar population simulations', *Monthly Notices of the Royal Astronomical Society* **439**(3), 2893–2902.

Beck, R. (2010), 'Square Kilometre Array'.
URL: *http://www.scholarpedia.org/article/Square_kilometre_array*

Beran, J., Feng, Y., Ghosh, S. & Kulik, R. (2013), Springer Berlin Heidelberg, Berlin, Heidelberg.
URL: *http://dx.doi.org/10.1007/978-3-642-35512-7*

Bhat, N., Cordes, J., Chatterjee, S. & Lazio, T. (2005), 'Radio frequency interference identification and mitigation using simultaneous dual-station observations', *Radio science* **40**(5).

Boyles, J., Lynch, R. S., Ransom, S. M., Stairs, I. H., Lorimer, D. R., McLaughlin, M. A., Hessels, J. W., Kaspi, V. M., Kondratiev, V. I., Archibald, A. et al. (2013), 'The green bank telescope 350 MHz drift-scan survey. I. Survey observations and the discovery of 13 pulsars', *The Astrophysical Journal* **763**(2), 80.

Bryant, R. E. (2011), 'Data-intensive scalable computing for scientific applications', *Computing in Science & Engineering* **13**(6), 25–33.

Carilli, C. L. (2014), Square Kilometre Array key science: a progressive retrospective, *in* 'Proceedings of Advancing Astrophysics with the Square Kilometre Array, Giardini Naxos, Italy, June 9–13, 2014', PoS(AASKA14)171.

Carilli, C. & Rawlings, S. (2004), 'Motivation, key science projects, standards and assumptions', *New Astronomy Reviews* **48**(11-12), 979 – 984. Science with the Square Kilometre Array.
URL: *http://www.sciencedirect.com/science/article/pii/S1387647304000880*

Champion, D., Petroff, E., Kramer, M., Keith, M., Bailes, M., Barr, E., Bates, S., Bhat, N., Burgay, M., Burke-Spolaor, S. et al. (2016), 'Five new fast radio bursts from the HTRU high-latitude survey at Parkes: first evidence for two-component bursts', *Monthly Notices of the Royal Astronomical Society: Letters* **460**(1), L30–L34.

Chatfield, C. (2016), *The analysis of time series: an introduction*, 6th edn.

Coenen, T., Van Leeuwen, J., Hessels, J. W., Stappers, B. W., Kondratiev, V. I., Alexov, A., Breton, R., Bilous, A., Cooper, S., Falcke, H. et al. (2014), 'The LOFAR pilot surveys for pulsars and fast radio transients', *Astronomy & astrophysics* **570**, A60.

Constine, J. (2012), 'How big is facebook's data? 2.5 billion pieces of content and 500+ terabytes ingested every day', *Retrieved from TechCrunch: https://techcrunch.com/2012/08/22/how-big-is-facebooks-data-2-5-billion-pieces-of-content-and-500-terabytes-ingested-every-day/* .

Cordes, J. (1980), 'Pulsar timing. II-Analysis of random walk timing noise-application to the Crab pulsar', *The Astrophysical Journal* **237**, 216–226.

Cordes, J. M., Freire, P., Lorimer, D. R., Camilo, F., Champion, D. J., Nice, D. J., Ramachandran, R., Hessels, J., Vlemmings, W., Van Leeuwen, J. et al. (2006), 'Arecibo pulsar survey using ALFA. I. Survey strategy and first discoveries', *The Astrophysical Journal* **637**(1), 446.

Cordes, J. M. & Lazio, T. J. W. (1997), 'Finding radio pulsars in and beyond the Galactic center', *The Astrophysical Journal* **475**(2), 557.

Cordes, J. & McLaughlin, M. A. (2003), 'Searches for fast radio transients', *The Astrophysical Journal* **596**(2), 1142.

Cordes, J. et al. (2010), 'The square kilometre array, project description for astro2010, response to program prioritization panels, 1 April 2009'.

Crawford, F., Kaspi, V. M., Manchester, R. N., Lyne, A. G., Camilo, F. & D'Amico, N. (2001), 'Radio pulsars in the magellanic clouds', *The Astrophysical Journal* **553**(1), 367.

CSIRO (2017), 'Australian Square Kilometre Array Pathfinder'.
URL: *https://www.csiro.au/en/Research/Facilities/ATNF/ASKAP*

De, K. & Gupta, Y. (2016), 'A real-time coherent dedispersion pipeline for the giant metrewave radio telescope', *Experimental Astronomy* **41**(1-2), 67–93.

de Vos, M., Gunst, A. W. & Nijboer, R. (2009), 'The LOFAR telescope: System architecture and signal processing', *Proceedings of the IEEE* **97**(8), 1431–1437.

Deneva, J., Stovall, K., McLaughlin, M., Bates, S., Freire, P., Martinez, J., Jenet, F. & Bagchi, M. (2013), 'Goals, strategies and first discoveries of AO327, the Arecibo All-sky 327 MHz drift pulsar survey', *The Astrophysical Journal* **775**(1), 51.

Dewdney, P. E., Hall, P. J., Schilizzi, R. T. & Lazio, T. J. L. W. (2009), 'The square kilometre array', *Proceedings of the IEEE* **97**(8), 1482–1496.

Dewdney, P., Stevenson, T., McPherson, A., Turner, W., Braun, R., Santader-Vela, J., Waterson, M. & Tan, G.-H. (2014), 'SKA1 System Baseline Design V2'.
URL: *http://skatelescope.org/wp-content/uploads/2014/03/SKA-TEL-SKO-0000308_SKA1_System_Baseline_v2_DescriptionRev01-part-1-signed.pdf*

Dewdney, P., Turner, W., Millenaar, R., McCool, R., Lazio, J. & Cornwell, T. (2013), 'SKA1 system baseline design', *Document number SKA-TEL-SKO-DD-001 Revision* **1**(1).

Dimoudi, S. & Armour, W. (2015), 'Pulsar acceleration searches on the GPU for the Square Kilometre Array', *arXiv preprint arXiv:1511.07343* .

Dobre, C. & Xhafa, F. (2014), 'Parallel programming paradigms and frameworks in big data era', *International Journal of Parallel Programming* **42**(5), 710.

Eatough, R. (2011), 'Pulsar acceleration processing'.
URL: *https://www.skatelescope.org/public/2011-04_Signal_Processing_CoDR_Documents/Presentations/34-SKA_CoDR_acceleration.pdf*

Eatough, R., Lazio, T., Casanellas, J., Chatterjee, S., Cordes, J., Demorest, P., Kramer, M., Lee, K., Liu, K., Ransom, S. et al. (2015), 'Observing radio pulsars in the galactic centre with the square kilometre array', *arXiv preprint arXiv:1501.00281* .

Eatough, R. P., , E. F. & Lyne, A. G. (2009), 'An interference removal technique for radio pulsar searches', *Monthly Notices of the Royal Astronomical Society* **395**(1), 410–415.

Fridman, P. (2000), 'Radio frequency interference rejection in radio astronomy receivers', *Astronomical and Astrophysical Transactions* **19**(3-4), 625–645.

Fridman, P. (2001), 'RFI excision using a higher order statistics analysis of the power spectrum', *Astronomy & Astrophysics* **368**(1), 369–376.

Fridman, P. (2008), 'Statistically stable estimates of variance in radio-astronomy observations as tools for radio-frequency interference mitigation', *The Astronomical Journal* **135**(5), 1810.

Fridman, P. & Baan, W. (2001), 'RFI mitigation methods in radio astronomy', *Astronomy & Astrophysics* **378**(1), 327–344.

Gantz, J. & Reinsel, D. (2011), 'Extracting value from chaos', *IDC iview* **1142**(2011), 1–12.

Grenier, I. & Harding, A. K. (2015), 'Gamma-ray pulsars: A gold mine', *Comptes Rendus Physique* **16**(6), 641 – 660. Gamma-ray astronomy / Astronomie des rayons gamma. **URL:** *http://www.sciencedirect.com/science/article/pii/S1631070515001486*

Hewish, A., Bell, S. J., Pilkington, J., Scott, P. F. & Collins, R. A. (1968), 'Observation of a rapidly pulsating radio source', *Nature* **217**(5130), 709–713.

Hey, T., Tansley, S., Tolle, K. M. et al. (2009), *The fourth paradigm: data-intensive scientific discovery*, Vol. 1, Microsoft research Redmond, WA.

Ifeachor, E. C. & Jervis, B. W. (2002), *Digital Signal Processing: A Practical Approach*, 2nd edn, Pearson Education, Harlow, England.

Jagadish, H., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R. & Shahabi, C. (2014), 'Big data and its technical challenges', *Communications of the ACM* **57**(7), 86–94.

Jouteux, S., Ramachandran, R., Stappers, B., Jonker, P. & Van Der Klis, M. (2002), 'Searching for pulsars in close circular binary systems', *Astronomy & Astrophysics* **384**(2), 532–544.

Karastergiou, A., Chennamangalam, J., Armour, W., Williams, C., Mort, B., Dulwich, F., Salvini, S., Magro, A., Roberts, S., Serylak, M. et al. (2015), 'Limits on Fast Radio Bursts at 145 MHz with ARTEMIS, a real-time software backend', *Monthly Notices of the Royal Astronomical Society* **452**(2), 1254–1262.

Keane, E., Barr, E., Jameson, A., Morello, V., Caleb, M., Bhandari, S., Petroff, E., Possenti, A., Burgay, M., Tiburzi, C. et al. (2017), 'The survey for pulsars and extragalactic radio bursts I: Survey description and overview', *arXiv preprint arXiv:1706.04459* .

Keane, E., Bhattacharyya, B., Kramer, M., Stappers, B., Bates, S., Burgay, M., Chatterjee, S., Champion, D., Eatough, R., Hessels, J. et al. (2014), 'A cosmic census of radio pulsars with the ska', *arXiv preprint arXiv:1501.00056* .

Keith, M. (2007), Ph.D. thesis,, PhD thesis, University of Manchester.

Keith, M., Jameson, A., Van Straten, W., Bailes, M., Johnston, S., Kramer, M., Possenti, A., Bates, S., Bhat, N., Burgay, M. et al. (2010), 'The high time resolution universe pulsar survey–i. system configuration and initial discoveries', *Monthly Notices of the Royal Astronomical Society* **409**(2), 619–627.

Kennea, J., Burrows, D., Kouveliotou, C., Palmer, D., Göğüş, E., Kaneko, Y., Evans, P., Degenaar, N., Reynolds, M., Miller, J. et al. (2013), 'SWIFT discovery of a new soft gamma repeater, SGR J1745–29, near SAGITTARIUS A', *The Astrophysical Journal Letters* **770**(2), L24.

Kitchin, C. R. (2013), *Data Processing*, Springer New York, New York, NY, pp. 197–212.

Kitchin, R. (2014), 'Big data, new epistemologies and paradigm shifts', *Big Data & Society* **1**(1), 2053951714528481.

Kramer, M. & Stappers, B. (2015), 'Pulsar science with the ska', *arXiv preprint arXiv:1507.04423* .

Laney, D. (2001), '3D data management: Controlling data volume, velocity and variety', *META Group Research Note* **6**, 70.

Lazarus, P., Brazier, A., Hessels, J., Karako-Argaman, C., Kaspi, V., Lynch, R., Madsen, E., Patel, C., Ransom, S., Scholz, P. et al. (2015), 'Arecibo pulsar survey using ALFA. IV. Mock spectrometer data analysis, survey sensitivity, and the discovery of 40 pulsars', *The Astrophysical Journal* **812**(1), 81.

Leskovec, J., Rajaraman, A. & Ullman, J. D. (2014), *Mining of massive datasets*, Cambridge university press.

Levin, L., Bailes, M., Barsdell, B., Bates, S., Bhat, N., Burgay, M., Burke-Spolaor, S., Champion, D., Coster, P., D'Amico, N. et al. (2013), 'The High Time Resolution Universe pulsar survey–VIII. the Galactic millisecond pulsar population', *Monthly Notices of the Royal Astronomical Society* **434**(2), 1387–1397.

Lorimer, D. (2001), Sigproc-v1. 0:(pulsar) signal processing programs, Technical report, Arecibo Technical Memo.

Lorimer, D., Faulkner, A., Lyne, A., Manchester, R., Kramer, M., McLaughlin, M., Hobbs, G., Possenti, A., Stairs, I., Camilo, F. et al. (2006), 'The Parkes Multibeam Pulsar Survey–VI. Discovery and timing of 142 pulsars and a Galactic population analysis', *Monthly Notices of the Royal Astronomical Society* **372**(2), 777–800.

Lorimer, D., Jessner, A., Seiradakis, J., Lyne, A., D'amico, N., Athanasopoulos, A., Xilouris, K., Kramer, M. & Wielebinski, R. (1998), 'A flexible format for exchanging pulsar data', *Astronomy and Astrophysics Supplement Series* **128**(3), 541–544.

Lorimer, D. R. (2011), Radio pulsar populations, *in* 'High-Energy Emission from Pulsars and their Systems', Springer, pp. 21–36.

Lorimer, D. R. & Kramer, M. (2005), *Handbook of pulsar astronomy*, Vol. 4, Cambridge University Press.

Lyne, A. & Graham-Smith, F. (2012), *Pulsar astronomy*, number 48, Cambridge University Press.

Lyon, R., Stappers, B., Cooper, S., Brooke, J. & Knowles, J. (2016), 'Fifty years of pulsar candidate selection: From simple filters to a new principled real-time classification approach', *Monthly Notices of the Royal Astronomical Society* **459**(1), 1104–1123.

Manchester, R. N., Hobbs, G. B., Teoh, A. & Hobbs, M. (2005), 'The Australia telescope national facility pulsar catalogue', *The Astronomical Journal* **129**(4), 1993.
URL: *http://stacks.iop.org/1538-3881/129/i=4/a=1993*

Mandelbrot, B. B. (1979), 'Fractals: form, chance and dimension.', *Fractals: form, chance and dimension., by Mandelbrot, BB. San Francisco (CA, USA): WH Freeman & Co., 16+ 365 p.* **1**.

Mandelbrot, B. B. (1983), *The fractal geometry of nature*, Vol. 1, Freeman, San Francisco.

Mandelbrot, B. B. & Van Ness, J. W. (1968), 'Fractional brownian motions, fractional noises and applications', *SIAM review* **10**(4), 422–437.

McConnell, D., McCulloch, P., Hamilton, P., Ables, J., Hall, P., Jacka, C. & Hunt, A. (1991), 'Radio pulsars in the magellanic clouds', *Monthly Notices of the Royal Astronomical Society* **249**(4), 654–657.

McMullin, J. P., Waters, B., Schiebel, D., Young, W. & Golap, K. (2007), CASA Architecture and Applications, *in* R. A. Shaw, F. Hill & D. J. Bell, eds, 'Astronomical Data Analysis Software and Systems XVI', Vol. 376 of *Astronomical Society of the Pacific Conference Series*, p. 127.

Mezger, P., Zylka, R., Philipp, S. & Launhardt, R. (1999), 'The nuclear bulge of the Galaxy. II. the K band luminosity function of the central 30 PC', *Astronomy and Astrophysics* **348**, 457–465.

Middleditch, J. & Kristian, J. (1984), 'A search for young, luminous optical pulsars in extragalactic supernova remnants', *The Astrophysical Journal* **279**, 157–161.

Miller, H. J. (2010), 'The data avalanche is here. shouldn't we be digging?', *Journal of Regional Science* **50**(1), 181–201.

Niamsuwan, N., Johnson, J. T. & Ellingson, S. W. (2005), 'Examination of a simple pulse-blanking technique for radio frequency interference mitigation', *Radio Science* **40**(5).

Nice, D., Fruchter, A. & Taylor, J. (1995), 'A search for fast pulsars along the Galactic plane', *The Astrophysical Journal* **449**, 156.

Nita, G. M., Gary, D. E., Liu, Z., Hurford, G. J. & White, S. M. (2007), 'Radio frequency interference excision using spectral-domain statistics', *Publications of the Astronomical Society of the Pacific* **119**(857), 805.

Offringa, A., de Bruyn, A., Biehl, M., Zaroubi, S., Bernardi, G. & Pandey, V. (2010), 'Post-correlation radio frequency interference classification methods', *Monthly Notices of the Royal Astronomical Society* **405**(1), 155–167.

Offringa, A., De Bruyn, A., Zaroubi, S., Koopmans, L., Wijnholds, S., Abdalla, F., Brouw, W., Ciardi, B., Iliev, I., Harker, G. et al. (2013), 'The brightness and spatial distributions of terrestrial radio sources', *Monthly Notices of the Royal Astronomical Society* **435**(1), 584–596.

Offringa, A., Van de Gronde, J. & Roerdink, J. (2012), 'A morphological algorithm for improving radio-frequency interference detection', *Astronomy & Astrophysics* **539**, A95.

Open Data Center Alliance (2012), 'Big data consumer guide'.
URL: *https://opendatacenteralliance.org/article/open-data-center-alliance-big-data-consumer-guide/*

Orginisation, S. (2015), *Advancing Astrophysics with the Square Kilometre Array*.

Pen, U.-L., Chang, T.-C., Hirata, C. M., Peterson, J. B., Roy, J., Gupta, Y., Odegova, J. & Sigurdson, K. (2009), 'The GMRT EoR experiment: limits on polarized sky brightness at 150 MHz', *Monthly Notices of the Royal Astronomical Society* **399**(1), 181–194.

Perlin, M. & Bustamante, M. D. (2016), 'A robust quantitative comparison criterion of two signals based on the Sobolev norm of their difference', *Journal of Engineering Mathematics* **101**(1), 115–124.

Press, W. H. (1978), 'Flicker noises in astronomy and elsewhere', *Comments on Astrophysics* **7**, 103–119.

Rane, A., Lorimer, D., Bates, S., McMann, N., McLaughlin, M. & Rajwade, K. (2016), 'A search for rotating radio transients and fast radio bursts in the Parkes high-latitude pulsar survey', *Monthly Notices of the Royal Astronomical Society* **455**(2), 2207–2215.

Ransom, S. (2008), 'The gbt 350 mhz surveys'.
URL: *http://www.lorentzcenter.nl/lc/web/2008/306/presentations/Ransom.pdf*

Ransom, S. (2011), 'PRESTO: PulsaR Exploration and Search TOolkit', Astrophysics Source Code Library.

Ransom, S. M. (2001), New search techniques for binary pulsars, PhD thesis, Harvard University Cambridge, Massachusetts.

Ransom, S. M., Eikenberry, S. S. & Middleditch, J. (2002), 'Fourier techniques for very long astrophysical time-series analysis', *The Astronomical Journal* **124**(3), 1788.
URL: *http://stacks.iop.org/1538-3881/124/i=3/a=1788*

Ransom, S. M., Stairs, I. H., Backer, D. C., Greenhill, L. J., Bassa, C. G., Hessels, J. W. & Kaspi, V. M. (2004), 'Green Bank Telescope discovery of two binary millisecond pulsars in the globular cluster M30', *The Astrophysical Journal* **604**(1), 328.

Rasmussen, C. E. (2006), Gaussian processes for machine learning, MIT Press.

Ray, P., Abdo, A., Parent, D., Bhattacharya, D., Bhattacharyya, B., Camilo, F., Cognard, I., Theureau, G., Ferrara, E., Harding, A. et al. (2012), 'Radio searches of Fermi LAT sources and blind search pulsars: the fermi pulsar search consortium', *arXiv preprint arXiv:1205.3089* .

Regulations, R. (2008), 'International telecommunication union', *Radiocommunication Sector. ITU-R. Geneva* .

Ridley, J., Crawford, F., Lorimer, D., Bailey, S., Madden, J., Anella, R. & Chennamangalam, J. (2013), 'Eight new radio pulsars in the Large Magellanic Cloud', *Monthly Notices of the Royal Astronomical Society* **433**(1), 138–146.

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N. & Aigrain, S. (2013), 'Gaussian processes for time-series modelling', *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* **371**(1984), 20110550.

Sanchita, G. & Anindita, D. (2016), *Evolutionary Algorithm Based Techniques to Handle Big Data*, Springer International Publishing, Cham, pp. 113–158.

Sclocco, A., Bal, H. E. & Van Nieuwpoort, R. V. (2015), Finding pulsars in real-time, *in* 'e-Science (e-Science), 2015 IEEE 11th International Conference on', IEEE, pp. 98–107.

Staelin, D. H. (1969), 'Fast folding algorithm for detection of periodic pulse trains', *Proceedings of the IEEE* **57**(4), 724–725.

Stovall, K., Lynch, R., Ransom, S., Archibald, A., Banaszak, S., Biwer, C., Boyles, J., Dartez, L., Day, D., Ford, A. et al. (2014), 'The green bank northern celestial cap pulsar survey. I. Survey description, data analysis, and initial results', *The Astrophysical Journal* **791**(1), 67.

Swiggum, J., Lorimer, D., McLaughlin, M., Bates, S., Champion, D., Ransom, S., Lazarus, P., Brazier, A., Hessels, J., Nice, D. J. et al. (2014), 'Arecibo pulsar survey using ALFA. III. Precursor survey and population synthesis', *The Astrophysical Journal* **787**(2), 137.

Taylor, A. R. (2012), 'The Square Kilometre Array', *Proceedings of the International Astronomical Union* **8**(S291), 337–341.

van Heerden, E., Karastergiou, A. & Roberts, S. (2016), 'A framework for assessing the performance of pulsar search pipelines', *Monthly Notices of the Royal Astronomical Society* **467**(2), 1661–1677.

van Heerden, E., Karastergiou, A., Roberts, S. & Smirnov, O. (2014), New approaches for the real-time detection of binary pulsars with the square kilometre array (SKA), *in* 'General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI', IEEE, pp. 1–4.

Walker, S. (2014), *Big data: A revolution that will transform how we live, work, and think*, Taylor & Francis.

Wilkinson, P. N. (1991), The hydrogen array, *in* 'International Astronomical Union Colloquium', Vol. 131, Cambridge Univ Press, pp. 428–432.

Winkel, B., Kerp, J. & Stanko, S. (2007), 'RFI detection by automated feature extraction and statistical analysis', *Astronomische Nachrichten* **328**(1), 68–79.

Wood, K., Norris, J., Hertz, P., Vaughan, B., Michelson, P., Mitsuda, K., Lewin, W., Van Paradijs, J., Penninx, W. & Van Der Klis, M. (1991), 'Searches for millisecond pulsations in low-mass X-ray binaries', *The Astrophysical Journal* **379**, 295–309.

Wu, X., Zhu, X., Wu, G. & Ding, W. (2014), 'Data mining with big data', *IEEE transactions on knowledge and data engineering* **26**(1), 97–107.

Zikopoulos, P., Eaton, C. et al. (2011), *Understanding big data: Analytics for enterprise class hadoop and streaming data*, McGraw-Hill Osborne Media.