



University
of Glasgow

Pearlstone, Brynley (2019) *Gravitational wave detector characterisation and transient searches*. PhD thesis.

<https://theses.gla.ac.uk/41205/>

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses

<https://theses.gla.ac.uk/>
research-enlighten@glasgow.ac.uk

DOCTORAL THESIS

Gravitational wave detector
characterisation and transient searches

Author:

Brynley PEARLSTONE

Supervisor:

Professor Graham WOAN

*Submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy*

Institute for Gravitational Research (IGR)

School of Physics and Astronomy

University of Glasgow



University
of Glasgow

April 17, 2019

Abstract

Recent gravitational wave observations have allowed for a new avenue through which to observe our universe. The detection of gravitational wave events has been possible by the advent of the advanced detector era and highly sensitive gravitational wave detectors, such as advanced LIGO and advanced Virgo. Spinning non-axisymmetric neutron stars present a promising source of gravitational waves, known as continuous waves. Such gravitational waves have yet to be detected, and are predicted to be extremely weak.

In order to increase the detectors' sensitivity to gravitational waves, they must be finely tuned to produce data without loud artefacts in either the time or frequency domains. The pursuit of identifying, characterising and removing these artefacts is known as detector characterisation. In the first portion of this thesis, we present a range of definitions, tools, and techniques that are used in detector characterisation.

I detail three case studies of work undertaken during the course of this research. These case studies draw on a broad knowledge base from staff scientists at both LIGO observatories, the detector characterisation community within the LIGO collaboration, and continuous wave search groups within the LIGO-Virgo collaborations.

First, I investigate a broadening of the 60 Hz noise line at the LIGO Livingston observatory during the final two observations of the initial LIGO detectors. Previous investigations by site staff linked short transients in magnetometer channels around the site with this line broadening. I examine this suspected coupling with more scrutiny, and fail to find an instantaneous coupling between broad magnetometer glitches and the broadening of the 60 Hz line. Secondly, we detail the creation of a list of known instrumental and environmental noise lines and combs in advanced LIGO's second observation. Thirdly, we present an investigation of a comb of noise lines prevalent in the low frequencies (≤ 100 Hz) of advanced LIGO's first observing run. In collaboration

with site staff, I identified the source of the comb, and the site staff took measures to reduce the contribution of the com to the strain channel.

In the second portion of the thesis, we focus on data analysis methods FOR “transient”-continuous waves. We examine several phenomena that can result in transient-continuous waves, and review a number of existing searches for transient-continuous waves. Then, we present a new method, reduced Bayesian blocks, built around the Bayesian blocks formalism, in order to determine whether a candidate continuous wave signal exhibits transient behaviour, and loosely constrain the beginning and ending times of any transient-continuous emission.

The methods involves coarsely chunking up gravitational wave observations into several chunks of equal time. These chunks are then recompiled into all allowable blocks, where a block is a series of contiguous chunks. In each block, the evidences that the block contains signal from a given source, or only Gaussian noise, are estimated. These evidences are then used to recompile a mixed-model description of the observation. We term each arrangement of signal- and noise-blocks as “intermittencies”. By comparing the posterior probabilities of all intermittencies in a given observation, we can determine whether an observation contains a transient-continuous signal from a given source.

We examine the performance of this new method, first on simple, short, simulated dataset then on a larger set of longer simulated signals. We see that the method produces results in line with our expectations. Then, we attempt to recover the injected intermittencies in the hardware injections of O2. Finally, we follow-up four outlying candidates from the first low-frequency all-sky search for continuous waves in advanced LIGO. In all four candidates, the reduced Bayesian blocks method supported the Gaussian noise model throughout the observation.

List of Abbreviations

GW	G ravitational W ave
CBC	C ompact B inary C oalescence
BH	B lack H ole
BBH	B inary B lack H ole
NS	N eutron S tar
BNS	B inary N eutron S tar
CW	C ontinuous W ave
SSB	S olar S ystem B arycentre
SNR	S ignal to N oise R atio
IFO	I nter F er O meter
LIGO	L aser I nterferometer G ravitational wave O bservatory
iLIGO	I nitial L IGO
aLIGO	A dvanced L IGO
LLO	L IGO L ivingston O bservatory
LHO	L IGO H anford O bservatory
adVirgo	A dvanced V irgo
DFT	D iscrete F ourier T ransform
SFT	S hort F ourier T ransform
ASD	A mplitude S pectral D ensity
PSD	P ower S pectral D ensity
CPSD	C ross P ower S pectral D ensity
PEM	P hysical E nvironment M onitor
ITM	I nput T est M ass
ETM	E nd T est M ass
VEA	V acuum E quipment A rea
LVEA	L aser and V acuum E quipment A rea
EBAY	E lectronics B AY
tCW	t ransient C ontinuous W ave
QPO	Q uasi- P eriodic O scillation
BB	B ayesian B locks
RBB	R educed B ayesian B locks
cSNR	c hunk S ignal to N oise R atio

Contents

List of Figures	xii
List of Tables	xv
1 Introduction: Gravitational waves and their sources	1
1.1 General relativity	1
1.1.1 The spacetime metric and gravitational waves	1
1.1.2 Gravitational wave interactions with matter	4
1.2 Sources of gravitational waves	6
1.2.1 Bayesian statistics	7
1.2.2 Transient sources	10
1.2.2.1 Compact binary coalescence (CBC) events	10
1.2.2.2 Supernovae	11
1.2.2.3 Unmodelled sources	12
1.2.3 Persistent sources	13
1.2.3.1 Stochastic gravitational wave background	13
1.3 Continuous gravitational waves	14
1.3.1 Continuous gravitational wave emission and morphology	14
1.3.1.1 Continuous wave morphology	15
1.3.1.2 Detecting continuous waves	16
1.3.2 Continuous wave search types	18
1.3.2.1 Targeted searches	18
1.3.2.2 Directed searches	22

1.3.2.3	All-sky searches	23
1.3.3	Follow-up techniques	24
1.3.4	Detection criteria	25
1.3.5	Detections of gravitational waves	27
1.4	Multimessenger astronomy	28
1.5	Summary	29
2	Gravitational wave detectors and data quality	30
2.1	Gravitational Wave Detectors	30
2.1.1	Resonant bar detectors	31
2.1.2	Interferometric gravitational wave (GW) detectors	31
2.1.3	Noise sources in interferometric gravitational wave detectors	32
2.1.4	First generation interferometric GW detectors	35
2.1.4.1	Initial LIGO	35
2.1.5	Second generation GW detectors	38
2.1.5.1	Advanced LIGO	38
2.1.6	Prospects for future detectors	41
2.2	Characterising aLIGO	43
2.2.1	Data in frequency and time-frequency domain representations	43
2.2.1.1	Data in the frequency domain	44
2.2.1.2	The time-frequency domain	46
2.2.2	Detector characterisation	47
2.2.3	Data quality	48
2.2.3.1	Lines	48
2.2.3.2	Combs	50
2.2.3.3	Wandering lines	51
2.2.3.4	Transient noise sources	51
2.2.4	Implications for long-duration searches	52
2.2.5	Sensors of the physical environment	53

2.2.6	Witness channels	55
2.2.7	Line and comb lists	56
2.2.8	Hardware mitigation	57
2.3	Summary	59
3	Detector Characterisation in S5, S6, O1 and O2	60
3.1	60Hz line broadening in S5 and S6	60
3.1.1	The Crab pulsar	61
3.1.2	The “Crab killer” noise	62
3.1.2.1	Previous investigations into the noise	63
3.1.3	Software mitigation of the Crab killer	65
3.1.4	Crab killer coupling in the time domain	68
3.1.5	Eventual mitigation of the Crab killer	74
3.2	Compiling a line list for aLIGO’s second observation	75
3.2.1	Motivation for creating a line list	77
3.2.1.1	Scale and scope of a line list	77
3.2.2	Discussion of O2	78
3.2.3	Methodology	79
3.2.3.1	Plotting the spectrum	80
3.2.3.2	Special cases: combs and violin modes	80
3.2.4	Results	81
3.3	Investigation of a 1Hz comb	82
3.3.1	Preliminary investigation of O1 data	83
3.3.2	Magnetometer coherence studies	83
3.3.3	aLIGO’s timing system: a candidate source	87
3.3.4	Testing a candidate coupling: Strain I/O chassis	90
3.3.5	Eventual suppression of the comb	92
3.4	Summary	92

4	Transient continuous gravitational waves – an introduction	94
4.1	Defining transient continuous waves	95
4.2	Sources of tCWs	95
4.2.1	Neutron star glitches	95
4.2.1.1	Defining a pulsar glitch	96
4.2.1.2	Glitches and continuous wave searches	97
4.2.2	Transiently accreting pulsars	100
4.2.2.1	Transiently accreting NSs and tCW detection	101
4.2.3	Magnetar burst quasi-periodic oscillations	102
4.2.4	Pseudo tCWs by spin wandering	103
4.2.5	Pseudo tCWs by template mismatch	104
4.3	Search methods for tCWs	106
4.3.1	Extensions of short-duration searches	107
4.3.1.1	STAMP search for magnetar bursts	107
4.3.1.2	Radiometer searches for very long-lived GW transients	108
4.3.2	Time-limited continuous GW searches	109
4.3.2.1	Transient \mathcal{F} -statistic	110
4.4	A tCW search set up	112
4.4.1	Prospects for tCW discovery	113
4.4.2	Best prospects for a search set up	113
4.4.3	Discussion of the transient timing parameters	114
4.4.4	A check on the persistence detection criteria	116
4.5	Summary	117
5	A tCW follow-up method: Reduced Bayesian blocks	118
5.1	Bayesian blocks	118
5.1.1	Bayesian blocks, a framework for tCWs	120
5.1.1.1	Defining the number of changepoints N_τ	120
5.1.1.2	Defining the time of a changepoint T_j	121

5.1.1.3	Defining a block B_j	122
5.1.1.4	The BB formalism	123
5.1.2	Bayesian blocks as tCW follow-up	125
5.1.2.1	Choice of models	125
5.1.2.2	Choice of priors for a tCW BB implementation	126
5.1.3	Advantages and disadvantages of BB as tCW follow-up	129
5.1.4	Building on the BB formalism	130
5.2	Reduced Bayesian Blocks	131
5.2.1	Intermittencies and data chunking	131
5.2.1.1	Number of possible blocks	133
5.2.2	Data models and likelihood estimations	134
5.2.2.1	Bayesian targeted heterodyne method	135
5.2.2.2	Signal log-likelihood estimation ($b(\alpha_i) = \blacksquare$)	135
5.2.2.3	Noise likelihood estimation ($b(\alpha_i) = \square$)	136
5.2.2.4	Model prior probabilities	137
5.2.3	Bringing it together: $P(\Gamma_p)$	137
5.2.4	Comparison with BB	138
5.2.4.1	Prior probabilities of Γ_p	140
5.2.5	Posterior probability of Γ_p	141
5.3	Algorithm output and the barcode plot	143
5.4	RBB: a summary	144
6	Results from RBB tests and candidate follow-up	148
6.1	RBB on simulated data	148
6.1.1	Description of simulated data	149
6.1.1.1	Gaussian noise generation	150
6.1.1.2	Signal parameters	150
6.1.1.3	Intermittency generation	151
6.2	Two chunk injections	152

6.2.0.1	Injected signals and intermittencies	153
6.2.0.2	Nested sampler realisations	154
6.2.0.3	Independent noise realisations	158
6.3	Eight chunk injections	160
6.3.1	Description of the simulated data	160
6.3.2	Defining recovery statistics	161
6.3.3	Describing the RBB configuration	162
6.3.4	Results: recovery of injected intermittencies	163
6.3.5	Findings from simulated datasets	163
6.3.5.1	Recovery fraction by cSNR	163
6.3.5.2	Posterior spread by cSNR	165
6.3.5.3	Ranked position spread by cSNR	168
6.4	RBB on hardware injections	173
6.4.1	Continuous wave hardware injections	173
6.4.1.1	Why have hardware injections?	174
6.4.1.2	Hardware injections in O2	175
6.4.1.3	Description of search set-up	176
6.4.2	Regime 1: Subset of O2, shorter chunks	178
6.4.2.1	Input intermittency	178
6.4.2.2	Recovery efficiency	178
6.4.3	Regime 2: All O2, long chunks	180
6.4.3.1	Input intermittency	181
6.4.3.2	Recovery efficiency	181
6.4.3.3	Is RBB behaving as it did with software injections?	182
6.5	RBB on O1 all-sky signal candidates	183
6.5.1	O1 all-sky search and outlying results	184
6.5.1.1	Describing the O1 low frequency All-sky search	184
6.5.1.2	Following up significant candidates	185

6.5.1.3	Four all-sky outliers	186
6.5.2	RBB configuration	187
6.5.3	Recovered intermittencies	188
7	Conclusions and future work	193
7.1	Conclusions	193
7.1.1	Data quality and detector characteristics	193
7.1.2	Transient continuous waves and reduced Bayesian blocks	196
7.2	Future work	198
A	Bayesian and RBB glossary	200
A.1	Bayesian terms: Glossary	200
A.2	RBB terms: Glossary	201
B	Lines list	203
C	Representative results for eight-chunk injections	207
D	Example .par and .priors file	212
D.1	Priors file	212
D.2	Par file	213
E	Python codes for RBB	214
E.1	do_splitting.py	214
E.2	timesplitter.py	217
E.3	write_analysis_args.py	220
E.4	create_analysis_subs.py	223
E.5	do_analysis.py	230
E.6	read_data.py	233
E.7	collate_evidences.py	235
E.8	RBB_summing_evidence.py	241

E.9	binary_structure.py	253
E.10	plot_functions.py	255

List of Figures

1.1	The effect of a GW on a ring of test particles	5
1.2	The antenna response of LIGO	17
2.1	The aLIGO noise curve.	35
2.2	A schematic of the iLIGO optical layout.	37
2.3	An aLIGO test mass and suspension system	39
2.4	A simplified schematic of the aLIGO optical layout.	40
2.5	An example of a spectrogram – a plot in the time-frequency domain . .	47
2.6	A line artefact in a time-frequency plot.	49
2.7	A site map of PEM sensors at LHO	54
3.1	The Crab killer noise around 60 Hz in S6	62
3.2	Coherence between a strain channel and magnetometer.	63
3.3	Spectrograms of the 60 Hz line at both iLIGO sites.	65
3.4	Results from the Crab killer glitch filter trial	68
3.5	Crab killer glitches in the magnetometer and strain channels.	69
3.6	Spectrograms of the Crab killer glitches in the strain and magnetometer channels.	70
3.7	Correlation between the strain and witness channel.	71
3.8	The CPSD during a Crab killer glitch, and at non-coincident times. . .	72
3.9	Crab killer glitch cross-power with the witness channel, and time-slides.	73
3.10	Crab killer glitch power and time-slide power ratios.	74
3.11	Crab killer noise, S56 versus O1	75

3.12	Lines and combs in the first week of O1.	76
3.13	Coherence of the 1 Hz comb with an EBAY witness channel	84
3.14	Coherence of the 1 Hz comb with a LVEA witness channel	84
3.15	Schematic layouts of the vertex and end stations EBAYs.	85
3.16	Combs founds in temporary magnetometer studies	86
3.17	Structure found in folded data studies of the PEM magnetometers. . .	87
3.18	An I/O chassis, the temporary magnetometer and a timing card LED. .	88
3.19	Combs found in the temporary magnetometer near the timing apparatus.	89
3.20	The first attempt to mitigate the comb's coupling.	91
4.1	A glitch causing a drop-out in the recovery of the signal.	97
4.2	A NS glitch causing a signal to be lost, or later recovered in a semi-coherent search.	99
4.3	An NS with spin-wandering causing intermittent recovery.	104
4.4	An intermittent recovery due to template mismatch.	106
5.1	An example application of Bayesian blocks	119
5.2	A barcode plot – the output of the RBB algorithm.	143
5.3	A flow-chart describing the processes involved in RBB.	146
6.1	Intermittency bit mismatch by injected cSNR.	164
6.2	Posterior probability of Γ_{rec} by cSNR.	166
6.3	$P(\Gamma_{\text{rec}})$ of correctly and incorrectly recovered intermittencies, by cSNR.	167
6.4	A 2D histogram of the rank of Γ_{inj}	169
6.5	A finer resolution 2D histogram of the rank of Γ_{inj}	170
6.6	The barcode plot of trial number 234 with cSNR = 5 and $\Gamma_{\text{inj}} = \Gamma_{13}$. . .	171
6.7	Two chunking regimes for the O2 hardware injections	177
6.8	The barcode plot for the tCW candidate J0709.	188
6.9	The barcode plot for the tCW candidate J1051.	189
6.10	The barcode plot for the tCW candidate J1901.	189

6.11 The barcode plot for the tCW candidate J2228.	190
--	-----

List of Tables

2.1	Seismic isolation technologies in GW detectors.	33
6.1	Mean and variance of the noise evidence in simple two-chunk tests . . .	155
6.2	log-Likelihood values for each intermittency in a two-chunk trial.	156
6.3	Mean and standard deviation of the nested sampler in two-chunk trials.	157
6.4	Mean and variance of the noise evidence over different noise realisations in two-chunk trials.	159
6.5	The Bayes factor of each block in the two-chunk trials.	159
6.6	Source parameters for 15 hardware injections in O2.	176
6.7	SNR and cSNR for 15 hardware injections in O2.	177
6.8	Splitting times for the final 96 days of O2.	179
6.9	Recovered intermittencies of the hardware injections in the first chunking regime.	180
6.10	Input intermittencies of the hardware injections in the second chunking regime.	182
6.11	Source parameters for four O1 all-sky outliers.	186
6.12	Chunk signal-to-noise Bayes factors for the candidates J0709 and J1051.	190
6.13	Chunk Bayes factors for the candidates J0709 and J1051.	191
B.2	Unvetted lines list from LLO, above 100 Hz in O2	206
C.1	Representative results from the eight-chunk simulations. Here I present 10 examples from each value of injected cSNR.	207

C.1	Representative results from the eight-chunk simulations. Here I present 10 examples from each value of injected cSNR.	208
C.1	Representative results from the eight-chunk simulations. Here I present 10 examples from each value of injected cSNR.	209
C.1	Representative results from the eight-chunk simulations. Here I present 10 examples from each value of injected cSNR.	210
C.1	Representative results from the eight-chunk simulations. Here I present 10 examples from each value of injected cSNR.	211

Declaration of work

I, Brynley Pearlstone, declare that the work presented in this thesis is my own. Figures and information derived from other sources are appropriately referenced.

In this work, I build on the work done by Robert Schofield, Anamaria Effler, Joshua Areeda and Joe Smith to characterise noise around the 60 Hz line at LIGO Livingston. This work was done with the advice of Graham Woan and Robert Schofield. With this advice, I coded a transient veto from the magnetometer channel, and further investigated the coupling between the magnetometer and strain channels.

Work on the O2 line list was done in collaboration with Evan Goetz, Pep Covas and Ansel Neunzert. This work was published in *Phys. Rev. D* as a part of a broader article ([155]). The work was divided roughly equally between the four of us. I searched for lines in the frequency range 100 Hz to 2000 Hz in the data from LIGO Livingston during O2. Ansel Neunzert covered the range 20 Hz to 100 Hz from the same observatory. Pep Covas and Evan Goetz searched LIGO Hanford at the same frequencies as myself and Ansel respectively.

In the work on the 1 Hz comb, I collaborated with Vincent Roma in investigating the comb. The investigation was supported by and guided by Robert Schofield. The studies of coherence between strain and magnetometer channels were carried out by myself and Vincent, as were the temporary magnetometer studies. The connection between the timing LED and the comb was suggested by Richard McCarthy, Verne Sanberg and Robert Schofield. The temporary power supply was a result of a conversation between myself and Robert Schofield, and was implemented by site staff. The firmware change

in the timing card LEDs was led by Ansel Neunzert and Daniel Sigg. This work was published as part of the same broad article [155] in Phys. Rev. D.

The work presented on the reduced Bayesian blocks method builds on the Bayesian blocks formalism. The method was designed by myself, under the guidance of Graham Woan and Matthew Pitkin. Matthew Pitkin wrote the nested sampling algorithm. I wrote code to chunk data, call the nested sampler in parallel, calls to read results, and process them to construct the reduced Bayesian blocks configuration. I also wrote the code to produce the barcode plots, and all other plots shown in chapter 6. This code is included as appendix E.

The short, two-chunk tests used for Bayesian blocks were suggested by Graham Woan. I produced the data for both the short two-chunk test and the larger, longer 8-chunk test, calling existing code to generate data both with and without signal, and using code of my own to compile these into larger datasets as desired.

The test of reduced Bayesian blocks against both the hardware injections and O1 candidate signals was suggested by Graham Woan. I performed the tests.

Chapter 1

Introduction: Gravitational waves and their sources

1.1 General relativity

General relativity (GR) [1] gives the most accurate theory of gravity to date, and predicts results which are consistent with experimental tests (for example [2–5]). This is true even in the strong-field regime [6–9]. GR suggests that gravitational attraction is not due to an instantaneous attractive force, but is as a result of the local curvature of spacetime. Further, as a result of non-Euclidean geometry in spacetime, GR can describe all manner of phenomena, including gravitational waves (GWs).

1.1.1 The spacetime metric and gravitational waves

In GR, it is the curvature of spacetime that influences the behaviour of matter [1]. Similarly, the matter in spacetime has an influence on its curvature. The curvature of spacetime is specified through the components of the metric tensor, $g_{\mu\nu}(\vec{x})$. In order to describe how this metric influences the matter within its spacetime, we must first define the Riemann curvature tensor, the Ricci tensor, the Ricci scalar and the stress-energy tensor. The Riemann curvature tensor, $R^{\rho}_{\sigma\mu\nu}$, can be calculated by combinations of

the metric coefficients and their first and second space and time derivatives [10]. The Ricci tensor $R_{\mu\nu} = R^{\alpha}_{\mu\nu\alpha}$, and the Ricci scalar $R = g^{\mu\nu}R_{\mu\nu}$, are contractions of the curvature tensor components with respect to pairs of its indices in combination with the metric tensor coefficients. The stress energy tensor $T_{\mu\nu}$ describes and governs the distribution and density of both the matter and energy within the spacetime. The Einstein equation

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi GT_{\mu\nu}}{c^4}, \quad (1.1)$$

allows us to solve for the metric, given different values in the stress energy tensor, and for different values in the Einstein tensor, $G_{\mu\nu}$.

The Λ term is the cosmological constant [11], and is important to our understanding of large-scale structure and cosmology. When solving the Einstein equation for gravitational waves from compact objects, the cosmological constant term can be neglected.

We can consider an empty spacetime to be locally flat. Such a region would be described by the Minkowski metric, $\eta_{\mu\nu}$,

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.2)$$

From any metric, we can recover the line element, which describes the proper distance of a line that is straight in a given spacetime, as

$$ds^2 = g_{\mu\nu}dx^\mu dx^\nu. \quad (1.3)$$

These line elements describe the intervals between two events in a spacetime.

We can consider other spacetime metrics, which themselves describe other phenomena allowable in by GR, as perturbations to the Minkowski metric, so long as the effects are relatively small.

$$g_{\mu\nu} = \eta_{\mu\nu} + \epsilon_{\mu\nu}. \quad (1.4)$$

One such perturbation is the plane wave solution, which gives rise to GWs. Here we will give the results of a plane wave solution, however, full derivations can be found readily in many textbooks such as [10] and [12]. We can consider a GW metric to arise from a small perturbation, $h_{\mu\nu}$ to the Minkowski metric $\eta_{\mu\nu}$.

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}. \quad (1.5)$$

If we consider a very weak perturbation ($|h_{\mu\nu}| \ll 1$), and expand it, keeping only terms linear in $h_{\mu\nu}$ and their derivatives, and shift to the Transverse Traceless gauge [10], we arrive at the GW perturbation metric as being given by

$$g_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ \exp[ic(z-t)] & h_\times \exp[ic(z-t)] & 0 \\ 0 & h_\times \exp[ic(z-t)] & -h_+ \exp[ic(z-t)] & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (1.6)$$

In this transverse traceless gauge, it is the coordinate velocity, dx/dt oscillates, meaning that light travel time between two points is altered by a passing GW.

In equation (1.6) above, the h_+ and h_\times components refer to the plus and cross polarisations of GWs respectively. When a GW has only the h_+ component, and $h_\times = 0$, we say it is plus-polarised, and similarly, a wave containing only h_\times components, and that has h_+ set to 0 is termed as cross-polarised.

1.1.2 Gravitational wave interactions with matter

As presented in equation (1.6), the GW metric is considered as a small perturbation to the flat space metric. To consider how the GW spacetime interacts with matter, we must consider the proper distance within this metric. Proper distance \mathcal{L} is defined as [10]

$$\mathcal{L} = \int_P \sqrt{g_{\mu\nu} dx^\mu dx^\nu}, \quad (1.7)$$

which is the square root of the line element defined in equation (1.3), integrated along a path. This can be explicitly expanded within a GW spacetime to

$$\mathcal{L} = \int_P \sqrt{-dt^2 + (1 + h_+)dx^2 + (1 - h_+)dy^2 + 2(1 + h_\times)dx dy + dz^2}. \quad (1.8)$$

The meaning behind equation (1.8) is the deformation to the local spacetime that a passing GW will produce. We can consider this by examining the effect that a gravitational wave, propagating along the z -axis, would have on a ring of freely falling test particles [10]. At first, we consider a scenario in which a plus-polarised GW passes through this ring. The ring of particles will be stretched by a factor of $\sqrt{1 + h_+}$ in the x -direction, and similarly compressed by a factor of $\sqrt{1 - h_+}$ along the y -direction. As the phase of the wave evolves, the stretching and compression of the ring of test particles also evolves, until the phase evolves by π radians, by which point, the stretching and compression components are exchanged. This effect is shown in the top row of test particles in figure 1.1

Similarly, a cross-polarised wave will transform the ring of test particles, stretching and squeezing diagonally along $x = y$ and $x = -y$, respectively. Again, after the phase evolves by π radians, the effects are swapped, so that the ring is stretched along $x = -y$, and compressed along $x = y$. This is shown in the second row of test particles in figure 1.1. It is these transformations that give the plus- and cross-polarisations their names.

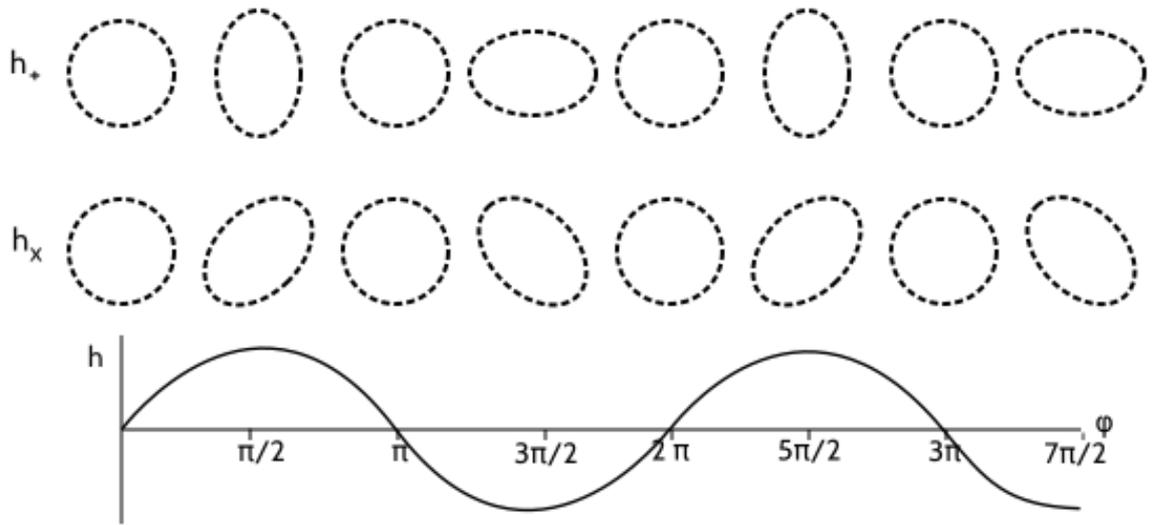


FIGURE 1.1: This cartoon shows how the plus and cross polarised gravitational waves deform a ring of freely falling test particles, and the amplitude (h) and phase (ϕ) of the associated gravitational wave.

These deformations are found by measuring the strain quantity, h , which is defined as the change in length of an object, ΔL , with respect to its length, L , such that $h = \Delta L/L$. Figure 1.1 illustrates the strain on circles of test particles. The top row of rings demonstrates the strain due to a plus-polarised GW, and the second row shows the strain due to a cross-polarised wave. The sinusoid below shows the strain amplitude, h , of the GW against its phase, ϕ at each step.

It should be noted that the effect of these GWs are notoriously weak. The cartoon in figure 1.1 shows these deformations exaggerated. The exact amplitude, waveform and polarisation of GWs depends on the parameters of the source that generated them. In section 1.2, we will discuss a number of these astrophysical sources, and the differences in the GWs that they generate.

1.2 Sources of gravitational waves

GWs in general permeate the universe across all frequencies. On a very large, low frequency scale, we can consider GWs from the early universe [13, 14], which have a wavelength in the order of Mpc, often due to the extreme redshifts they have experienced. Conversely, we could consider local gravitational waves, from anthropogenic sources. However, the spacetime metric is sufficiently stiff that such small perturbations are unlikely to yield gravitational waves carrying enough energy to be resolved by current gravitational wave detectors [15].

The amplitude of a GW is proportional to the second derivative of the mass quadrupole moment, $Q_{\mu\nu}$, with respect to time. The quadrupole moment of a body with density distribution $\rho(x_\mu)$ is given by [10]

$$Q_{\mu\nu} = \int \rho \left(x_\mu x_\nu - \frac{1}{3} |x^\mu x_\mu| \right) d^3x. \quad (1.9)$$

Approximately, the relationship between $Q_{\mu\nu}$ and $h_{\mu\nu}$ is given by the quadrupole approximation [15]:

$$h_{\mu\nu} \propto \frac{1}{d} \frac{d^2 Q_{\mu\nu}}{dt^2}, \quad (1.10)$$

with d being the distance between the source and the GW detector. There are a number of known phenomena that result in a relatively high GW amplitude and that are of astrophysical interest. These sources of order a few to tens of solar masses, and include neutron stars (NSs) and black holes (BHs). Similarly, supernovae are likely to contain enough energy to disturb the metric enough to produce short-lived and predictable GW strains [16, 17].

In this section, we will briefly discuss a number of sources that are important astrophysical targets for current ground-based GW detectors. We will apply a particular focus onto sources of continuous gravitational, as it will become a focus for this work after

chapter 4. However, prior to discussing sources, we must first discuss a key component in many of the detection methods, namely Bayesian statistics.

1.2.1 Bayesian statistics

Bayes' theorem is the basis of many statistical techniques used throughout GW analyses [18–20]. It is often used in the context of parameter estimation, wherein we may be attempting to estimate the range of possible values of a set of parameters, X given a dataset d . Bayes' theorem can be used to express the probability, or degree of belief that the values of X found by the analysis of d reflect the truth about those variables. Bayes' theorem states that

$$P(X|d, I) = \frac{P(X|I)P(d|X, I)}{P(d|I)}, \quad (1.11)$$

where:

I represents our information about the analysis, such as the signal model, particular details about the observation and detectors that may affect the analysis. This term is often dropped for ease of reading, and we will do the same here.

$P(X|d)$ gives the posterior probability, or degree of belief that the value of X found is true, given that it arose from an analysis of the data d .

$P(X)$ is our prior knowledge about X , which includes previously held information about X by independent observations, and mathematical boundaries (such as an angle θ is bounded by $[0, 2\pi]$). The range of a variable allowed by its prior is termed the “prior range”. For several variables, we refer to the n-dimensional prior range as a “prior volume”.

$P(d|X)$ describes the likelihood that the data d could be generated given the obtained value of X .

$P(d)$ is the evidence, often thought of as the degree of trust in the data. It is the probability that the data d could be obtained, regardless of the parameter X .

Often, the evidence is difficult to calculate, as it involves calculations such as

$$P(d) = \int_{-\infty}^{\infty} P(X)P(d|X)dX, \quad (1.12)$$

We can think of the posterior probability as the fractional area under the curve $P(X)P(d|X)$ for a given value of X . From this view point, we can see a tendency to return higher posterior probabilities for values of X drawn from a small (or sparsely populated discrete) range. This behaviour is known as the trials factor. It arises from the coarser or finer discretisation of a dataset. The evidence term can become impractical if there is no analytic solution, or the integrand has a high dimensional parameter space. As such, Bayes' theorem is often stated as a ratio of two posterior probabilities, which is known as a Bayes factor.

$$B_{1,2} = \frac{P(X_1|d)}{P(X_2|d)}, \quad (1.13)$$

$$= \frac{P(X_1)P(d|X_1)}{P(X_2)P(d|X_2)}. \quad (1.14)$$

The distribution of the prior probability of a variable can be key when trying to estimate its value. When little is known of about the variable, the least informative prior distribution is often the safest choice. The form of this safe choice depends highly on the characteristics of the parameter.

In addition, a model of the data is often dependent upon parameters over which we have no control or knowledge. These nuisance parameters can be removed from an analysis by marginalisation. We integrate the joint likelihood of our parameter of interest and nuisance parameter, so that all possible values of the nuisance parameter are calculated

and summed. As $\int P(a)da = 1$, this removes the nuisance parameter.

$$P(X|d) = \int_{-\infty}^{\infty} P(X, Y|d)dY. \quad (1.15)$$

Often, Bayes' theorem is used to discriminate between differing models of the data, which are inherently enfolded into our background information I . When choosing between two models, M_1 and M_2 , it can be worthwhile to look again at the Bayes factor,

$$B_{1,2} = \frac{P(M_1)P(d|M_1)}{P(M_2)P(d|M_2)}, \quad (1.16)$$

as calculating the evidence in a model selection problem similarly involves summing all possible discrete models,

$$P(d) = \sum_k P(M_k)P(d|M_k). \quad (1.17)$$

We can extend this framework to test our belief that a GW signal is found in an analysis. We purpose equation (1.16) towards discriminating between a GW model, M_{GW} , and a pure noise model, M_{N} . For any GW signal in the data, we require its amplitude h to be non-zero, so we can generally say that $P(d|M_{\text{GW}}) = P(d|0 < h \leq h_{\text{max}})$. Similarly, a model of the data which contains only noise must imply that $h = 0$. This leads to the signal-to-noise Bayes factor, $B_{\text{S,N}}$, which is

$$B_{\text{S,N}} = \frac{P(0 < h \leq h_{\text{max}})P(d|0 < h \leq h_{\text{max}})}{P(h = 0)P(d|h = 0)}. \quad (1.18)$$

This is the figure of merit for many searches, particularly when claiming a detection. This formalism is flexible to different data and signal models. As we will see in sections 1.2.2 and 1.3, signal morphologies and detection techniques vary, though many of the analyses share this underlying statistical practice.

Several new terms have been introduced in this section, and will be used throughout this work. A glossary of terms introduced here is included in appendix A for reference.

1.2.2 Transient sources

Transient GW events are those that produce short-lived signals in a GW detector, from less than a second up to a few minutes in length. In order to produce detectable strains over the short time scale, they are among the most powerful in the observable universe [21]. In part, this is a selection effect: among the short-lived events (which last between about 0.1 s and 100 s), only the most powerful can transfer sufficient energy into gravitational radiation, such that current GW detectors are sensitive enough to detect. Further, current ground-based GW detectors have their most sensitive frequencies between 10 Hz and 2000 Hz, which further limits the events which are of interest, given the state of current GW detectors. The events of interest are supernovae, binary neutron star (BNS) inspirals, binary black hole (BBH) inspirals, and neutron star - black hole (NSBH) inspirals [17].

1.2.2.1 Compact binary coalescence (CBC) events

A CBC event occurs when two massive, compact objects bound gravitationally together, orbiting each other, eventually collide. If the objects are both sufficiently compact, these events can produce GWs with high enough amplitude to be detected by current ground-based GW detectors in their final inspiral phase of their orbits.

There are three primary sources of CBC signals that we can expect to detect: BNS, NSBH, and BBH inspirals. Each event presents up to 15 different parameters. Some of these parameters pertain to the source system, including the unitless spin parameter of each body, the system's precession, and the chirp mass of the system

($\mathcal{M} = [m_1 m_2]^{3/5} / (m_1 + m_2)^{1/5}$ [22]), which is a function of the two bodies individual masses). Others are parameters relating to the observation, including the observed sky position and inferred distance to the source.

The waveform of CBC events can be calculated using post-Newtonian approximations of GR [23], but these calculations tend to be computationally expensive. In order to reduce overall computational cost, a set of such waveforms that best cover the parameter space are calculated, and used as templates in a matched filter search [24, 25]. In this method, the templates are compared to GW data at every time step. If the data and template are similar, then the time, data and matching waveform are flagged as a trigger. A trigger's sky position can be estimated by tracing rays between the detectors, using the arrival time of the trigger at each detector [24]. This allows for analysis to occur in real time, and very low latency follow-up of the trigger to occur. This sky location information, and information about source's masses can be fed forward to partners with electromagnetic observatories, in order to begin searching for a counterpart [26, 27].

1.2.2.2 Supernovae

We also expect to detect GWs from supernovae within the Milky Way [28]. In order for a supernova to create a GW, it must have spherical asymmetry, resulting in a non-zero $d^2 Q_{\mu\nu} / dt^2$ term. Further, these signals are short [29, 30], and the total GW energy radiated is far less than in a BNS signal. As a result, we only expect to be able to detect galactic supernovae, which also draws an upper limit on the detection rate of supernova events to the rate of galactic supernova – about once every 100 years [31].

There are a number of models to describe processes occurring during a supernova that are in tension with each other [29, 32, 33]. These tend to predict differing GW waveforms. As such, GWs may be key in discriminating between the models. As current GW detectors are only sensitive to supernovae within our own galaxy, we are

afforded the opportunity of a triggered search [34]. In such a search, an electromagnetic or neutrino observation of a galactic supernova would inform the start time and sky position of the supernova in a search. This type of feed-back search is not often afforded by other searches, where GW observations typically trigger an electromagnetic follow-up. As the waveforms are not well constrained, we use techniques similar to those in unmodelled searches in order to search for GWs from supernovae in the data.

1.2.2.3 Unmodelled sources

Additionally to the sources listed above, we may expect to detect gravitational waves from yet unknown or unmodelled sources [16, 35]. Such a search is attractive, as often new insights are found from unexpected discoveries.

Searches for short unmodelled transient GWs must be both robust against instrumental noise artefacts, and flexible so as not to throw out potentially interesting signals. A search robust against instrumental noise can be achieved by requiring such a signal to be present in two independent detectors, with a time delay less than $\Delta t = (\Delta x/c) \cos \theta$, where the two detectors are separated by a straight line of length Δx , with the source at an angle θ from this line. The time delay is a result of the time-of-flight time of the GW between the two sites. The longest allowable time delay is between two detectors is $\Delta t = \Delta x/c$, and any signal with a time delay longer than this can be ruled out, as it must be moving slower than c . We can also impose that an astrophysical signal must not be present in any auxiliary channels that are witness to instrumental noise sources. See section 2.2.6 for a more in-depth discussion of witness channels.

An unmodelled search by definition does not have a template, so the same matched filter searches cannot be used to search for these signals. Instead, an effective search method is to search for excess power, correlated in time between detectors [35].

Notable among the unmodelled burst algorithms is the “coherent WaveBurst” method [36]. This low-latency pipeline was the first to detect GWs. Additionally, the “STAMP” method [37] is notable in its broad applicability to both shorter and longer signals.

1.2.3 Persistent sources

We also expect GWs that last much longer in time than those discussed in section 1.2.2. These GWs are expected to last longer than any observation, such that they are functionally persistent. However, due to the effective stiffness of spacetime, we expect such signals to have a comparatively low amount of power in GWs when compared to the transient sources discussed above [16]. As such, detection of these weaker sources will require their signals to be integrated over a much longer time period. Ultimately, this means that it is often more efficient to search for these signals in the frequency and time-frequency domains, rather than the time domain.

1.2.3.1 Stochastic gravitational wave background

The stochastic GW background refers to the unresolvable transient GW sources, which form a background noise [16, 38]. This background, viewed as a single, persistent emergent signal is itself detectable. It is likely to become a source of obfuscating noise in the future, for more sensitive detectors.

The stochastic GW background is broadband, and exists across all frequency scales. This provides the opportunity for detection by many means – by pulsar timing arrays, ground-based GW detectors and even in imprints upon the cosmic microwave background [39]. Ground-based GW detectors can hope to detect this background, and have placed firm upper limits on its amplitude [40].

The core method for stochastic detection pipelines is cross-correlation. This method essentially looks for correlation in the data in the frequency and time-frequency domains.

To this end, the first generation of LIGO GW detectors included a third detector, H2, that was co-located with LIGO Hanford observatory (LHO). The two detectors shared the same local environment, position on Earth, and had similar sensitivity to GWs, and importantly, the noise in both detectors is correlated. As such, cross-correlating the data between the two detectors at LHO would be particularly illuminating [41]. Cross-correlating data between two detectors that are not co-located requires deconvolving the antenna pattern of each detector from the data, and taking special care mitigate local noise in both locations. The co-located were able to improve upon the upper limits set on the stochastic GW background in this search.

1.3 Continuous gravitational waves

1.3.1 Continuous gravitational wave emission and morphology

One source of persistent GWs is thought to be found in rotating NSs. By considering the properties of such a source, we can model its GW emission. In order to emit GWs, it must have an accelerating quadrupole moment, thus it must have spherical asymmetry about its axis of rotation. Usually, we consider a triaxial NS as this is the simplest formulation which can create GWs [42]. Further, we can impose that this triaxial NS does not precess, partly as there is little evidence in observations that NSs precess, and in part as the precessing behaviour would be quickly damped by the NS's rotation [43].

A rotating NS which emits a beam of electromagnetic radiation, and which rotates in such a way that it sweeps over an observer is known as a pulsar. Observations of pulsars have shown that their rotation rate, f_{rot} , slows with time [44, 45], though there is no consensus as to what drives it [46]. This behaviour is often called “spin-down”, and is given as \dot{f}_{rot} . Though this amount is usually very small (between about 10^{-15} Hz/s and 10^{-10} Hz/s), if it is neglected, it can lead to a large mismatch between expected frequency and observed frequency over the course of a long observation. When trying

to model its waveform, it is also important to consider higher orders of the frequency derivative with time, $(\ddot{f}_{\text{rot}}, \ddot{\dot{f}}_{\text{rot}} \dots)$, which for a known pulsar, can be obtained by fitting values to observations of the rotation rate [47].

1.3.1.1 Continuous wave morphology

Triaxial stars have three orthogonal axes of rotational symmetry, each with symmetry with rotations of π^c . As such, it is associated with a GW emission f_{GW} at twice its rotation frequency, such that $f_{\text{GW}} = 2f_{\text{rot}}$. Similarly, the time derivatives of the rotation frequency relate to the same derivatives of the GW frequency as $\frac{d^n f_{\text{GW}}}{dt^n} = 2 \frac{d^n f_{\text{rot}}}{dt^n}$.

We can expect the GWs from such a rotating non-precessing triaxial NS to have the form [47]

$$h_+(t) = h_0(1 + \cos^2 \iota) \cos \Phi(t, \vec{n}_0), \quad (1.19)$$

$$h_\times(t) = h_0 \cos \iota \sin \Phi(t, \vec{n}_0, \vec{r}_d), \quad (1.20)$$

where the parameter ι is the angle between the source's rotational axis and the line of sight from the detector to the source. Φ gives the phase evolution of the source in a time frame, t , at the solar system barycentre (SSB). The phase Φ is given as [47]

$$\Phi = \phi_0 + 2\pi \sum_{k=0}^s f_0 \frac{t^{k+1}}{(k+1)!} + \frac{2\pi}{c} \vec{n}_0 \cdot \vec{r}_d(t) \sum_{k=0}^s f_0 \frac{t^k}{k!}, \quad (1.21)$$

where f_0 denotes the instantaneous frequency at $t = 0$ at the SSB. Similarly, ϕ_0 denotes the instantaneous phase of the continuous wave (CW) at a reference time ($t = 0$) at the SSB. The k^{th} time derivative of f_0 is denoted by $f_0^{(k)}$. The vector quantity \vec{n}_0 is the constant unit vector in the direction of the source in the frame of the SSB, and \vec{r}_d is the position vector of the GW detector in that same frame [47].

The parameter h_0 denotes the amplitude of the GW received at the detector. It is determined by the NS's intrinsic physical parameters, such as its moment of inertia about its axis of rotation I_{zz} , and its distance d from the detector. For a maximally-oriented NS (ie. one whose orientation maximises the value of h_0), the exact form is given by

$$h_0 = \frac{4\pi^2 G I_{zz} f^2}{c^4 d} \epsilon, \quad (1.22)$$

where its ellipticity, ϵ , is given by

$$\epsilon = \frac{|I_{xx} - I_{yy}|}{I_{zz}}. \quad (1.23)$$

1.3.1.2 Detecting continuous waves

Current ground-based detectors are laser interferometers, which have an antenna response pattern with broad acceptance lobes. The response pattern of the detectors is different for plus- and cross-polarised waves. These response functions, F_+ and F_\times are given by given by [47]

$$F_+ = \sin \zeta [a(t) \cos 2\psi + b(t) \sin 2\psi], \quad (1.24)$$

$$F_\times = \sin \zeta [b(t) \cos 2\psi - a(t) \sin 2\psi] \quad (1.25)$$

ζ gives the angle between the two arms of an interferometer (IFO), which for most current interferometric detectors is $\pi/2$ (the exception being GEO600, whose internal angle is 95.33° [47]). The polarisation angle ψ dictates the contributions of the plus- and cross-polarisations in the GW. By convention, ψ is given as the angle (clockwise about the wave propagation direction \vec{k}) from $\vec{k} \times \vec{z}$ to $\vec{k} \times \vec{s}$, where \vec{z} is the North Celestial Pole, and \vec{s} is the direction of rotation of the source[49].

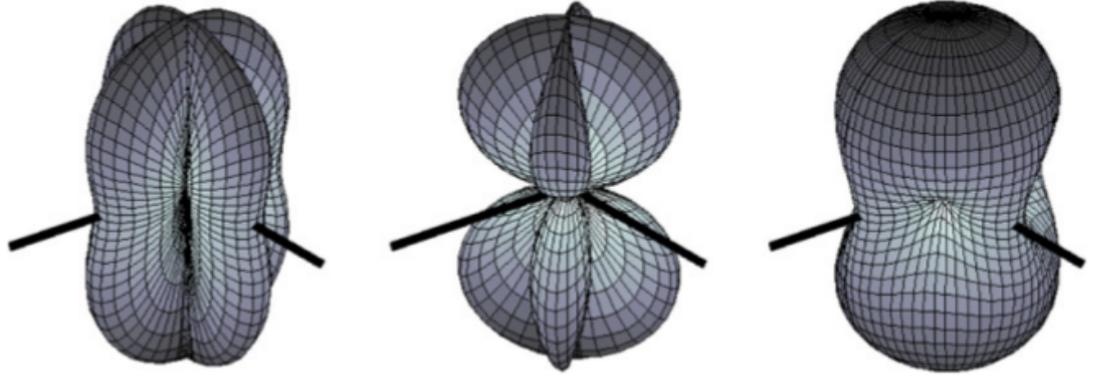


FIGURE 1.2: The antenna response of a LIGO detector to $+$ -polarised (left), \times -polarised (centre) and unpolarised GWs [48]. The vertex of the detector is at the origin, The thick black lines indicate the orientation of the IFO arms. The distance from the origin to the surface of the shape indicates the amplitude of the response function.

These antenna response patterns can be plotted as 3D shapes, as in figure 1.2, where the distance from the origin in any direction represents the amplitude of the response.

$$h(t) = h_+(t) \cos^2 2\psi + h_\times(t) \sin^2 2\psi \quad (1.26)$$

ζ gives the angle between the two arms of an IFO, which for most current interferometric detectors is $\pi/2$. The parameters $a(t)$ and $b(t)$ are a combination of the sky position of the source (α and δ), the position of the detector on Earth (L and λ) and the orientation of its arms (γ). The exact form of the parameters $a(t)$ and $b(t)$ are given in [47]. These response functions couple with the GW waveforms as

$$h(t) = F_+(t)h_+(t) + F_\times(t)h_\times(t). \quad (1.27)$$

The response pattern is fixed relative to the detector, however, as the detector rotates on the surface of Earth once per sidereal day, and around the solar system barycentre once per year, the response function varies with time. Similarly, sources in different sky positions will result in different detector response evolutions. The effect of the detector response is to modulate an incoming CW, as shown in equation (1.27).

Altogether, this presents a search with a wide array of parameters to search over: sky position (α and δ) and source GW parameters ($f_{\text{GW}}, \dot{f}_{\text{GW}}, \ddot{f}_{\text{GW}} \dots, \cos \iota, \phi_0, \psi, h_0$).

1.3.2 Continuous wave search types

Unlike transient searches, which can be run “online” – as the data are generated [50] – CW searches usually start only after the data for an entire observation has finished, and the data are well understood (for a full discussion on data quality, see chapter 2).

In order to claim a detection of CWs, we approach the problem from many angles. In this section, we discuss the three principle search tactics. We will begin with precise targeted searches, in which we examine the expected signals for known pulsars. Next we will discuss the directed searches, in which we suspect a CW source is emitting, but have looser constraints on the source parameters. Finally, we will discuss the all-sky searches, broad searches that look for signals for which there are not yet electromagnetic counterparts, or necessarily any evidence for a GW signal. Finally, we will discuss the candidate vetting processes and detection criteria which append each of the three search strategies.

1.3.2.1 Targeted searches

Targeted searches are CW detection pipelines whose focus is on known pulsars, for which many parameters are well constrained by independent electromagnetic counterpart observations. These searches must find the GW amplitude parameters $\lambda = \{h_0, \phi_0, \psi, \cos \iota\}$, which cannot be determined electromagnetically [47].

As targeted searches for CWs have a small parameter space, these searches have a much lower trials factor than a broader search, making them more sensitive to GWs from a single source. There are a number of ways in which targeted searches can be performed [18, 51], however, in this section, we will focus on only one: the Bayesian heterodyne

search [18]. In this section, we will outline the search, and later, in chapter 5 we build upon this search method.

The Bayesian heterodyne search is made of 4 stages: heterodyne stage, downsampling stage, applications of a low-pass filter, and a nested sampler search. We can consider the first three of these stages a pre-processing procedure, to prepare the data for the nested sampling search. To begin, we can consider that the data $d(t)$ consists of the sum of an unmodelled noise contribution, $N(t)$, and a GW signal, which we take to be given by equation (1.27), such that

$$d(t) = N(t) + h(t). \quad (1.28)$$

Here, we contract $A_+ = h_0(1 + \cos^2 \iota)$ and $A_\times = h_0 \cos \iota$ so as to focus on the phase evolution of the GW. As such, the data can be represented as

$$d(t) = N(t) + F_+(t)A_+ \cos \Phi(t, \vec{n}_0) + F_\times(t)A_\times \sin \Phi(t, \vec{n}_0). \quad (1.29)$$

We can express the sine and cosine functions in their exponential forms,

$$d(t) = N(t) + \left(F_+(t)A_+ \frac{e^{i\Phi(t)} + e^{-i\Phi(t)}}{2} \right) + \left(F_\times(t)A_\times \frac{e^{i\Phi(t)} - e^{-i\Phi(t)}}{2i} \right). \quad (1.30)$$

Here, we have dropped Φ 's dependance of \vec{n}_0 for ease of reading. We can expand each instance of $\Phi(t)$ to $\phi_0 + \phi(t)$, as in equation (1.21).

The heterodyne is performed by multiplying the data by an evolving phase, matching the CW evolving phase term $\phi(t)$

$$d'(t) = d(t) \exp -i\phi(t) \quad (1.31)$$

Examining only the plus-polarised term, we can see that this results into two phase components.

$$F_+ h_+ e^{-i\phi(t)} = F_+ A_+ \frac{e^{i\phi_0} e^{i\phi(t)} + e^{-i\phi_0} e^{-i\phi(t)}}{2} \times e^{-i\phi(t)}, \quad (1.32)$$

$$= F_+ A_+ \frac{e^{i\phi_0} e^0 + e^{-i\phi_0} e^{-2i\phi(t)}}{2}. \quad (1.33)$$

The same splitting occurs in the cross-polarised component. In the fraction in equation (1.33), the first term in the numerator gives a DC offset, and the second term gives a component evolving at twice the source frequency. As the noise has also been multiplied by the evolving phase, it has also been shifted by $\phi(t)$.

In order to remove much of the noise, as well as the high-frequency component evolving at $2f$, we apply a ninth-order Butterworth low-pass filter, with knee frequency $f_{\text{knee}} = 0.25$ Hz. This is an all-pole filter with nine poles. Each pole is distributed around a unit circle of radius $\pi/2$ filter has a response function $G(f)$ that scales as $G(f) \propto 1/(\frac{f}{f_{\text{knee}}})^{18}$.

Finally, the data $d'(t)$ are re-sampled to a rate of 1/60 Hz, by averaging the data within each minute. The result of this are the pre-processed and heavily down-sampled data, which we call B_k .

This pre-processed data can be analysed using a nested sampling parameter estimation script, `lalapps_pulsar_parameter_estimation_nested` [52], which is included as a part of the `lalapps` software package. In order to find the set of parameters with the highest likelihood, the nested sampler begins by drawing a number of samples. This number, N_{live} , is the constant number of samples considered by the method at all times. Each sample is a vector, which contains a value for each parameter, drawn independently from that parameters' prior distribution. For each sample, the likelihood is calculated, and the sample with the lowest likelihood is discarded. To replace the discarded sample, a new sample is drawn. In order to draw new samples,

one of a several sample proposal methods (including an ensemble walk [53] and an independent draw from the parameter priors) is used [54, 55]. By this method, the samples tend towards the maximum likelihood. A full treatment of this package can be found in [52], though here will will discuss only its core functionality.

This pipeline calculates the log-quotient of the evidence for signal and the evidence for noise ($\ln(B_{S,N})$) given the data presented. To find this, we must first calculate the log-likelihood that there is either signal in the data B_k , or only noise. These log-likelihoods are given by [52]

$$\ln(P(d(t)|\phi(t), I)) = \begin{cases} \sum_{v=1}^C (\mathcal{A}_v - m_v \ln [\sum_{n=n_0}^{n_0+m_v-1} |B_{k,n}(t)|^2]), & \text{if } h_0 = 0, \\ \sum_{v=1}^C (\mathcal{A}_v - m_v \ln [\sum_{n=n_0}^{n_0+m_v-1} |(B_{k,n}(t) - h(\phi(t)_v)|^2]), & \text{if } h_0 \neq 0. \end{cases} \quad (1.34)$$

These are calculated by performing a nested sampling routine over the parameter space described by the prior probabilities assigned to the parameters folded into $\phi(t, \rho)$ and $h(t)$. Equation (1.34) describes how the total likelihood are calculated. First, the data are chunked up into C chunks of length m_v , as the noise within each shorter chunk is presumed to be stationary. Within each chunk, the internal calculation is performed for each case. In this internal calculation, the running index n runs from the beginning of a chunk $n_0 = \sum_{l=1}^v 1 + m_l - 1$ up until the block's end, $n_0 + m_v - 1$, \mathcal{A}_v is given as

$$\mathcal{A}_v = \ln([v - 1]!) - \ln 2 - m_v \ln \pi, \quad (1.35)$$

and

$$\ln(B_{S,N}) = \ln(P(d(t)|\phi(t), h_0 \neq 0, I)) - \ln(P(d(t)|\phi(t), h_0 = 0, I)), \quad (1.36)$$

where $B_{s,n}$ is again the signal-to noise Bayes factor. The contributions from each chunk are then multiplied together (equivalent to summing in log space). During this part of the routine, the nested sampling algorithm explores the parameter space by drawing

samples as described in [54, 55]. The third part of the pipeline examines the samples drawn from the calculated evidences in order to estimate the amplitude parameters.

Importantly, there are other targeted search methods, which similarly utilize the modelled waveform to best recover a signal from the data.

As CW searches tend to occur only after the data are generated, we can update the phase evolution model with observational data throughout the GW observation. This leads to very accurate models for the signal throughout an observation, reducing uncertainty in the reconstructed parameters. These accurate templates are often vital to the search as targeted methods are typically very narrow-band searches, over only a small prior volume. However, a small mismatch between the signal model and a GW brings only slight detriment to search prospects [56]. As a result of the narrow search, for the same number of core processing unit hours (CPU-hours), a targeted search is more sensitive than a directed or all-sky search method. Similarly, a targeted search for a single source can be run more quickly than a directed search, since the targeted search has a lower prior volume to search.

1.3.2.2 Directed searches

In a directed search, some (but not all) of the source parameters are known. This includes searches for CWs from the galactic core [57], and searches for CWs from Scorpius X-1, a NS accreting its companion star [58, 59].

Generally speaking, the directed search algorithms used do not differ much from those utilised in all-sky searches, outlined in the next section. What is different is that each method is specially tuned to its suspected source [60], which is not possible in a blind, all-sky search.

Having a larger parameter space to search than a targeted search tends to lead to less sensitive results. However, the flexibility of a directed search increases the number of

possible candidates greatly.

1.3.2.3 All-sky searches

An all-sky search for CWs is a very wide search over the entire sky for CWs from unobserved sources. Arguably, the first detected CWs are likely to come from a previously unknown source. We can expect this as there are expected to be between 10^8 to 10^9 NSs in the galaxy [61], of which only about 10^3 have been observed to date [62]. Any of these 10^8 NS could be emitting GWs. Of these, some are likely to be young, and a few of the younger NSs are likely to be close to Earth. In order to detect these unknown NSs, an all-sky search tactic was devised, which searches over the entire sky for a CW signal [16].

An all-sky search for CWs is a computationally expensive problem, due to the size of the parameter space to search over. To facilitate the enormous computing cost, the Einstein@Home (E@H) distributed computing infrastructure was created [63]. E@H uses the idle time of volunteered computers to perform parallel analyses of signal templates against the GW data. There have been many all-sky searches performed using E@H [64–69].

In order to distribute the computing cost, E@H first splits the observation into several smaller segments of several hours each, and each of these time segments into frequency bands of 0.5 Hz width. Each computing node receives a banded segment of data to analyse. The parameter space for the signal models is also subdivided into bins, to be distributed among the computing nodes. Once divided up, each node calculates the \mathcal{F} -statistic over its template-parameter space for its banded segment [47, 70]. The \mathcal{F} -statistic is a coherent CW analysis statistic which can be analytically computed. It is a function of the maximal likelihood ratio with respect to h_0, ϕ_0, ψ and $\cos \iota$. A candidate whose \mathcal{F} -statistic passes a threshold can be considered for detection [47].

For each set of parameters, the segments are re-compiled, to produce a semi-coherent \mathcal{F} -statistic for a given model over an entire observation. The signal templates are ranked by their values of $2\mathcal{F}$, and candidate CW sources are generated by examining the distribution of these $2\mathcal{F}$ values [47]. The parameters for each of these possible candidates are then fed forward into the several stages of follow-up procedure, which are outlined below in 1.3.3.

Not every all-sky search method relies on the distributed computing of E@H. Several others, including TwoSpect [71], PowerFlux, Hough, and StackSlide methods [72, 73], have been used to generate results independent of E@H. However, these tend to use shorter coherence times, resulting in less sensitive searches. As a result, these searches rely more heavily on a hierarchical approach, in which each subsequent analysis is more stringent than the last.

1.3.3 Follow-up techniques

Following up candidate signals is key to the observation of CWs and their sources, as it provides a series of robust and independent analyses to check the same templates. The follow-up procedure usually falls into two stages [64]: search refinement and re-analysis. These two stages are repeated a number of times until the number of remaining candidates is sufficiently small for follow up by hand [67, 74]

In the first part of the follow-up procedure, the parameter space around the candidate is updated. This involves reducing the searched space by narrowing the window around the candidate, and refining the parameter space to allow for a finer search. Along with this, the coherence time used for the subsequent analysis is also often increased to increase sensitivity. The exact way in which the parameter space's grid is refined is often dependent upon the source parameters themselves, the details of the observation and the method used for the re-analysis section [75].

Secondly, the data are re-analysed in this smaller parameter space. This is likely to be more sensitive, partly due to the decreased trials factor associated with a narrowed search window, partly due to having more precise templates and thus less mismatch between the search template and the true signal in the data (termed template mismatch) [56], and partly due to having increased coherence time [64]. Any one of the analysis pipelines previously mentioned could be used for re-analysis of a candidate signal.

Once the re-analysis is complete, if the candidate no longer shows evidence for a CW, then the candidate is discarded. Otherwise, parameter estimation is performed again to more precisely constrain the parameters of the candidate, so that in the next iteration of follow-up, the parameter space can be further narrowed.

With each successive iteration of follow-up, the search grid becomes finer, the parameter spaces searched become narrower, fewer candidates survive, and the surviving candidates are more well constrained. A typical follow-up procedure may take three or four rounds of follow-up search, which will often reduce the number of viable candidates from around 10^4 to 10 [64, 74]. At this point, the surviving candidates can be checked by hand for coincidence with known instrumental lines (see section 2.2.3.1) or injected signals. Such coincident signals are discarded, and any remaining signals remain as outliers, which must meet a set of criteria for detection, or be similarly discarded.

1.3.4 Detection criteria

A candidate CW, whether from an all-sky, directed or targeted search, having survived several rounds of follow-up is not necessarily a detection unless it conforms to a number of other criteria which we can expect a CW to conform to:

These criteria include:

- match a signal model given its source parameters,
- have constant amplitude h_0 ,

- be continuous,
- be coherent over long analyses,
- be coherent between independent detectors.

Much of the hardest work here is necessarily performed by the analysis pipelines. The pipelines and follow-up procedures are designed to accept signals which match a given signal model [47], which generally assume a constant value of h_0 (once the antenna response has been accounted for) and a continuous signal throughout the observation.

However, there is a further statistic which we can look to: the measured SNR of the signal. We expect background noise in a GW detector to be Gaussian-distributed, so the noise contribution to the SNR sums proportionally to $\sqrt{T_{\text{obs}}}$. The signal for a CW is coherent, and so the signal contribution to the SNR sums proportionally to T_{obs} . This leads to an overall measured SNR that scales with $\sqrt{T_{\text{obs}}}$.

In the case of a possible detection, we expect that the majority of the source parameters would be well constrained, allowing for the optimal SNR to be calculated for any values of T_{obs} or h_0 .

Similarly, the final analysis can be run a number of times, each calculating the measured SNR value found. These different instances would all start at the beginning of the observation, T_0 , however, the amount of data to be analysed would be varied on each repeat, so as to find the measured SNR against observing time. This measured SNR should trace the optimal SNR calculated through different values of T_{obs} .

Such a check puts the second, third and fourth detection criteria under scrutiny. Should a candidate CW be coherent, we expect the measured SNR to rise with $\sqrt{T_{\text{obs}}}$. Should a candidate have a value of h_0 which varies with time, we would expect the measured SNR to evolve as $h_0(t)\sqrt{t}$. Conversely, a candidate with constant h_0 , as is expected, is predicted to evolve with $h_0\sqrt{t}$. Should a candidate detection not be truly continuous,

we would expect the measured SNR to rise with \sqrt{t} whilst a CW signal is present, and to fall when absent.

To date, there have not been any detections of CWs, however there have been upper limits set on the strain amplitudes of many known pulsars [76]. Now, in the age of advanced GW detectors, detection prospects are higher than any time in history [77].

1.3.5 Detections of gravitational waves

After two observing runs with Advanced LIGO (aLIGO) [78] (O1 and O2), there have been six detections of GW events [79–84]. Of these six, five have been BBH events, and one was a BNS event.

The first direct detection of GWs occurred on September 14th 2015 [79]. The detected system was a BBH event, with the progenitor masses being $36 M_{\odot}$ and $29 M_{\odot}$, and the remnant BH hole having a mass of $62 M_{\odot}$ [85]. One further event from aLIGO's first observing run was detected, GW151226 [80], another BBH with significantly lower mass BH progenitors. One further high significance trigger occurred, LVT151012 [86], however the event was not sufficiently significant to claim a detection.

aLIGO's second observing run yielded three more BBH events (GW170104 [81], GW170608 [82] and GW170814 [83]), with the latter being significant in that its three-detector observation (in conjunction with the advanced Virgo(adVirgo) detector) [87] led to a vastly improved localisation than was previously possible. O2 also brought about the first GW detection of a BNS system [84].

The first BNS observation allowed for several science outcomes, including the confirmation the confirmation of a kilonova associated with a BNS event [88]. Detections of the kilonova led to the observations of heavy metal production in a BNS event [89]. The precise localisation of the BNS allowed for an independent measurement of the Hubble

constant [90]. The lack of an associated neutrino burst was found to be consistent with model predictions [91].

1.4 Multimessenger astronomy

Of the sources discussed in sections 1.2 and 1.3, most involve a counterpart observation by another astrophysical messenger, be that electromagnetic or neutrino [92, 93]. Targeted CW searches necessarily rely upon pulsar observations to build a signal model [47], galactic supernovae would be bright in the sky for a few days, visible even to the naked eye, and certainly a high priority target for many observatories [94] and neutrino experiments [94, 95]. Importantly, we do not expect counterpart radiation from BBH events, which make the majority of detected GW events at the time of writing.

The BNS event GW170817 is an example of why this multifaceted approach can be of interest. GWs detected from the BNS’s inspiral was detected as early as about 100 s before the merger in current detectors [84]. This sort of “early warning” trigger can be sent out to astronomers to follow up the event. In the case of an ultra-violet (UV) counterpart, this early warning is critical, as the UV counterpart is expected very soon after the merger [96]. GW170817 was observed by many telescopes [96, 97].

Multimessenger astronomy could lead to many more exciting opportunities to gain new insights beyond what is possible with electromagnetic observatories alone. Observations of galactic supernovae in GWs are likely to be able to discriminate between the current models in tension [29, 32, 33]. Further observations of NSBH and BNS events, and the detection of CWs are likely to constrain the NS equation of state [98–101].

1.5 Summary

In this chapter, we describe how GWs are a result of GR, by linearising a perturbation, and changing into the transverse-traceless gauge. We describe several sources of GWs, including CBC events and CWs from rotating NSs. We discussed in greater detail a number of search strategies and analysis techniques for CWs. Then, we outline the state of GW detections, including five BBH events and one BNS.

Chapter 2

Gravitational wave detectors and data quality

In this chapter, we discuss gravitational wave detectors. We will touch on the history of GW detection attempts, and detectors, from early resonant bars through to current advanced detectors. Later, in section 2.2 we will examine the data created by these detectors, the types of noise which can limit search sensitivity for persistent sources, and how to prepare it to achieve the best results with the analysis methods discussed in section 1.3.

2.1 Gravitational Wave Detectors

GW detectors must be sensitive to the strain that a passing GW would apply to the detector. As strain is defined as $h = \Delta L/L$, detectors have relied upon the ability to measure lengths with a high degree of accuracy and precision. The methods by which these measurements have been made have been developed through the years, as the field of GW detection matured [102].

2.1.1 Resonant bar detectors

Among the earliest GW detectors were resonant bar detectors, such as the Weber bar [103], which consist of a cylinder of metal, often a few metres in length, and often weighing several tons [104]. The principle concept behind these bars was that a passing GW with a frequency evolution that overlaps the length-mode resonance of the bar would couple into the detector, and the resonance of the detector would amplify the strain in the bar. By measuring accurately the position of each end of the bar, its length can be monitored. The bars length was often measured by piezoelectric sensors at either end, sensitive to displacements as low as $\approx 10^{-16}$ m [105]. Throughout their use, bar detectors evolved to include laser length measurements, enhanced seismic isolation, and even cryogenically cooled experiments [106].

Weber claimed detection of a GW in a bar detector, but his claim was proved false by a lack of a coincident or subsequent detection. This spurred on the development of more sensitive GW detectors.

2.1.2 Interferometric GW detectors

Towards the end of the 20th century, the concept of an interferometric GW detector began to gain popularity [107]. The core concept for this type of detector is a Michelson interferometer, in which a laser beam is split at a central beam-splitter and reflected off of two mirrors at the end of equal-length arms positioned at right angles to each other. The light then recombines at the beam splitter. The arms' lengths are tuned such that the dark fringe of the recombined beam arrives onto the a sensor, and its brightness is read off at the detector's output port [108]. As a GW passes in the transverse traceless gauge, the coordinate velocity of the light in each arm changes. In this coordinate system, as the coordinate velocity in one arm is reduced by a GW, the velocity in the other is increased. This results in a difference in the light travel time in each arm. The

coordinate velocity in each arm oscillates at the frequency of the GW in anti-phase with each other. Measuring the brightness of the beam spot at the output port acts as a proxy measurement for the difference in coordinate velocity, as the relative phase of each part of the laser beam at recombination is dependant the light travel time.

Early models of GW IFOs showed promise, with prototypes in Glasgow [109], 30Garching [110], Caltech [111], and MIT [112]. These prototype detectors showed that larger, more sensitive detectors were feasible, and highlighted the technologies which would be required in order to bring them into operation, and the limiting noise factors that such detectors would need to overcome.

2.1.3 Noise sources in interferometric gravitational wave detectors

GW IFOs are the most sensitive GW detectors to date, however to attain that sensitivity, they must minimise broadband noise from a number of sources. Here, we explore some limiting noise sources. We will see how different IFOs implemented these techniques in different combinations to achieve their desired sensitivity in sections 2.1.4 and 2.1.5. Then, later, in section 2.2.2 we will explore a number of more narrowband and specific noise sources, which may be local to a given detector, before discussing different, more directed methods of mitigating them.

Seismic noise Ground-based GW detectors must minimise the effects of ground motion on the optical components [113], affecting the frequency range from DC to about 50 Hz. In order to isolate optics from horizontal ground motion, most optics are suspended on pendulum-like isolators. A simple pendulum couples power from ground motion to the horizontal swinging mode of the suspended mass at a frequency f above the suspension's resonant frequency with a scaling of $1/f^2$ [114]. By designing pendulum suspensions with horizontal mode resonances at

particularly low frequencies, this method has been shown to be very effective. Additional to the pendula, several technologies have been used to further isolate components from ground motion. Several of these are listed in table 2.1.

Technology	Detector
Passive vibration isolation stack [115]	iLIGO
Active hydraulic isolation (HEPI) [116]	aLIGO
Active in-vacuum isolation [117]	aLIGO
2 stage steel wire pendulum [114]	iLIGO
4 stage Monolithic suspension [118]	aLIGO
Blade springs [118, 119]	aLIGO
Underground installation [120, 121]	Kagra, ET

TABLE 2.1: Seismic isolation technologies in GW detectors.

Thermal noise These detectors are also sensitive to the random motion of particles in the bulk, coatings and suspensions of the optics [122]. The noise is currently below sensitivity, but limits sensitivity in the band from 10 Hz to 500 Hz [123]. To effectively combat this thermal noise, we carefully select materials to make the mirror substrate, coatings and suspensions. Each element must also be selected for its other criteria: the mirror coatings must be selected for low absorption of light with the input laser’s wavelength, and the required transitivity for each optical component [124, 125]. The low absorption of the mirror coating is important as thermal Brownian noise of the mirror coating is a limiting noise source at peak detector sensitivity. An alternative method to combat thermal noise is to cryogenically cool the optical components of GW detectors [126, 127]. This, necessitates a different choice of mirror and suspension materials, which perform well at cryogenic temperatures. It can also bring about its own source of noise associated with the cooling system itself [128].

Quantum noise Quantum noise is composed of two primary effects: photon shot

noise, and radiation pressure fluctuations. Photon shot noise is an effect related to the Poisson-statistics of counting photons at the dark fringe of the IFO's output, where the strain is read out. Radiation pressure fluctuations are similarly an effect of the number of photons. It is fluctuations in the transfer of the photon momentum to the IFO's mirrors [129]. Where radiation pressure fluctuations will be limiting to future detectors below 40 Hz [123], photon shot noise a limiting noise source above 100 Hz [123]. Quantum noise is limited by the standard quantum limit (SQL). The limit is associated with the uncertainty principle relating to the photons in the system [129]. It can be thought of as a trade-off between radiation pressure fluctuation noise and photon shot noise. The SQL noise can be minimised at a given frequency by tuning the laser power of an IFO. However, it is possible to beat the SQL in a band by introducing a squeezed vacuum state into the optical cavities [130].

Gravity gradient noise The gravity gradient noise describes fluctuation on the local gravitational acceleration, g due to density fluctuations in the Earth [131]. It will likely be a limiting noise source for future ground-based GW detectors. Gravity gradient noise is a consideration below about 15 Hz [131]. Gravity gradient noise can be mitigated by building the IFO a few tens to hundreds of metres underground. For this reason, future detectors, such as Kagra and the Einstein Telescope are to be built underground

These noise sources, as well as specific environmental noise sources associated with a given detector will combine to define the sensitivity of any detector to GWs. We typically characterise this with a sensitivity curve. This shows the frequency profile we expect from a detector during normal operation while there are no astrophysical signals in the detector. Figure 2.1 shows a projected noise source for aLIGO as a sum of its principle limiting noise sources.

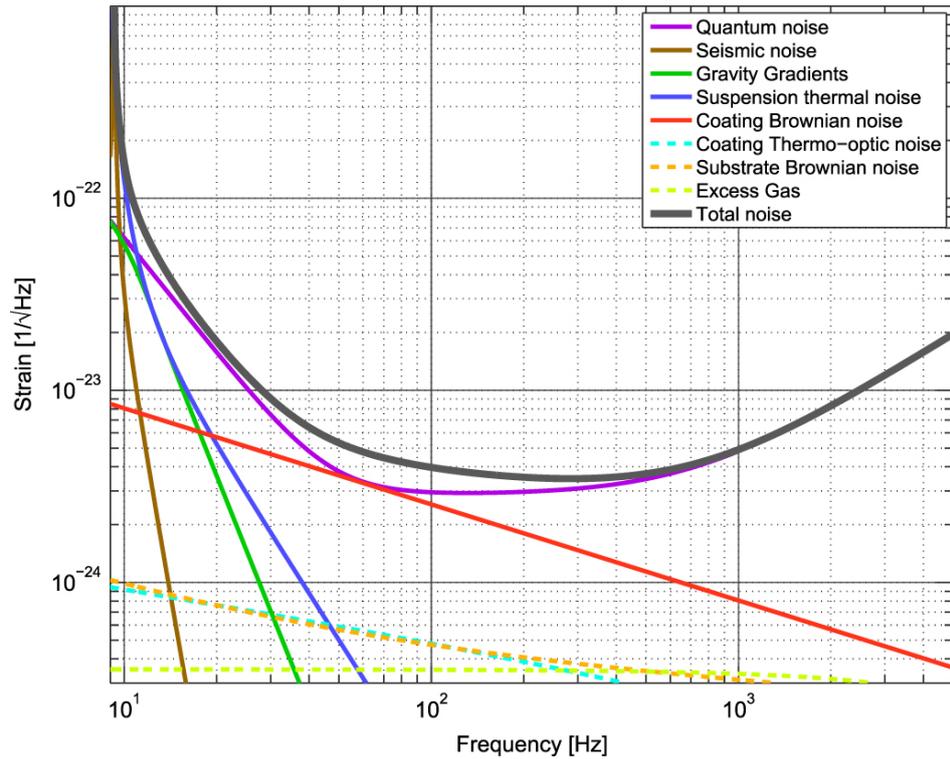


FIGURE 2.1: Advanced LIGO’s expected noise curve (black) as a sum of its constituent noise sources at design sensitivity. [132].

2.1.4 First generation interferometric GW detectors

A growing network of GW IFOs was developing throughout the 1990s and 2000s, including: TAMA300, a 300 m long detector based in Japan [133]; AIGO, the Australian IFO GW observatory; GEO600, a 600 m long IFO in Germany [134]; Virgo, a kilometre-scale IFO in Italy [135]; and Initial LIGO (iLIGO).

2.1.4.1 Initial LIGO

The laser IFO GW observatory (LIGO) network consists of three principal GW detectors on either coast of the USA, LHO in Washington state in the north west held two of them, and LIGO Livingston observatory (LLO) in Louisiana in the south east. The two detectors at LHO was to allow for two independent observations to occur with the same noise background. This allowed for a greater sensitivity to a gravitational wave background. Each detector had arms of length 4 km [136]. All three detectors were

nearly identical in design, so that each detector could provide independent, coincident observations of any given GW source with similar sensitivity. Since both detectors have a broad antenna response pattern, a strong GW event in the vast majority of positions in the sky will generate a response in both detectors (see figure 1.2).

The design of iLIGO drew from prototypes, like those at Caltech [111], Garching [110], MIT [112] and Glasgow [109], as well as earlier detectors designs, by incorporating a number of optical and material techniques in order to increase its strain sensitivity, and minimise sources of local environmental noise.

The laser used in each iLIGO detector was a pumped 1064 nm Nd:YAG laser with a 2D Gaussian beam profile. Prior to entering the IFO, the laser is frequency and amplitude stabilised, and is fed through a mode cleaner [137], to remove spatial modes of the laser other than the desired Gaussian modes. The mode cleaner is made of one or more high-finesse triangular cavities, with length an integer multiple of the desired laser mode's wavelength. This exploits the slight difference in wavelength of the higher order spatial modes in order to reduce their amplitude. In order to increase the strain sensitivity of such a GW detector, high laser power within the arms is preferred, as the optimal power to balance the radiation pressure fluctuation noise and photon shot noise is at 15 kW. In order to achieve this, iLIGO employed a number of techniques, including the introduction of a power recycling mirror, and the use of Fabry-Perot cavities in each arm. Together, these are capable of increasing the laser power from 5 W input power to 15 kW circulating laser power. A reduced schematic of the iLIGO optical design is shown in figure 2.2.

Each arm had two test mass optics, the input test mass (ITM), which was a partially transmissive mirror, and the end test mass (ETM) was fully reflective. The two were used in conjunction to create the Fabry-Perot cavity within each arm. Both the ETM and ITM in each arm were formed of a 10.7 kg coated fused-silica substrate [137]. The coatings in each case were required to have low optical absorption, low scattering, low

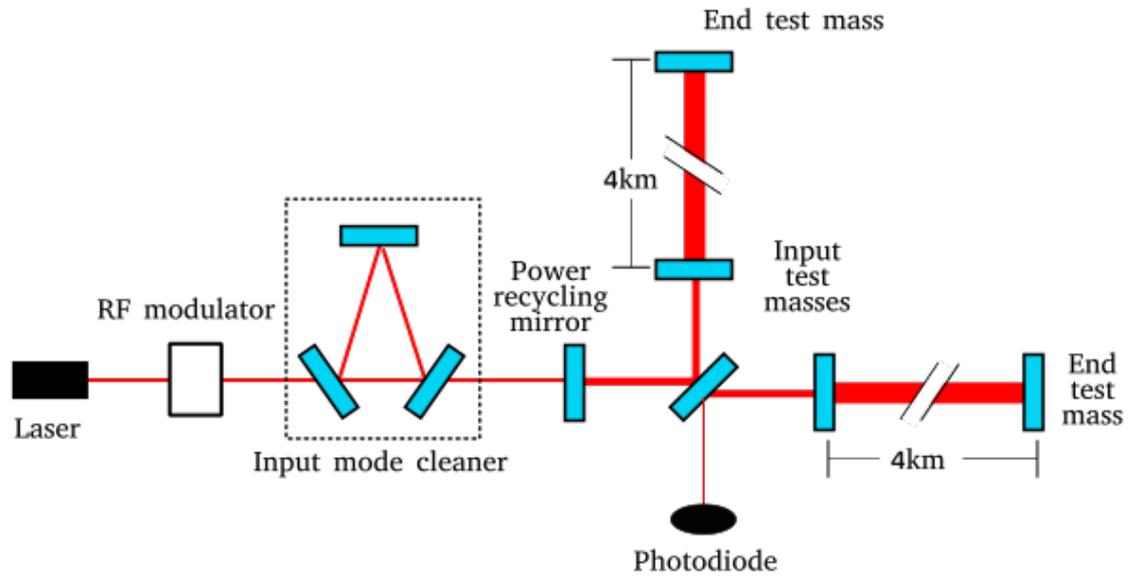


FIGURE 2.2: A simplified schematic of the iLIGO optical layout. Each test mass weighs 10 kg.

thermal noise, and high reflectivity. All optical components in iLIGO were isolated from the ground by a steel wire pendulum, hanging from a frame isolated from vertical seismic motion by mass-spring isolation stacks [137]. This seismic isolation provided for a response that scaled with the frequency f of the ground motion as f^{-2} at low frequencies, and f^{-8} below 100 Hz [115].

Observations made by iLIGO spanned from 2003 to 2010 [138, 139], with a total of six observations, termed “science runs”, each lasting longer than the previous observation. The fifth, S5, was the first such observation during which iLIGO was at design sensitivity, and ran between September 2005 to October 2007. Between the fifth and sixth science runs, the detector was subject to enhancements beyond the original scope of its run [140]. This included the addition of a squeezed vacuum state, an in-vacuum readout and an output mode cleaner.

The fifth and sixth science runs of iLIGO were coincident with the first three observing periods of Virgo [141, 142], as well as the continued operation of the GEO600 detector, both of which had comparable sensitivity. This allowed for several three- and four-detector analyses, resulting in greater sensitivity than any pairing could achieve. With

the end of iLIGO's operation, plans to upgrade both iLIGO detectors were already in place. This upgrade would bring in the second generation of GW detectors.

2.1.5 Second generation GW detectors

This second generation of GW detectors, aLIGO [132], was built on the foundations of the first generation. The internal components for both iLIGO detectors were overhauled. Construction for the second generation began in 2010, and was completed in 2015. The new aLIGO detectors began operation in 2015.

2.1.5.1 Advanced LIGO

Built on their predecessors, aLIGO was composed of two detectors, one at each of LLO and LHO, and each detector is a 4 km Michelson IFO as before. However, major upgrades have affected most key components.

The electronics at both sites were overhauled: All switched mode power supplies and DC/DC converters were eliminated; all data acquisition hardware was replaced; digital and analog electronics were separated into different enclosures.

The test masses used in aLIGO were increased from a mass of 10.7 kg each to 40 kg each. These are still coated fused-silica substrates, however, the coating materials and techniques have been improved since iLIGO. All of the optics in aLIGO are similarly suspended above the ground in order to suppress seismic noise. Rather than a single steel wire pendulum, aLIGO suspensions were a specially designed four-stage pendulum system. The top three stages of pendulum are steel wire, the final stage is fused-silica fibre, chemically bonded to the test mass [118]. Figure 2.3 shows such a test mass and its suspensions being installed. Besides the benefits of a seismic response to ground vibrations above 10 Hz scaling as f^{-8} , this monolithic construction also reduces the mechanical losses in the pendulums, which confines the thermal noise closer to the 1 Hz

pendulum resonance, and reducing the broad noise floor in the band between 30 Hz and 8 kHz.

Beyond the new monolithic suspensions, the vibration isolation system was also overhauled. Hydraulic platform pre-isolators were installed, and active seismic isolation replaced the previous passive stack system [117].



FIGURE 2.3: A photograph of a test mass and its suspensions being installed [143].

The input lasers of aLIGO are also an advancement on the iLIGO configuration. The wavelength used remains at 1064 nm, however the newer input optics are capable of a tunable output power of up to 125 W [144]. The Fabry-Perot cavities are designed for a circulating power of 750 kW [145]. The optical layout now includes a signal recycling mirror at the dark port, and an output mode cleaner [145].

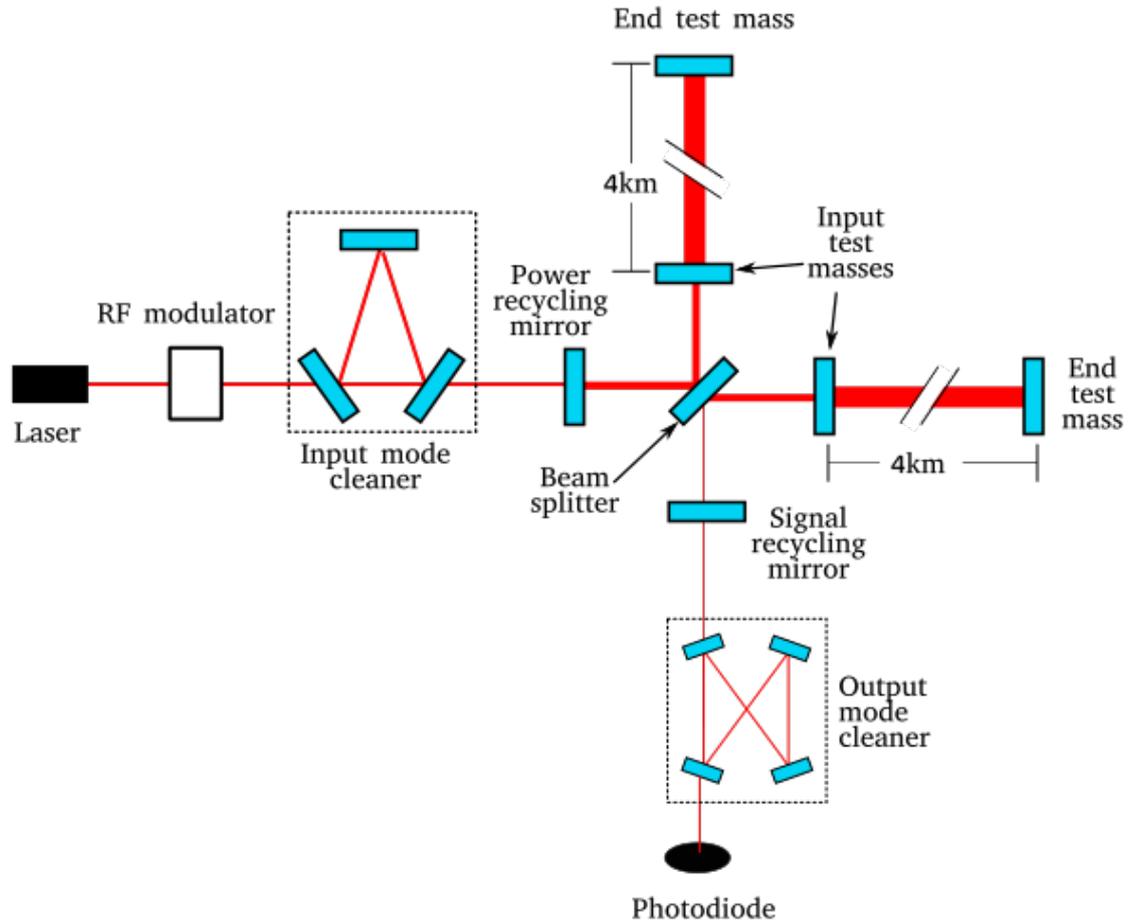


FIGURE 2.4: A simplified schematic of the aLIGO optical layout.

This higher circulating power in the cavity causes thermal lensing on the test masses. In order to combat this, both ITMs are equipped with a ring heater and an auxiliary CO₂ laser system [146]. New mirror coatings were designed, tested and selected to reduce the effect of the coating's thermal noise [145], and to reduce optical absorption so as to reduce the test mass heating. Another effect of the high laser power is parametric instabilities, excitation of body vibration modes of the test mass [147]. These parametric instabilities can be damped by precisely actuating the excited test masses exactly out of phase with the instability [148].

At the time of writing, the aLIGO detectors have completed two observation periods, (O1 and O2), and are currently undergoing upgrades ahead of their third observation.

During O2, aLIGO was joined in observation by advanced Virgo (adVirgo), a second-generation upgrade of the Virgo detector. During these observations, aLIGO and adVirgo were able to detect a number of CBC sources, outlined in section 1.3.5.

2.1.6 Prospects for future detectors

Now, in the age of GW detections, increased sensitivity of GW detectors can bring exciting new observations. Careful choice of detector location, configuration and data analysis can aim to increase the GW horizon further into the universe, and increase the observation rate of GW sources, as well as bringing about detections of other sources detailed in sections 1.2 and 1.3.

First expected of these future detectors is KAGRA [126]. Based in the centre of Japan, KAGRA will be the first “2.5 generation” detector. Its location in Kamioka, close to other cutting edge experiments such as Super KamiokaNDE and KamLand, highlights its aim to be part of a suite of multi-messenger detectors. KAGRA is a 3 km subterranean interferometer, whose key optical components are cryogenically cooled. The fused-silica optics and suspensions used in aLIGO are not ideal for use at cryogenic temperatures, and as such the suspension fibres and test masses are made of sapphire (aluminium oxide) [149]. KAGRA is due to join aLIGO and adVirgo for observing in 2020.

Following aLIGO’s scheduled observations, due to finish around 2022, there are already plans to further upgrade these detectors. The first upgrade, called A+ [150], is analogous to enhanced LIGO’s operation at the end of the iLIGO observations. A+ is set to join KAGRA as a 2.5 generation detector. Later, A+ will be further upgraded to LIGO Voyager (LV) [151]. This cryogenic interferometer will likely involve changing to silicon test masses and suspensions, necessitating a change of laser wavelength also. Much of the specifics for LV are only loosely defined, and there are a large number of variables which may drastically change much of any potential design. It is likely that

LV will be operating in the late 2020s. Following LV, the Cosmic Explorers are set to drastically change the LIGO IFOs, with plans including 40 km IFOs and 320 kg test masses [152].

Following on from adVirgo in a similar sense, there are plans for a third-generation GW detector in Europe. This pan-European detector is to be called the Einstein Telescope (ET) [153]. A final location and configuration for ET has not yet been chosen, and many of the specifics for the design are not yet finalised. ET is likely to be a triangular formation – an IFO with 60° internal angles. Each side of the triangle is to be 10 km long, and the entire detector will be subterranean. Each vertex will house two IFOs, a low-power, cryogenic configuration, and a high-power room temperature configuration. The cryogenic configuration will be designed for low-frequency sensitivity, and the room temperature configuration will compliment it with high frequency sensitivity. Currently, the project aims to have ET operating before 2030.

In the age of GW detections, the primary goal for third generation detectors is to expand the GW horizon. As the sensitivity of a detector increases, the distance out to which it can detect a given GW increases linearly. The volume which this represents scales as d^3 , the cube of the observing range for a given source. An increase in sensitivity to a given source by a factor of 2 corresponds to an increased detection rate for similar sources of $2^3 = 8$. Until these new detectors are in operation, we can still improve current detectors.

Between each observing run, aLIGO goes through a round of upgrades and maintenance. While this aims to improve the sensitivity, it often introduces a large number of tunable parameters which affect the sensitivity, whether within a narrow band of interest, or broadly across the spectrum.

For the remainder of this chapter, we will examine how we can characterise a detector's sensitivity, and how environmental noise can be tuned out of the detectors for more effective data analysis.

2.2 Characterising aLIGO

The analysis of the data from a GW detector relies upon the data being of a form which we understand. Often, noise is modelled as Gaussian white noise, then rescaled to achieve the detectors frequency response curve. In part, this is because such a model of the noise is what is expected from the detector to a good approximation, and in part because it is a convenient approximation, with Gaussian noise having a well defined statistical distribution, and being a commonly observed distribution of noise in real instruments. However, noise from a GW detector also contains time varying, transient, and non-Gaussian noise components. Beyond the limiting broad noise sources mentioned in 2.1.5, there can be any number of local environmental noise sources, which create non-Gaussian data in the detector.

2.2.1 Data in frequency and time-frequency domain representations

When looking at GW data in order to examine the data quality, it often helps to visualise the data differently to highlight poor data quality. When considering data quality which affects searches for persistent GWs, we are mostly interested in long time duration noise sources with a relatively narrow frequency profile. GW detectors generate data in the time domain, and aLIGO creates strain data with a sampling rate of 16 384 Hz. Naively, we may think plotting the data against the time of its creation may be most useful. However, it is rarely the case that such a plot is intuitive or instructive to assessing the data quality. For this reason, it is often more useful to view the data in the frequency domain, or the time-frequency domain, than in the time domain.

2.2.1.1 Data in the frequency domain

Data produced by GW detectors is produced in the time domain (denoted by $D(t_k)$), which is not necessarily the best way to represent the data when looking for instances of noise. We can instead consider a frequency domain representation.

The data are not taken continuously, but at regular intervals, at times $t_k = t_{\text{start}} + \Delta k$, and where $k = 0, 1, 2, \dots, L_D - 2, L_D - 1$, Δ is the time between subsequent samples, and t_{start} is the start time of a dataset. As such, we use the discrete Fourier transform (DFT) to approximate the Fourier transform. The DFT estimates the frequency content, F_n , of discretely and regularly sampled data. The frequency content is estimated at regular frequency intervals. In a dataset with L_D samples, with a sampling frequency of f_{samples} , the DFT will estimate the contribution in $L_D/2$ frequency bins equally spaced between DC (0 Hz) and $f_{\text{samples}}/2$, the Nyquist frequency. The DFT is defined as [154]

$$F_n = \sum_{-k=0}^{L_D-1} D(t_k) \exp(-2\pi i k n / L_D). \quad (2.1)$$

As the data $D(t_k)$ are real values, the values of F_n resulting from the DFT are complex. The resulting frequency contributions will be at frequencies [154]

$$f_n = \frac{n}{L_D \Delta}. \quad (2.2)$$

In order to calculate the power spectrum, P_n , from the DFT, we calculate the modulus squared.

$$P_n = F_n F_n^*. \quad (2.3)$$

Further, we can define the cross-power as the power computed by two sets of data. The cross-power can highlight similarities and differences in the frequency components in two. We can consider two datasets, $D(t_k)$ and $B(t_k)$, and take the product of their frequency components, $F_{D,n}$, $F_{B,n}$, to calculate the cross power. $C_{D,B} = F_{D,n} F_{B,n}^*$.

In equation (2.2), we see that datasets with more samples result in finer frequency resolution. We often end up with more frequency bins than we require, which may themselves be noisy. Instead, we can create a frequency representation with a preferred frequency resolution by splitting the dataset into segments $d_T(t)$ of length $L_d = 1/\Delta f$, such that L_D/L_d is an integer. We can then perform the DFT on these $d_T(t)$ instead, to produce $d_T(f)$.

However, the DFT treats a finite dataset as though it were periodic, with a periodicity of the length of the dataset. Therefore, the DFT will also calculate frequency components across this supposed (but not real) period boundary. In order to combat this, we can choose to “window” the data. A windowing function is a method of weighting the data in each chunk $d_T(t)$, such that the data smoothly evolves to zero at each end of the dataset. The Hann window, $w(t)$, one of many viable choices, is often used, and is defined for a data set of length L as

$$w(t_k) = \sin^2\left(\frac{\pi\Delta k}{L-1}\right). \quad (2.4)$$

The Hann window applies a weight to the end samples of the data with a weighting of $w(0) = w(L-1) = 0$, and the central sample a weighting of $w((L-1)/2) = 1$. To prevent any one data sample to be more important than any other, we can choose to chunk the data such that there is an overlap fraction of 0.5, meaning that the second half of one chunk shares its samples with the first half of the next. In terms of the Hann window, this is equivalent to a phase shift of $\pi/2$ for the shared samples. The total weight W applied to each sample is given by

$$W(t_k) = \sin^2\left(\frac{\pi\Delta k}{L-1}\right) + \sin^2\left(\frac{\pi}{2} + \frac{\pi\Delta k}{L-1}\right) \quad (2.5)$$

$$= \sin^2\left(\frac{\pi\Delta k}{L-1}\right) + \cos^2\left(\frac{\pi\Delta k}{L-1}\right), \quad (2.6)$$

and as $\cos^2(x) + \sin^2(x) = 1$, then all samples are given equal weighting, with the

exception of the first and last chunks. We choose this distortion of the dataset ends over the distortions due to the assumed periodicity of the dataset, as this drawback of windowing the data is usually a smaller effect.

We apply this window to each of the shorter datasets $d_T(t)$, and calculate the DFT on the windowed data, which are often called short Fourier transform (SFT).

$$d_T(f_n) = \sum_{-k=0}^{L_d-1} [d_T(t)w(t)] \exp(-2\pi i k n / L_d). \quad (2.7)$$

We are able to increase the accuracy of the frequency contribution estimate by averaging together the estimates from several smaller datasets $d_T(t)$. By averaging together the modulus-squared the SFT we can estimate the power in the data. This is known as the Welch estimator of the power spectral density (PSD) [154]

$$\langle D(f_n) \rangle = \frac{L_d}{L_D} \sum^T (d_T(f_n) d_T^*(f_n)). \quad (2.8)$$

In practise, an algorithm called the fast Fourier transform (FFT) is used to estimate the DFT. This algorithm greatly increases on the speed of calculating the DFT.

In the context of a GW detector, the frequency representation (in particular, the PSD) can be understood as a broad saahape, attributed to the noise floor of the detector, and many sharp, narrow artefacts attributed to calibration lines or noise sources (which may have a known or unknown origin), and intermediate,, broader structures, some well understood, others less well characterised.

2.2.1.2 The time-frequency domain

Instead of averaging together several SFTs, we can instead take the modulus of each SFT, and stack them next to each other to create a coarse time series of frequency components. We can create a 3D plot of this data, with time along the x-axis, frequency

along the y-axis, and amplitude up the z-axis. The product is usually expressed as a colour map image. This is called a spectrogram.

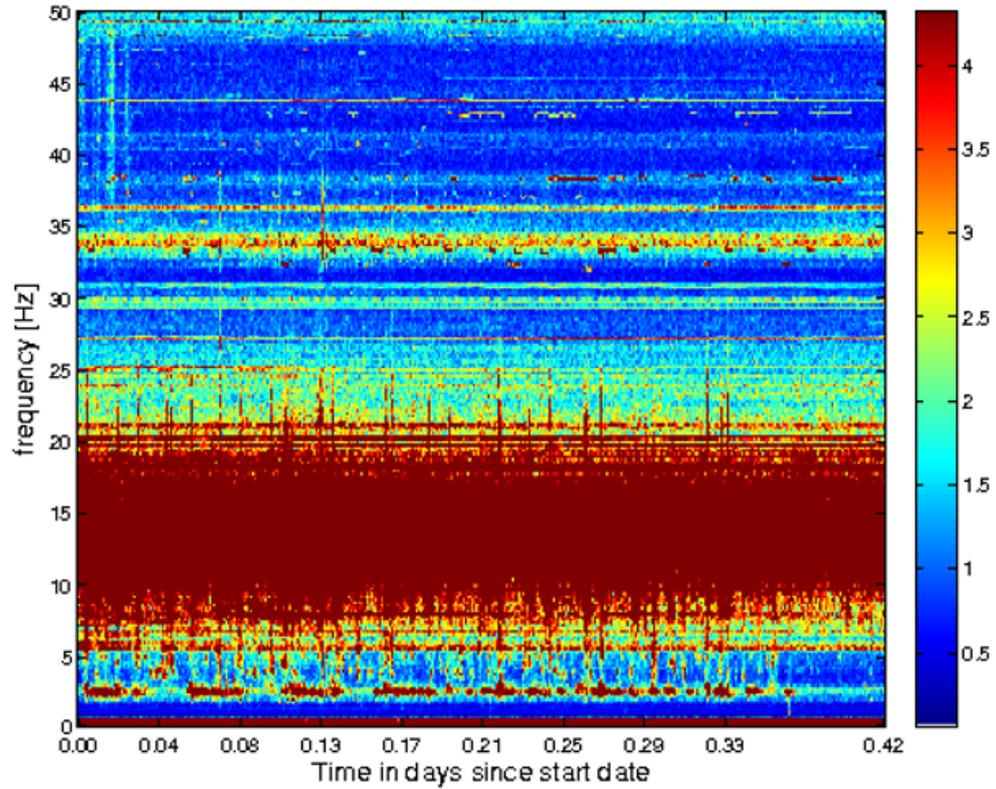


FIGURE 2.5: An example of a spectrogram, which plots data in the time-frequency domain. Each column of pixels is a single SFT represented by the colour key. Each subsequent column shows another SFT at another time.

The bins of a spectrogram obey an uncertainty product, due to the relationship in (2.2). If the width of a bin in time is Δt , and the width of a bin in frequency is Δf , then their product $\Delta t \Delta f = 1$. The shape of a bin can be tuned to suit a purpose.

2.2.2 Detector characterisation

Detector characterisation is the process of deriving information about the quality of the data from the detectors for the purpose of identifying science-quality segments of data for analysis, and for identifying and assisting in the removal of noise artifacts. Where

mitigation of a noise source is not possible or practical, we are able to flag sections of the data with poor data quality as such, so that they can be excluded from searches.

Data from the detector is acquired through several data “channels”. Each data channel delivers a data stream from different instrument, readout or observable within the detector. Data from several channels are examined with low latency, for the operation of the detector, and the data from every channel is stored for later use.

The first step in the detector characterisation process is to study the quality of the data from the detector.

2.2.3 Data quality

Both the frequency- and time-frequency domain provide intuitive ways to review GW data for its quality. By reviewing a PSD or a set of spectrograms, we can explore a number of common data quality issues which particularly impact searches for persistent GWs. In this section, we will examine different types of narrowband noise, which appear above the noise floor, namely lines, combs and wandering lines.

2.2.3.1 Lines

First, the most common persistent narrow noise source is called a “line”. A line refers to a long lasting noise source, which oscillates with a fixed frequency. In a spectrogram, lines simply appear as narrow regions of increased PSD at a fixed height in the plot, giving them their name.

A line is easily described by its central frequency f_c , and the width of the line on both the left (Δf_l) and right (Δf_r) of the central frequency. The frequencies affected by a line are a range from $f_c - \Delta f_l$ to $f_c + \Delta f_r$.

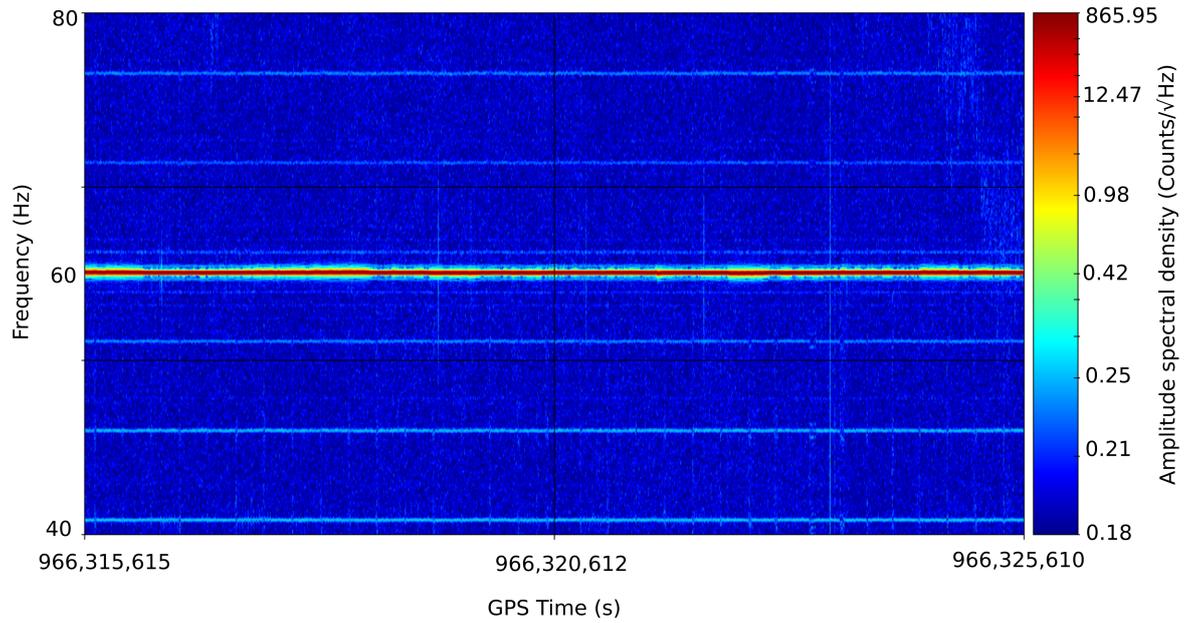


FIGURE 2.6: An example of a line in a time-frequency plot. This plot shows the 60 Hz line at LHO, seen as a horizontal strip of red pixels [155].

Unwanted lines can arise from any number of sources, and any number of coupling mechanisms. For example, figure 2.6 shows a noise line at LHO at $f_c = 60$ Hz. This line is the result of the mains frequency in the USA. Some lines, such as the detector’s calibration lines are purposefully introduced into the detector.

Similarly to the calibration lines, another set of lines are knowingly allowed in the detector. The suspensions from which the test masses are hung in aLIGO introduce a set of lines as the fibres vibrate thermally. These vibrations mechanically couple to the test masses, which couples them into the GW data. It is advantageous that these vibrations are in a known, narrow frequency band, rather than broad noise over a wide frequency range, as the former will contaminate far less data with this instrumental noise. These vibrations (at the fundamental frequency as well as higher order harmonics), and their associated lines are known as violin modes, which introduce a region of roughly 20 Hz heavily polluted by 16 lines (one for each fibre, four for each of the four test masses), and their side-bands, introduced by upconversion of loud very low-frequency noise.

2.2.3.2 Combs

A related noise sources appears as a set of lines which appear at regular intervals of frequency. Such a set of lines is called a “comb”. These noise sources can be very problematic, as they can affect a much broader range than a single line. However, combs are much less common in data than single lines. The name “comb” comes from a number of single lines appearing as teeth of a comb in the FFT of data. As such, a single line in a comb is referenced as a “tooth”.

A comb can be described by the width between the central frequency of each tooth (Δf_c), the smallest positive offset from 0 Hz that a tooth could appear (f_0 , the 0th harmonic), the visible harmonics of the teeth (n_i denotes the harmonic number for the i^{th} tooth), and the widths of each tooth on the left and right of its central frequency (Δf_l and Δf_r respectively). As any given comb can contain hundreds of teeth, it is not always possible to define the widths for each tooth. In this case, the widest values for Δf_l and Δf_r are used for each tooth. For a comb, the affected frequencies f_{aff} to the left and right of each tooth are given by

$$f_{\text{aff}}^l = f_0 + (n_i \Delta f_c) - \Delta f_l, \quad (2.9)$$

$$f_{\text{aff}}^r = f_0 + (n_i \Delta f_c) + \Delta f_r, \quad (2.10)$$

respectively.

Data which is periodic but not sinusoidal in the time domain will result in a comb. We can think of the comb as the fourier components of this data. The hard edge of a square or sawtooth wave will result in a comb with infinite teeth.

2.2.3.3 Wandering lines

Some persistent noise sources appear to be stationary in frequency. Unlike lines and combs, “wandering lines” are a persistent noise source with some smooth, time dependent evolution with time. These sources can be among the most difficult persistent sources to describe, as the class allows for such variability. Fortunately, this type of persistent noise source appears to be less common than even combs.

A wandering line must smoothly evolve. A line with discontinuities either in frequency or in time can not be considered a single wandering line without evidence from other auxiliary channels to back up the claim.

2.2.3.4 Transient noise sources

While persistent noise is of primary concern to searches for persistent GWs, any extraneous noise in the detector can impact a search. Not all environmental noise sources can be neatly organised into lines, combs and wandering lines. Additionally, transient noise artefacts which have a well defined start and end time, can be important considerations when discussing data quality for CW analyses. These transient noise artefacts often are much louder than a persistent noise source, but as they are limited in time, they typically interfere less with a CW analysis than a persistent noise source. Typically, transient environmental noise is broader in frequency than persistent noise. In cases where a transient noise source repeatedly and frequently affects a given frequency band being searched for CW signals, then we consider the noise source to be of concern.

Characterising transient noise is usually done by studying it in SFT plots at first. We also look at correlation plots between the strain and physical environment monitor (PEM) channels, also plotted as a cross-power spectrogram. Once a consistent set of behaviours for this noise, including a valid witness, has been found, candidate coupling methods can be investigated.

2.2.4 Implications for long-duration searches

These narrow band persistent noise sources can negatively impact searches for persistent GW signals (CW and stochastic searches alike), in a number of ways.

In searches for transient GW signals, we require that a coincident signal be present in multiple GW detectors before it can be further investigated. The analogue of this for persistent GW signals is that a consistent signal model signal should be coincident in multiple detectors (however, this can be relaxed if the signal model is present in many observations from one detector). The presence of a line or wandering line in the data, common between detectors (or observations) also fulfils this criteria for an unmodelled search (such as a Viterbi search, eg [156]) without a line-robust statistic. For example, the 60 Hz line is present in both aLIGO detectors, though its frequency evolution between detectors is not consistent with an astrophysical source. This kind of “false positive” source, where a persistent artefact in the data mimics a persistent GW, is likely to generate a candidate GW source which must then be followed up, and eventually discarded.

A further concern is that a persistent coherent noise source (such as a line or comb tooth) overlaps the GW frequency of a true astrophysical source. In such a case, the GW strain amplitude may be concealed by a greater strain induced by the noise, and thus not likely found. This can occur for a number of reasons.

First among them is that a persistent coherent noise source is likely to limit the detector’s sensitivity at the target frequency. At the detector’s noise floor, we model the noise as Gaussian distributed, meaning that each sample is independent of every other sample, and as such, the data are incoherent. By contrast, persistent GWs are coherent over time, as each sample is related to every other by its relationship to the signal model. This gives the \sqrt{t} relationship described in section 1.3.4, which tells us

that the SNR of a signal, ρ scales with the square-root of the observation time.

$$\rho \propto \sqrt{T_{\text{obs}}}, \quad (2.11)$$

To compound the issue, it is very likely that the louder source would be the noise source. We have constrained the upper limits of GW amplitude of a number of high priority sources to within a few percentages of their total energy dissipation budget [157]. Conversely, persistent environmental noise can only be diagnosed when it can be detected, whether by long integration or shorter (though generally, it is much more convenient to investigate louder noise, which requires shorter integration time to become visible). The means that known noise artefacts are almost certainly expected to have a higher strain amplitude than a CW signal.

Another method by which persistent narrow noise artefacts can negatively impact persistent GW searches is the way in which the lines, combs and wandering lines are handled. Frequency bins which contain noise classified as a known narrow artefact, and adjacent bins, are often excluded from persistent CW searches. This is a common practice, in order to minimise the number of false-positive candidate signals, which require a lot of expensive follow-up.

In a persistent GW search, we usually analyse data after the observation has finished, meaning that there is little scope to remove environmental sources of noise from the analysis. However, by scheduling successive observations around several-months long commissioning periods, we are able to treat known noise sources for future observations.

2.2.5 Sensors of the physical environment

Each detector with aLIGO is comprised of more than just the equipment which delivers the GW data, the detector must also produce metadata for monitoring the performance of each subsystem.

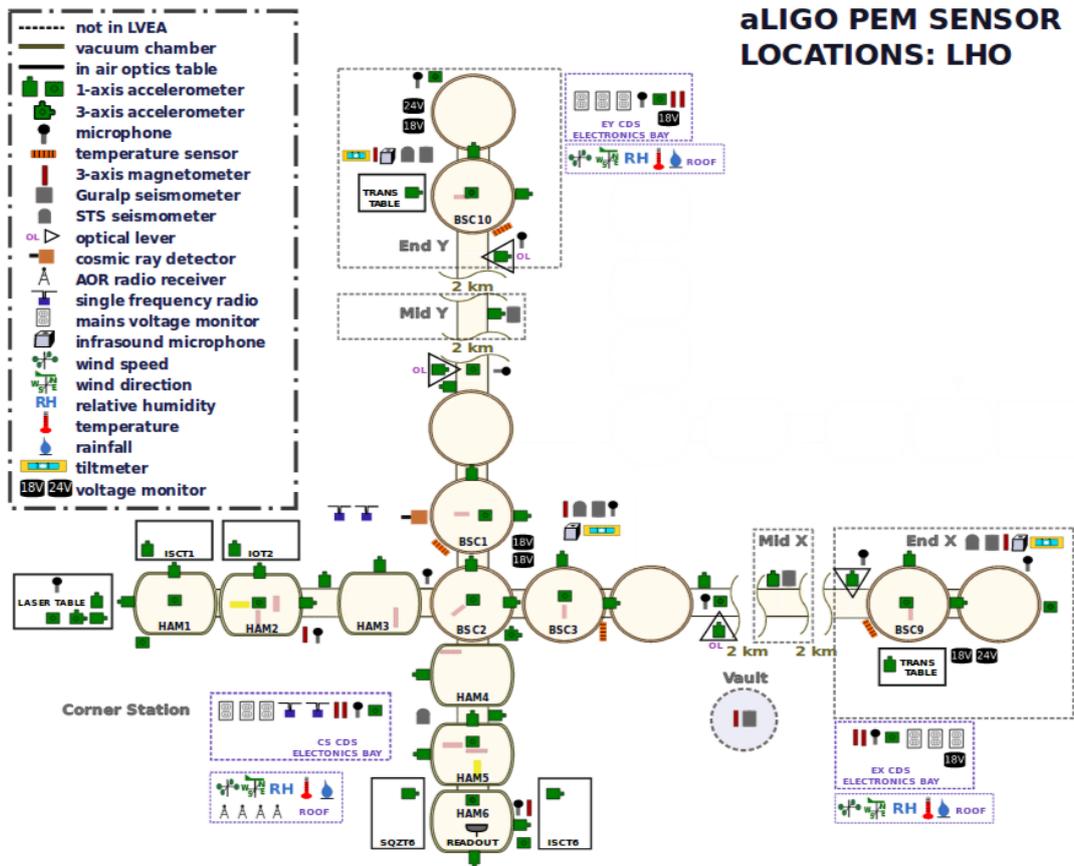


FIGURE 2.7: A map of LHO with each PEM sensor mapped. The mid- and end- stations of each arm are labelled. All other sensors are in the corner station at the vertex of the interferometer [158].

Particularly important for detector characterisation are the PEM channels. The PEM sensors consist of a varied set of monitors, which are distributed around the site of each detector. They serve to accurately and reliably record the local environment, so that should some noise source arise, these auxiliary channels can be searched for similar noise signals. Additionally, PEM channels must be insensitive to astrophysical GWs. This allows a PEM sensor to be an independent witness to the detectors environment. The types and positions of PEM sensors at LHO are shown in figure 2.7. Among the most important and most frequently queried PEM sensors and channels are

Seismometers Distributed throughout the detector are a number of seismometers of different types, both outside of each of the main buildings, and inside the

building's experimental areas. These arrays are able to witness earthquakes, and other, more localised seismic activity, which disrupt the detector. Their sample rate of 256 Hz limits their frequency response to a maximum of 128 Hz, which is well above the region where seismic noise is dominant, up to about 20 Hz in aLIGO.

Accelerometers A large number of 1- and 3-axis accelerometers monitor the motion of every optical component, as well as any motion of the beam tube, and the floor of most experimental areas. They are able to witness noise up to 4 kHz.

Magnetometers Inside each electronics bay (EBAY), vacuum equipment area (VEA) and laser and vacuum equipment area (LVEA) are a number of magnetometers with three-dimensional sensitivity. They can witness a range of noise sources often produced by the data acquisition hardware, coupling into the detector electronically or magnetically.

For any environmental noise source in the GW data, there should be at least one PEM witness channel which is able to identify the noise as being local to the detector, or local to within a location within the detector.

2.2.6 Witness channels

PEM channels offer a host of possible witness channels to any local noise source, which are sensitive to the detector's environment, but not to GWs. An artefact in the GW channel with significant frequency (or time-frequency in cases of transient noise) overlap with a channel which is insensitive to GWs should be flagged for further investigation. Such channels insensitive to GWs are called witness channels.

In order to determine whether a PEM channel is a viable witness to a noise source, we can consider the cross-channel coherence, C between the GW data and a potential

witness. The cross-channel coherence between two channels D_1 and D_2 given by

$$C^2 = \frac{\langle D_1 D_2^* \rangle^2}{\langle D_1 \rangle^2 \langle D_2 \rangle^2}, \quad (2.12)$$

where $\langle D_1 D_2^* \rangle$ is the cross-power spectral density of D_1 and D_2 , and $\langle D \rangle^2$ is the PSD of the channel D . The cross-channel coherence describes a coherence between two channels in the frequency domain.

The cross-channel correlation between the GW channel and a witness should resemble the profile of the noise being witnessed. The strength of the correlation indicates the coupling strength between the noise source seen by the witness, and the noise in the GW data. While there may be many witnesses for any given noise source, often the witness with the highest cross-channel correlation indicates the witness most closely related to the coupling mechanism between the source and the GW data. Tying a noise artefact to a witness channel firmly identifies the noise as environmental or instrumental, and not as an astrophysically interesting source.

In order to monitor the noise in the detector, we use a combination of frequency and time-frequency representations of the GW and PEM data in order to easily and quickly pick out likely instances of noise.

2.2.7 Line and comb lists

Simply identifying the source for noise is not sufficient. In order to disseminate information about affected frequency bins, we construct a list containing the lines and combs in a dataset. A list of known lines is created for each detector in every observing run. In order to be a part of this list, we require that lines on this list have a witness channel, or to be a known injection (which are well documented, and have their own witnesses also). Further, combs are added to this list, as we don't expect combs from current astrophysical models of rotating NSs.

A line list details the key frequencies, widths and harmonics detailed in sections 2.2.3.1, 2.2.3.2 and equations (2.9) and (2.10). With this information, we are able to recover the affected frequencies. This information is made available to each search group, and often the affected frequency bins are excluded from searches.

2.2.8 Hardware mitigation

While data is being taken during observing runs, there is very little scope to mitigate noise sources, as doing so requires physically altering components, either in the LVEA, VEA or the EBAY. As such, between observing runs is a good opportunity to investigate and mitigate as many sources of lines and combs as possible. Issues which are not mitigated are likely to reappear in the data obtained in subsequent observations and, as the sensitivity of the detectors are increased between observations, more noise sources are likely to be exposed. As such, the mitigation of persistent noise sources is an important part of detector characterisation.

Mitigation of a noise source can only occur once the source and coupling method are known – knowing only one or the other is not enough. While having a witness channel is a key clue in determining the source and coupling method, by offering a possible coupling channel, it is not enough on its own. We can infer the nature of the source of the noise by examining the noise itself. A wandering line or a broad line is likely to couple mechanically – through ground motion, or vibrations in the beam tube casing, for example. This is because mechanical couplings are dependent on the physical environment around the coupling site. For example, the temperature of the beam tube can cause the housing to expand, and alter its resonant frequencies. A very narrow line is much more likely to be coupled electromagnetically. Further, if the frequency is very precisely regular (for example at exactly 16 Hz), or the time series of the noise repeats at regular intervals, exactly on the boundary of a second, then the source is likely associated with a timing component. This is due to the precise timing

infrastructure that is used throughout the site. The infrastructure uses 1 Hz and 16 Hz fundamental architecture, derived from GPS clocks and quartz oscillators trained on the GPS clocks.

The channel which serves as witness to the noise is also very informative as to the source. For example, if there are many witness channels all over the site (such as a number of seismometers inside of the end stations), or in one isolated location (for example, a microphone only in one end station). In either case, without any more information than the location and nature of the witness, there is often enough information to begin following up the witness with further investigation. The investigation procedure for each noise source will necessarily vary with each source, and each coupling mechanism.

An investigation of any given line or comb should lead to a number of potential coupling mechanisms. Once a candidate coupling has been considered, it can be tested. In order to test a coupling, we can either attempt to reproduce the noise seen in the GW data, by introducing a known input into the coupling mechanism, and measuring the output through the GW channel. Alternatively, we can test the coupling by attempting to remove or bypassing the coupling, and measuring any change in the noise seen in the GW data.

The first method is useful if the noise is no longer present in the detector. This would be the case for transient noise, or if the detector is operating at reduced sensitivity whilst it is out of observing mode. Whilst this is useful for identifying the noise coupling, in order to mitigate the noise source and stop the noise from reoccurring, the coupling of the noise into the GW channel must be removed.

The second method is particularly preferable if the noise is continuous, and still visible in the data whilst the detector is not observing. If the detector is able to maintain sensitivity when removing or bypassing the coupling, then identification of the coupling and removal of the coupling are both achieved at once.

In either case, if the candidate coupling method is not correct, then an alternative hypothesis must be formed, and another candidate coupling tested.

As outlined so far, detector characterisation encompasses a broad range of hardware at the detector, software techniques for viewing the data from GW data, and its relationship to many witness sensors, and through them, the environment at large. In chapter 3, we will discuss how these detector characterisation techniques operate when applied to real-world noise sources affecting the detectors.

2.3 Summary

In this chapter, we briefly discussed the history of gravitational wave detectors, from resonant bar detectors, to early GW IFOs, and ending with current second generation IFO detectors, such as aLIGO and adVirgo. We outlined some of the limiting noise sources for GW IFOs, and methods of reducing the noise where possible.

Then, we turned to focus on data quality issues that affect persistent GW searches. We started by describing frequency and time-frequency domain representations of detector data, and how persistent noise artefacts (lines, combs and wandering lines) appear in these representations.

The PEM sensor suite was described, and its role in witnessing local noise, but not GWs. These sensors can be used to investigate the source and coupling of noise sources. Once a noise artefact has an associated witness, mitigation becomes much easier, and the noise can be flagged in a catalog, such as a line list.

Chapter 3

Detector Characterisation in S5, S6, O1 and O2

The broad range of possible noise sources, analysis techniques and mitigation strategies that are involved in detector characterisation requires flexible treatment of noise. We describe three detector characterisation studies. In section 3.1, we look at a noise source resulting in the broadening of a 60 Hz line in S5 and S6. We show how we might hope to remove contributions due to this source after the observation is complete, and attempt to investigate the coupling mechanism using the existing data from S5 and S6. In section 3.2, we explore the process of compiling a list of known instrumental and environmental lines and combs in aLIGO's O2. Finally, from a similar list compiled for O1, we then demonstrate the characterisation, investigation and attempted mitigation of a 1 Hz comb in section 3.3.

3.1 60Hz line broadening in S5 and S6

Data from iLIGO is still be useful to searches for persistent GWs. Though the detectors were not as sensitive to GWs as the advanced detectors are, persistent signals detected in advanced detectors should still be present in the initial detectors. We can use the initial detector data to marginally increase the significance of a source, and to show

the source’s continuity over a period of several years. One such source for which we can hope to show this is the pulsar PSR J0534+2200, known as the Crab pulsar.

3.1.1 The Crab pulsar

The Crab pulsar is a high priority source for targeted searches for CWs [157]. It is a young pulsar, only about 1 000 years old [159]. Its age makes it a particularly exciting potential CW source, as young rotating neutron stars are thought to be more likely deformed than older stars, and therefore more likely to emit detectable GWs [160]. Additionally, the Crab is a well studied pulsar, with very accurately known position and timings, which makes searching for a CW signal a much smaller task, given the narrow prior volume. The Crab pulsar is known to “glitch” [161] (we discuss NS glitches in more depth in chapter 4). For the purpose of the prospect of CWs from the Crab, its glitching implies that the NS is still young enough to be active, another indication that it is a likely CW emitter.

As a relatively interesting, young, glitching galactic pulsar, its frequency evolution has been well monitored since long before iLIGO began taking data, and its frequency evolution remains well tracked. The Crab’s rotation frequency, f_{rot} , at the time of iLIGO’s S5 and S6 evolved between 29.746 Hz and 29.72 Hz [62]. As the GWs emitted from a simple triaxial rotating NS (as discussed in section 1.3.1) are at $2f_{\text{rot}}$, it would be picked up at the iLIGO detectors at between 59.492 Hz and 59.44 Hz. In an ideal detector, this would pose no issue at all, however for iLIGO, this lies very close to one of the loudest environmental noise lines; the line caused by the local 60 Hz AC mains frequency.

3.1.2 The “Crab killer” noise

In iLIGO’s data, the PSD at LLO around 60 Hz differs from that at LHO. They both see a strong, narrow peak at 60 Hz, which corresponds to the mains power transmission frequency in the US, however LLO’s 60 Hz line sits on a “hump” in the PSD, which is not present in the LHO data. This hump has been called the “Crab killer” noise, because at LLO, it serves to raise the noise floor at $\approx 60 \pm 5$ Hz, potentially obfuscating GWs from the Crab pulsar. So much so, in fact, that in the final iLIGO upper limit estimates for the Crab’s GW emission, LLO data did not greatly contribute [142].

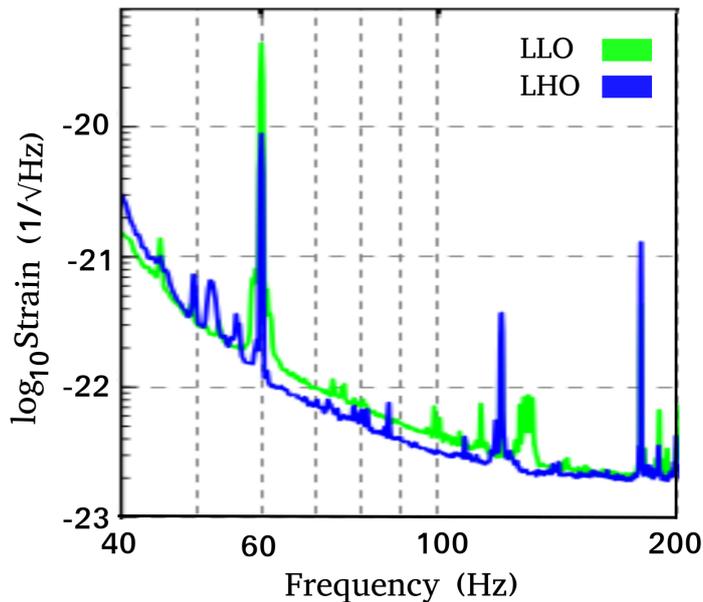


FIGURE 3.1: A spectrum of strain taken from iLIGO’s sixth science run. The 60 Hz line is broadened at LLO (green), but not at LHO (blue). This is thought to be due to the Crab killer noise under examination [162].

A brief note on terminology, as the mains electricity cables are often called power lines, which is likely to be confused here with the terminology for instrumental noise lines. The delivery method for electricity shall be called the “mains transmission cables” or “power transmission cables”, whereas the detector noise will be called the “60 Hz line”, or “60 Hz noise line”.

3.1.2.1 Previous investigations into the noise

An investigation into the physical source of the 60 Hz line broadening was conducted some time ago by site staff at LLO [155]. In this subsection, I will summarise this investigation, and its findings.

As the noise is located around 60 Hz, the first and most obvious potential source for the broadening was in the mains power transmission. LLO is situated relatively near to a power substation, and a power transmission cable runs over both of the arms of the IFO.

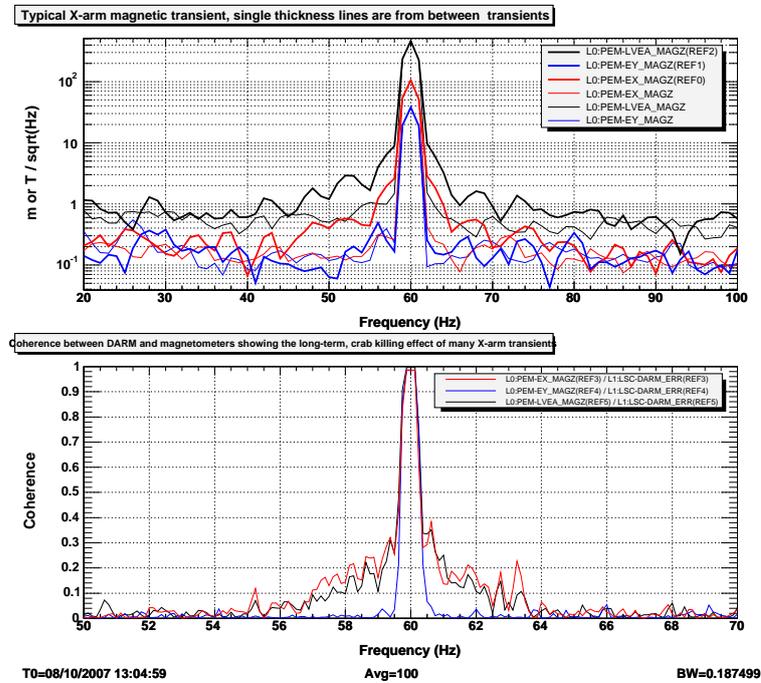


FIGURE 3.2: The Cross-channel coherence between various magnetometer channels and the DARM_ERR uncalibrated strain channel at LLO. The raised hump around 60 Hz implies noise coupling between these two channels [155].

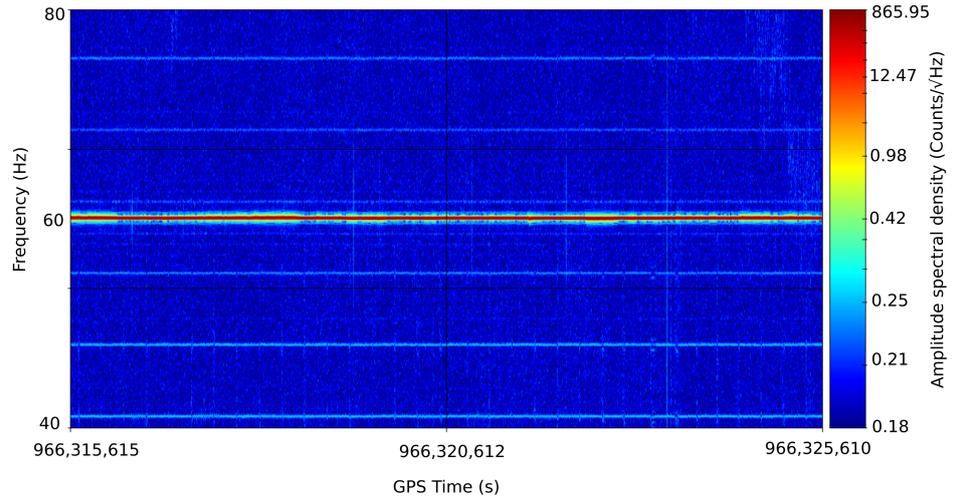
The cross-channel coherence between the PEM magnetometers and the GW data channel was calculated and plotted by Robert Schofield [155]. The plot, shown in the lower panel in figure 3.2 shows a broad base to the 60 Hz peak, corresponding to the hump seen in the spectrum in figure 3.1. Additionally, the magnetometer data at the LVEA

(located in the corner station) and the X-arm end station EBAY both showed loud, broad features near the 60 Hz line, which we call “glitches”. The Y-arm end station also showed glitches, though they were not as loud.

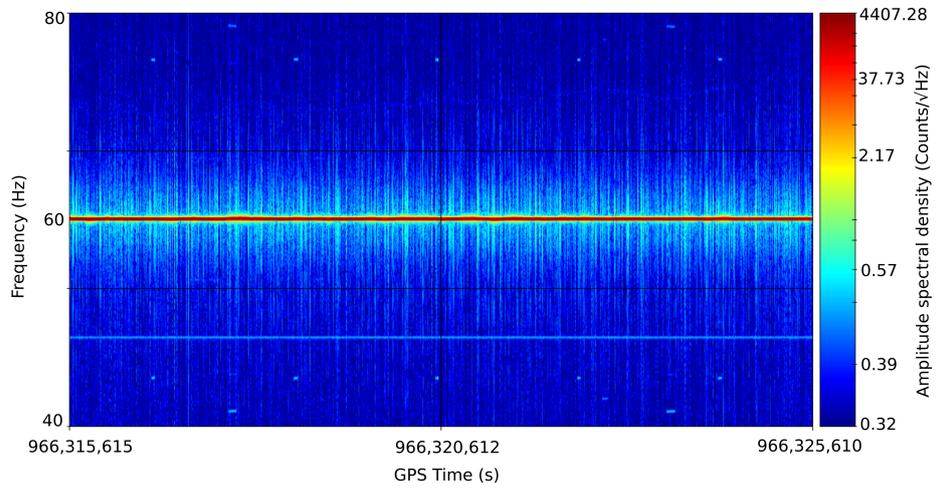
Continuing on the investigation, Robert Schofield and Anamaria Effler placed temporary magnetometers along LLO’s X-arm. The power in a narrow band around 60 Hz showed the presence short transients, matching those seen in the PEM magnetometers at the X-arm and corner station. Additionally, it was found that the strengths of these glitches running through the beam tube was significantly stronger than when observed from as little as 10 mm away [155], indicating that currents were travelling through the beam tube.

Further effort by Josh Smith and Joe Areeda was then put into characterising these glitches, and their rates. Spectrograms from the LVEA magnetometer and X-arm magnetometer channels were made for coincident times at both LLO and LHO. There was a clear difference in the width of the 60 Hz line in each site, with LHO’s being clean, with very little modulation or broadening, whilst LLO’s was noisy, and its power spread wide, as seen in figure 3.3.

A glitch counter was created, which heterodyned the magnetometer data to a frequency close to 60 Hz, and then high-pass filtered the resulting data, such that the slow evolution of the 60 Hz noise line was removed. A threshold was arbitrarily set on this modified data, and any glitches which exceeded the threshold power were counted. With this, histograms of glitch count were created in 30-minute and 24-hour bins. The transients showed clear diurnal behaviour, with the glitch rate peaking before 9am locally, and falling off after 6pm. This showed that the source was ultimately anthropogenic, and indicated that the nearby power transmission cables were likely the cause. Further investigation found that the Crab killer glitches were coincident with similar glitches at a local power substation. Grounding faults at this local substation caused a current to conduct through a petroleum pipeline which runs underneath LLO. These



(A) LHO



(B) LLO

FIGURE 3.3: Spectrograms of the 60 Hz line at (a) LHO and (b) LLO. LLO has a much higher noise level around the 60 Hz line – these are caused by the glitches [155].

ground currents coupled into the detector through the beam tube.

3.1.3 Software mitigation of the Crab killer

Though the S5 and S6 observations have long since finished, and the instrument has undergone a major upgrade since, there is still the prospect of removing the glitches from the S5 and S6 data. This could yield more sensitive upper limits on the Crab pulsar’s h_0 value by combining data from O1 and O2 with S5 and S6, potentially

improving the detection prospects for the Crab pulsar in S5 and S6 data. In this section, I detail a potential new data based veto I created, aimed at mitigating these glitches.

One possible method to remove the crab killer glitches from the data would be by removing the data at times corresponding to the glitches, via a data veto. Such a veto would use a thresholding method similar to the glitch counter in order to identify a glitch. The algorithm would calculate the power in a band around 60 Hz in a witness magnetometer channel at a time t_i , $\{P(t_i)\}$. The power is then compared to a threshold power, T . From this comparison, we create a masking function $m(t_i)$, such that

$$m(t_i) = \begin{cases} 1 & \text{if } P(t_i) \leq T, \\ 0 & \text{if } P(t_i) > T. \end{cases} \quad (3.1)$$

The mask can then be applied to the strain data as in equation (3.2), such that

$$d(t_i) = m(t_i)h(t_i). \quad (3.2)$$

This mask is tantamount to replacing the data its expected mean value, 0. The amount of data removed can easily be tuned by raising and lowering the threshold power. The length of time removed for each transient event is such that so long as the power $P(t_i)$ remains above the threshold T , the data will be masked. The amount of data masked should be minimized, so that the analysis pipeline still has a large portion of the data to analyse. The increase in sensitivity to the Crab should outweigh the losses associated with reducing integration time, else the pursuit would be fruitless.

Creating such a filter, I applied it to a 24 hour-long set of data taken from iLIGO's S5. This length of time was chosen as the Crab killer hump could be seen with 24 hours of integration time. I was able to check whether the filter may have any significant effect on the sensitivity of a CW analysis. In order to determine whether or not the filter had

any effect, I simulated a rudimentary CW search centred on the Crab pulsar's phase evolution to first order ($\Phi_C = \Phi(\phi_C, \dot{\phi}_C, t)$). In this rudimentary search, I simulated a lightweight version of the Bayesian targeted search pipeline described in section 1.3.2.1, and considered the sum of the heterodyned data as a figure of merit. This rudimentary heterodyne was calculated by multiplying the masked and resampled strain by a time-evolving phase,

$$c_i = d_i \exp[-i\Phi_C(t)], \quad (3.3)$$

which approximated the Crab's evolution. As described in section 1.3.2.1, I then applied a ninth-order Butterworth low-pass filter with a knee frequency of 0.25 Hz. This filter is an all-pole filter, in this case, there are nine poles, that lie around a circle of radius $2\pi f_{\text{knee}} = \pi/2$. In order to see the effects of the data-based veto, the heterodyne process was repeated on the same data, which had not been masked. The two heterodyned time series were then compared to each other. In order to compare exactly the difference between the filtered and unfiltered data, their differences were calculated as a time series.

The difference between the masked and unmasked data in all test cases was found to be exactly zero. When plotting the coherence between the masked strain and magnetometer, the same noise structure was seen around the base of the 60 Hz line.

The assumption up until this point was that there was a direct coupling between the magnetometer glitches and the broadening of the 60 Hz line. A filter designed to remove the effects of the glitches did not have any impact on the data. Rather than pursuing further, I instead decided to focus on verifying that the magnetometer was truly a witness to the Crab killer noise, and to better understand the coupling between magnetometer glitches and the elevated noise around 60 Hz, in particular, in the time domain.

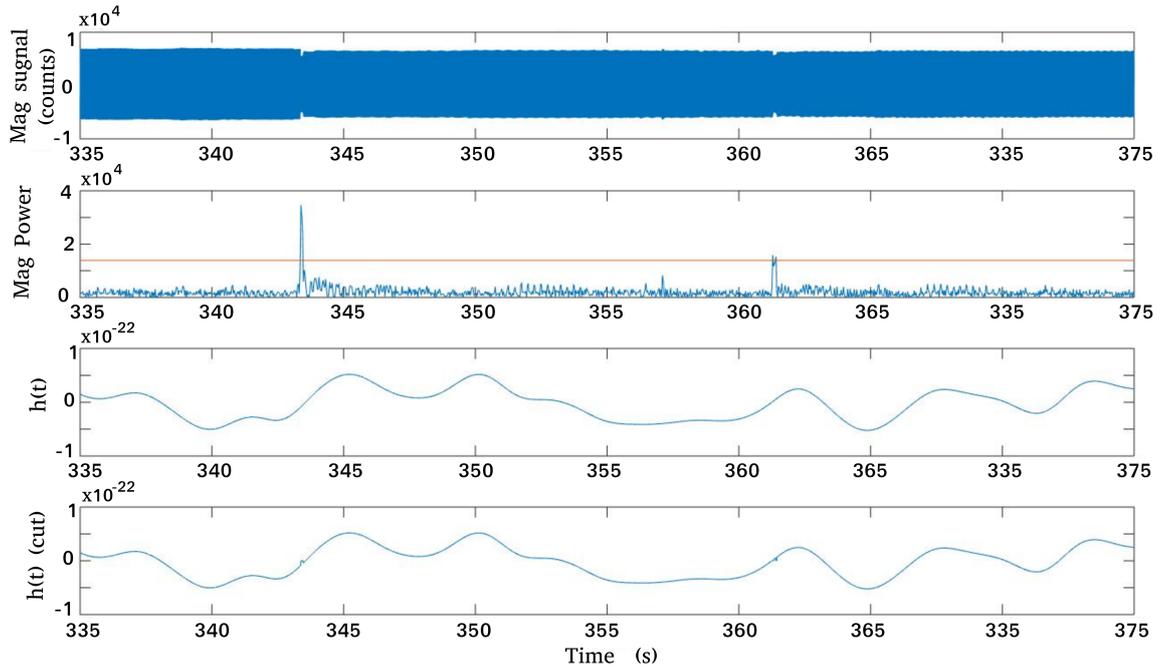


FIGURE 3.4: Results from the data-based veto. The top plot shows the magnetometer time series. The second shows the high-pass filtered magnetometer power in the 40 Hz - 80 Hz band. The red line in this plot is the threshold power. The third plot shows the output of the heterodyne and strong low pass filter for data without the masking function applied, and the fourth shows the same with the masking function applied. All four plots share a common time axis.

3.1.4 Crab killer coupling in the time domain

Whilst a sensible candidate source of the transients has been identified, the coupling between these transients and the excess power in strain is based on their correlation in the frequency domain. We must also find evidence of this coupling in the time-domain if we are to apply a data-based veto as described in section 3.1.3.

To begin, I simply examined raw data from the X-end EBAY magnetometer and the strain channel in the time domain. I used the thresholding method implemented in the filter in order to select a short time around a glitch. Figure 3.5 shows 20 seconds of data, including a glitch at about 752 s.

We see no obvious glitch in the strain data, in the third panel of figure 3.5. This is not surprising, as the strain data is not dominated by the glitch, whereas we expect the

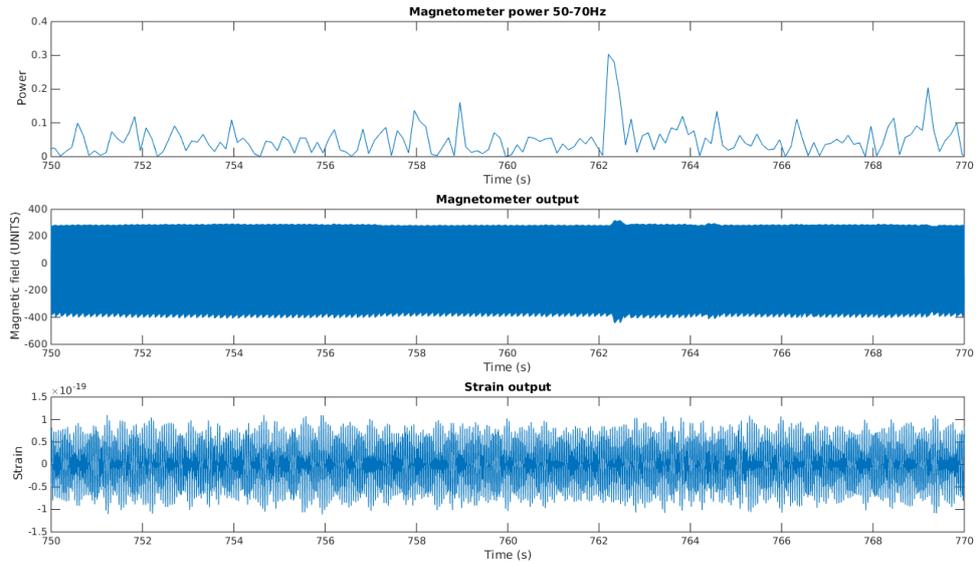


FIGURE 3.5: By looking at transient events in the time domain, we see the glitches in the magnetometer power (top panel), which can be seen in the raw magnetometer data (second panel), but not the strain channel (third panel).

magnetometer to be dominated by the 60 Hz line, and therefore, the glitch.

It may be more instructive to view this in the time-frequency domain. I created spectrograms of the strain and magnetometer data (shown in figure 3.6). I used a time resolution of 0.5 s, and a frequency resolution of 2 Hz to create these spectrograms. The top panel in figure 3.6 shows the frequency composition of the magnetometer channel. We can clearly see the glitches here as wide lobes on either side of the 60 Hz line. The lower plot shows a spectrogram of the strain channel during the same time period. A low pass filter has been applied to the raw strain to attenuate the low frequency noises, and clean up the plot. In addition, the amplitude range of the spectrogram has been chosen on a logarithmic scale in such a way as to accentuate the desired features. The strain spectrogram did not show similar glitch features. This may imply that the coupling is not straight forward, or that this effect may be much smaller than anticipated.

One further test was to produce a correlation plot of the two channels – simply a scatter

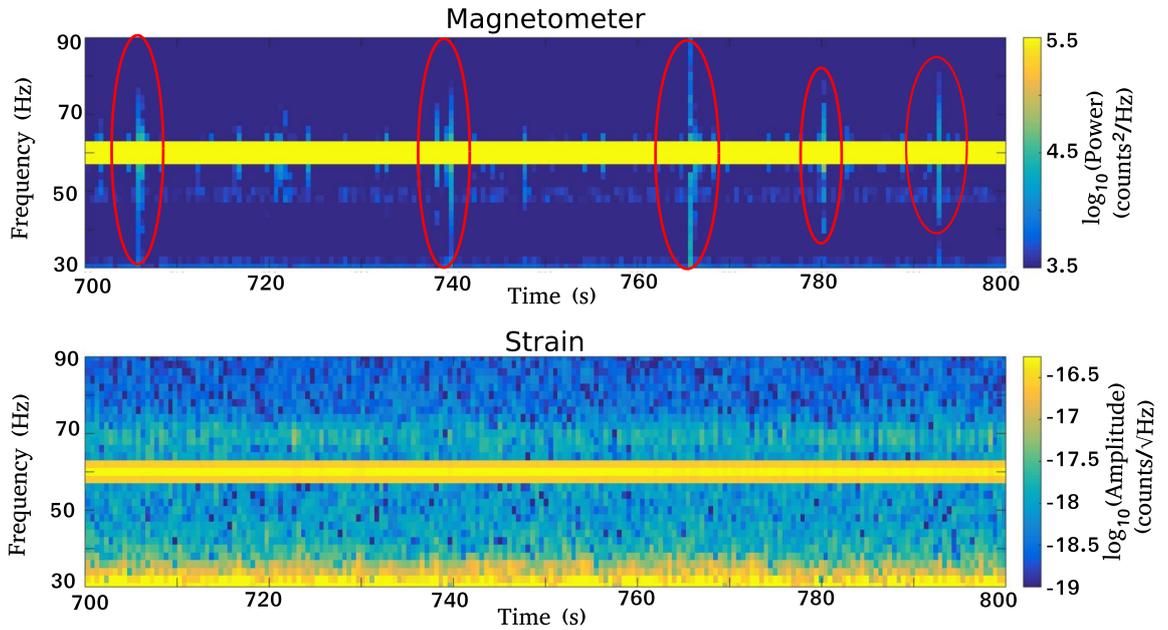


FIGURE 3.6: Spectrograms of (top) the LVEA magnetometer channel and (bottom) the strain channel. Transients seen in the magnetometer channel (circled red) cannot be seen in the strain channel.

graph of samples, at the same sample times, from the two channels. In addition, in order to only find a correlation for the glitches, each channel was heterodyned to a frequency of 59.5 Hz, and with a strong 0.25 Hz filter, removing the contribution from the 60 Hz line. An instantaneous coupling of the loud magnetometer glitch to a similar glitch in strain would manifest itself in the data as a correlation between the x and y coordinates of the samples, with a positive gradient. As shown in figure 3.7, there is no clear trend, implying that the data are not strongly correlated, as we would expect.

Rather than continuing to search for a correlation between the glitches in the magnetometer and excess noise in the strain data, I decided instead to broadly divide the data between times concurrent with a magnetometer glitch, and times without a glitch. To divide the times, the same heterodyne, high-pass filter and thresholding was applied. Strain data corresponding to a glitch, along with the timestamps of those samples, were added to a data series called $\{d_G\}$. All other data and timestamps were added to a data series called $\{d_{NG}\}$. The subscripts G and NG stand for “glitch” and “no

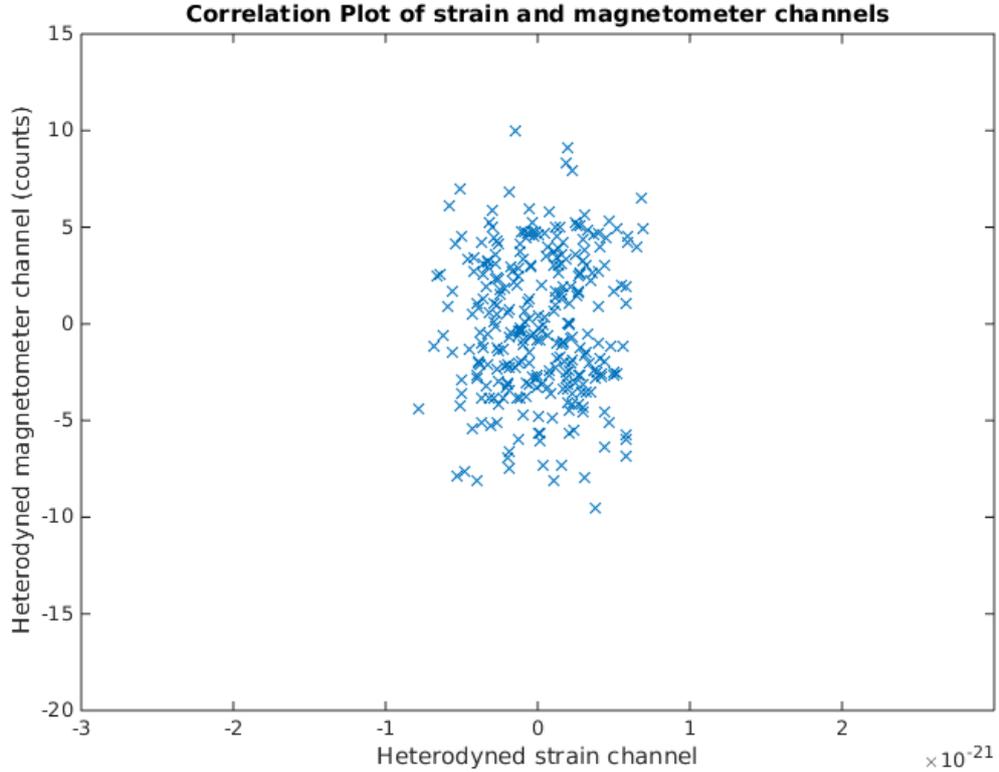


FIGURE 3.7: A correlation plot of data heterodyned to a frequency of 59.5 Hz. We expect transient effects, causing a positive correlation between a sample's x and y position.

glitch” respectively. The cross-power spectrum between the magnetometer and strain channels in each stream was calculated. These can be seen in figure 3.8.

The cross-power of the glitch-stream data was found to be greater than that of the other stream, which is consistent with the findings of previous studies by Robert Schofield and Annamaria Effler [155].

Next, in each stream of strain data, the power at 61 Hz was calculated, P_G^h and P_{NG}^h . We expect the power at 61 Hz during a glitch to be higher than at times without a glitch. The ratio $R_{NG}^G = P_G^h/P_{NG}^h$ was calculated, and found to be 1.14, indicating a slight excess of power in the glitches in the strain data.

Alone, this quantity tells nothing, as it says nothing about the expectation or background value of such a ratio. I performed a time-slide analysis in order to obtain such

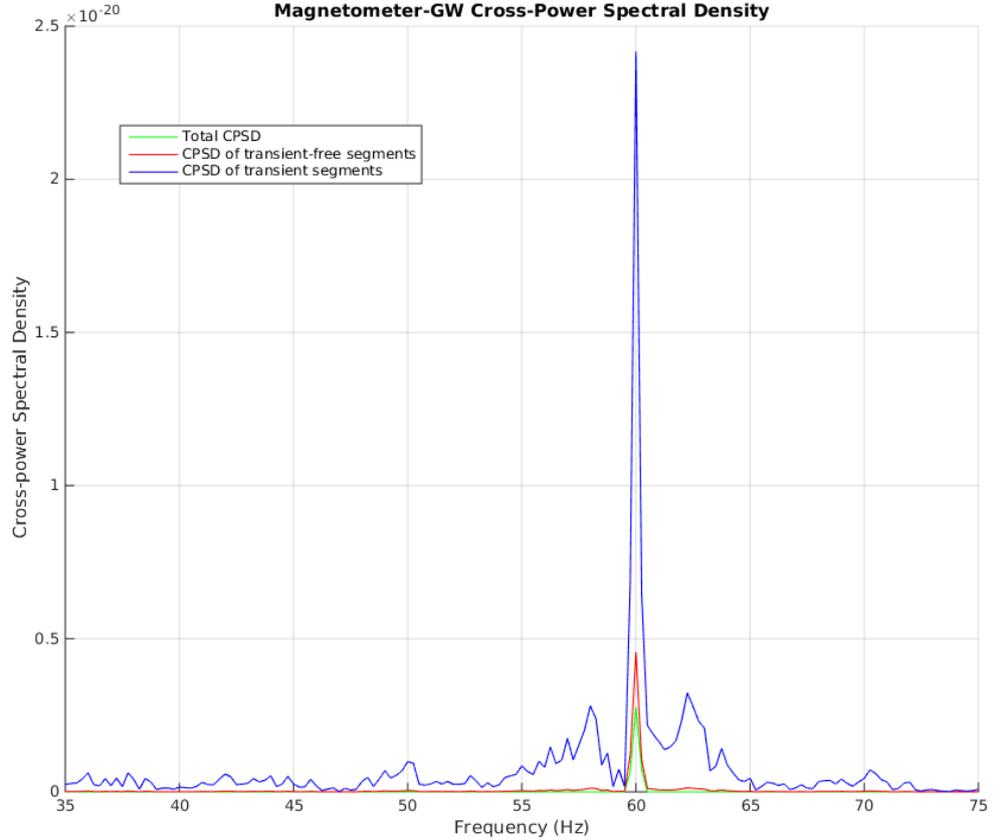


FIGURE 3.8: The plot shows the normalised CPSD for each of the bins, and the total CPSD.

a background value for R_{NG}^{G} , where P_{G}^h does not coincide with glitches. A time-slide analysis involves misaligning two datasets in time, and repeating an analysis at several different time-misalignments. It is a method of obtaining a background coincidence rate between two sets of data. During a time-slide analysis, the first step in which I determined the time of a glitch was altered. The magnetometer power data is padded with leading zeros whilst being compared to a threshold value, whilst the strain is padded with trailing zeros in order to create a positive time-slide. This means that when the mask is applied to the strain, it is now masking out a time with no real significance to the time of the transient.

The power of the time-slide data were calculated as before, with the ratio R_{NG}^{G} recalculated at each time-slide, and the ratio R_{NG}^{G} was plotted against the time-slide in figure

3.9.

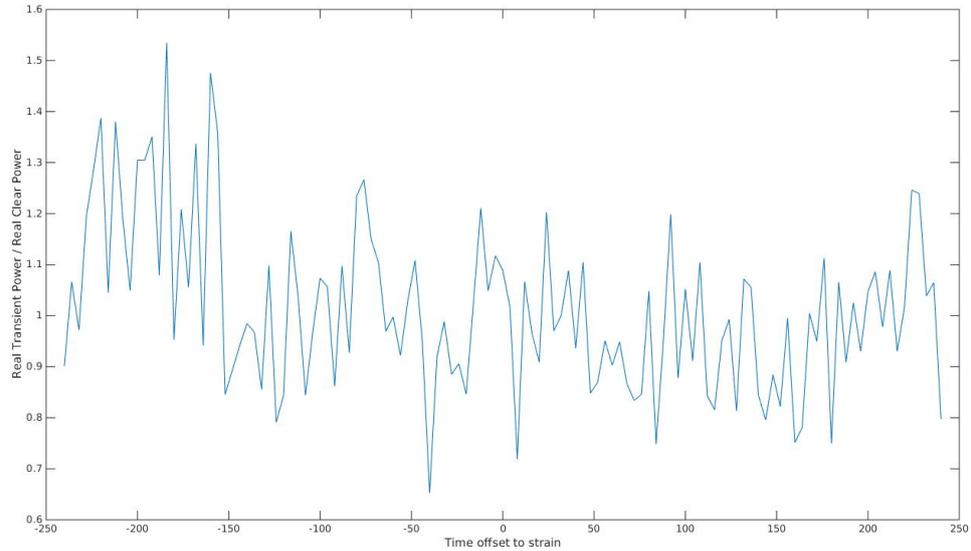


FIGURE 3.9: This plot shows the time-slide size against the ratio R_{NG}^{G} . The plot shows no peak at zero, implying that the glitches in the magnetometer are not coincident with excess power in the GW data channel.

Additionally, I applied this time-slide methodology to examine the cross-power spectral density of time-slide data streams. Figure 3.10 shows the cross-power of different time-slides. In each case, the elevated hump around the base of the 60 Hz line is present.

These tests have not shown any coincidence between excess power in the magnetometer in a band around 60 Hz, and a similar excess of power at the same time around 60 Hz in the GW data channel. Instead, it seems to show that there are fluctuations in the power around 60 Hz in the GW data stream (as shown in figure 3.9), and that the cross-power between the magnetometer and strain channel always shows an elevated noise floor around 60 Hz. I concluded that the coupling between the magnetometer and GW data is not sufficiently well understood to create a data-based veto which would enable us to mitigate the broadening of the 60 Hz line. However, the magnetometers at LLO's LVEA and X-arm EBAY are likely witnesses to the noise source causing the broadening of the 60 Hz noise line, associated with the mains transmission cables. It is

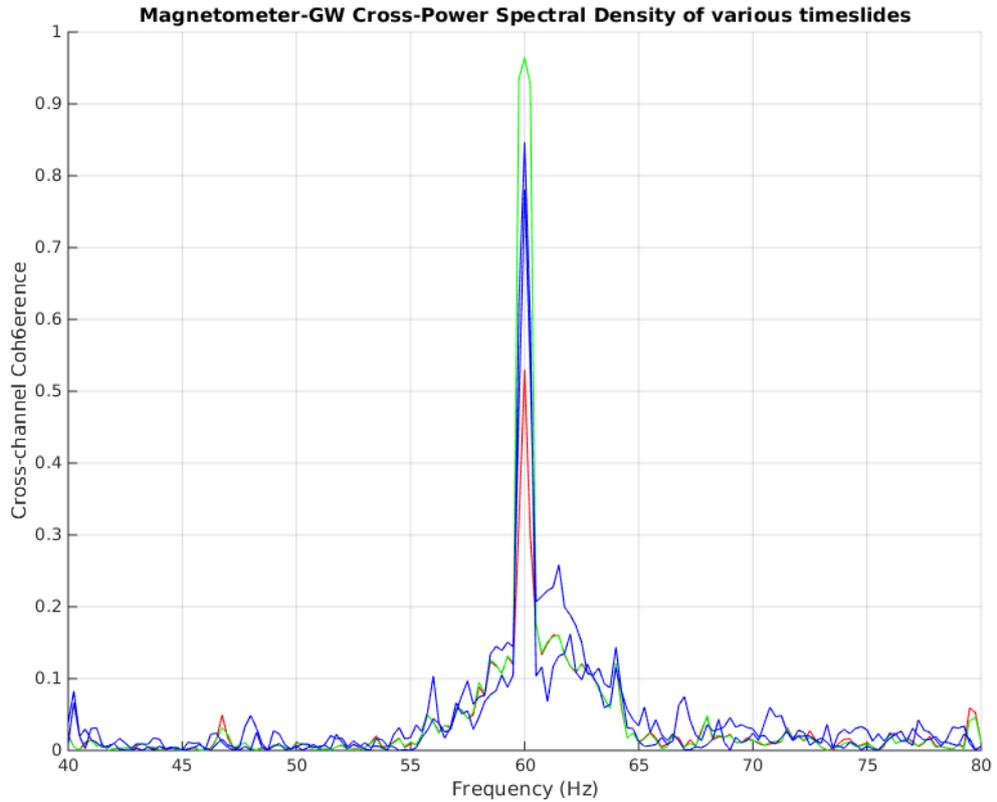


FIGURE 3.10: Each trace in this plot shows the cross-power spectral density of the glitch-only data stream. Each trace is a different time-slide. The green trace corresponds to no time-slide, whilst the blue traces correspond to time-slides of ± 5 s.

likely that the elevated cross-power was caused by the increased magnetometer signal, as cross-power is simply a product of the amplitudes of two channels.

3.1.5 Eventual mitigation of the Crab killer

During the upgrades between iLIGO's S6 and aLIGO's O1, the grounding along both of LLO's IFO arms, and the power transmission cables were addressed. By comparing the coherence of a witness magnetometer and the GW channel in S5 to the coherence of an analogous magnetometer and the GW channel in O1, I checked that the crab killer noise was mitigated in hardware.

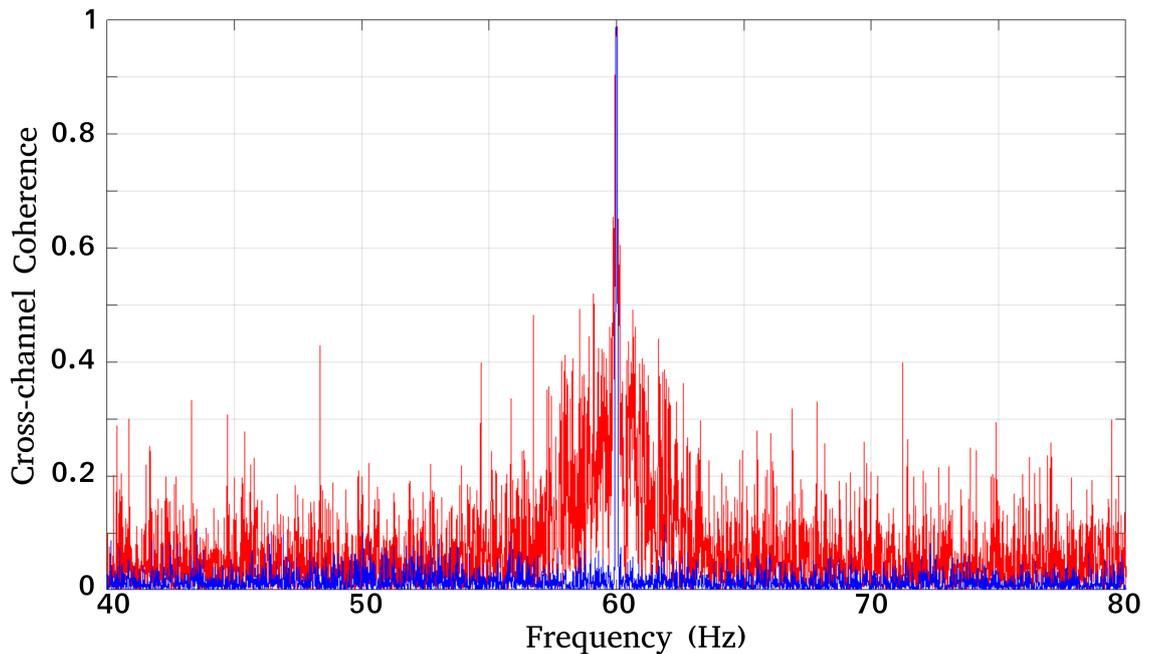


FIGURE 3.11: This plot shows the coherence between a witness magnetometer and the GW channel in (red) S5 and (blue) O1. The raised noise floor around 60 Hz which is present in S5 is no longer present in O1.

We see that in O1 data, the raised noise floor around 60 Hz is no longer present. This is seen in figure 3.11. Though I was unable to remove the Crab killer noise from iLIGO data for re-analysis, the noise is no longer present in aLIGO, increasing the prospects for more sensitive upper limit calculations for h_0 from the Crab pulsar.

3.2 Compiling a line list for aLIGO's second observation

The first observing period of aLIGO (O1) took place between September 2015 and January 2016. This was the first observation by a second generation GW observatory, and at the time, it was the most sensitive observation so far. By iLIGO's fifth and sixth observations, the iLIGO detectors had been well characterised, and their data was well understood. Conversely, prior to O1, there was very little study of the data produced

by the aLIGO detectors. As a result, the data quality throughout O1 was generally worse than that of S5 and S6. Figure 3.12 shows the contamination of the low-frequency band (≤ 100 Hz) by lines, which are visible with just one week’s integration of O1 data.

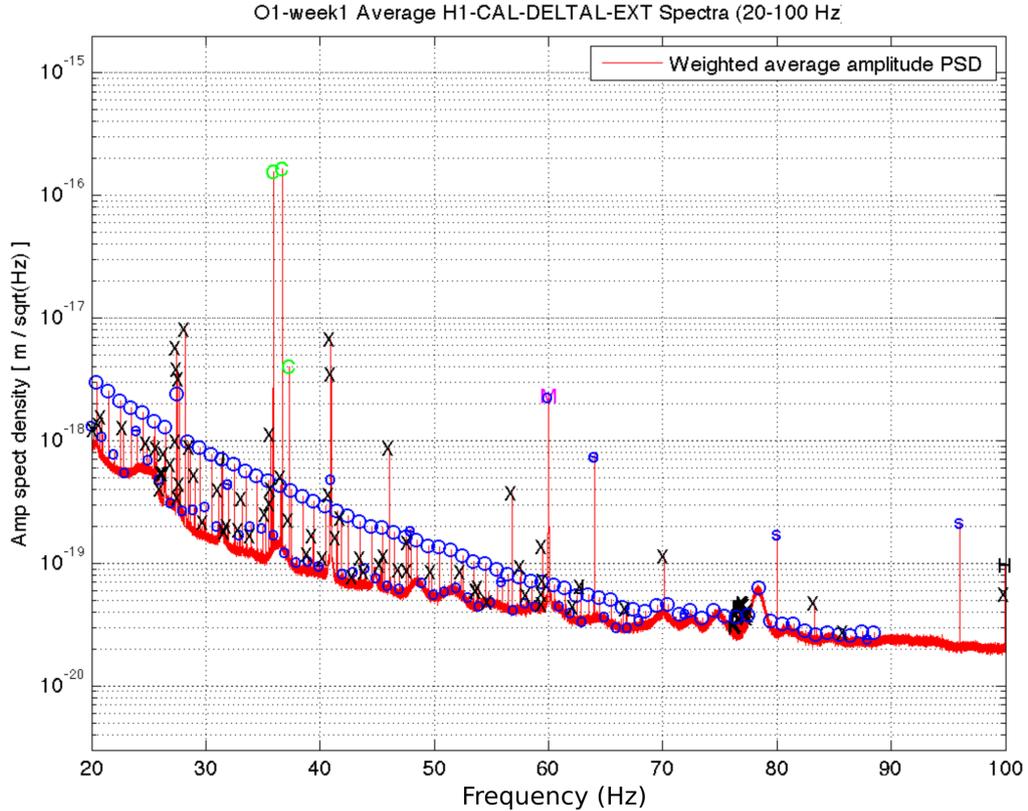


FIGURE 3.12: The first week of O1 data at LHO shows a high volume of loud lines in the data, marked in this plot by “O”s and “X”s [163]. These markings are not relevant for our purposes.

Many of the most troubling combs that ran throughout the GW band were investigated in the interim period between O1 and O2, aLIGO’s second observation (which we explore in greater depth in section 3.3). However, during this time between observations, both LHO and LLO saw upgrades and maintenance. When the observations began again on December 1st 2016 for O2, the detector was not in the same state that it was at the end of O1. In order to analyse the O2 data, we should first characterise the data. After the close of O2 in August 2017, work began to study the data quality of O2 for narrow artefacts.

3.2.1 Motivation for creating a line list

O2 was longer than O1 by a factor of about two, though the strain sensitivity was roughly the same. This allows for an analysis roughly $\sqrt{2}$ times more sensitive to CWs, and similarly, reveals instrumental lines with $\sqrt{2}$ times more sensitivity. Given the population of lines visible during O1, we expected that there were likely to still be several instrumental lines in the data, particularly at low frequencies (≤ 100 Hz). However, many of the most pervasive combs visible in O1 had been mitigated, and so we expected the data to be generally cleaner than O1's. It remained important to characterise lines and combs which could limit the sensitivity of searches in the O2 data. Whilst there are a few methods that search pipelines use to deal with instrumental lines (notching, vetoing), this relies on the lines being known, and associated with local environmental and instrumental noise sources. As such, they rely on a list of lines to inform the searches. Further, as we discuss in section 3.3, a line list provides a solid beginning to any mitigation attempts for lines and combs, which can be critical for the improvement of the next observation's data quality.

3.2.1.1 Scale and scope of a line list

When compiling a list of known lines, it is not sufficient to simply note the frequencies of narrow peaks in the data. Besides a line's width, it is also important to note any and all witness channels, and, if the line is not present throughout the observation, to note the start and end times of the line. Similarly, it is also worthwhile to note any combs, giving the width of each tooth, the tooth spacing, offset from 0 Hz and the visible harmonics.

In order to confirm that a line is non-astrophysical, any line in the line list should have a witness channel from the array of PEM channels. Any line without a corresponding PEM witness cannot be deemed safe to remove from the analysis, as there

is no reason to believe it to be non-astrophysical simply because it is loud. Combs are not consistent with expected astrophysical signals, and so we do not require witness channels for combs. However, if a witness for a comb exists, it should be noted as it can greatly help to have a witness when attempting to remove the comb's source and coupling from the detector. The witness channels are generated by tools such as NoEMi [164], which finds and tracks noise lines in the data, by calculating the coherence between the strain channel and a wide range of PEM sensors.

3.2.2 Discussion of O2

The second observation with aLIGO was not without its issues. Prior to beginning the observation, during the commissioning period between O1 and O2, both sites encountered problems when increasing the power of the input laser. O1 had used an input power of 25 W, and the plan for O2 was originally to double that to 50 W. This resulted in the laser power at LHO remaining at 25 W for O2, though squeezing was implemented. Conversely, LLO achieved 50 W input power, though squeezing was not implemented. The result of this difference was that LHO was generally slightly less sensitive than LLO throughout O2.

The sensitivity of LHO further suffered through O2. LHO's desert environment provides no buffer for high winds, which introduce low frequency noise into the detector, by pushing against the housing structures, and applying a small tilt to the portion of the IFO within. This can cause the detector to fall out of observing mode (or "lose lock"). Additionally, LHO's location in the pacific north-west of the USA places it much closer to several tectonic fault lines than its counterpart LLO in Louisiana. As such, LHO's sensitivity and duty factor (the proportion of time for which the detector is operating) is limited. During O2, and following an earthquake near Alaska, LHO's data quality was particularly bad for a period of a month, between March 14th and

April 14th. When creating a list of known lines, due to the limited time available, this period of poor data quality was omitted.

LLO was not affected nearly as much by local environmental factors, and as such, generally had better data quality during O2. However, low frequency seismic noise is a major concern at the site, coupling from micro seismic activity caused by human activity in the nearby forests. Nevertheless, this did not represent a limiting noise source during O2. Similarly, the period of poor data quality at LHO was not mirrored at LLO, and data from the entire observation was used to create a line list for LLO.

3.2.3 Methodology

Though tools exist to find transient noise events and persistent noise lines, correlating to PEM channels, as we saw in section 3.1, a correlation does not always tell the entire story. As such, we require human action to examine the spectrum of a given detector, and identify lines and combs in the spectrum. Compiling a list of all observed lines in the detector is the most time consuming portion of this exercise, as the entire CW analysis band has to be studied, from 20 to 2000 Hz. Once a list of narrow artefacts in the detector has been generated, each line can be checked against existing lists from previous observations, and tools such as NoEMi. Only artefacts which appear both in the spectrum, and on an independent source (NoEMi, BruCo, previous line lists) are included in a list of known instrumental lines. Combs can be added to this list also. A wandering line may appear as a broader “bump” in the spectrum, which look distinctly different than a line or comb tooth. If a wandering line is also present in both the spectrum and in independent list, then it can also be added to the list of known lines, importantly noting its width. This known lines list contains all lines which can be safely vetoed from searches, though through the process, there will likely be a number of lines which are not included in this list.

In order to identify lines in O2, an effort, led by Evan Goetz, and including myself, Pep Covas and Ansel Neunzert was undertaken. Based on the detectors' performance throughout O2, a "stable" period was determined (by analysing the electronic logbooks at each site [165, 166]) by Evan Goetz which would be searched for lines.

3.2.3.1 Plotting the spectrum

Once the segments of viable data were decided, Pep Covas plotted data from each detector in the frequency domain. The configuration used was a Tukey windowed 7200 s long FFT. For each detector, the spectrum was split into bands of 100 Hz width for convenience.

The work of searching the detector spectrum was split between the four researchers, each given a band in a detector. Each detector was split into a low- and high-frequency band. Between 20 Hz and 100 Hz was considered low frequency. This region was searched by Evan Goetz (LHO) and Ansel Neunzert (LLO). Between 100 Hz and 2000 Hz was considered to be high-frequency. These bands were searched by Pep Covas (LHO) and myself (LLO). We chose to split the section around 100 Hz as, in O1, below about 100 Hz was the most polluted region of the spectrum. For each artefact, the central frequency, left- and right-widths and artefact type (line, comb or wandering line) were recorded.

3.2.3.2 Special cases: combs and violin modes

At low frequencies in the spectrum, below about 150 Hz, combs were common. Often these combs were easy to find. At higher frequencies, there were very few combs in LHO, however several combs were clearly visible in LLO. Most common among them were several combs with spacing of 0.6 Hz, visible at many offsets from 0 Hz, and sporadically clustered in groups between 440 Hz and 1600 Hz. Each tooth on this comb

was very narrow, suggesting an electronic coupling, however there was no witness to any of the combs found in NoEMi.

Additionally, the higher frequency band contains the suspension violin modes, and their harmonics. Each violin mode is individually resolvable in the spectrum, resulting in a broad footprint of the violin modes on the spectrum. In total, there are 16 violin modes for the test masses (as there are four test masses, each with four fibres), as well as the beam splitter’s violin modes, which occur at lower frequencies. The regions containing these violin modes tends to contain a number of additional lines, as other modes of the suspensions (such as the bounce, roll, pitch and yaw modes, which are low frequency, about 5 Hz by design) and seismic noise peaks upconverted to the frequency of the violin modes. Additionally, the violin modes themselves are broader than many other lines, as they represent a direct mechanical coupling into the detector. Conveniently, as the violin modes for the test masses are all within a band roughly 20 Hz wide, we choose to veto the entire width of the violin mode band.

At higher frequencies, harmonics of the violin modes also couple into the data, however, they appear more spread out. The noise floor between the higher violin modes at higher harmonics is clearly visible, where it was not at the fundamental frequency. Rather than reporting increasingly wide bands to veto with each harmonic, each resolvable violin mode harmonic is reported as independent entries into the line list.

3.2.4 Results

The resulting line lists were reported in the appendix of [167]. In the first table, each line and comb (noting the width of each tooth, spacing between teeth and offset from 0 Hz) constitute a single entry. The table includes known lines and combs for both O1 and O2. The final vetted line list for LHO during O2 consisted of 140 entries, reduced from an initial list of 688 unvetted entries. Similarly, LLO’s line list was reduced from 353 unvetted entries, to 88 entries in the final list. We see that although LHO had

many more individual entries, LLO had many more combs at high frequencies. My contribution to the lines list is included in appendix B.

Although little characterisation of these lines was done beyond their presence in the data and witness channels, the line list gives a good point from which to start an investigation into a given line or comb.

3.3 Investigation of a 1Hz comb

Following O1, Keith Riles created a similar line list for O1, detailing the many instrumental lines and combs found contaminating the data, some of which are shown in figure 3.12. Among the loudest and most pervasive combs at low frequencies was a comb described by $f_0 = 0.5$ Hz and $\Delta f_c = 1$ Hz (where these parameters are described in 2.2.3.2). Teeth of this comb were visible at every harmonic between 10.5 Hz and 133.5 Hz. The comb was present in both aLIGO sites, and was present in the GW data channel with just one week’s worth of integrated data.

This comb became a high-priority target for mitigation following O1, due to its pervasiveness, and the density of its teeth. Key stochastic GW searches veto bands of 0.25 Hz around each known noise line, which in this case, removes much of the low frequency spectrum. It also posed a problem for several high-priority known pulsar targets, and posed an issue to all-sky and unmodelled searches below 133.5 Hz.

In this section, I detail an investigation into this comb and its source. The investigation was instigated during a visit to LHO by myself and Vincent Roma, and was completed following our departure from the site by Ansel Neunzert. The investigation is summarised and published in [167].

3.3.1 Preliminary investigation of O1 data

An investigation of this comb was begun by myself and Vincent Roma. We began investigating this comb, beginning with the line list, and FFT data integrating the first week of O1. The FFT revealed that each tooth of the comb was narrow, within a single frequency bin (≤ 0.0006 Hz wide) on either side of the central frequency. With only one week's integration time, not all teeth of the comb were visible, though most between 15.5 Hz and 78.5 Hz were above the noise floor, which provided a good basis to begin the investigation. A number of teeth were obscured by more prominent features, such as the 60 Hz line. The narrow width of each tooth suggested that the coupling method was electrical, and not mechanical. The line list for O1 included several witness channels for the comb, all of which were magnetometers in various locations about the site.

3.3.2 Magnetometer coherence studies

In order to verify this coherence between the GW channel and the magnetometers at the comb frequencies, we created coherence plots between each magnetometer channel and the GW data using data taken during O1. The work was split between Vincent Roma and I, dividing up the magnetometer channels to plot.

The magnetometers within each EBAY showed a higher coherence with the GW channel than magnetometers inside the LVEA, though the coherence inside the LVEA was non-zero on a comb. Further, as each magnetometer was composed of an x , y and z component, we observed a directionality towards the source. In order to determine that the coherence observed was due to a physical source, and not an artefact of a faulty magnetometer, we rotated magnetometers in the EBAY of the corner station by an angle of 90° about their axis. Following this, their coherence with the GW channel was measured again, and it was found that the directionality of the source remained

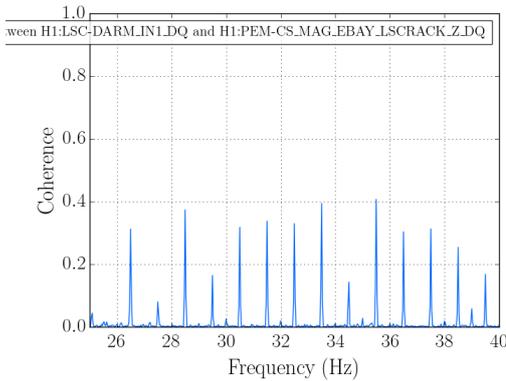


FIGURE 3.13: The coherence between the GW channel and the corner station ISC-rack PEM magnetometer.

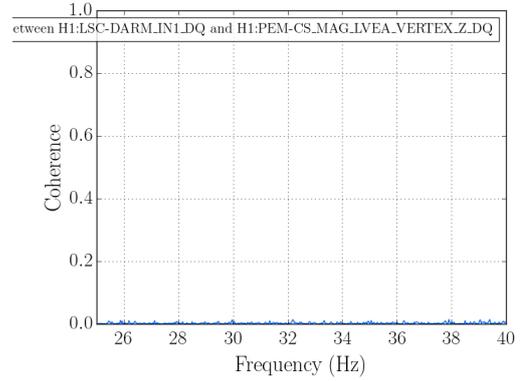


FIGURE 3.14: The coherence between the GW channel and the magnetometer in the LVEA at the corner station.

consistent between the two magnetometer frames. This pointed towards the source of the comb being physically present in the EBAY, and not an artefact of faulty equipment. Additionally, we found that this coherence was sufficiently prominent that the comb structure was clear with very little integration time.

This short integration time allowed us to take several short data sets quickly. This allowed us to take a portable magnetometer into the EBAY at the corner station, in order to measure the magnetic field in various locations. We began by positioning the portable magnetometer in various locations around the EBAY, and accumulating 15 minutes of data, and repeating this process to get a reasonable spread of locations throughout the room. The magnetometer data was recorded on an unused auxiliary channel, so that it could be easily accessed later. Before moving the temporary magnetometer from one position to another, the data were given a precursory check, by eye-balling a FFT. We repeated this process at the EBAY at the Y-arm. Figure 3.15 shows a schematic of the layout of both the corner station's and Y-end station's EBAY, including the positions of the permanent PEM magnetometers, and of the sampling locations of the temporary magnetometer.

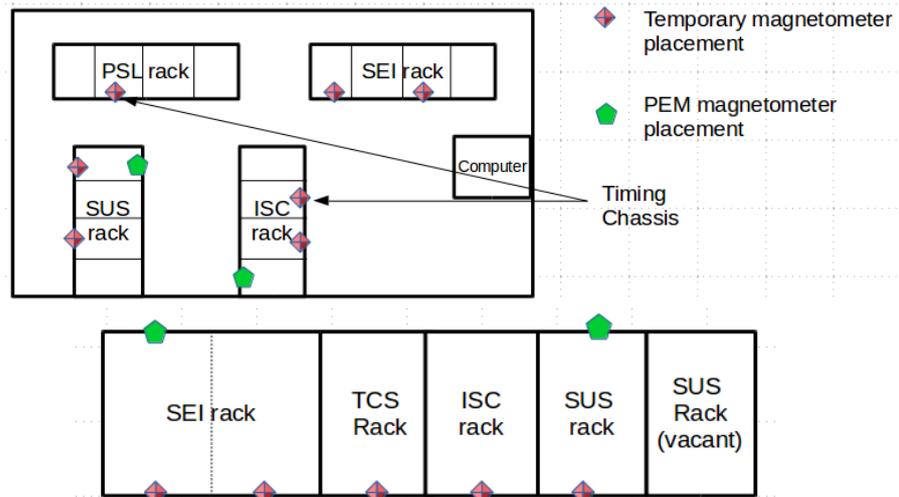


FIGURE 3.15: Above: A schematic layout of the corner station EBAY. Below: A schematic layout of the Y-arm EBAY. The green diamonds mark the magnetometers within PEM sensor suite, and the purple diamonds mark the locations of the temporary magnetometer.

At this time, upgrades to the input optics meant that there was no coincident strain data being generated by the detector. As such, we had only magnetometer data to study, which contained more features than the combs alone. A subset of the results are shown in figure 3.16.

The temporary magnetometer showed a much stronger comb when placed nearest to the timing equipment in each EBAY. Additionally, we saw a clear trend in the time domain data of the magnetometer, which showed a periodic signal with a period of 2s. In order to further examine this, Riles and Liu [168] used a “folding” technique on the magnetometer data. This involves dividing the data from a given magnetometer channel into several segments of 8s, and calculating the average 8s of data from the segments. The folding study (shown in figure 3.17) revealed a clear underlying structure, which had a 2s periodicity. One the exact boundary of a GPS second, the magnetometer would see an impulse, and a steady decay, before repeating at the next second boundary.

The location of the loudest comb being near the timing equipment in the EBAY, and the exact second-boundary impulse shown in the magnetometer data was enough

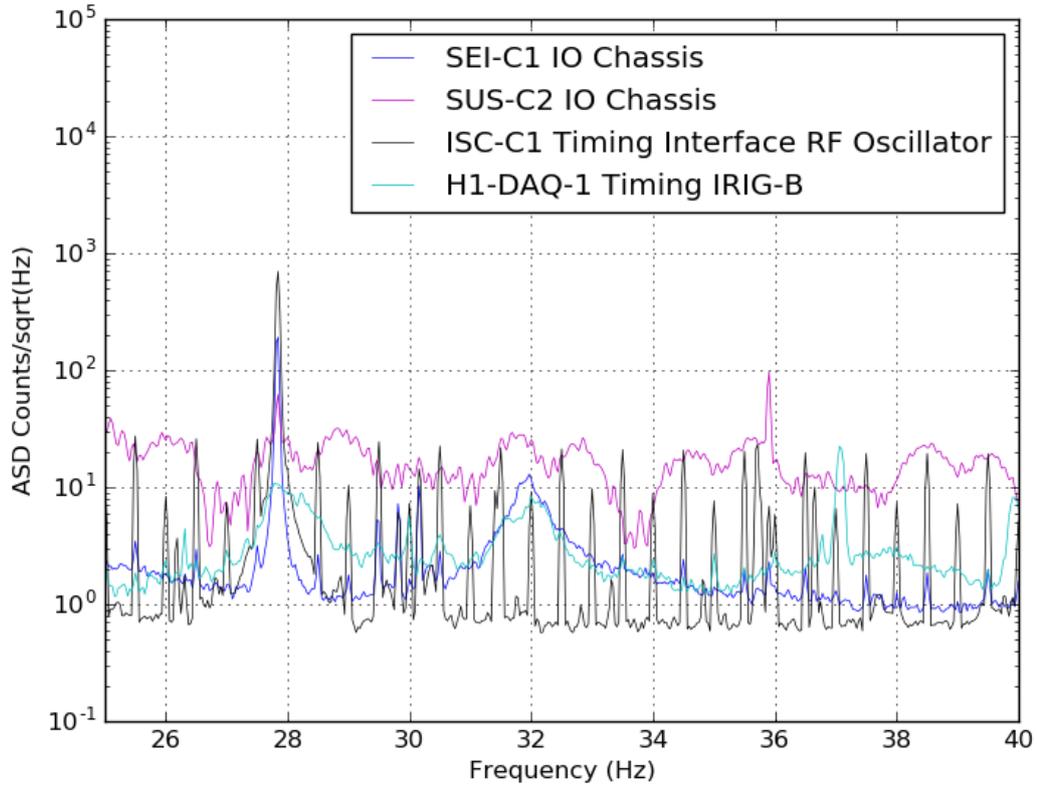


FIGURE 3.16: Each trace shows the spectrum of the temporary magnetometer data in a different location around the EBAY.

information for the timing equipment to become the focus of the investigation. The involvement of the timing system in the coupling of the comb was also hinted in the structure comb itself. As we will discuss in section 3.3.3, the timing system synchronises computer clocks across the site accurately to 10^{-6} s. This is reflected in the precision of the comb: Each tooth is centred at precisely $N + 0.5$ Hz; each tooth is very narrow, within a single bin width; and each comb is spaced 1 Hz apart. The precision in the placement of the comb's teeth is only as accurate and precise as the mechanism that couples the comb into the detector.

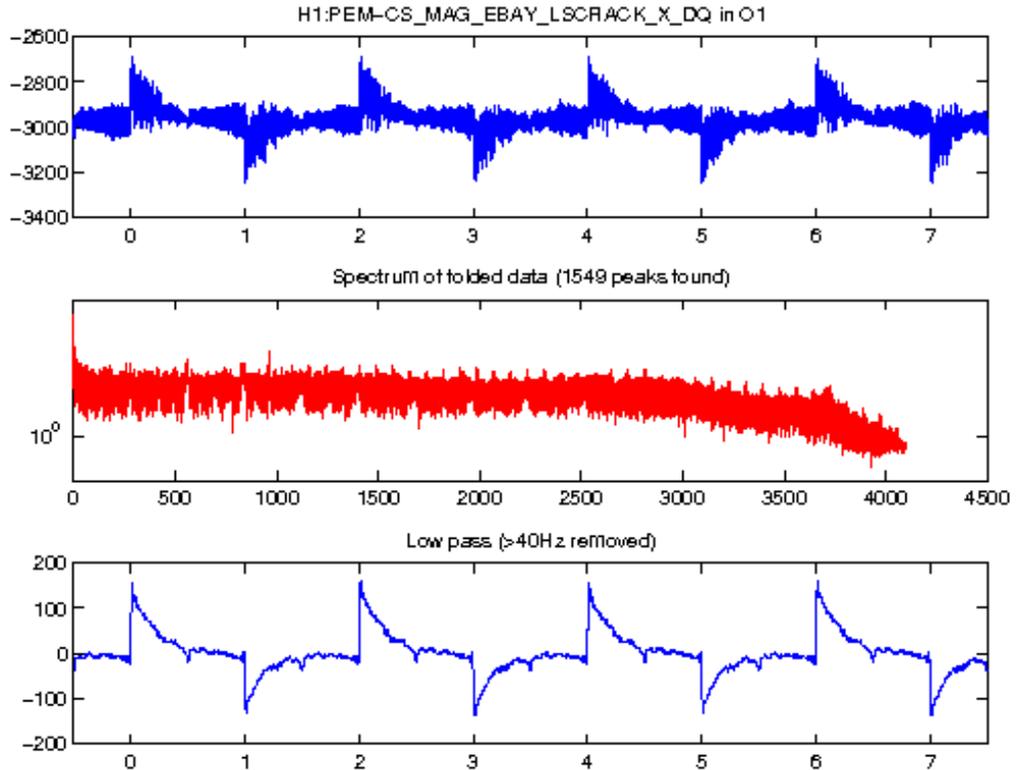


FIGURE 3.17: The three-panel folded magnetometer data generated by Riles and Liu [168] shows (top panel) an 8 s time series of folded magnetometer data, (middle panel) the spectrum of the folded panel and (lower panel) only the low frequency components of the folded magnetometer data in the time domain. In this folded data, we can clearly see a two-second periodic structure.

3.3.3 aLIGO's timing system: a candidate source

The timing system at aLIGO is a key component for taking precise data at all times. The timing system is synced to a GPS clock, maintaining accurate time sampling of the GW data, as well as the environmental data, monitoring the site. There are three GPS antennae at each site. From these antennae, the timing signal must be distributed to each electronic chassis which processes any data channel before it is recorded. This keeps each chassis synchronised to the GPS clock. The timing signal which synchronises the electronics is distributed through a system of fibre-optic cables, as pulses of light. At each EBAY, there is a timing comparator. Upon receiving a

pulse of light, the comparator compares this GPS signal to a local fixed oscillator timing device, ensuring the signal is accurate. The pulses of light are then directed to the timing signal fanout, which distributes the timing signal to the many electronic I/O chassis within each EBAY. Each chassis contains a timing card, which receives the timing signal, and synchronises its clock to it.

At every interface, where a timing signal pulse is either received or emitted, there was a timing card. The timing card showed an LED, which was used to indicate the status of the timing card. One such LED is shown circled in yellow in figure 3.18. If the LED was not illuminated, then the card was off. A continuously lit LED indicated that the card was on, but out of sync with the GPS clock. An LED blinking with two second periodicity (on for one second, off for one second) indicated that the timing card was operating well.



FIGURE 3.18: The figure shows as I/O chassis, with the timing LEDs on the left (circled in yellow), above and below the fibres carrying the timing signal. Additionally, the temporary magnetometer is seen (circled in blue) above the I/O chassis.

A square wave with periodicity of two seconds (such as the blinking of the LEDs), has the property that its FFT contains only the odd harmonics of the square wave's frequency. As a periodicity of two seconds corresponds to a frequency of 0.5 Hz, this

corresponds exactly to the comb seen in the GW data. Any one of the several timing cards within any EBAY could be the primary coupling mechanism into the detector.

In order to verify that the population of LEDs within a given area of the EBAY correlated to the strength of the comb in the EBAY, we placed the temporary magnetometer in locations which were near the timing signal fanout (which has the most timing cards of any chassis within the EBAY), near another I/O chassis, which has only one or two timing cards, and away from any electronics, where there should be no timing cards. We found that as predicted, the comb was strongest in the magnetometer when it was nearest to the most timing cards. The results of this are shown in figure 3.19, in which the dark green and purple traces, taken close to two different timing fanouts show a strong comb with 1 Hz spacing and 0.5 Hz offset, where the other sets, taken on other racks show little evidence of this comb.

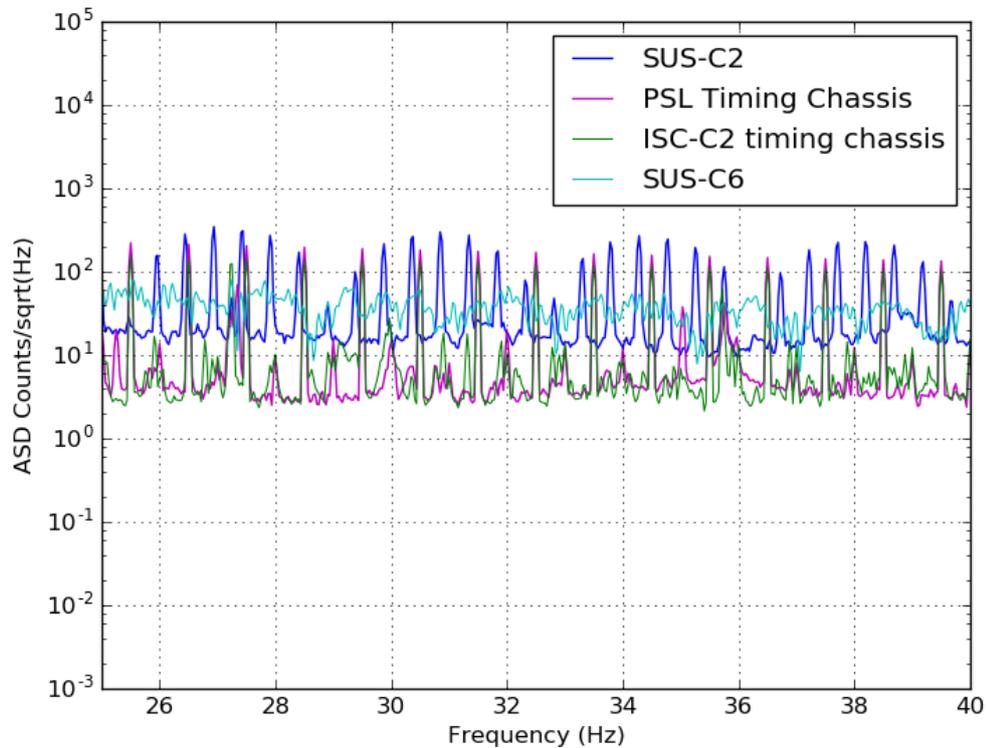


FIGURE 3.19: Four traces show the spectrum of the temporary magnetometer when placed near a timing fanout (purple and green), and far from the timing fanout, but near an I/O chassis (blue and teal).

From this observation, several candidate couplings between the comb in the magnetometer and the comb in the GW channel were suggested.

3.3.4 Testing a candidate coupling: Strain I/O chassis

Robert Schofield suggested a number of candidate couplings, the first of which was through the I/O chassis which handles that GW data. This I/O chassis timestamps the raw GW data, and applies a gain. It is termed the CS ISC I/O chassis, and as with any other I/O chassis, contains a timing card. Each timing card draws its power from the same source as the chassis which houses it. As such, the power needed to periodically illuminate the LED could be applying a load onto the gain applied to the GW signal, creating a mirrored square-wave into the gain. In order to test this first candidate, electronic and data engineers at the site shut down the CS ISC I/O chassis, disconnected the timing card's power supply from within the chassis, and connected it to an external power supply, independent to the chassis. This set up is shown in figure 3.20. They then restarted the chassis, so that we could collect some more strain data to check this coupling.



FIGURE 3.20: This photograph shows an external power supply (left) connected to the timing card within the CS ISC I/O chassis (bottom right) in order to test the first coupling hypothesis.

During the next sensitive operation of the detector, several months later, the comb was found to still be pervasive [169], and the external power supply was removed again by electronic and data engineers at the site. The I/O chassis and timing card were reset to their original configuration.

3.3.5 Eventual suppression of the comb

After Vincent Roma and I had left the site, Ansel Neunzert followed up on the investigation, and a new approach was taken. The LED's behaviour at large was to be addressed. In order to do this, the firmware controlling the LED within each of the timing cards was reprogrammed by electronic engineers at the site. This was to exchange the roles of the blinking and continuous LED behaviour. This work was carried out around the entire LHO site [170], and led by Ansel Neunzert.

The results of this alteration showed a clear reduction in the strength of this comb during the next sensitive operation of the detector [171]. The same firmware upgrade was used to also reduce an analogous comb in LLO, which was present throughout O1 in that detector [172].

Though the comb is not entirely removed from the LHO or LLO detectors, its impact is greatly reduced thanks the efforts of several months of research, by several researchers involved. The process for mitigating the 1 Hz comb with a 0.5 Hz offset involved consulting a line list (which itself is an involved process), studying several magnetometer channels, examining each EBAY with greater scrutiny, a data folding study, studying the site's timing system, implementing a trial mitigation attempt, and eventually forcing a site-wide upgrade of hundreds of timing cards.

3.4 Summary

I applied detector characterisation techniques to three different problems.

First, I examined a glitch around the 60 Hz line in S5 and S6 at LLO. I attempted to mitigate the effects using a data-based veto of glitched data, but the veto had no effect. On closer investigation of the glitch, I found that glitches in the magnetometer did not correspond to higher power in the strain channel.

Secondly, I was a member of a small group, compiling a line list for O2. Using spectra generated for the task, I examined the region 100 Hz - 2 000 Hz at LLO. The lines and combs in this section can be found in appendix B.

Thirdly, I investigated and determined the cause of a 1 Hz comb at LHO that was prevalent through O1. By looking at cross-channel coherence between the strain and magnetometer channels, and following up with a temporary magnetometer, I was able to pinpoint the timing system as the source of the comb. This led to the eventual suppression of the comb before the start of O2.

Chapter 4

Transient continuous gravitational waves – an introduction

The pursuit of GW detector characterisation is not an end unto itself, but a means by which to achieve data that is more sensitive to a broader range of GW sources. In section 1.2, we discussed a few of the most interesting sources of transient gravitational radiation, including CBC inspiral events and supernovae. In addition, and quite separately, we discussed sources of persistent GW radiation, such as the stochastic GW background, and CWs. However, signals which lie between short transient signals (lasting up to a few seconds), and those which are persistent, are not without astrophysical interest. However, they do not simply fall into either the short transient nor persistent definitions.

We term these long duration transients as tCWs. A transient continuous wave (tCW) is a GW with morphology similar to that of a CW, but with a limited emission period. This chapter explores emission mechanisms and detection methods for tCWs. Later, in chapter 5, we will build on the concepts and techniques introduced here in order to develop a new tCW analysis technique, and in chapter 6, we examine the performance of such a method.

4.1 Defining transient continuous waves

A tCW is a transient GW with morphology similar to CW morphology whilst emitting. As such, a tCW must have a start time, t_0 , and end time, t_f . We can define the duration of a tCW, τ , to be $\tau = t_f - t_0$.

As we know CW (and tCWs) are extremely weak, it will be very difficult to accurately constrain t_0 and t_f for a tCW. To demonstrate this, we can consider two similar tCWs, one with duration τ_{tCW} , and the other slightly longer, with length $\tau_{\text{tCW}} + \epsilon$. Assuming that each tCW is contained within a single observation, the SNR now scales as $\sqrt{\tau}$, the square root of the tCW duration. Taking the ratio $\sqrt{\tau_{\text{tCW}}}/\sqrt{\tau_{\text{tCW}} + \epsilon}$, so as to discriminate between the two, and expanding, we see that

$$\frac{\rho_{\text{tCW}}(\tau_{\text{tCW}} + \epsilon)}{\rho_{\text{tCW}}(\tau_{\text{tCW}})} = \frac{\sqrt{\tau_{\text{tCW}} + \epsilon}}{\sqrt{\tau_{\text{tCW}}}}, \quad (4.1)$$

$$= \frac{\sqrt{\tau_{\text{tCW}}(\sqrt{1 + \epsilon/\tau_{\text{tCW}}})}}{\sqrt{\tau_{\text{tCW}}}}, \quad (4.2)$$

$$= (1 + \epsilon/\tau_{\text{tCW}})^{0.5} \quad (4.3)$$

$$\approx 1 + \frac{\epsilon}{2\tau_{\text{tCW}}}. \quad (4.4)$$

4.2 Sources of tCWs

4.2.1 Neutron star glitches

Observations of pulsars since their discovery [173] show a highly regular behaviour [174]. Simply put, they spin, and they slow down. Observationally, a pulsar's signal comprises short pulses of light with a period of T_{pul} , further, the period between pulses generally increases by \dot{T}_{pul} (plus higher order terms). Pulsars with very little timing noise, such as millisecond pulsars [175, 176] (so named for their short period) are used as reference clocks to attempt to measure very low frequency GWs in pulsar timing experiments

[177, 178]. Importantly, it is this highly predictable property of NS frequency evolution which allows us to generate a CW signal model from which to draw accurate signal templates, and perform CW searches. However, a subset of pulsars do not strictly behave in this way. A population of pulsars have been observed to have unpredictable deviations from a simple frequency evolution, known as “glitches” [179].

4.2.1.1 Defining a pulsar glitch

A glitch is characterised by an instantaneous increase in rotation frequency of the pulsar, with relative size $\Delta f/f \approx 10^{-11}$ to 10^{-5} [180]. Following a glitch, the pulsar has a “relaxation period”, during which its spin-down, \dot{f} , is initially increased by a relative factor of $\Delta \dot{f}/\dot{f} \approx 10^{-4}$ to 10^{-3} , decaying away exponentially. After the relaxation period, often the pulsar’s spin down rate settles near to its value prior to the glitch, with differences in the pre-glitch and post-relaxation spin down rate ($\Delta \dot{f}$) ranging from $\approx 10^{-19}$ to 10^{-11} [181].

The mechanisms responsible for NS glitches are not well constrained, though there are two candidate models in tension [182], crust-quake driven glitches [183] and vortex pinning [184]. Crust quakes are sudden changes in the NS crust, causing the mass distribution to change, which results in a sudden change in the angular velocity of the star [183]. The vortex pinning model supposes that quantised cells of angular velocity in the NS superfluid core drift towards the crust, and exchange their angular momentum with the NS.]

Additionally, pulsars with a higher spin-down rate are observed to have more frequent glitches than those with lower spin-down values [180, 181], and the relationship between spin down rate and glitch activity has been found to be linear. As a pulsar’s spin down rate is used as an indicator of its age, this leads us to believe that younger pulsars are more likely to experience glitches than older pulsars [185]

4.2.1.2 Glitches and continuous wave searches

Energetic arguments relating a pulsar’s spin down rate to its radiated GW energy [186] suggest that during a pulsar’s relaxation period, a portion of the pulsar’s excess rotational energy is radiated away by GWs, namely these GWs are the tCWs generated by a pulsar glitch. This requires a non-zero second derivative of the mass quadrupole (see section 1.2), which can be caused by a wide range of different phenomena [187]. As the NS relaxes, the asymmetries driving the GWs also relax, and the NS settles back into a stable state, so that the tCW radiation ceases. These tCWs induced by pulsar glitches are expected to last between seconds and weeks, depending on the glitch and relaxation model [186–188].

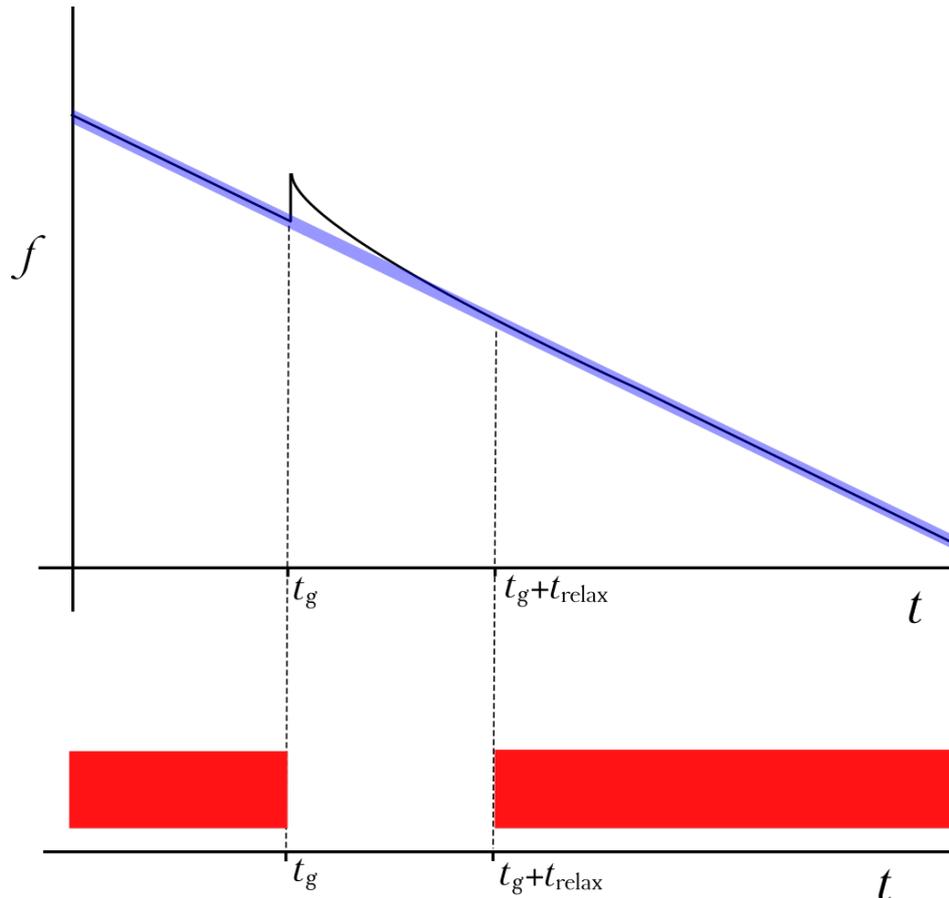


FIGURE 4.1: Here, in the upper panel, we show a schematic of the source frequency evolution (solid line) and that of the search template (blue highlighted area). Below, we depict the periods of the observation in which the source is visible to the search as red rectangles.

Glitches observed in known pulsars are often well studied, as they present a window into the equation of state of its host [189, 190]. As such, glitches in high priority known pulsars (such as the Crab pulsar [191]) often have accurate timing throughout the glitch and relaxation period. This allows for accurate frequency tracking of glitching known pulsars for use in a targeted search [181]. Conversely, for directed searches, or follow-up studies on all-sky candidates, very often there are no electromagnetic counterparts to track glitching behaviour. In general, we assume that there are no glitches in these sources during an observation. This presents a problem. If a detected source glitches during an observation, the frequency evolution of the candidate will deviate from a smoothly evolving template. We show in figure 4.1 how the difference in templates can effectively reduce the integration time of the signal, resulting in a drop-out of signal during a glitch.

Additionally, we can consider a longer type of pseudo-transient CW signal caused by a glitch. It is likely that following a glitch, the values of many of the key CW signal template parameters (though in particular f , \dot{f} , ϕ_0 and h_0) are discontinuous from their values prior to the glitch [181]. This would create a mismatch between the template and true phase evolution of the source. Semi-coherent search methods (which allow for the values of ϕ_0 and h_0 to be discontinuous between segments of data) are less sensitive to this effect than fully coherent methods (which maintain consistent values for its search parameters over an observation) [181]. Though these are not physically transient GWs, their dropping out of a search leads them to resemble a transient source. The observed result is that the searches have a lower total integration time with the signal, resulting in a less sensitive search.

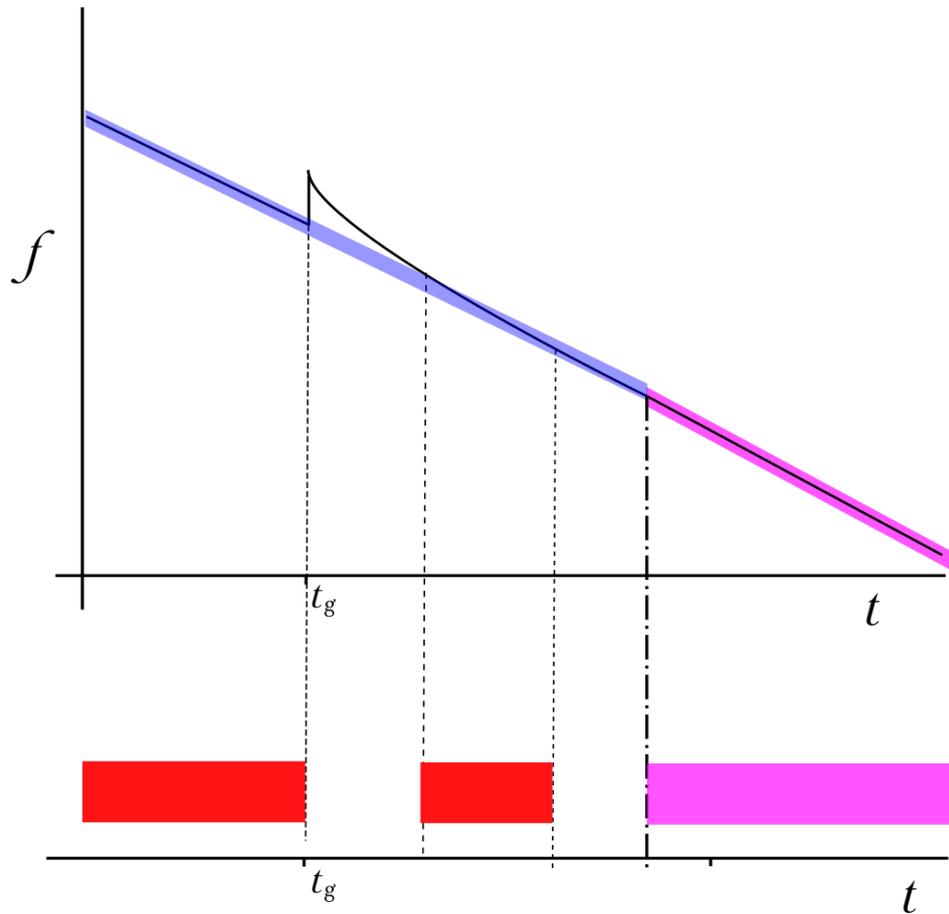


FIGURE 4.2: We show here that a pulsar whose post-glitch frequency evolution can further complicate its apparent transient behaviour. The top plot shows a schematic source and template frequency evolution. We picture here a semi-coherent search, in which a new segment begins at the dot-dashed line. Below, we show a time-line of the search. Periods during which the source is detectable are shown by the red and magenta rectangles. A fully coherent search would only see the red rectangles, however a semi-coherent search can pick the signal back up in a new segment, and detect both the red and magenta rectangles.

We show this effect in figure 4.2, which shows a similar set up to that seen in figure 4.1. Here, we see that a coherent search template and the source’s evolution differ following the glitch. It also shows how a semi-coherent search is inherently able to recover the signal, as each segment contains an independent template. A fully coherent search may never recover the signal following a glitch, however a semi-coherent search may recover the signal in the next segment (vertical dot-dashed line).

4.2.2 Transiently accreting pulsars

There is also a small sub-population of NSs in binary systems which have been observed to accrete matter transiently [192]. Interesting among them is PSR J1023+0038 [193], which has been observed as both an X-ray emitter and a radio emitter, though not both at the same time. Whilst the pulsar is in an X-ray emitter phase, it is well modelled by an accreting NS, and can be considered a low-mass X-ray binary (LMXB) [194]. When it is not accreting (it is in its quiescence), the low density of accreted matter around the NS allows its magnetosphere to expand, and for the magnetic field to resume accelerating particles [195]. This causes the radio emission to resume, and the quiescent NS is best described as any other NS in a binary system [196].

These “multiphase pulsars” change between these two emission phases on time scales of years [196, 197], and have not been shown to be predictable. Typically, a pulsar’s rotation frequency is increased by accretion [198], however in observations of PSR J1023+0038, it is seen to be spinning down more quickly during its accretion powered X-ray phase, with the difference between the two spin down rates found to be $\dot{f}_{\text{diff}} = -6.428 \times 10^{-16}$ Hz/s.[193].

The higher spin down rate of PSR J1023+0038 in its X-ray phase is unintuitive, and hints at an unknown energy dissipation method. One particularly interesting theory is that infalling matter is concentrated into a small patch of the NS’s surface, creating a mountain, and a mass asymmetry in the NS. We can estimate a conservative upper limit of the strain of associated GWs from PSR J1023+0038 by considering the gravitational wave strain if all of its energy budget were radiated as GWs. This is given as [199]

$$h_{\text{sd}} = \left(\frac{5}{2} \frac{GI_{zz}}{c^3 d^2} \frac{|\dot{f}|}{f} \right)^{1/2}, \quad (4.5)$$

and is known as the “spin-down limit”. By this argument, as the spin-down rate of PSR J1023+0038 has been observed to be higher in the X-ray phase, then we can expect

the spin-down limit in that phase to also be higher. However, we must be careful here. The spin-down limit for a NS is valid for a NS in an energy conservation framework. In an accreting NS, the accretion inserts energy into the NS's system. As such, we can argue that the energy budget of PSR J1023+0038 is greater than the spin down energy. Without knowing the details of the energetics of the system, we are left without a better estimate for the energy radiated away from PSR J1023+0038.. It is sensible then to continuous to use the spin-down limit as given in equation (4.5) as an estimate for the upper-limit of the strain from PSR J1023+0038.

From the case of PSR J1023+0038, we can expect there to be other such transiently accreting pulsars, whose energetics are likely to change between accretion and quiescent phases. It is this change of signal model between the two NS phases which define the tCW behaviour for this type of source. Though not all transiently emitting NSs spin down faster during accretion (as PSR J1023+0038 does), the arguments about the energetics of the system hold, and we may expect different values of h_0 in each of the quiescence and accretion phases.

4.2.2.1 Transiently accreting NSs and tCW detection

GWs from a transiently accreting pulsar can be expected from both phases, though as argued, we can expect GW's from the accretion phase to have a higher value of h_0 than during the quiescence phase, as the energetics associated to the spin down are no longer zero-summed. During the transition between accretion and quiescence, we might expect only the source's spin down and strain to change. Nevertheless, a CW search template which does not account for this change will become mismatched with the source's evolution.

In order to search for GWs from both phases of such a NS, we can consider a search which is allowed to switch between two signal models. We can consider the two signal models as corresponding to the quiescent and accretion phases of the source. In each

model, we would allow for values of \dot{f} and h_0 to have values independent from their values in the other models, in other phase. We could impose that $\phi_0, f, \psi, \cos \iota$ and the sky position of the source remain consistent between models, as we do not necessarily expect them to change between the source phases.

In targeted searches, where a transition between two emission phases is observed, the phase change can be tracked relatively straightforwardly. We can impose a known time at which the system transitions between the emission phases, and select a model corresponding to both before and after the transition.

This becomes more complicated when following up a candidate from directed or all-sky searches, for which there is no electromagnetic counterpart. We can consider a case in which the source is a transiently accreting NS, but we do not have information time of the transition between the phases. In such a case, we can consider adding a variable transition time, t_{tx} , and we allow the values of \dot{f} and h_0 to be independent on either side of t_{tx} .

4.2.3 Magnetar burst quasi-periodic oscillations

Magnetars are a class of NSs which have magnetic fields 10 to 1000 times stronger than those of other NSs [200]. Their strong magnetic fields are thought to power energetic bursts of electromagnetic radiation [201, 202], in the form of X-ray and soft gamma-ray bursts.

Studies of these bursts have revealed that underlying the burst is a structure of quasi-periodic oscillations (QPOs) [203]. It is thought that these QPOs are symptomatic of vibrational and spherical modes of the NS, excited by the burst [204]. In particular, the (fundamental) f-modes [205], Alfvén modes [206] and torsional modes are capable of producing GWs if sufficiently excited [207–209]. F-modes and torsional modes are oscillations of the crust of an NS, with torsional modes in particular being caused by

shear forces on the star’s surface, and driven by elastic forces in the crust [205]. Alfvén modes are resonances of the star corresponding to waves whose restoring force is the NS’s magnetic field [206]. Torsional and Alfvén modes, if responsible for the QPOs, may last longer than observable QPOs [208, 209], which themselves have been observed to last up to a few hundred seconds.

These tCWs are expected to be shorter in duration than those produced by glitches or transiently accreting NSs, and their GW frequency is expected to be at the frequency of their f-modes, Alfvén modes and torsional modes, rather than at $2f_{\text{rot}}$, as with other tCW sources discussed.

4.2.4 Pseudo tCWs by spin wandering

In section 4.2.2.1, we touched on NSs in a binary system, which could be a tCW source due to changing strain amplitudes between their quiescence and their accretion. However, there is another mechanism which may lead to an apparent tCW in binary sources. NSs in a binary system have been observed with frequencies which evolve in a way that can not be predicted. This behaviour is known as spin wandering [210]. NSs which exhibit spin wandering (wandering sources) can not be matched to a signal template without consistent observations, tracking their frequency evolution.

However, it is possible that a wandering source will at some point, for some short time period, evolve similarly to a given template, wandering in and out of the sensitive frequency band. This will appear to the search that a transient source has begun, however, in truth a continuous source has entered its sensitive band, as we show in figure 4.3.

As spin wandering is an inherently unpredictable process, it is not possible to constrain the characteristic time of these pseudo-tCWs. A template-driven tCW search is likely to be able to constrain the times during which a modelled search is sensitive to such a

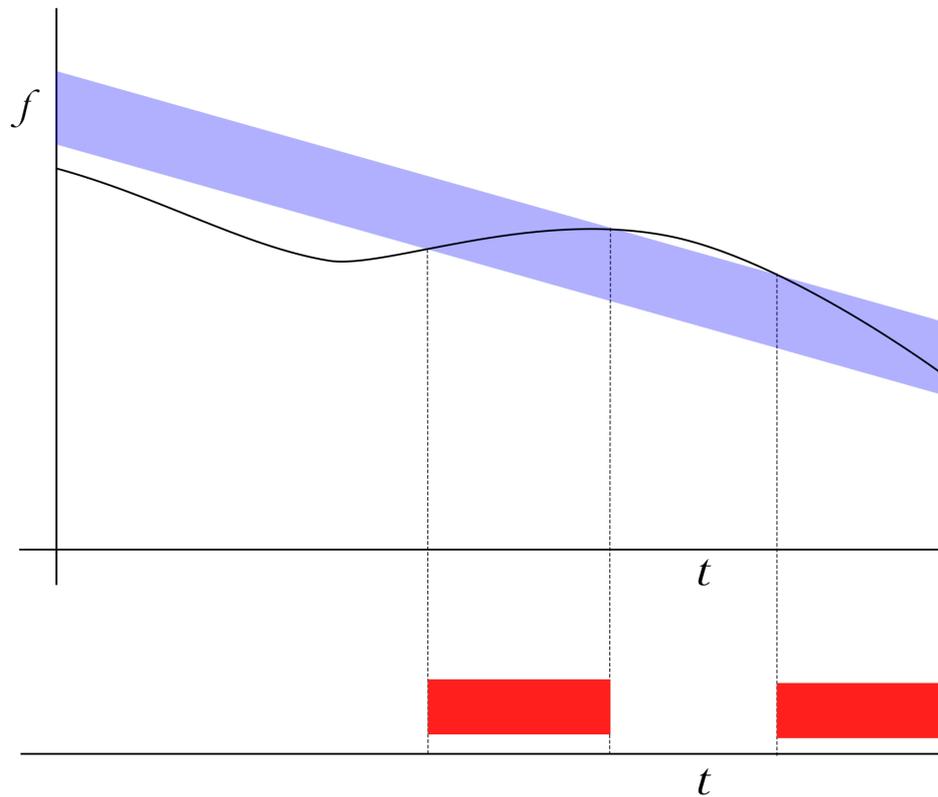


FIGURE 4.3: In this schematic, we show how a spin-wandering neutron star (solid black line) can move in and out of a CW search band (blue highlighted area), resulting in apparent transient signals (red boxes) on the time-line below.

wandering source, though it is not likely to be able to identify the signal as a wandering source. Conversely, unmodelled search methods, such as the Viterbi algorithm [156] (an efficient track-finding algorithm, which searches for an optimal path through data in the time-frequency domain) are able to track the frequency evolution of these sources.

4.2.5 Pseudo tCWs by template mismatch

Additionally, we can identify a set of CWs which can appear as tCWs not due to any transient behaviour of their source, but as a result of a mismatch between the search templates and the source's frequency evolution. analysis.

Most CW searches (besides unmodelled searches) make use of a template, or set of templates, that track the expected GW signal from a source [18, 47]. In directed and all-sky searches, there are often no observations from which to draw values of the source’s emission parameters. We must search over a region in parameter space using several signal templates.

In all-sky searches, the sky position parameters cover the entire celestial sphere, by decomposing it into several patches. The size of these patches necessarily varies with frequency and spin-down. Similarly, we search the entire range of sensitive frequencies of the GW detectors. Often, we break this into distinct frequency regions (such as the low frequency all-sky search [211]) in order to tune the analyses to best detect sources in these frequency ranges.

The spin-down range of any search is the only parameter that is truly chosen without it being otherwise constrained. As such, the range of spin-downs searched should be physically motivated. We see in electromagnetic observations of pulsars that observed pulsars tend to have spin-down rates clustered between about $10^{-15} \text{ Hz s}^{-1}$ up to about $10^{-10} \text{ Hz s}^{-1}$. We can see this in figure (2) of [212] (bearing in mind that the quoted frequencies are rotational frequencies, and GW emission is expected at $f_{\text{GW}} = 2f_{\text{rot}}$). Further, we can make arguments regarding the energy budget of a pulsar (as we did in section 4.2.2) that those sources with the highest spin-downs are expected to produce the most GW strain, and thus are more likely to result in a detection of CWs.

Due to this effect, all-sky searches tend to search for signals with higher spin-downs. While the most stringent searches do not impose a lower bound on the range of allowed spin-downs, others (that are limited by computing time or resources) must impose both upper and lower limits on the value of spin-down templates searched [211, 213].

It is possible therefore, that the frequency evolution of a NS with spin-down outside of the ranges searched will at some time evolve into and out of a given template. This is described pictorially in figure 4.4.

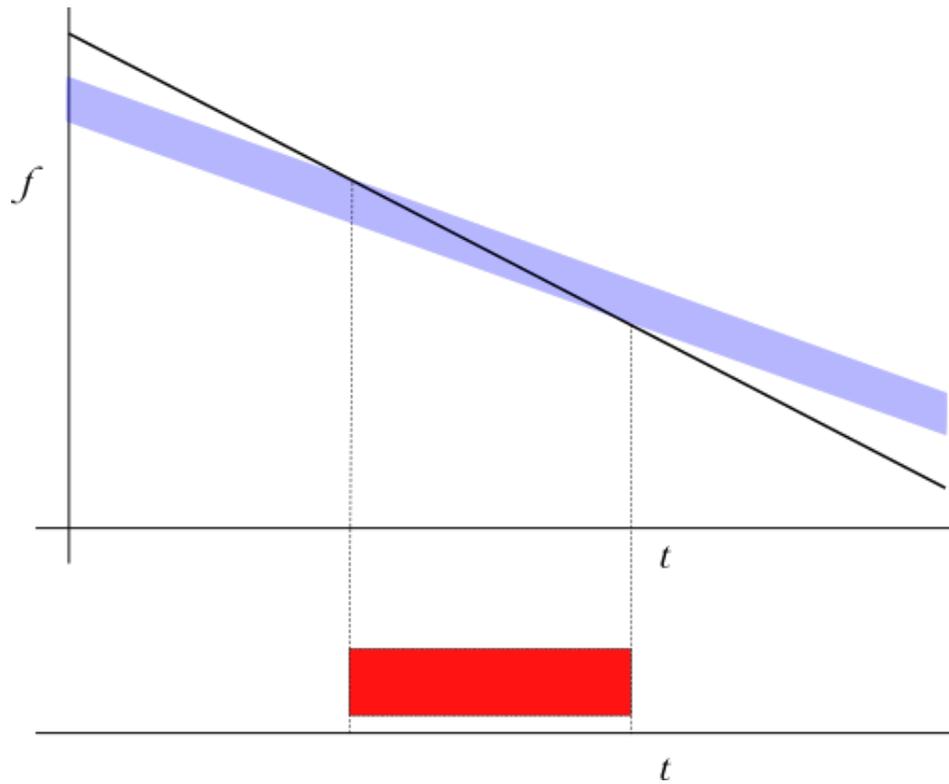


FIGURE 4.4: This simplified drawing shows how a template mismatch may lead to a pseudo-transient continuous wave. The frequency evolution of a search window (shaded blue area) evolves differently than a source, leading the search window and source frequency to overlap. The source is as a dotted red line when the search is not sensitive to the source, and as a solid black line when the search is sensitive to the source.

This becomes of particular interest when we consider a sufficiently strong GW that falls inside the sensitive region of a search for a limited period. Such a source may be found as a significant signal candidate whose associated templates do not match the true evolution of the source. Relative to these templates, the signal appears transient as it passes through the sensitive band. However, if a template better matching the source's evolution is found, its true persistent nature can be revealed.

4.3 Search methods for tCWs

When considering search methods for tCWs, we can consider two approaches. We can either extend short-duration searches, such as unmodelled burst searches, and extend

them ever longer and longer, until they become sufficiently sensitive to tCW signals. Alternatively, we can take CW searches, and impose upon them start and ending points, making them into a transient-continuous version of the same search.

4.3.1 Extensions of short-duration searches

STAMP searches compute the power within pixels of a spectrogram in order to produce a detection figure of merit. Secondly, we will consider a longer duration radiometer search, which is sensitive to signals of length from hours to weeks.

4.3.1.1 STAMP search for magnetar bursts

The first long duration STAMP search was conducted on iLIGO data for a specific source of long duration bursts: magnetar bursts.

The STAMP search method utilises a radiometry technique, which has been well explored in searches for both CWs [214] and the stochastic GW background [215, 216]. Radiometry employs a method of cross-correlating strain data between two detectors, such that the cross correlated data $p(t, f)$ is calculated by [217]

$$p(t, f) = \kappa \frac{s_1^*(t, f) s_2(t, f)}{\sqrt{P_1(t, f) P_2(t, f)}}, \quad (4.6)$$

where in equation (4.6), $s_D(t, f)$ is the discrete Fourier transform of the strain from detector D, and $P_D(t, f)$ is the power spectrum of the detector D. Note that the product of $s_1(t, f)^*$ and $s_2(t, f)$ is equivalent to the cross-power spectral density (CPSD) of detectors 1 and 2, as defined in equation (3.8). Further this search method is directionally sensitive. The sky position of the source, α, δ , can be encoded within the value of $p(t, f)$ as a complex phase Ψ_Ω . In this phase, the Ω term is a complicated function of both α and δ . [217]

$$\Psi_\Omega = -2\pi f_{\text{GW}} \Omega \frac{\Delta_D}{c}. \quad (4.7)$$

This complex phase also depends upon the source’s GW frequency f_{GW} and the the straight line distance between the two detectors used for the analysis, Δ_D .

As indicated by equation (4.6), this method is conducted in the time-frequency domain. As such, the pixel size and spectrogram duration become a tunable factor which affects the sensitivity of the search, and must be carefully chosen to suit both the target source and the detectors which are to be correlated. A typical search may use pixels which span 100 s, and which are 1 Hz wide in frequency.

In the STAMP search for magnetar bursts, template models are relegated, and instead pattern recognition algorithms are used in order to detect signals which resembled magnetar bursts. The advantage of this is that algorithms which are more efficient for different signal models can trivially be substituted in. For example, where STOCHTRACK integrates along Bezier curves, an alternative, such as the Radon algorithm [217] integrates only along straight lines within $p(t, f)$.

4.3.1.2 Radiometer searches for very long-lived GW transients

Search methods for extending the window for transient GW astrophysics ever wider must eventually borrow techniques used principally in searches for persistent GWs. Here we outline a radiometer search method for long lived transients presented in [218].

The STAMP method must inherently include a large amount of spurious data within each spectrogram. Though this gives the search a high degree of flexibility, it also presents the search with a problem: this method, though sensitive, is computationally expensive to perform in full. In order to increase the time which can be analysed in an analysis beyond $\mathcal{O}(100\text{ s})$, the data to be analysed must be compressed.

To do this, the search employs a technique known as coarse-graining, which was first described in [219]. The coarse-graining procedure in essence averages together several

frequency bins of the spectrogram to create one wider bin. In order to reduce spectral leakage due to possible edge effects of the coarse-grained frequency bin, a weighting factor $w(f)$ is included for each bin. Frequency bins of width δf which fall entirely within the wider coarse-grain bin of width Δf are summed with a weighting factor $w(f)$ of exactly 1. Frequency bins which lie on the edge of a coarse-grain bin are weighted by a factor equal to the proportion of the smaller bin that lies within the coarse-grain bin. Formally, the coarse-grained CPSD $C_{i,j}^{\text{CG}}$ can be expressed as

$$C_{i,j}^{\text{CG}} = \frac{\delta f}{\Delta f} \sum_{f=f_{\text{CG}}-\Delta f/2}^{f=f_{\text{CG}}+\Delta f/2} w(f) \tilde{s}_1^*(f, t) \tilde{s}_2(f, t), \quad (4.8)$$

where the centre of a coarse-grained bin is at f_{CG} .

Once this pre-processing of data is complete, the search employs an existing pattern recognition algorithm, based around seedless clustering techniques [217], in order to pick out tracks within the spectrograms as candidate tCW signals. A proposed seedless clustering algorithm, STOCHTRACK [220] integrates over a number of Bezier curves within $p(t, f)$, which are generated within the constraints of the search, namely a minimum integration time, t_{min} , and a search frequency range.

None of the formalisms within this search are new, but by combining the radiometry and clustering techniques from the STAMP method with compression techniques from persistent GW search methods, we are able to better search for tCW signals.

4.3.2 Time-limited continuous GW searches

The other approach towards the detection of tCWs involves in limiting the “persistent” criteria for CW searches. If we expect tCWs to have a morphology similar to that of a CW, then adapting searches which are already optimised for CW signals is likely to yield high sensitivity to tCWs also. In targeted searches for GWs from known pulsars,

we can expect that the pathways by which we expect tCW emission (NS glitches, multiphase pulsar emission) would be detected by electromagnetic observatories, allowing for a triggered tCW search, where we have prior information about tCW emission start and end times. However, in all-sky searches, for which there can be no counterpart observations to trigger a tCW method, tCW searches become much larger problems.

All-sky searches are already extremely large searches, as discussed in section 1.3.2.3. Adding in extra parameters to that search (such as the tCW emission start and end times) only serve to make such a search larger. It is important, therefore, to consider how a search for tCWs can be simplified, and their parameter spaces reduced. In doing so, we must justify any assumptions we make, and consider how they may help or harm a search strategy.

4.3.2.1 Transient \mathcal{F} -statistic

The transient \mathcal{F} -statistic search is an adaptation of the \mathcal{F} -statistic search [47]. The \mathcal{F} -statistic has long been a staple detection method in the field, and we do not re-derive it here. A full derivation of the \mathcal{F} -statistic can be readily found in [47]. The \mathcal{F} -statistic is given as a principle figure of merit in similar search methods, and estimates the value of the likelihood of the data for a given signal model, maximised over that model's parameters. Here, the authors of [221] define a new figure of merit, the Bayes' factor B_{GS} , which is closely related to the \mathcal{F} -statistic, and whose calculation follows a similar procedure.

Implicit to the calculation of B_{GS} is the signal model of the source, and the signal templates $(\Psi(h_0, \psi, \cos \iota, \phi_0)h(t, \lambda))$ which can be drawn from the model. For readability, we will drop the arguments of Ψ for the remainder of this section. The signal templates are dependant upon the amplitude parameters encoded into Ψ , as well as the Doppler parameters, λ , which are resultant of the detector's motion with the Earth,

and the source's sky position. The principle alteration made to the continuous \mathcal{F} -statistic method in order to arrive at the transient \mathcal{F} -statistic method is in the signal model provided. The transient \mathcal{F} -statistic method introduces a windowing function, $g_w(t, t_0, \tau)$, which is multiplied into a CW template $\Psi h(t, \lambda)$ in order to describe the start time t_0 and characteristic duration τ of the tCW signal model. In [221], the authors define two window functions, the first being a rectangular window, g_r , defined as

$$g_r(t, t_0, \tau) = \begin{cases} 1 & \text{if } t_0 \leq t \leq t_0 + \tau \\ 0 & \text{otherwise.} \end{cases} \quad (4.9)$$

With this windowing function applied, the signal template amplitude vanishes, until $t = t_0$, at which point the tCW begins. Between $t = t_0$ and $t = t_0 + \tau$, the signal model is described by a canonical CW signal ($\Psi h(t, \lambda)$), until $t > t_0 + \tau$, whereupon the template amplitude vanishes once more.

The second windowing function is described as an exponentially decaying window, g_e , which is similarly defined as

$$g_e(t, t_0, \tau) = \begin{cases} \exp -(t - t_0)/\tau & \text{if } t_0 \leq t \leq t_0 + 3\tau \\ 0 & \text{otherwise.} \end{cases} \quad (4.10)$$

With this exponentially decaying function applied, prior to $t = t_0$, the template amplitude vanishes as with the rectangular window. During $t_0 \leq t \leq t_0 + 3\tau$, the template resembles a canonical CW signal modulated by an exponential envelope. At time $t = t_0 + 3\tau$, the amplitude of the exponential envelope has dropped to below 5% of its initial value, and the CW is truncated. After this time, the template amplitude again returns to zero.

The transient \mathcal{F} -statistic is presented as applicable in both the fully coherent and semi-coherent regimes [221]. However, as the inclusion of two extra timing parameters, t_0 and τ , as well as the selection of window functions g_w create a larger parameter

space than is expected for the CW variant of the \mathcal{F} -statistic method, the authors note that extending the fully coherent analysis to an all-sky search is computationally impractical. However, a fully coherent analysis of known sources is possible with a reasonable computational cost, as the Doppler parameters λ are known, and need not be maximised. As such, the fully coherent transient \mathcal{F} -statistic search is best suited for targeted searches for known pulsars, and late-stage follow-ups for all-sky candidates. A semi-coherent version of this method has also been developed [222], whose figure of merit, $B_{StS/GLtL}$ is sensitive to both CW and tCW signals. Such a semi-coherent search is far less computationally expensive than a fully coherent search, and so can be feasibly extended to larger search ranges.

In tests performed on tCW signals injected over Gaussian noise, the transient B_{GS} search outperformed the continuous \mathcal{F} -statistic search as expected. The authors of [221] argue that the characteristic time scale for transient signals for this search is of order several hours to a few weeks. This statement may be motivated by astrophysical arguments, imposed by limitations on the computing power required to extend the range of the parameter τ , or on the diminishing returns on the improved sensitivity compared to a persistent CW search as the characteristic time scales approach the length of an observation.

Recent work on a graphical processing unit (GPU) implementation of this search [223] has shown parallelisation of this method can lead to greatly reduced calculation times, allowing for searches for tCWs over broader ranges of τ .

4.4 A tCW search set up

As we have seen in section 4.2, it is thought that that many of the interesting possible sources of tCWs have an expected morphology similar to that of CWs. Existing search methods, as discussed in section 4.3, are able to search both for modelled signals

(assuming a CW-like signal model) as well as unmodelled signals, and able to probe tCWs with lengths from seconds to weeks. In this section, we attempt to distil elements from existing CW, tCW and transient searches in order to describe the framework for search for tCWs with CW-like morphology, regardless of search method.

4.4.1 Prospects for tCW discovery

Irrespective of search method, we must consider the search conditions under which we expect to find tCWs. Known pulsars present reasonable candidates for tCW sources, as young known pulsars, which are high priority CW targets, are more likely to exhibit glitches [180, 181]. Several glitches have been observed in both the Crab and Vela pulsars [180, 191].

However, the population of known galactic pulsars is only one in $\mathcal{O}(10^6)$ of the total expected population of galactic pulsars [224]. As such, we expect unobserved young, glitch prone pulsars, a key target of GW searches, and a key source of tCWs, to be numerous throughout the galaxy. The search profiles of all-sky searches are already biased towards such sources, as they favour higher spin down rates. .

4.4.2 Best prospects for a search set up

An idealised tCW search should be able to explore candidates from both young targeted pulsars, and from all-sky searches. However, methods which are effective at deeply exploring a narrow parameter space (targeted searches) are inherently expensive to run on a large parameter space, such as an all-sky search. Similarly, methods which are able to efficiently explore a large parameter space (all-sky methods) are unable to achieve similarly sensitive results as targeted methods.

As such, rather than considering an all-sky tCW search, we can consider a tCW sensitive method of following up candidate signals from all-sky searches. This allows for the

search to explore a parameter space much smaller than the all-sky parameter space, as additional stages of candidate follow-up serve to further constrain the candidate's source parameters. Such a narrow follow-up search may also be able to probe known targets with similar sensitivity to targeted searches.

Searches for tCWs present an additional set of parameters into a search associated with the beginning and end (or duration) of a tCW. These transient timing parameters, \mathcal{T} , are the distinguishing difference between a tCW search and a CW search.

4.4.3 Discussion of the transient timing parameters

We can express the transient timing parameters one of two ways, either $\mathcal{T} = \{t_0, t_f\}$, or $\mathcal{T} = \{t_0, \tau\}$, where $t_f = t_0 + \tau$. In this section, we will work in the $\{t_0, t_f\}$ framework.

First, we consider that a tCW can not end before it begins, and so we impose that

$$t_0 < t_f. \tag{4.11}$$

Next, we can consider that a tCW is ambivalent to the observations of GW detectors, and so we cannot expect that all tCWs will neatly begin and end within a single observation. However, a tCW that begins prior to an observation and ends following an observation is indistinguishable from a persistent CW. As such, we impose that a tCW must be described at least one of t_0 or t_f . It must either, begin, or end, or both, within an observation.

We can consider four signal models: One in which a signal begins prior to the observation and ends during an observation, one in which a signal both begins and ends within a single observation, and finally one in which a signal begins during an observation and ends following the end of the observation. The fourth signal model is one in which a signal begins before the observation and ends after an observation. This is consistent with a persistent CW, and though other CW searches may be better optimised for the

detection of such a signal model, its inclusion allows us to discriminate between a CW signal and a tCW signal.

The first signal model we consider is the one in which a tCW begins prior to the observation, and ends during the observation. This is equivalent to the tCW beginning exactly at the same time as the beginning of the observation, T_{start} . The only constraint on t_f is that it must be strictly greater than t_0 . We can consider that a tCW ending time of T_{end} (the end of the observation) is consistent with the description of a CW. As such, we also impose that $t_f < T_{\text{end}}$. Without knowledge of the starting time of the tCW, and due to the wide variability in the duration of tCWs (as discussed in 4.2, we need not restrict the range of values for t_f any further. Explicitly, in this case, the parameters of \mathcal{T} are:

$$t_0 = T_{\text{start}}, \quad t_f = [T_{\text{start} + 1}, T_{\text{end} - 1}]. \quad (4.12)$$

In the third case, in which a signal begins during an observation and ends following the end of the observation, we can make similar arguments to those made previously. Through this we arrive with the parameter space for the timing variables:

$$t_0 = [T_{\text{start}+1}, T_{\text{end}-1}], \quad t_f = T_{\text{end}}. \quad (4.13)$$

The second signal model, in which a signal begins and ends within a single observation is a little more complicated as both t_0 and t_f are variables. Due to their relationship (equation (4.11)), the allowed range of each variable is bound to rely on the value for the other.

$$t_0 = [T_{\text{start} + 1}, t_f - 1], \quad t_f = [t_0 + 1, T_{\text{end} - 1}]. \quad (4.14)$$

Importantly, we can allow that more than a single tCW is observed in a single observation. We represent the timing parameters of additional tCWs as additional parameters, \mathcal{T}_n for the n^{th} tCW in an observation for a single source. Consider the case represented in figure 4.1, in which a glitch signal is recovered following the relaxation period. This

can equivalently be thought of as two transient signals, occurring around the glitch. In such a case, we impose that the first tCW finishes before the second can begin, otherwise the two events are indistinguishable.

$$t_0^{n+1} > t_f^n. \quad (4.15)$$

In the time between t_0 and t_f , we expect a signal. As we expect signal morphology similar to a persistent CW, we can employ any search method as before, however now the signal template $\mathcal{H}(t, \mathcal{T}_n, \Psi, \lambda)$ is dependant upon the timing parameters, such that

$$\mathcal{H}(t, \mathcal{T}_n, \Psi, \lambda) = \begin{cases} \mathcal{H}(t, \Psi, \lambda) & \text{if } t_0^n < t < t_f^n, \\ 0 & \text{otherwise.} \end{cases} \quad (4.16)$$

4.4.4 A check on the persistence detection criteria

The inclusion of the fourth signal model, in which we assume a persistent CW signal, allows for us to perform a check of the persistence detection criteria on a candidate. The persistence criteria states that a CW signal's SNR should grow with $\sqrt{T_{\text{obs}}}$, however this is not the case with a tCW signal. The SNR of a tCW signal grows as

$$S_{\text{tCW}} \propto \sqrt{t} \quad \text{if } t_0 < t < t_f. \quad (4.17)$$

Using model comparison techniques discussed in section 1.2.1, we can compare a persistent CW model to all other tCW models.

The remainder of this work is centred around a proposed tCW method which is designed as a late-stage follow up for candidate signals from all-sky searches, as well as interesting candidates from within the population of known pulsars. In chapter 5, we outline the search, and justify the methods by which we reduce the computing cost of an idealised search. Then, in chapter 6, we discuss the method's performance on simulated data

containing transient signals, as well as on some candidate CW signals brought forward from O1 all-sky searches.

4.5 Summary

In this chapter, we introduced tCWs, long transient GWs with morphology similar to that of CWs. We outlined several mechanisms that may result in tCWs, including pulsar glitches, magnetar bursts and transiently accreting NSs.

Then we discussed existing methods of data analysis for tCWs, some of which were designed by elongating the emission period of transient searches, and others by curtailing existing CW searches. Finally, we distilled the requirements of a tCW search from these, in order to outline the most important aspects of such a search in order to create a new method, outlined in chapter 5.

Chapter 5

A tCW follow-up method: Reduced Bayesian blocks

Building upon the definition of tCWs, and the aspects of a tCW search discussed in chapter 4, we aim to describe a follow-up method sensitive to tCWs in this chapter. To begin, in section 5.1, we will explore the Bayesian blocks (BB) formalism, and outline why this framework for partitioning data into blocks may be useful. We touch on the benefits and drawbacks of BB as a means to ascertain the starting and ending points of tCWs. We will then build upon the BB formalism to develop a method, reduced Bayesian blocks (RBB), that we propose is more suitable for this task. We discuss the similarities between RBB and BB, and detail the compromises made in the formulation of RBB.

5.1 Bayesian blocks

Originally described by Scargle [225], BB provides a versatile method to flexibly describe a data set by a number of different models. The core concept behind BB relies on the ability to split a dataset containing L samples, at “change-points” τ_j . An example of a situation in which this becomes useful is in a dataset wherein the mean of the Gaussian background noise changes part way through. In this case, we could envisage a

change point separating the data into two blocks, each with a different value of the variance of the noise, as we show in figure 5.1. By varying the position of the change point, and allowing for different model parameters on either side, we can achieve a better fit to the data (relative to the fit with a single parameter set) with only a slightly more complicated description. This is described in equation (5.8).

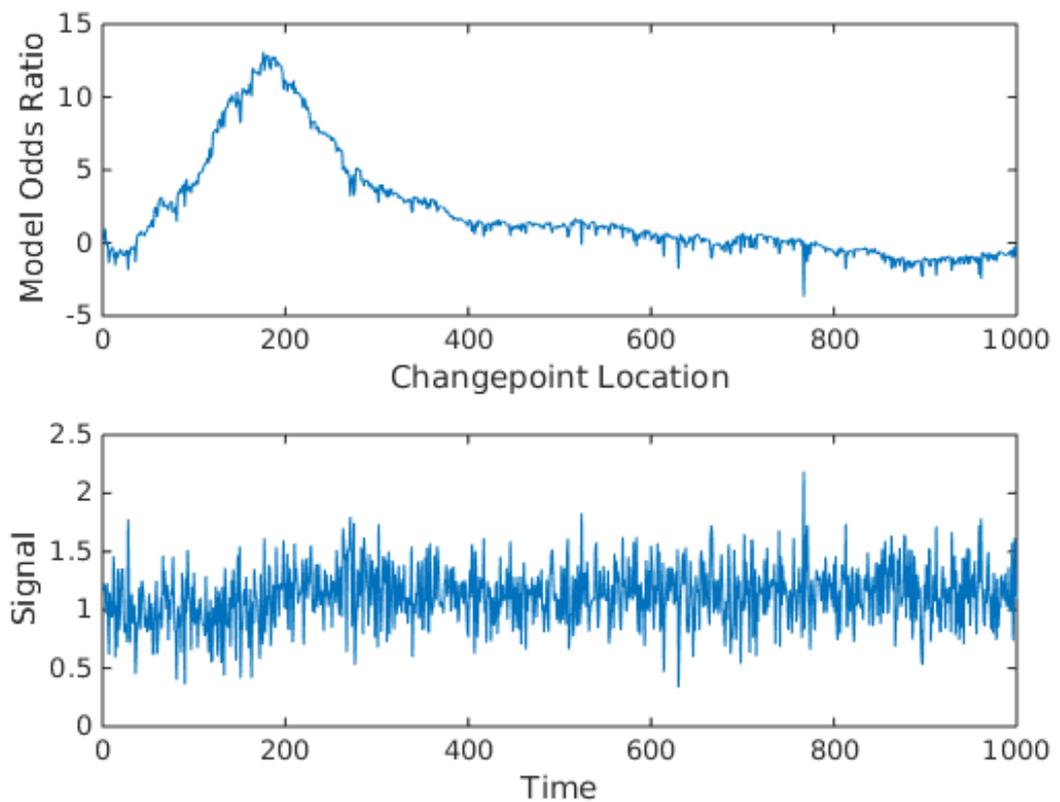


FIGURE 5.1: Here, we show an example application of BB. The dataset in the lower panel consists of Gaussian noise. At $t=200$, the mean of the noise is slightly increased. The upper panel shows the likelihood of a change points in each position in the dataset.

The data are split into different blocks B_j , each block being bound either by two change points, or the ends of the dataset. The number of change points and their positions can all be determined in a fully Bayesian framework, with the condition that change points can only be placed between two adjacent samples.

As we are working with data in the time domain, we will exclusively consider BB as applied to a time series in this work. This means that changepoint times T_τ must occur between time samples ($t_i < T_\tau < t_{i+1}$), so that there is no ambiguity into which block a given sample falls, and all samples fall into only one block.

Each block can be described by any one of a number of models of the data, M , with the added constraint that two adjacent blocks may not be described by the same model ($M_j \neq M_{j\pm 1}$). We can select the preferred model in each block by calculating the Bayes factors for each model against some reference model. This, combined with the inherent flexibility in the data contained within each block allows for a maximally matched set of models given the data.

Exactly how BB achieves this flexibility is to be explored in the following section. Once we have described the elements which define the BB method, we discuss how to consider each element in the context of a tCW method.

5.1.1 Bayesian blocks, a framework for tCWs

5.1.1.1 Defining the number of changepoints N_τ

As the number of changepoints, N_τ , is a variable like any other, we treat it as we do in any Bayesian framework. First, we can decide a prior range, prior probability distribution for N_τ , and a likelihood function for N_τ .

In discussing the range of allowed values of N_τ , we consider the two extreme cases: the minimum and maximum number of changepoints that are possible. In the minimum case, we can allow the number of changepoints in a data set, N_τ^{\min} , to be 0, so that the entire data set is a single block, and described by a single data model. In the other extreme case, a maximally split data set minimises the length of each block to a single sample, and so $N_\tau^{\max} = L - 1$. All values of N_τ are allowable between these two values.

We can assign prior probabilities $P(N_\tau)$, so long as the prior is bounded by $N_\tau^{\min} = 0$ and $N_\tau^{\max} = L - 1$. Naively, if we have no information about how many changepoints to expect, we can supply a uniform prior distribution across all values of N_τ . A uniform prior distribution P_U on a variable X between two values X_{\min} and X_{\max} is defined as

$$P_U(X) = \begin{cases} \frac{1}{X_{\max} - X_{\min}} & \text{if } X_{\min} \leq X \leq X_{\max} \\ 0 & \text{otherwise.} \end{cases} \quad (5.1)$$

5.1.1.2 Defining the time of a changepoint T_j

First of all, we define two fixed points with respect to the data, the beginning and end of the dataset. These points behave similarly to changepoints, in that they define the beginning and end of blocks, however they are not variables. We can define these as the $0^{\text{th}}(T_0)$ and $(N + 1)^{\text{th}}(T_{N+1})$ changepoints. We define that T_0 occurs before the first data sample is taken, and T_{N+1} occurs after the final data point is taken.

As with the number of changepoints, the location of each changepoint T_j is also a variable in BB. A framework with N_τ changepoints introduces N_τ new variables, each indexed as T_j , and each with its own allowed range of values and prior probability distributions. We can implement some common sense rules. Firstly, changepoints can occur only between adjacent data samples. Secondly, two changepoints can not occur in the same position. Finally, the indexing of changepoints are interchangeable. To avoid double counting configurations, we can therefore impose that $T_{j+1} > T_j$.

The range of allowable values for T_j is easy to find if we consider beginning with $N_\tau = 1$, then expand to cases where $N_j > 1$. In the first case, where $N_j = 1$, we can allow the changepoint T_1 to occur between any two adjacent samples. The earliest a changepoint can occur is between the first and second samples. This creates two blocks, B_0 and B_1 , where B_0 contains exactly one sample, and B_1 contains $L - 1$ samples. The latest that a changepoint can occur is between the $(L - 1)^{\text{th}}$ sample and the L^{th} sample. Similarly,

this also results in the data being split into two blocks, this time B_0 contains $L - 1$ samples, and B_1 consists of only the final sample. Between these two extrema, we can allow any value of T_1 . As the changepoints fall between samples, we define their position by the sample that they follow. In this convention, the allowed values of T_1 are every value within $[1, L - 1]$.

Considering a case where $N_\tau > 1$, we can consider similar arrangements. Clustering the changepoints near the start, we can place T_1 between the first and second data samples, T_2 between the second and third, and so on up to T_N lying between the N^{th} and $(N + 1)^{\text{th}}$ samples. Similarly, clustering the changepoints at the end of the dataset, we arrive at T_N between the $(L - 1)^{\text{th}}$ and L^{th} sample, and so on, until we find T_1 between the $(L - N)^{\text{th}}$ and $(L - N + 1)^{\text{th}}$ sample. By this reasoning, the allowed values for the j^{th} changepoint range between $[j, L - N_\tau + j]$.

With these constraints on the allowed values, we can define the prior probability for the locations of any changepoint. A uniform prior on T_j is the least informative prior distribution, and is the most suitable without additional information about the time of a changepoint.

5.1.1.3 Defining a block B_j

A block is defined as the data that falls between any two consecutive changepoints, such that the block B_j is bounded by the changepoints at T_j and T_{j+1} . This definition includes the two fixed points T_0 and T_{N+1} . Blocks are treated as independent datasets in BB. For example, the block bounded by the changepoints at T_0 and T_1 is the 0^{th} block, B_0 , and will start at the beginning of the full dataset. Any dataset is made up of $N_\tau + 1$ blocks. A data set with no changepoints consists of one block, the entire dataset. One changepoint splits the data into two blocks.

5.1.1.4 The BB formalism

Often the formalism is applied in order to separate the data into blocks defined by different models. This is done using Bayesian model selection techniques, such as those described in section 1.2.1, in order to generate a Bayes factor between models with two different sets of changepoints, or different numbers of blocks. To find this, we begin with the posterior probability that a block B_j is described by a model M_j , which is given as

$$P(M_j|B_j) \propto P(B_j|M_j) P(M_j). \quad (5.2)$$

When there are several models to describe the data, in order to find the best description of the data, we select the model which maximises this posterior,

$$P_{\max}(M_j|B_j) \propto \max_M [P(B_j|M_j) P(M_j)]. \quad (5.3)$$

We can apply this maximisation across all blocks in the dataset in order to assign the most likely model to each block, given the set of changepoints. If we apply this to an example with only one changepoint at a time $T_1 = t$, we see

$$P(\{M_0, M_1\}|\{B_0, B_1\}, T_1 = t) \propto P(M_0|B_0, T_1 = t) \times P(M_1|B_1, T_1 = t), \quad (5.4)$$

where

$$P(M_0|B_0, T_1 = t) \propto \max_M [P(B_0|M_0, T_1 = t) P(M_0)], \quad (5.5)$$

$$P(M_1|B_1, T_1 = t) \propto \max_M [P(B_1|M_1, T_1 = t) P(M_1)]. \quad (5.6)$$

In order to expand this into a general case, with a generalised N_τ , we will next consider a case where we are given the positions T_j of the N_τ changepoints τ_j .

$$P(\{M_j\}|\{T_j\}, \{B_j\}) \propto \prod_{j=0}^{N_\tau} \left[\max_{M_j} P(B_j|M_j, T_j, T_{j+1}) P(M_j) \right], \quad (5.7)$$

In equation (5.7), the expression within the product is very similar to the components shown in equations (5.5) and (5.6) – with the additional term for the the $(j+1)^{\text{th}}$ changepoint (which defines the end of the j^{th} block). This was not necessary in equations (5.5) and (5.6) as each block was bounded by a fixed point, the start or end of the dataset.

We can generalise this further by allowing the values of T_j to vary. We can instead treat them as parameters to be estimated, or as nuisance parameters to marginalise over, depending on the goal of the analysis. If we wish to find the best fitting positions for the changepoints, we must consider the value of equation (5.7), maximised over the position of the changepoints. As before, we are trying to find the most probable configuration of changepoints and models, and so we will also be maximising over the positions of the changepoints.

$$P(\{T_j\}, \{M_j\}|\{B_j\}) \propto \max_{T_j} \max_{T_{j+1}} \left\{ \prod_{j=0}^{N_\tau} \left[\max_{M_j} P(B_j|M_j, T_j, T_{j+1}) P(M_j) P(T_j) P(T_{j+1}) \right] \right\}. \quad (5.8)$$

However, if we are ambivalent about the positions of the changepoints and only want to know the posterior for some value of N_τ , we can marginalise over the values of T_j .

$$P(N_\tau, \{M_j\}|\{B_j\}) \propto \prod_{j=0}^{N_\tau} \left[\sum_{T_j} \sum_{T_{j+1}} \max_M (P(B_j|M_j, T_j) P(M_j) P(T_j) P(T_{j+1})) \right]. \quad (5.9)$$

In order to use this, we generally attempt to calculate the Bayes factor between two models, as often calculating the evidence is prohibitively expensive.

This formalism can be quite straight forwardly applied to a tCW search, as we will

discussion in section 5.1.2. We will apply a particular focus on some assumptions we can make which can inform the prior probabilities of the variables involved, and address some important advantages and drawbacks inherent to adapting BB for this purpose.

5.1.2 Bayesian blocks as tCW follow-up

Continuous GW candidates, for which there would be some evidence for GW emission in the data, need to be followed up with a more sensitive search than an all-sky search can provide. A follow-up which is maximally sensitive to tCWs is designed to be able to either verify that the source is truly continuous, constrain the start and stop times of tCW emission, or rule out a signal entirely. When considering beginning and ending times of emission, the ideas of changepoints and the method in BB are a useful and convenient tool.

At its core, this technique as applied to a tCW follow-up would split the data around changepoints, which we can think of as candidate times for t_0 or t_f for a tCW. The models that could be tested against the data in each block would describe Gaussian noise in the detector, or a CW-like signal.

A follow-up search using BB such as this would need to prioritise determining the number of changepoints and their locations with some accuracy. It may be less important to accurately estimate the source's GW parameters, as there are other well established follow-up methods (as discussed in 1.3.3) which efficiently manage that. In order to begin, we will first consider the data models M around which we will test in this analysis.

5.1.2.1 Choice of models

A simple search for tCWs might only use two models – a model for a signal on top of Gaussian noise, as described in section 1.3.1.2, and a model describing only Gaussian

noise. We can label these models M^S and M^N respectively.

The signal model, M^S is informed by a set of parameters, whose values have been bounded by prior follow-up stages ideally. As discussed in section 4.4, it benefits a tCW follow-up such as this to have a relatively narrow parameter space to search. As discussed in section 1.3.1.2, CW searches ask whether there is signal in the data that lies within the prior range of the source parameters. These source parameters in this case are the amplitude parameters $\Psi = \{h_0, \phi_0, \psi, \cos \iota\}$ as well as some uncertainty in the template parameters $\lambda = \{f, \dot{f}, \alpha, \delta \dots\}$ in addition to the changepoint parameters $\mathcal{C} = \{N_\tau, T_j\}$ make for a relatively large parameter space.

We can see this by adapting equation (5.8), and expanding upon the likelihood term, $P(B_j|M_j, T_j, T_{j+1})$. Doing this explicitly for the signal model, we see that

$$P(B_j|M^S, T_j, T_{j+1}) = \sum_{\Psi} \sum_{\lambda} P(B_j|M^S, T_j, T_{j+1}, \Psi, \lambda)P(\Psi)P(\lambda), \quad (5.10)$$

which requires an often expensive estimation of the marginalised likelihood of signal. We must either explore the parameter space in relatively fine detail, summing the contributions of each position in the parameter space, or have a robust way to estimate the full marginal likelihood given far fewer samples of the parameter space.

However, there already exist several CW searches that do this effectively, and which we can use to estimate the maximal likelihood. As a bonus, these searches also have methods of estimating the noise likelihood, for example in equation (1.34).

5.1.2.2 Choice of priors for a tCW BB implementation

Important to any Bayesian analysis are the prior probabilities assigned to variables of the search. This is no different in this case. If we are importing other searches to perform the likelihood estimation, we can also import their treatments of the amplitude

priors. Additionally, as a late follow-up to a candidate signal, we can use the information gained by previous rounds of follow-up to inform our priors here. The question remains though, how to treat the priors on the changepoint parameters \mathcal{C} . We discuss the aspects of the priors on both N_τ and T_j in this section.

First, we will consider the prior distribution of N_τ . This prior informs the search of how many times we might expect to detect unique tCWs from our target source. We covered the characteristic durations of several important tCW sources in section 4.2. The least informative prior for this parameter is a uniform prior distribution across all values within $N_\tau = [0, L - 1]$, however we will discuss why this is an infeasible parameter space to cover.

A BB set up with N_τ changepoints results in $N_\tau + 1$ blocks. Each block must have independent maximal likelihood estimates performed. This must be repeated for every allowed combination of values of T_j (which we will discuss below). All this makes for a very expensive search.

Additionally, as each block represents an independent set of data, each block carries with it an independent set of source parameters for that block is introduced. As the number of blocks becomes large, so too does the prior volume of the search. This increased prior volume has the effect of lowering the evidence for any one given configuration by an increased trials factor. We call this the Occam factor, as it has the effect of favouring simpler signal models (those with fewer changepoints) to complicated ones (those with many changepoints). The Occam factor suppresses the probability of high values of N_τ particularly strongly, and so we need not consider values of N_τ greater than around 10.

We can inform the upper value of this prior distribution by the length of an observation, as four changepoints in the space of a four month observation (an average spacing on one month) is optimised to far shorter signals than four changepoints in the space of a two year observation (whose average spacing is six months).

Combining these two aspects of the form of the prior distribution and its limits, we can impose that the prior distribution on N_τ is a uniform prior between 0 and $\mathcal{O}(10)$.

Next, we can consider the prior distribution for the time of a given changepoint. Naively, we may consider that a changepoint is equally likely at any time. In the case of each of the tCW sources outlined in section 4.2, the beginning of a tCW is unpredictable. As such, the least informative prior is a uniform prior on T_j between $[j, L - N_\tau + j]$.

Similarly to the way in which BB disfavours large numbers of N_τ via the Occam factor, it also disfavours values of T_j that are close to other changepoints. This tendency extends also to the pseudo-changepoints T_0 and T_{N+1} , the beginning and end of the observation. Clustering of changepoints creates shorter blocks, which results in inherently lower likelihoods. By this method, without ruling out short blocks by imposing a minimum block duration, they are disfavoured inherently.

The value of the prior for any changepoint depends on the positions of the other changepoints. For example, consider a dataset with L samples, and a BB configuration with 2 changepoints. If we consider first that the first changepoint is placed at its first possible location, $T_1 = 1$. The second changepoint can be placed in any other position, $T_2 = [2, L - 1]$, and so the height of the uniform prior according to equation (5.1) is given by $P_U(T_2) = 1/[(L - 1) - 2]$. Whereas if we consider a similar case where now T_1 is located at a position $L - 3$, as we have imposed that $T_{j+1} > T_j$, this leaves only two possible positions for T_2 : $L - 2$ and $L - 1$. In this case, the height of the prior is given by $P_U(T_2) = 1/[(L - 1) - (L - 2)] = 1/2$. In general, we can express this as

$$P_U(T_j) = \begin{cases} \frac{1}{(T_{j+1}-1)-(T_{j-1}+1)} & \text{if } T_{j-1} + 1 < T_j < T_{j+1} - 1 \\ 0 & \text{otherwise.} \end{cases} \quad (5.11)$$

As such, we can only express the prior probability for a given changepoint as a set of coupled priors which describe the positions of all changepoints in a configuration.

5.1.3 Advantages and disadvantages of BB as tCW follow-up

The BB formalism as described is very flexible in allowing for different combinations of blocks containing CW-like signals or noise. By making use of existing methods of likelihood estimation (for examples, the Bayesian heterodyne method described in section 1.3.2.1) we can achieve similar sensitivity to current CW searches. Additionally, the fine flexibility in the location of any given T_j allows for the starting and ending times of a transient to be found with relatively high precision.

We might expect the of introduction extraneous changepoints to achieve a better fit to the data. As discussed, this is suppressed by the Occam factor. The inclusion of an upper limit to N_τ serves only to cement this tendency into the analysis by prohibiting large numbers of N_τ .

Additionally, as BB can perform a complete search over the changepoint variables \mathcal{C} , we could construct the posterior probability for the location of the changepoints for a given value of N_τ , or the posterior probability of the value of N_τ . These posterior probabilities are often simpler to interpret, as they express the probability of a given BB configuration.

On the other hand, a full BB implementation tends to be very expensive. In the case of a tCW search, each time the analysis encounters a new block, it must perform either a relatively straight forward noise likelihood calculation, or a costly signal likelihood estimation. A full treatment of BB has $\mathcal{O}(L^{N_\tau})$ terms associated. In large data sets, such as the years-long observing runs of aLIGO and AdVirgo at design sensitivity, even when limiting N_τ to $\mathcal{O}(10)$, this method becomes very expensive indeed.

Further, this method, allowing a changepoint to occur between every time sample, is not optimal. As mentioned in previously in section 4.1, the SNR of a tCW signal is proportional to $\sqrt{T_{\text{overlap}}}$. The argument presented in section 4.1 about small changes in the searched time leading to even smaller increases in SNR remains true for a BB search for tCWs. As such, by considering changepoints between every sample, the location of true starting and ending times for a tCW are somewhat over-calculated for the increase in SNR that is likely achievable by the small increase.

5.1.4 Building on the BB formalism

The BB formalism gives a firm base for a tCW analysis, though in this form, its disadvantages are that it is too expensive to implement. We can build on the BB in order to address this expense by compromising on a few of the benefits of BB. In this next section, we propose a new method, called RBB, with slightly different aims than a full BB tCW analysis, and which is consistent with a BB formalism, with a few alterations to its implementation.

The aims of RBB are not to exactly constrain the starting and ending times of a tCW, as we see that this is computationally expensive with a maximally flexible search. Instead, RBB aims to discriminate between a persistent CW signal, a tCW signal and the lack of any astrophysical signal. We will see in 5.2 that we propose a method which tests for intermittent signals on a broad time scale. The intermittent nature of a candidate signal can then be fed forward into a tCW search with higher time resolution, such as a BB search, or those outlined in section 4.3. Including an RBB step prior to a full tCW search could reduce the parameter space for the timing parameters \mathcal{C} of a BB tCW search, if one is required.

5.2 Reduced Bayesian Blocks

The computing cost of a full BB treatment of a CW follow-up makes is prohibitive for a realistic application to this problem. In particular, we highlight two key disadvantages of a full BB treatment together. The high time resolution of the T_j leads to a small gain in SNR of a potential tCW signal, and the high cost of re-running several CW analyses for each new configuration of blocks. Both could be considered as a problem of oversampling.

We could reduce the cost of a BB search (or any other tCW search) by reducing the prior range of the timing parameters \mathcal{T} , by ruling out chunks of data in which we do not expect a changepoint. In this section, we describe a tCW analysis method which, we show, is consistent with an altered BB treatment despite beginning very differently. This alternative starting point allows us simply to justify the choices made in the process of developing this method.

Throughout this section, I will be introducing and using several terms specific to this analysis. For clarity and ease of reading, I have included a glossary of these terms, and Bayesian terminology in appendix A.

5.2.1 Intermittencies and data chunking

We start by defining the intermittency, Γ_p , a bitwise description of a transient intermittency hypothesis. Γ_p is a bitwise number of length N_b , and is composed of bits $\{\gamma_j\}$. In an intermittency hypothesis, the value of each bit denotes a signal model: a value of $\gamma_j = 1$ denotes the signal model, and a value of $\gamma_j = 0$ denotes the noise model. In this way, an intermittency represents a set of signal- and noise-models in a sequence.

We propose a notation of filled and unfilled boxes as a shorthand to more easily parse an intermittency. These are a filled box, representing signal ($1 = \blacksquare$), or an unfilled

box, denoting only noise, ($0 = \square$). A string of filled and unfilled boxes therefore represents the bitwise intermittency Γ_p . For example, an 8-bit intermittency which describes signal in the first four places and noise in each of the remaining places can be expressed as

$$\Gamma_{240} = \blacksquare\blacksquare\blacksquare\blacksquare\square\square\square\square. \quad (5.12)$$

The index on Γ_p refers to the binary number expressed. For any value of N_b , there are 2^{N_b} possible intermittencies, so the index under Γ_p must run between 0 and $2^{N_b} - 1$.

Similarly, a full data set D is split into N_b chunks of equal length, denoted as $\{d_j\}$, such that each data point falls into one and only chunk. For each bit in the intermittency profile γ_j , there is a corresponding chunk of data d_j , which is in the same position relative to the data set.

From an intermittency, we can define two key structural elements. Firstly, we define changepoints: a changepoint divides adjacent 2 bits whose expressed values differ. Changepoints fall between bits, and their positions are denoted by the index of the bit that they follow. Secondly, we define blocks: a block, α_i , is a series of contiguous bits whose values are the same. A block is bounded on either side by changepoints (or by the start and end of the intermittency). For example, our intermittency Γ_{240} in equation (5.12) has one changepoint (which follows the fourth bit), and contains two blocks:

$$\alpha_1 = \{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}, \quad (5.13)$$

$$\alpha_2 = \{\gamma_5, \gamma_6, \gamma_7, \gamma_8\}, \quad (5.14)$$

Similarly, these chunks of data $\{d_j\}$ can be compiled into blocks of data, denoted by β_i .

A block is the concatenation of one or more chunks of data in ascending index order. The configuration of blocks of data $\{\beta_i\}$ are identical to the blocks $\{\alpha_i\}$ expressed in the intermittency.

We define the data model hypothesis as the data model expressed by the value of the intermittency bits within each block $b(\alpha_i)$. The model hypothesis is either signal (if $b(\alpha_i) = \blacksquare$) or noise (if $b(\alpha_i) = \square$). The model hypothesis dictates the data model which is used to analyse a given block. For example, in our intermittency Γ_{240} (equation (5.12)), the first block α_1 corresponds to $b(\alpha_i) = 1$, which is the signal model. As such, the first block of data $\beta_1 = \{d_1, d_2, d_3, d_4\}$ is interrogated against a signal model of the data. We will explore exactly what this means later in section 5.2.2. Similarly, the second block α_2 contains bits whose value is \square , dictating that the block of data $\beta_2 = \{d_5, d_6, d_7, d_8\}$ be interrogated against a noise model.

Finally, as the bitwise expression of the intermittency varies with each iteration of Γ_p , so does the configuration of blocks, changepoints and model hypotheses. By considering all 2^{N_b} possible intermittencies, we explore all possible combinations of blocks, changepoints and data hypotheses.

Explicitly, there exists an intermittency, $\Gamma_{15} = \square\square\square\square\blacksquare\blacksquare\blacksquare\blacksquare$, in which the blocks and the data within the blocks are identical to Γ_{240} . The difference in this case is that Γ_{15} and Γ_{240} containing opposing signal models in every chunk. This is the opposite hypotheses, $\Gamma_{\tilde{p}}$, to Γ_{240} .

5.2.1.1 Number of possible blocks

By chunking the data the up into N_b chunks, we limit the number of possible ways to make up a block. The total number of blocks it is possible to make is given by $N_b(N_b + 1)/2$. For a configuration with $N_b = 10$, we can expect only to have to construct $110/2 = 55$ different blocks of data.

This method of coarse chunking is sufficient to test the permanence detection criteria. We know from sections 1.3.4 and 4.1 that the addition of a small amount of integration time has the effect of raising the SNR only slightly. Similarly, a signal which only drops out for a small amount of time only lowers the SNR slightly. Coarsely chunking the data as described gives us greater resolution in SNR for a much lower computing cost by sacrificing time resolution, as we will discuss in section 5.2.2.

5.2.2 Data models and likelihood estimations

As with any analysis, the heavy lifting lies in the likelihood estimation. The important figure in this method is not a model comparison between signal and noise as in many CW searches. Instead, we present a method in which we chose between several intermittencies Γ_p . As discussed, each Γ_p is constructed of several blocks, which are in turn composed of chunks of data. In a given Γ_p , each block is tested against a data model, $M_{b(\alpha_i)}$, dictated by the value of the intermittency in that block. By this, we mean that the likelihood that the data is described by the model, $P(\beta_i|M_{b(\alpha_i)})$ is estimated.

In this section, we define exactly the term $P(\beta_i|M_{b(\alpha_i)})$ for both $b(\alpha_i) = \square$ and $b(\alpha_i) = \blacksquare$. We use an existing tool for estimating the evidence of a signal model in data, and for calculating the evidence for a noise model. In particular, we use the evidence estimates used in the Bayesian targeted heterodyne method outlined in section 1.3.2.1, which we will touch on once more in this section. This method utilises a single template method, which is appropriate for the late-stage follow up of an all-sky candidate, whose source parameters have been well constrained through several stages of follow up.

5.2.2.1 Bayesian targeted heterodyne method

Though the method was described in detail in section 1.3.2.1, here we briefly review it. The Bayesian targeted heterodyne method involves two key components: a pre-processing stage and a Bayesian evidence estimation stage. The latter stage includes a parameter estimation portion, however, in this method, this portion of the evidence estimation is ignored.

The pre-processing stage consists of a heterodyne and a strong low-pass filter. The heterodyne transformation multiplies the data by a complex phase evolution $\exp[-i\phi(t, \lambda)]$, where $\phi(t, \lambda)$ is the phase evolution of the target source, as shown in equation (1.31). In essence, this “unwinds” the phase evolution of the source, changing the source’s frequency f to a very low frequency component at 0 Hz, and a high frequency component at $2f$. We then implement a ninth order low-pass Butterworth filter in order to remove this high frequency component. This data is then heavily down sampled to a rate of one sample per minute.

We perform this heterodyne and filtering procedure to each block of data β_i . However, we can equally perform this procedure to the entire dataset D prior to chunking the data, as the targeted search uses only a single signal template.

The second component of the Bayesian targeted heterodyne method is the likelihood estimates.

5.2.2.2 Signal log-likelihood estimation ($b(\alpha_i) = \blacksquare$)

Firstly, we define the log-likelihood given that the model is M_{\blacksquare} , the signal model. In this case, we use the log-Bayesian evidence, which is the logarithm of an estimate of the likelihood multiplied by the priors on each parameter, marginalised over the searched parameters.

The marginal likelihood estimate within a single block of data, given the signal model is given by [52]

$$\ln(P(\beta_i|\phi(t), M_{\blacksquare})) = \sum_{v=1}^C \left(\mathcal{A}_v - m_v \ln \left[\sum_{n=n_0}^{n_0+m_v-1} |\beta_n(t) - h(\phi, t)_v|^2 \right] \right). \quad (5.15)$$

In equation (5.15), the block of data is divided up into C stretches of data, each indexed by v , and whose lengths are m_v . The running index n runs from the beginning of a stretch of data $n_0 = \sum_{l=1}^v 1 + m_l - 1$ up until the block's end, $n_0 + m_v - 1$. The prefactor term \mathcal{A} is given as

$$\mathcal{A}_v = \ln([v-1]!) - \ln 2 - m_v \ln \pi. \quad (5.16)$$

It is important to note here that the stretches of data C used to determine the Student's t-likelihood, are not the same as the data blocks β_i , but subdivisions of the blocks.

While equation (5.15) gives the likelihood of signal in the data for a given set of parameters, we must explore the parameter space in order to marginalise this over the prior volume. The method described utilises a nested sampler to explore the prior volume of the parameters h_0, ϕ_0, ψ and $\cos \iota$. These techniques are detailed in [52], and touched upon earlier in section 1.3.2.1.

5.2.2.3 Noise likelihood estimation ($b(\alpha_i) = \square$)

The likelihood that the data in a given block is Student's t-distributed [52] is calculated by

$$\ln(P(\beta_i|\phi(t), M_{\square})) = \sum_{v=1}^C \left(\mathcal{A}_v - m_v \ln \left[\sum_{n=n_0}^{n_0+m_v-1} |B_{k,n}(t)|^2 \right] \right). \quad (5.17)$$

The terms \mathcal{A}_v , n_0 and m_v in equation (5.17) are consistent with their definitions in section 5.2.2.2.

Unlike the signal estimate, this can be exactly calculated relatively quickly. As there is no signal template (and thus no prior volume to explore), we do not need nested sampling techniques to calculate this value.

5.2.2.4 Model prior probabilities

In this formulation, we only allow two data models, a signal model and a noise model. As we intend the method to discriminate between the two models, we can use the model selection form of Bayes' theorem seen in equation (1.16). In this formulation, we must supply each model with a prior probability. In this case, we choose to supply the least informative prior, one that is uniform over both models. As we have only two models, whose uniform priors must sum to unity, then the prior probability $P(M_{b(\alpha_i)}) = 0.5$ in each case.

5.2.3 Bringing it together: $P(\Gamma_p)$

In RBB, as each block of data is treated as an independent observation, we are able to combine the evidence values from each block in order to find an overall likelihood value for a given intermittency. The log-evidence that are obtained for each block are summed to give

$$\ln P(\{\beta_i\}|\{b(\alpha_i)\}) = \sum_i \{\ln [P(\beta_i|b(\alpha_i))] + \ln P(b(\alpha_i))\}. \quad (5.18)$$

We can consider that Γ_p is constructed from several bits, $b(\alpha_i)$, so we can substitute $\{b(\alpha_i)\}$ with Γ_p . Similarly, as a complete set of blocks $\{\beta_i\}$ is the entire dataset, D , divided into convenient sets, the terms $\{\beta_i\}$ and D are interchangeable here.

We can estimate values of $\ln P(D|\Gamma_p)$ for each value of p from $p = 0$ to $p = 2^{N_b} - 1$. This range of intermitencies covers the range of possible combinations of blocks and models that are able to expressed in a N_b -bit intermittency.

As discussed in section 5.2.1.1, a configuration with N_b chunks contains $N_b(N_b + 1)/2$ unique blocks. These blocks can be constructed, and their log-likelihoods estimated and recorded before we reconstruct Γ_p from them. Once the evidence calculations are performed for the $N_b(N_b + 1)/2$ blocks, calculating the log-likelihoods of the intermitencies is a trivial summation.

At this point in the analysis, we have an expression for the log-likelihood of a given intermittency, however, if we are to treat this as a fully Bayesian analysis, we must also assign prior probabilities to each Γ_p . Additionally, we aim to show that this analysis is analogous to a BB formulation. We will see in section 5.2.4 that it is precisely the prior values for Γ_p that bring this method to parity with a similar BB formulation.

5.2.4 Comparison with BB

In BB, which we describe in detail throughout section 5.1, the key concepts are also changepoints and blocks. Here, changepoints are inserted in order to divide a dataset into blocks. In equation (5.7), we arrived at the expression for a BB posterior probability, given the positions of the N_τ changepoints at times T_j , and selecting the configuration of models which maximises the posterior probability. For convenience, we restate it here, as

$$P(\{M_j\}|\{B_j\}, \{T_j\}) \propto \prod_{j=0}^{N_\tau} \left[\max_{M_j} P(B_j|M_j, T_j, T_{j+1}) P(M_j) \right], \quad (5.19)$$

In section 5.1.2.1, we stated that for a tCW follow-up, we can use two models only, a signal model and a noise model. In BB, as the models in two adjacent blocks must be different, with a choice from only two models, for a given set of changepoints T_j , there are only two possible configurations of models. Switching the models in any one block must necessarily switch the models in every block. For each set of changepoints T_j , the two possible configurations are one in which the first block's model is the signal

model M^S , and another in which the model in the first block is the noise model M^N . We can define these two configurations as $M^{(1)}$ and $M^{(2)}$ respectively. Thus, can write this explicitly in the calculation of $\ln [P(\{M_j\}|\{T_j\}, \{B_j\})]$ as

$$\ln [P(\{T_j\}|\{B_j\}, M^{(1)})] \propto \sum_{j=1}^{N_\tau} \{ \ln [P(B_j|M^{(1)}, T_j, T_{j+1})P(M^{(1)})P(T_j)P(T_{j+1})] \}, \quad (5.20)$$

$$\ln [P(\{T_j\}|\{B_j\}, M^{(2)})] \propto \sum_{j=1}^{N_\tau} \{ \ln [P(B_j|M^{(2)}, T_j, T_{j+1})P(M^{(2)})P(T_j)P(T_{j+1})] \}. \quad (5.21)$$

In RBB, we defined blocks as both a series of contiguous intermittency bits with the same value, and as a series of contiguous chunks of data in those same positions. We see that for a given intermittency Γ_p , there is a fixed configuration of blocks. In RBB, the selection of models is emergent from Γ_p , dictated by the values of b_{α_i} . Whereas in BB we can define a set of models $M^{(1)}$ and its opposite configuration $M^{(2)}$, in RBB, an opposite configuration is the result of a different value of Γ_p altogether. For each Γ_p , there exists such an opposite configuration, $\Gamma_{\bar{p}}$.

With this in mind, we can compare the likelihood of a RBB intermittency Γ_p to the BB likelihood of a given set of changepoints, given a model configuration (either $M^{(1)}$ or $M^{(2)}$), as given in 5.20, as they express the same concept – the likelihood given a set of changepoints and models (as chosen in BB, and as dictated by Γ_p in RBB). For convenience, we restate the likelihood of Γ_p as given in equation(5.18)

$$\ln P(\{b(\alpha_i)\}|\{\beta_i\}) = \sum_i \{ \ln [P(\beta_i|b(\alpha_i))] + \ln P(b(\alpha_i)) \}. \quad (5.22)$$

In both RBB, we estimate the value of $P(\beta_i|b(\alpha_i))$, (the likelihood of the model choice, equivalently $P(B_j|M^{(1)}, T_j, T_{j+1})$ in BB) using established likelihood estimators. Similarly, as in both cases, we only use two data models, a signal model and a noise model,

we assigned a prior of $P(M) = 0.5$ to each model.

The only difference between the expressions is in the dependence of the BB expression on its changepoints. As blocks, and therefore changepoints in RBB are emergent from the choice of intermittency, this is equivalent to a dependence on the intermittency itself. Instead of defining prior probabilities of the locations of changepoints T_j and N_τ , we can instead consider a prior probability for each Γ_p .

Thus, up to these prior terms, this RBB formulation is equivalent to the BB formulation discussed in section 5.1.2, with the added constraint that the positions of T_j are constrained to only occur at times $T_j = nL/(N_b - 1)$ where n is any integer between 1 and $N_b - 1$. Though we prescribe the RBB method with about ten chunks of data, the analysis holds for arbitrary numbers of chunks, meaning that for signals which we expect to have higher amplitude, we can increase the number of blocks (and thus time resolution), by decreasing the length of each block.

5.2.4.1 Prior probabilities of Γ_p

We have freedom in the definition prior probabilities, $P(\Gamma_p)$, to choose what configurations we favour. As the intermittency determines both the number of blocks (and therefore changepoints), as well as their positions, we can consider this a joint prior on both of these. Naively, we may consider a uniform prior across all intermittencies, however, as we will see, this tends to favour some values of the number of blocks. Instead, if we wish to find again the BB priors on N_τ and T_j , we must have a more complicated prior probability distribution on Γ_p .

We can consider the distribution of number of changepoints from all intermittencies. We can consider first that irrespective of N_b , the number of bits, there can only ever be two intermittencies with $N_\tau = N_b - 1$, these are configurations where each block is a single bit. Additionally, there are only two configurations with $N_\tau = 0$, as in each case, the block spans the entire dataset, which can take one of two values, namely \blacksquare and \square .

Further, there are only $N_b - 1$ configurations with $N_\tau = 1$, as there are only $N_b - 1$ positions to place a single changepoint. Similarly, there are only $N_b - 1$ configurations with $N_b - 2$ changepoints, as there are only $N_b - 1$ positions from which to remove a changepoint relative to the configuration with $N_\tau = N_b - 1$. Carrying along this exercise, we see that the number of configurations available for intermittencies with b bits and τ changepoints is given by C_b^τ , the binomial coefficient.

A prior which is uniform over all intermittencies will result in a distribution over N_τ which peaks at $N_\tau = N_b/2$ due to there being more of these configurations than those with higher or lower values of N_τ . Instead, in order to bring the effective prior on N_τ to a uniform distribution between $N_\tau = [0, N_b - 1]$, we apply a prior probability of

$$P(\Gamma_p) \propto \frac{1}{C_b^\tau}. \quad (5.23)$$

Similarly, we should consider the distribution of the changepoint positions within the RBB formulation. However here, the distribution of changepoint position within the set of all intermittencies is uniform, and so we do not need to modify our prior term on Γ_p in order to arrive at a uniform prior on the position of the changepoints.

With the prior probability for Γ_p (as in equation (5.23)) being equal to a uniform prior over N_τ and T_j in a BB formulation, we are now ready to find the figure of merit for this method: the posterior probability of Γ_p .

5.2.5 Posterior probability of Γ_p

The RBB method tests an exhaustive list of all Γ_p for a given N_b , and as such, we are able to calculate a posterior probability of each. The likelihood value for a given Γ_p is given in (5.18). Additionally, the prior probability of Γ_p , given in equation (5.23), balances out the natural weighting of the values of N_τ that emerge from the set of all Γ_p . We can put these elements together in Bayes' theorem (as in equation (1.11)),

which states that

$$P(\Gamma_p|\{\beta_i\}) = \frac{P(\Gamma_p)P(\{\beta_i\}|\Gamma_p)}{P(\{\beta_i\})}. \quad (5.24)$$

Here, the evidence term on the denominator is given by

$$P(\{\beta_i\}) = \sum_p [P(\{\beta_i\}|\Gamma_p)P(\Gamma_p)]. \quad (5.25)$$

As we naturally calculate the likelihood of each Γ_p , it is trivial to calculate the posterior probability as

$$P(\Gamma_p|\{\beta_i\}) = \frac{P(\{\beta_i\}|\Gamma_p)P(\Gamma_p)}{\sum_p [P(\{\beta_i\}|\Gamma_p)P(\Gamma_p)]}, \quad (5.26)$$

or, as we arrive at the log-likelihood, the log-posterior is given by

$$\ln[P(\Gamma_p|\{\beta_i\})] = \ln[P(\{\beta_i\}|\Gamma_p)] + \ln[P(\Gamma_p)] - \ln\left[\sum_p [P(\{\beta_i\}|\Gamma_p)P(\Gamma_p)]\right]. \quad (5.27)$$

As described, the RBB method can approximate BB, by searching coherent blocks of data for a number of models (in this case, 2 models, signal and noise), and by testing all possible combinations of blocks and models. This creates an analysis in which much of the expensive nested sampling can be done in advance, and the number of blocks to be analysed is given by $N_b(N_b + 1)/2$. In the pursuit of better sensitivity to the presence of tCWs, the algorithm loses precision in the beginning and ending times of a tCW signal. However, we posit that this round of follow-up is better able to indicate transient behaviour of a candidate signal, allowing other more expensive tCW searches (such as those outlined in 4.3), which are better optimised to accurately determine the t_0 and t_f of a tCW. It also is able to determine the persistence of a CW candidate, in order to check against the persistence detection criteria outlined in section 1.3.4. The figure of merit is the posterior probability of the intermittencies, $P(\Gamma_p)$, which can be estimated by performing well reviewed CW analyses on combinations of chunks of data in a prescribed manner.

5.3 Algorithm output and the barcode plot

The key outputs from the RBB algorithm must be illustrative as to the true underlying transient nature of the signal, which we express as the posterior probabilities of its intermittencies. We present these figures of merit within the RBB “barcode plot”, which shows the intermittencies, ranked by their posterior probabilities, and plotted alongside their posteriors. We show an example of such a plot in figure 5.2.

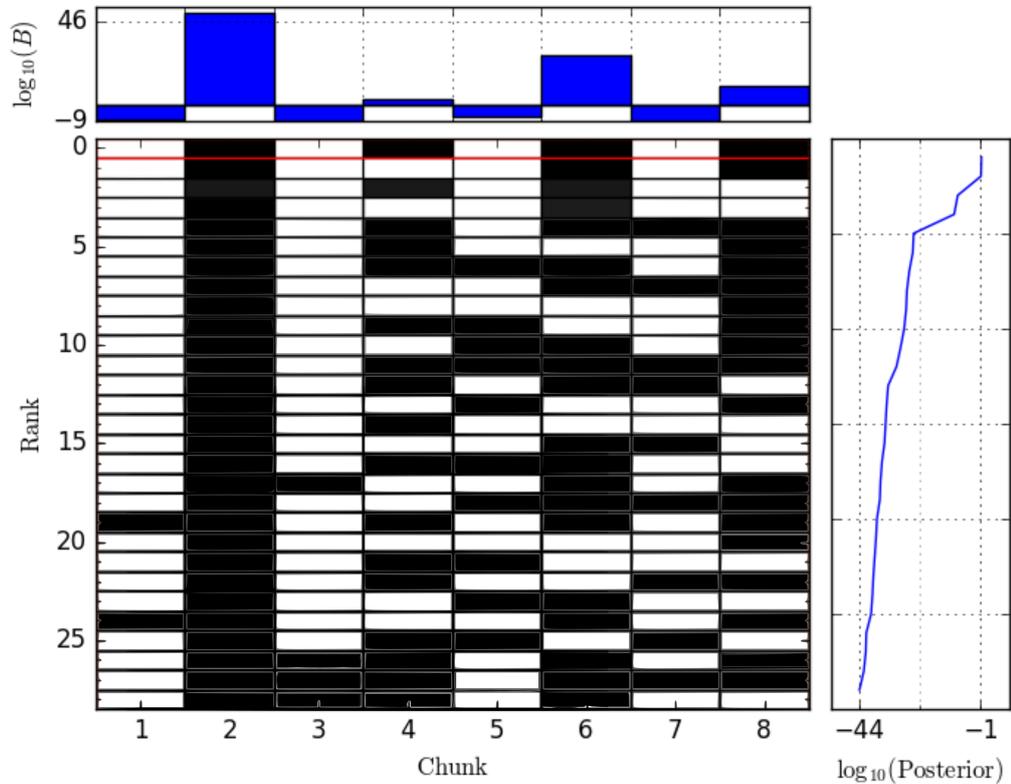


FIGURE 5.2: The plot shows an example of a barcode plot. At the top of the plot, the bar chart shows the \log_{10} Bayes factor of each chunk of data. The central panel shows the intermittencies, represented by their binary expressions in terms of \blacksquare and \square , which rank within the top 30 in terms of the value of $P(\Gamma_p)$, in descending order from top to bottom. The rightmost panel shows the $\log_{10}(P(\Gamma_p))$, for each intermittency in the central panel.

This example shows the top 30 ranked intermittencies (centre) against their posteriors (right), and the \log_{10} signal-versus-noise Bayes factor for each chunk (top). In this

case, with $N_b = 8$, there are a total of $2^{N_b} = 256$ possible intermittencies, though only the top 30 are shown for clarity. In this example, we test for an injected signal with exactly the intermittency that is recovered. We will explore in much greater detail this, and other injected signals against which we tested the algorithm in chapter 6. The red rectangle around the topmost row in the central panel highlights a chosen intermittency hypothesis. In this case, as we knew the injection intermittency, we chose to highlight that. In general, we can choose to highlight any intermittency.

The advantage of this plot is that it shows succinctly any “structure” within the most probable intermittencies. For example, in figure 5.2, the second chunk has a very high SNR. As we look at the central panel, we see that the bit in the second position of each row is consistently represented as \blacksquare . Conversely, the chunk in the fourth position has the lowest positive $\log_{10}(\text{SNR})$ of any chunk. In the preferred intermittency, it is represented by \blacksquare , however in the next intermittency down, the fourth chunk is altered to \square . It represents the lowest “penalty” for the intermittency assuming the wrong model of any chunk, and is therefore the next-most probable.

Additionally, as the Bayesian targeted heterodyne analysis is an effective parameter estimator for signal parameters, we can recover the most likely parameters within each block. If the barcode plot shows good evidence for transient structure in a candidate, we can recover some signal parameters in any given block of data from the nested sampling algorithm. For example, we can recover the strain h_0 of a strong candidate in any block whose most probably bitwise intermittency value is \blacksquare .

5.4 RBB: a summary

The RBB method is a late-stage follow-up method for CW and tCW candidates. The RBB method is a configuration of BB, with block lengths constrained to multiples

of a fundamental chunk length, and two data hypothesis: \blacksquare and \square , corresponding to signal and noise respectively.

The evidences for each data hypothesis are estimated for each allowable block, and then combined in all allowable ways, to create an exhaustive list of data hypotheses and intermittencies. A prior probability value is calculated for each intermittency, dependant on the number of changepoints in that intermittency. The log-prior and data hypothesis log-evidences are then summed to generate the intermittency log-evidence.

As the list of intermittencies is exhaustive, we are able to calculate the posterior probability for each intermittencies. The intermittencies are ranked by posterior, and the barcode plot is generated.

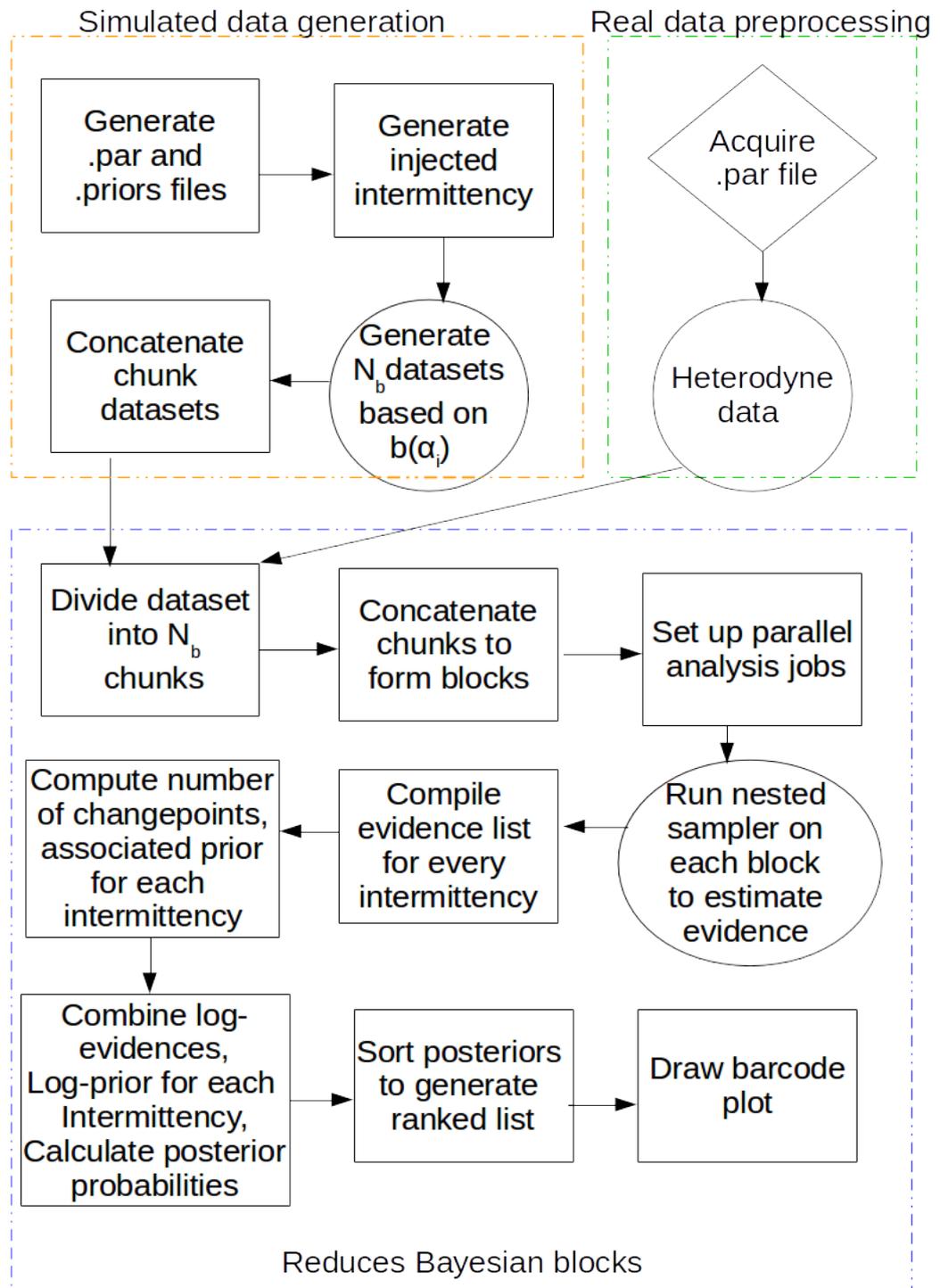


FIGURE 5.3: This flow-chart describes the tasks involved in the RBB method. The chart includes simulated data generation (inside orange box), detector data handling (green box), and RBB processes (blue box). The shapes of each task indicate the origin of the code used to carry out that task. Rectangle: My own code, ellipsoid: pre-existing code, diamond: usually done by hand.

This process, as well as the process for simulated data generation and real data handling prior to RBB is summarised in figure 5.3. In the figure, tasks done using pre-existing code are depicted within an ellipsoid, and tasks which require code that I wrote are described in rectangles. Tasks which are often done by hand are inside diamonds.

Chapter 6

Results from RBB tests and candidate follow-up

Having outlined RBB in chapter 5, in this chapter, we examine the validity of an RBB implementation by detailing a simple test of RBB on simple, simulated data. Following that, we extend the tests of RBB to the detection of signals with known intermittencies in data from GW detectors in section 6.4. Finally, in section 6.5 we apply RBB as a follow-up algorithm to viable tCW candidate signals from the O1 all-sky low frequency search.

6.1 RBB on simulated data

In order to validate RBB, we can implement it with data which we can understand exactly. By far the simplest way to do this is often to produce our own data. In this section, we detail several sets of simulated data against which RBB was tested. Each of these validations is performed on a set of simulated signals injected onto Gaussian noise. To begin with in section 6.2, we split each simulation into only two chunks, which we analysed in detail. Following this, in section 6.3 we test RBB against several longer datasets, each of which is divided into eight chunks. This second, larger set of

injected data covers a range of injected signal SNRs, signal parameters and injection intermittencies. A total of 2500 simulated datasets were analysed.

6.1.1 Description of simulated data

In the following simulations, we attempt to simulate transient signals in aLIGO data. As such, in each simulation, we use two data streams, one to mimic LLO data, and another to mimic LHO. The simulated data on which the RBB algorithm was first tested can be considered as comprised of three components, a noise component, n_i , a signal component, h_i , and an intermittency, I_i . The injection intermittency can be considered as a windowing function which modulates the signal. The final data, d_i can be considered a combination of the three components, as

$$d_i = n_i + I_i h_i. \quad (6.1)$$

In this section, we will examine each of these three components individually, and the way in which they are combined in order to create the simulated tCW signals.

Each simulation was comprised of a number of chunks of data, each chunk spanned two days. The choice was made to use only short observing periods here in order to keep the total amount of data relatively low. As each observation must be split into its chunks, and assembled into blocks, the amount of data used for each simulation is proportional to $N_b(N_b + 1)/2$ times larger than each simulation. The simulations assumed a 100% duty factor (both observatories were observing at all times). Whilst the amount of data used in a two-chunk set remains relatively low (with only three possible blocks), the eight-chunk sets would each need to be composed into 36 different blocks. The simulated data were generated as fine-heterodyned data, with one sample each minute.

6.1.1.1 Gaussian noise generation

Each simulated observation generated datasets from the two aLIGO observatories, LLO and LHO. The simulations approximate the noise expected in these detectors. As such, our noise was drawn from a Gaussian distribution. The frequency profile of the noise was given by the aLIGO “design sensitivity” curve, the noise levels that aLIGO is expected to achieve by its fourth observing run [145]. Independent noise samples were drawn from a stationary Gaussian distribution with a mean of $S_h(f)$, the spectral density of the detector at the frequency of the source. The simulated data had a duty factor of 100% – meaning that there were no data drop-outs in the simulated observation.

6.1.1.2 Signal parameters

The signal injected into each simulated observation was generated from a single parameter `.par` file. An example of a `.par` file can be found in appendix D. Each parameter file contained a sky position (given as right ascension $[\alpha]$ and declination $[\delta]$), source frequency, source spin-down, polarisation angle, initial phase and orientation angle. Based on this set of source parameters, a coherent signal was generated within each chunk. Additionally, the generated signal was coherent between chunks, such that when any two adjacent chunks were compiled into a block, the signal was coherent within the block. Similarly, a single dataset injects a signal that is coherent between the simulated detectors, meaning that signals from the same source parameters were injected into the data in each detector.

As the integration time for these signals was very short (each chunk was only two sidereal days long), the amplitude of the simulated GW signals were re-scaled to a predetermined SNR value when integrated coherently over the chunk. This chunk-SNR (cSNR) dictated the h_0 value of the GW, given the other amplitude parameters

$\{\phi_0, \psi, \cos \iota\}$ which were drawn from uniform distributions, as $\phi_0 \in [0, 2\pi]$, $\psi \in [0, \pi/2]$ and $\cos \iota \in [-1, 1]$.

The cSNR value of each chunk was the same across all chunks containing signal in any given simulation. As the injected signals were coherent across chunks, and any given block of data was integrated coherently using the Bayesian targeted heterodyne method, the total SNR of a given block α_i , made of only chunks with injected signal, is given by the square root of the number of chunks in the block, L_α , as

$$\text{SNR}_\alpha = \text{cSNR} \sqrt{L_\alpha}. \quad (6.2)$$

As such, transient injections with different intermittencies may have different total SNRs. However, this is acceptable, as transient signals with the same GW amplitude and differing lengths will also have different SNRs. This rescaling to a fixed cSNR simply allows us to find signals in shorter datasets.

In these simulations, we did not apply a window to smooth out the data between adjacent \blacksquare and \square chunks. At the boundary between two blocks, the data are not continuous.

6.1.1.3 Intermittency generation

Finally, here we discuss the injection intermittencies used in these simulations. As discussed in section 5.2.1, the injection intermittency reflects the transient nature of a source. It is expressed as a bitwise number, $\Gamma_p = \{\gamma_i\}$, easily expressed as filled ($\gamma_i = \blacksquare$) and unfilled ($\gamma_i = \square$) boxes, signifying that a chunk of data contains signal and no signal respectively.

In the injection stage, these intermittencies acted as a windowing function to modulate the amplitude of the simulated signal. In a given chunk of data, the signal was multiplied either by 0 if $\gamma_i = \square$, or 1 if $\gamma_i = \blacksquare$.

For these simulations, in order to test RBB in recovering an injected intermittency, we used a configuration in which the injection intermittency could be exactly matched to a recovery intermittency. Injected signals were modulated by intermittencies with the same number of chunks with which they were recovered with RBB. As such, we injected signals based on possible RBB recovery intermittencies with $N_b = 2$ and $N_b = 8$ in each of the two and eight chunk cases respectively.

The prior probability on the intermittency Γ_p , as discussed in section 5.2.4.1, is such that the prior probability for the number of changepoints (which is encoded within Γ_p) is uniform. However, we did not inject signals with the intermittency Γ_{255} , as this is the intermittence of a truly CW. Other than this omission, in order to reflect that prior in the simulations, the injection intermittencies were also uniformly distributed in the number of changepoints. The result of this is that some simulations (for example in the eight chunk trials, Γ_0 , Γ_{85} and Γ_{170} , which represent the only cases where N_τ are either one or seven) are injected more frequently.

6.2 Two chunk injections

We begin by looking at a set of short, easily calculable datasets. The goal of testing this dataset is to deeply probe the mechanics behind RBB. This can inform us as to how the likelihood estimator we have chosen to use, the targeted Bayesian heterodyne method, affects the evidence values given for the signal and noise estimations. It also allows us to explore how the different noise realisations and nested sampler realisations can affect the evidence values used in RBB.

6.2.0.1 Injected signals and intermittencies

The low number of chunks in these simulations leads to a small set of simply constructed intermittencies. These were

$$\Gamma_0 = \square\square \quad (6.3)$$

$$\Gamma_1 = \square\blacksquare \quad (6.4)$$

$$\Gamma_2 = \blacksquare\square \quad (6.5)$$

$$\Gamma_3 = \blacksquare\blacksquare. \quad (6.6)$$

This small set of possible intermittencies makes the process far easier to follow than the more complicated arrangement in higher values of N_b . As we see in equations (6.3) to (6.6), in these simulations, we need only consider three blocks, which we can define as

$$\alpha_1 = \{b_1\} \quad (6.7)$$

$$\alpha_2 = \{b_2\} \quad (6.8)$$

$$\alpha_3 = \{b_1, b_2\} \quad (6.9)$$

In these simulations, as there are only four intermittencies, we chose to run 10 simulations each with the injected intermittencies Γ_0, Γ_2 and Γ_3 . We did not inject Γ_1 as it provides no information about the data that Γ_2 does not. RBB still attempts to recover all four intermittencies, independent of injection.

Additionally, in order to simplify the tests, the signal parameters of the injected signals were not changed between different realisations. Additionally, we examined simulations which used the same noise realisations over the 10 simulations. In this case, it is the same as running RBB and in 10 independent instances over the same dataset. This allows us to examine the effects of different nested sampling realisations on RBB, due

to the stochastic nature of the sampling methods employed.

In another instance, we injected the same signal over 10 different noise realisations. This allows us to characterise the effects that the noise has on RBB.

In all of these analyses, we used a nested sampler set-up with 2048 live points. A full discussion on the effects of vary the number of live points is explored in [52].

Additionally, we used narrow priors on h_0 . The prior range on h_0 was determined by running the nested sampler over the injection once, and the mean posterior sample (μ_h) of h_0 and the standard deviation (σ_h) were found, and used to calculate a narrow prior range for the investigation. The prior range for each trial recovery was between $\mu_h - 3\sigma_h$ and $\mu_h + 3\sigma_h$.

6.2.0.2 Nested sampler realisations

The first test was to analyse the same dataset 10 times, with independent nested sampler realisations. We examined the log-evidence values found by the likelihood estimator. The absolute value of the evidence of any given configuration is not particularly illuminative. Instead, we find that the Bayes factors of two hypotheses, or the variance, is the most illuminating.

However, one simple check that we can perform with the log-evidences is that RBB is summing them correctly. In particular, we can check that the noise evidence for the first chunk and second chunk sum to produce the evidence in the block containing both chunks, as it should.

In table 6.1 we present the mean noise evidence values for each chunk, as well as its variation for each of the three intermittent injections

We see in table 6.1 that the noise evidence has no (or close to no) variance. In the one case where the variance is non-zero, it is likely due to some rounding difference in the calculation, and is several orders of magnitude smaller than the evidence value, and

TABLE 6.1: This table shows the mean and variance of the noise evidence for each chunk in two-chunk simulations, over three injected intermitten-
cies. The value μ gives the order of magnitude of evidence in both α_1
and α_2 , and the second column shows the difference between the two
means, to illustrate the finer detail. We see that this difference is much
larger than the variance.

		μ	$\langle\alpha_1\rangle - \langle\alpha_2\rangle$	σ_1	σ_2
Injection	Γ_0	1.38×10^6	97.6	4×10^{-10}	0
	Γ_2	1.37×10^6	-16	0	0
	Γ_3	1.36×10^6	98.7	0	0

is therefore negligible. We expect this, as the noise evidence is analytically calculable, and need not be estimated by nested sampling.

The values of the log-evidence are of the order 10^6 , meaning that the evidence values are $10^{(10^6)}$. This high value of evidence reflects the precision of the data. In general, an evidence value indicates how well data constrains a hypothesis, in this case, how well this simulated data constrains the signal or noise hypothesis. As the aLIGO detectors (and thus the simulated data based on their sensitivity) are designed to take particularly precise data, they constrain both hypotheses very well. We see in equation (1.34) that the likelihood is the negative log of the sum of the data. This results in large positive numbers for very precise data.

As these noise log-evidence values have zero variance, we can also test that the noise evidence values in the chunks sum to the noise evidence value in the block which contains both chunks. We present these in table 6.2. Namely, we are checking that the noise evidence values in each row of table 6.1 sum to produce the noise evidence values in the first column of table 6.2. We can see that they do in each of the three cases presented.

The last important figure to look at in this set of data is the standard deviation among the different nested sampler runs. As in each case, we are examining the same noise realisation and signal in the data, the standard deviation of the noise reflects only the variation in the nested sampling routine.

TABLE 6.2: Here, we present the log-likelihood values for each Γ_p tested. The first section shows the mean values over 10 nested sampler runs, and the second shows their standard deviation

		Recovered intermittency			
		□□	□■	■□	■■
		Mean log-Bayes factor vs □□			
Injection	Γ_0	0	-12.3	-6.27	-15.6
	Γ_2	0	-67.0	111.6	37.9
	Γ_3	0	102.7	101.7	204.9
Standard deviation of log-evidence values					
Injection	Γ_0	4.65×10^{-10}	0.0816	0.0587	0.0959
	Γ_2	0	0.0593	0.0346	0.0621
	Γ_3	0	0.0212	0.0195	0.0339

The standard deviations of the nested sampler evidences are given in the second section of table 6.2. As the standard deviations (and therefore variances) of all of the noise evidence values is zero, we can take the standard deviation of each intermittency as the standard deviation of only the signal evidence estimation. We can see in all instances that the standard deviation of the evidence due to the nested sampler is several orders of magnitude smaller than the log-Bayes factors (the difference between the log-evidence values), and certainly negligible relative to the evidence values. As the standard deviation due to the nested sampler is so small, it is unlikely to affect the RBB analysis, or impact the determination of the most likely intermittency.

Finally, we can show that in each of these injections, RBB correctly recovered the injection intermittency. In these simulations, there were three injection intermitencies, namely Γ_0, Γ_2 and Γ_3 , as given in equations (6.3) to (6.6). RBB was allowed to search for any of the four possible intermitencies in the simulated data. RBB correctly recovers an intermittency when the input intermittency is the same as the intermittency recovered with the highest posterior probability.

To show this, we will look at the mean posterior probabilities of each recovered intermittency, and the standard deviation of the 10 repeat trials. These are presented in table 6.3.

TABLE 6.3: The mean and standard deviation of the posterior probabilities found in the 10 repeat trials of two-chunks injections. For brevity, we have capped the precision presented to three significant figures.

		Recovered intermittency			
		□□	□■	■□	■■
		Mean posterior probability			
Injection	Γ_0	0.940	4.08×10^{-3}	0.0553	9.73×10^{-4}
	Γ_2	1.81×10^{-19}	1.69×10^{-23}	1	8.01×10^{-13}
	Γ_3	5.82×10^{-37}	5.36×10^{-20}	3.35×10^{-20}	1
Standard deviation of posterior probabilities					
Injection	Γ_0	0.0212	1.44×10^{-3}	0.0194	3.34×10^{-4}
	Γ_2	5.69×10^{-19}	5.34×10^{-23}	0	2.49×10^{-12}
	Γ_3	1.83×10^{-36}	1.40×10^{-21}	1.14×10^{-21}	0

We see three key features in table 6.3, firstly, that in each case, the injected intermittency was correctly recovered by RBB, secondly, the injection intermittency’s posterior probability in the Γ_0 case is lower than those for Γ_2 and Γ_3 , and thirdly, the posteriors for the injection intermittencies in the other two cases are very high indeed.

To expand on the first point above, we see that for each injection, RBB correctly recovered the injection. Furthermore, we can consider an “opposite” intermittency, where the value of each bit in the injected intermittency is changed. We see that the posterior probability of this opposite intermittency, in each case, is the lowest posterior probability.

The second feature from table 6.3 is that the posterior probability for Γ_0 is lower than those for Γ_2 and Γ_3 . In order to understand this, we must consider the next most probable intermittency, Γ_2 . In this intermittency, the algorithm calculates the likelihood of a signal in the second chunk, however, the amplitude parameters of that signal are dependent on a prior probability distribution.

In this case, the prior distributions for the parameters ϕ_0, ψ and $\cos \iota$ are uniform over their allowed range, and h_0 is given a prior range of 1×10^{-24} to 2.5×10^{-24} . The most likely sample comes from the centre of this distribution, at 1.26×10^{-24} . When comparing this to the other injections, which include signal, the strain of the most

likely sample were 9.28×10^{-24} in Γ_2 and 9.44×10^{-24} in Γ_3 , each at least seven times higher than the most likely value of h_0 in the Γ_0 injection. It is much more likely that a weak signal can be mistaken for Gaussian noise than a strong signal.

If we instead selected a sample for Γ_0 with a comparable h_0 value to those selected for Γ_2 and Γ_3 , we can expect the posterior probability of the injected intermittency to be much higher.

Thirdly, we can see in table 6.3 that the posterior probability for the injected intermittency in the Γ_2 and Γ_3 intermittencies are given as 1, with no variance in their values over 10 analyses. This is mostly due to the strength of the signal. The injected signals in each chunk have an integrated SNR of 10, and so is easily distinguished from Gaussian noise. The precision of the output of the algorithm is not enough to distinguish the posterior probability of the injected intermittency from 1. It is for this reason that in the barcode plots (such as shown in figure 5.2), the right-most panel plots the \log_{10} values of the posterior probability.

6.2.0.3 Independent noise realisations

In addition, we examined a similar set of simulations, in which we injected an identical signal over 10 different realisations of Gaussian noise. This can inform us as to the effect of different noise realisations, and its effect size relative to those discussed previously. In this case, we are concerned with the evidence calculated in each chunk by the nested sampler, rather than the intermittency posteriors. We present these evidences in table 6.4.

The most obvious feature in table 6.4 is that the standard deviation of evidences among the 10 noise realisations is far larger than those found among 10 independent runs of the nested sampler, which are shown in table 6.2. We also see that the difference in the signal log-evidence between the two chunks is largest in Γ_2 , which we would expect. This is because Γ_2 is the only injection with a different bit-value in each chunk.

TABLE 6.4: This table shows the mean and variance of the noise evidence for each chunk in two-chunk simulations, over three injected intermittencies. Within each intermittency, the log-evidence values are taken across 10 analyses of the same signal injected over different Gaussian noise realisations.

		α_1		α_2	
		μ	$\langle\alpha_1\rangle - \langle\alpha_2\rangle$	σ_1	σ_2
Noise log-evidence					
Injection	Γ_0	2.61×10^5	-1.769	27.077	27.218
	Γ_2	2.57×10^5	-1.22×10^3	26.729	27.218
	Γ_3	2.57×10^5	-1.994	26.729	26.163
Signal log-evidence					
Injection	Γ_0	2.61×10^5	-1.748	27.256	27.563
	Γ_2	2.57×10^5	-1185.932	26.904	28.586
	Γ_3	2.57×10^5	-5.617	26.932	20.725

Perhaps more informative than the log-evidences in this case are the log-Bayes factors, given by the difference between the log-evidences for signal and noise. We present these values in table 6.5

TABLE 6.5: In this table we present the means and standard deviations of the signal-vs-noise log-Bayes factors in each block in a two-chunk RBB simulation.

		α_1		α_2		α_3	
		μ	σ	μ	σ	μ	σ
Injection	Γ_0	-1.69	1.27	-1.71	1.22	-2.89	0.86
	Γ_2	20.8	5.27	-17.5	4.31	1.34	3.20
	Γ_3	20.8	5.22	20.6	4.90	41.2	3.75

These log-Bayes factors contextualise the signal and noise log-evidences relative to each other. We see that the RBB algorithm tends to correctly identify the contents of each individual chunk, however is less decisive about the block containing both chunks (α_3) in the transient injection Γ_2 , which is likely because the truth is neither all signal or all noise, but the combination of the two.

It is important to recognise that these log-Bayes factors are in \log_{10} , and so a relatively small number, like -1.692 in fact represents a Bayes factor of about 0.02, meaning that there is roughly 50 times more evidence for noise than signal.

In blocks containing an injected signal, we see that the mean Bayes factor is very large, around 10^{20} , and though the standard deviation σ is also large, the results are still expected to be a long way from contentious. However, the same is not true of Γ_0 . In that case, the Bayesian targeted parameter estimator, contributions to the signal evidence are likely to be largest from the lowest values of h_0 .

Seeing, in these sets of two-chunk simulated data, that RBB behaves as we expect it to, we can now test it on a larger, more complicated set of injections, in order to characterise its performance. In section 6.3, we extend these simulated data sets from two chunks of two sidereal days to eight chunks of two sidereal days each.

6.3 Eight chunk injections

6.3.1 Description of the simulated data

In these simulations, we generated data in the same way as described in 6.1.1. The length of each chunk was two sidereal days, sampled once per minute. Each data set consisted of eight chunks, for a total of 16 days observing. The duty factor of the simulated data was 100%. The noise was generated to reflect the noise levels in the aLIGO detectors at design sensitivity, and the injected signals were the templates as seen at each aLIGO detector.

We injected each simulated dataset with a tCW signal with a given chunk-SNR (cSNR). The cSNR took integer values between 1 and 10, with 250 simulations at each value of cSNR. Additionally, a signal in any simulation is coherent across all chunks with signal; the signal in an intermittency with a drop-out (such as $\blacksquare\square\blacksquare$) is coherent across the drop-out.

In total, 2500 simulations were generated in this way. In each simulation, we recorded the injected intermittency and signal parameters, in order to aid in characterising the recovery efficiency of RBB.

6.3.2 Defining recovery statistics

The input intermittencies for each individual simulated dataset were known, which allowed us to measure the success of RBB simply as an injected intermittency recovery fraction. This is defined as

$$r_{\text{rec}} = \frac{N_{\text{rec}}}{N_{\text{sim}}}, \quad (6.10)$$

the number of simulations with correctly recovered intermittencies N_{rec} divided by the total number of simulations N_{sim} .

Additionally, if RBB did not recover the injected intermittency, we can characterise the inaccuracy of RBB as both the posterior probability of the highest ranked intermittency, which we denote as $P(\Gamma_{\text{rec}})$ and the posterior probability of the injected intermittency, denoted as $P(\Gamma_{\text{inj}})$. We can also consider the ranked position of Γ_{inj} . If $\Gamma_{\text{inj}} = \Gamma_{\text{rec}}$, then by definition, Γ_{inj} is the highest ranked intermittency, with a rank of 0. A higher rank number corresponds to the injected intermittency being further down the ranked list.

Finally, we can define the number of mismatched intermittency bits n_{mis} between the injected and recovered intermittency. When RBB correctly recovers the injected intermittency, by definition, the value of $n_{\text{mis}} = 0$. The value of n_{mis} is always an integer value between $n_{\text{mis}} = 0$ and n_b . It can be found by comparing the injected and recovered intermittencies bitwise, and in each position that their b_i values differ, the value of n_{mis} increases by one.

6.3.3 Describing the RBB configuration

In order to attempt to recover the injected intermittency, we used a RBB configuration with chunk sizes that matched exactly to those in the injection intermittency. There were equal length chunks of two sidereal days. For each simulated dataset, a parameter (`.par`) file contained the simulated frequency, spin-down and sky position used to inject the signal. A separate file contained three of the four amplitude parameters, ϕ_0, ψ and $\cos \iota$, with the strain parameter h_0 being rescaled in each case in order to meet the injected cSNR. In each simulation, the `.par` file was given to the likelihood estimator in order to search for the correct signal as injected.

In these studies, we used the same nested sampler configuration for each trial. We used 256 live points. As discussed in [52], a higher number of live points results in a lower spread in evidence values obtained by the nested sampler. However, we found that the run time of the nested sampler on so many datasets, and blocks within the datasets, was prohibitive, and as discussed in section 6.2, the variance in evidence due to the nested sampler is far lower than the variance in different noise realisations.

Additionally, we defined the prior distributions on all four amplitude parameters, with the prior distributions were given in a separate `.priors` file. An example of such a file can be found in appendix D. The prior probabilities of these parameters were

$$P(h_0) = P_U(h_0 = [10^{-25}, 10^{-20}]), \quad (6.11)$$

$$P(\phi_0) = P_U(\phi_0 = [0, \pi]), \quad (6.12)$$

$$P(\psi) = P_U(\psi = [0, \pi/2]), \quad (6.13)$$

$$P(\cos \iota) = P_U(\cos \iota = [-1, 1]). \quad (6.14)$$

where $P_U(X = [a, b])$ signifies a uniform distribution of values of X between a and b . These are the full ranges of the amplitude parameters for a rotating simple triaxial star, which reflects the injected signal model.

6.3.4 Results: recovery of injected intermittencies

6.3.5 Findings from simulated datasets

In this section, we will present the key results of RBB having performed analyses on the simulated datasets described at the beginning of section 6.3. A sample of the results presented in a readable format can be found in C.

6.3.5.1 Recovery fraction by cSNR

We plotted the mismatched bits as a set of stacked-bar charts. Each column in figure 6.1 represents all of the trials with a given cSNR value. Each colour of bar represents a different value of n_{mis} , and its height is proportional to the number of trials which were found to have that value of n_{mis} . Additionally, a blue dashed line traces the percentage of correctly recovered injections.

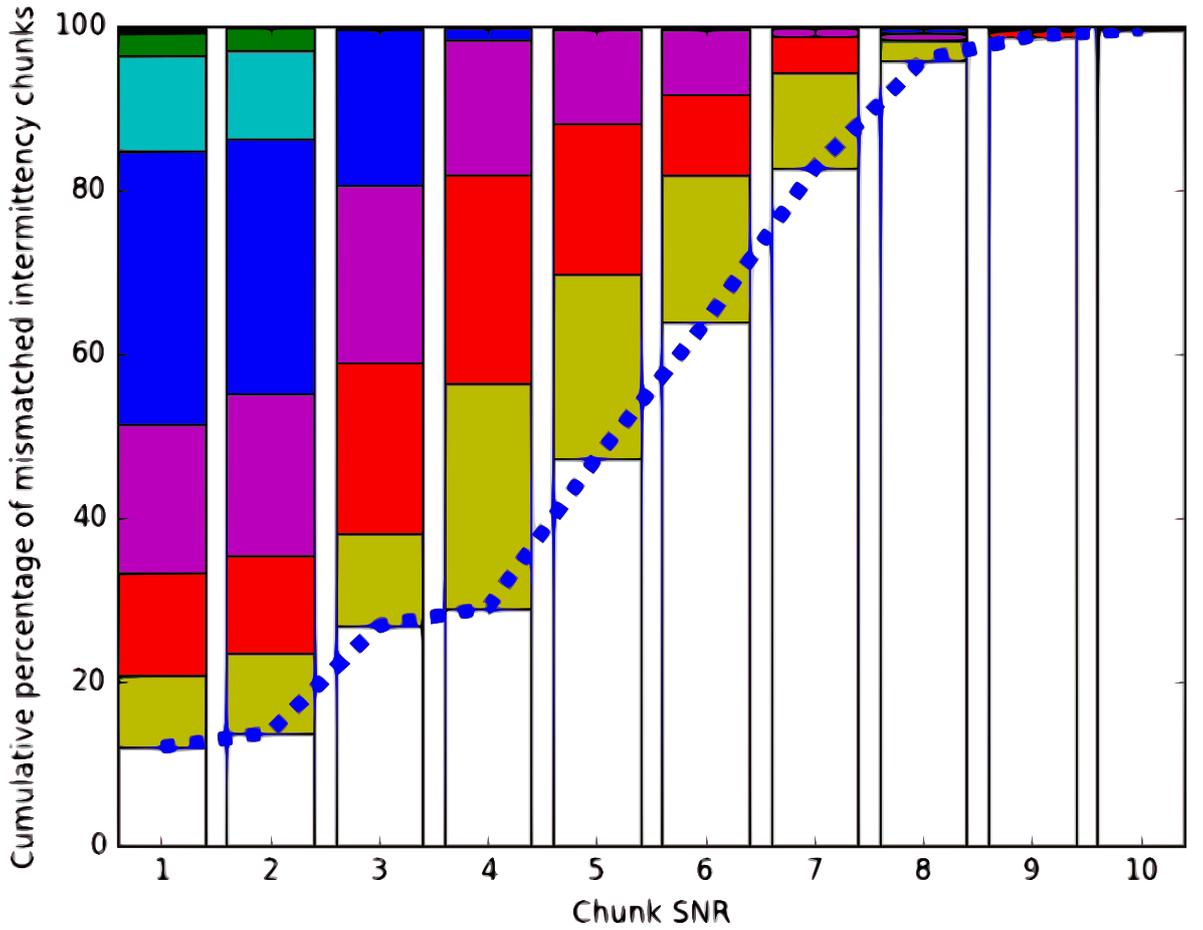


FIGURE 6.1: A stacked bar chart showing the mismatch. White bar: $n_{\text{mis}} = 0$, gold: $n_{\text{mis}} = 1$, red: $n_{\text{mis}} = 2$, violet: $n_{\text{mis}} = 3$, blue: $n_{\text{mis}} = 4$, teal: $n_{\text{mis}} = 5$, green: $n_{\text{mis}} = 6$, black: $n_{\text{mis}} = 7$. The blue dashed line traces the percentage of trials correctly recovered at each cSNR.

As we can see in figure 6.1, the first, top level take away is as the cSNR of injected signals increases, the value of r_{rec} tends towards unity. We might expect this, as a louder signal should be easier to detect.

Additionally, at these low cSNR values, we see that the distribution of n_{mis} is similar to a binomial spread of all values of n_{mis} . This is a reflection of the injected spread of Γ_P . Injections with cSNR of 1 are almost indistinguishable from datasets of only noise. As such, we would expect that RBB would prefer intermimacies made only of \square . So, the number of mismatched bits corresponds to the number of injected bits with \blacksquare . These follow a binomial law, as there are only eight ways to inject signals with exactly one

■, 28 with exactly two ■, and so on.

In fact, we see similar results for the injections with $\text{cSNR} = 2$. In all cases, the recovered intermittency is Γ_0 , resulting in a similar binomial distribution across n_{mis} . As we get to injections of $\text{cSNR} = 3$, we see the algorithm beginning to pick out signals more often. The number of trials with $n_{\text{mis}} = 6$ and 7 drops out almost entirely. From here, with each increase in cSNR , the values of $n_{\text{mis}} > 0$ decrease, and drop out entirely, as we expected.

6.3.5.2 Posterior spread by cSNR

Similarly, we can examine the spread of posterior probabilities of the recovered intermittencies. We show this as a set of histograms in figure 6.2.

In figure 6.2, we can see the distribution of posterior probabilities of the recovered intermittencies. Each of the 10 histograms corresponds to a value of cSNR . We see that the vast majority posterior probabilities in the $\text{cSNR} = 10$ trials are about a value of 0.95. We see that in figure 6.1 all but one of these are also correctly recovered. Similarly, we know that most trials with $\text{cSNR} = 1$ were recovered with the intermittency Γ_0 . We also see in figure 6.2 that RBB is relatively confident in this, as the posteriors are, for the most part, about 0.9. At a $\text{cSNR} = 2$, RBB tends to prefer recovering Γ_0 still, though the increased spread in the posterior probabilities shows that with louder signals injected, the evidence for signal is growing, and so the recovery of Γ_0 is somewhat a more conservative recommendation from RBB.

We see that the posterior probability is most spread out at $\text{cSNR} = 3$, just as RBB starts making slightly better informed guesses as to the input intermittency. It's at this cSNR that RBB begins recovering intermittencies other than Γ_0 more consistently. As the cSNR increases above 3, we see that the posterior probabilities of the recovered intermittencies tend to rise. By $\text{cSNR} = 7$, the majority of trials are recovered with a posterior greater than 0.9.

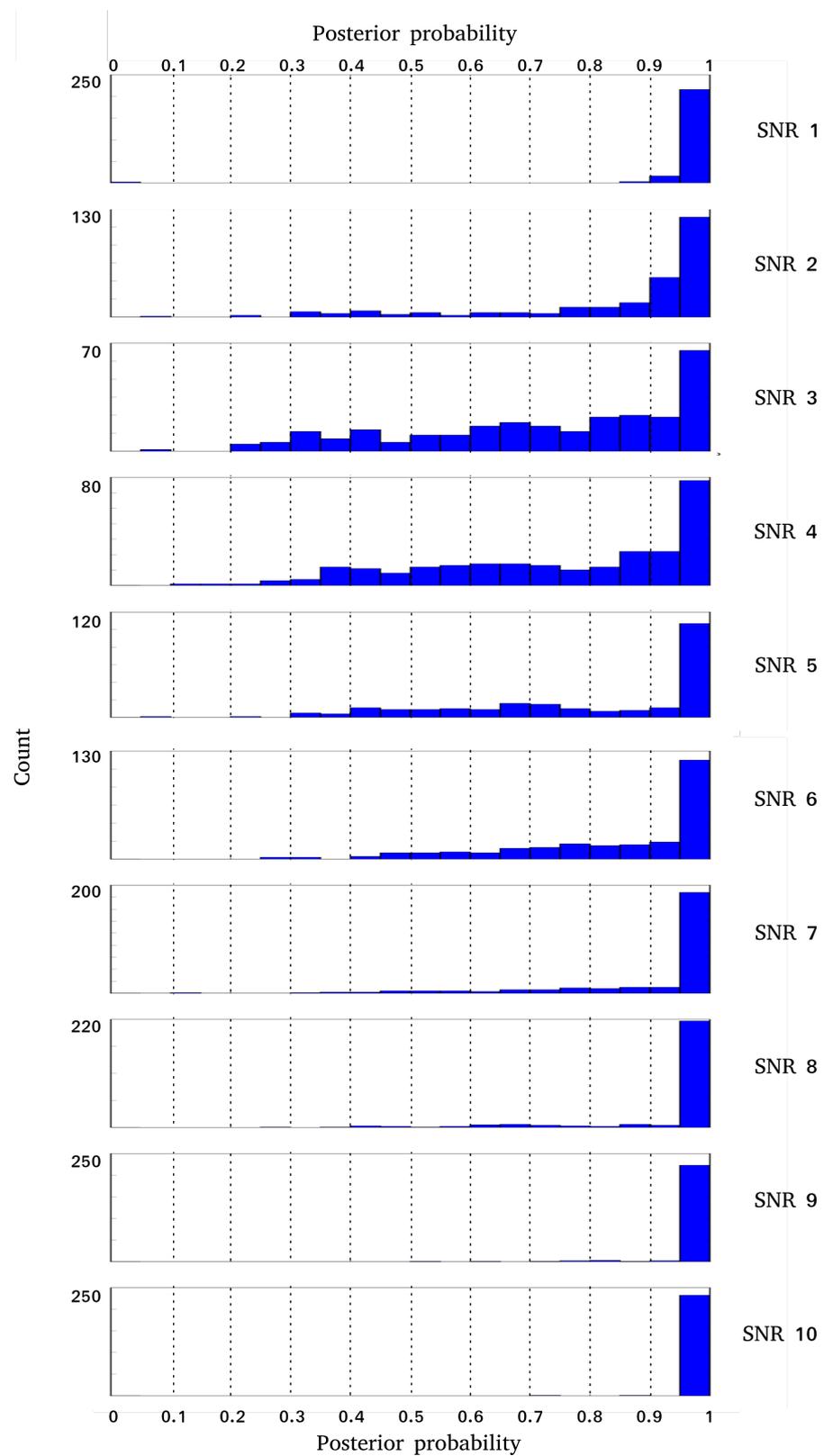


FIGURE 6.2: These histograms show the posterior probability for the recovered intermittency in all trials, with one plot for each value of injected cSNR.

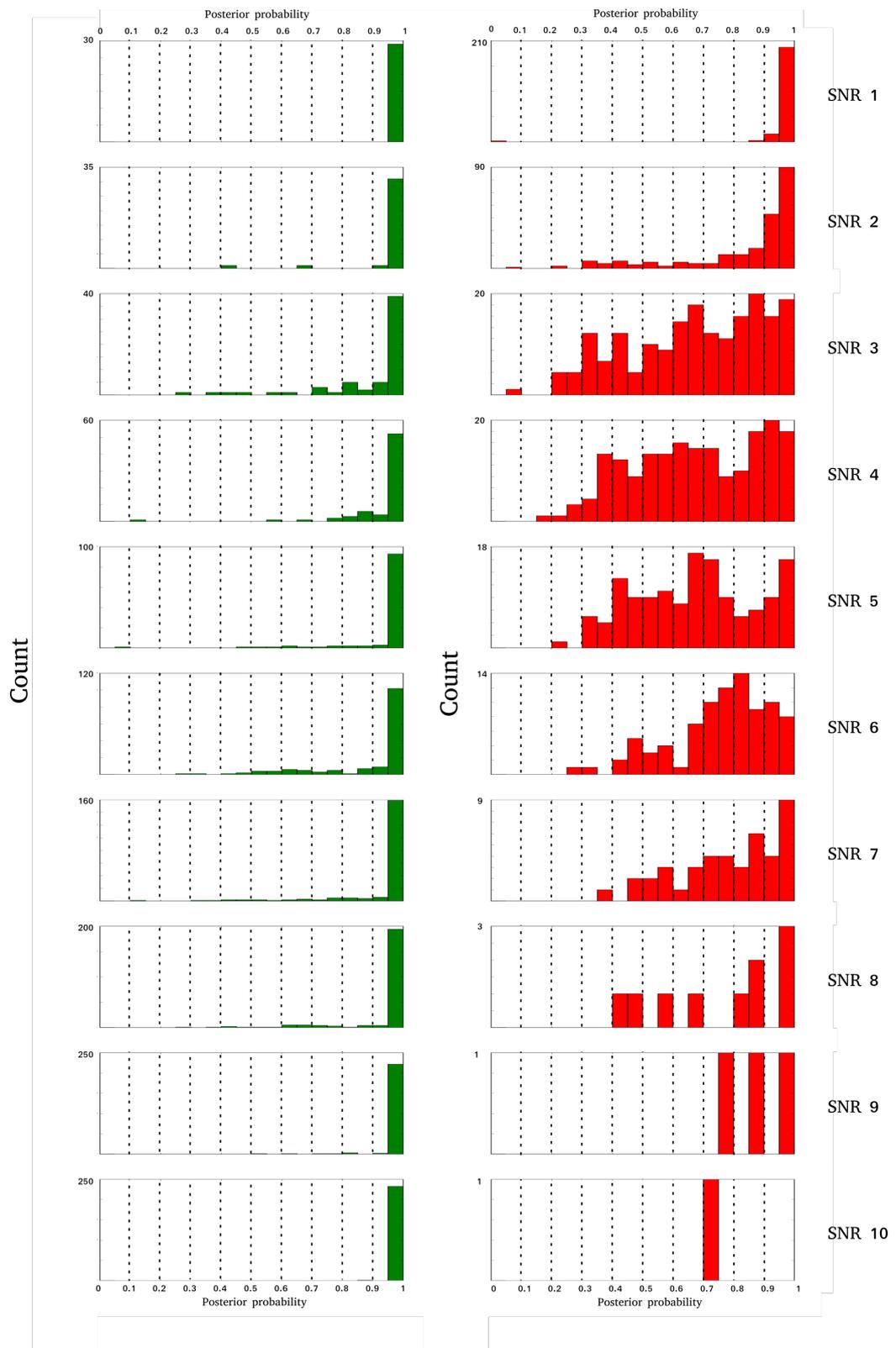


FIGURE 6.3: Here we present the histogrammed posterior probability of the recovered intermittency. The left column of histograms (in green) shows the count of those which were correctly recovered. The right-hand column (in red) shows the number of those which were incorrectly recovered. The posterior probabilities are binned into 20 equally sized bins, each is 0.05 wide.

Further, it may be illustrative to compare the posteriors of correctly and incorrectly recovered intermittencies. In figure 6.3, we can see two features. First of all, injections with a cSNR of 1 have similar posterior distributions regardless of whether or not they were recovered correctly. We know that these were recovered with the intermittency Γ_0 in each case, whether correct or not. At this low injected cSNR value, this is to be expected, as the signal is no louder than the noise.

On the other hand, above cSNR = 3, we see two different distributions in the correctly and incorrectly recovered intermittency's posteriors, which is clearly seen in cSNR of 3, 4, 5, 6 and 7, whilst the number of incorrectly recovered intermittencies is relatively high. In the correctly guessed intermittencies, we see that the vast majority of recoveries are made with a posterior probability above 0.95. As the cSNR increases, we see that fewer correctly recovered intermittencies are found with a probability less than 0.95. The incorrectly recovered intermittencies are found with a distribution which is much more uniform across posterior probabilities greater than $P(\Gamma_{\text{rec}}) \approx 0.25$. We do not see this at the high cSNR trials, as the number of incorrectly recovered jobs is too low. There are only three incorrectly recovered trials with cSNR 9 and one with cSNR of 10.

This tells us that RBB makes relatively confident selections of sensible intermittencies given the data. Data with a signal too quiet to be picked out over the noise will be identified as noise with high certainty. Louder intermittent signals will be recovered correctly with high certainty too. Where RBB is uncertain, we should expect the recovered intermittency to not mirror the input intermittency.

6.3.5.3 Ranked position spread by cSNR

Finally, we can examine the rank of the input intermittency, in the list of intermittencies sorted by their posterior probability. By definition, if an input intermittency is correctly recovered, it will be at the top of this ranked list, with a rank of 0. If it was the second

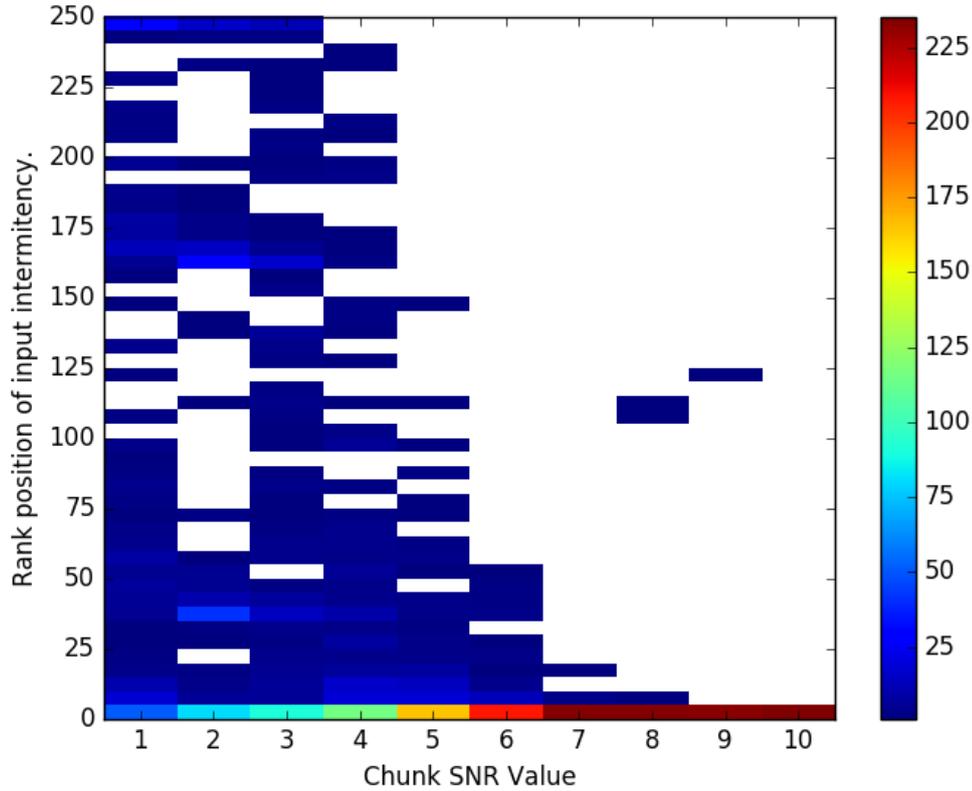


FIGURE 6.4: Here, we show a 2D histogram of the injected cSNR against the position of the injected intermittency in the ranked list of recovered intermittencies. Each bin in the vertical axis spans five ranks in the list. The colour of each pixel relates to the population of that bin, given by the colour bar on the right of the plot.

most likely intermittency, it will have a rank of 1, and so on. We can examine the distribution of the recovered ranks for these injections. Alongside this discussion, we will be examining how this rank manifests itself in the barcode plots, (which can be seen in the central panel of barcode plots such as 5.2), and what the input intermittency's position in the barcode plot, and the input intermittencies' nearest neighbours in the plot, might be able to tell us.

Figure 6.4 shows a 2D histogram of the position in the ranked list of the input intermittency. The height of the bins is represented by colour, as shown in the colour bar on the right of the figure. By definition, where the intermittency is correctly recovered, the ranked position is 0, at the top of the list. Figure 6.4 shows coarse bins in list

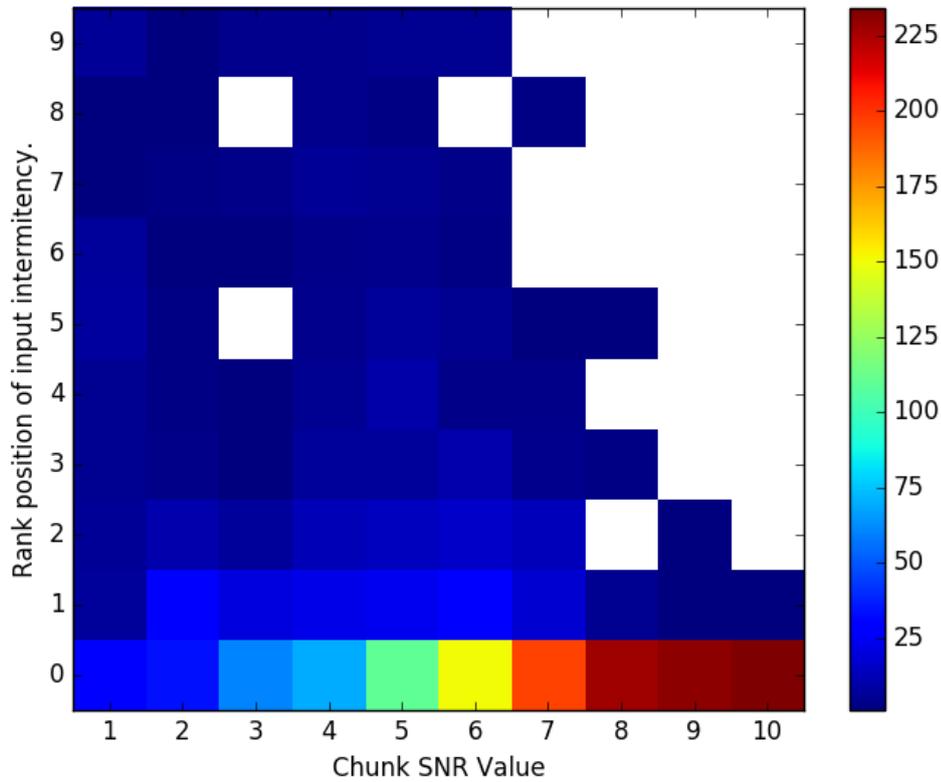


FIGURE 6.5: Here, we show a 2D histogram of the injected cSNR against the position of the injected intermittency in the ranked list of recovered intermittencies, limited to only the top ten ranks. We show fine detail here, with each bin on the vertical axis having a width of one rank position only. The colour of each pixel relates to the population in each bin, given by the colour bar on the right of the plot.

position, with a width of 5. As such, we do not see the very fine structure at the top of the lists. We show this instead in figure 6.5. In the coarse binned histogram, we see that at the lowest cSNR, that the list position of the input intermittency is relatively evenly spread throughout the ranked list. We expect this spread for the same reason we expect a spread in the number of mismatched bits.

At $\text{cSNR} = 5$, more than two thirds of the input intermittencies are in the top five intermittencies, ranked by posterior. As the cSNR increases, this proportion grows, until by $\text{cSNR} > 7$, all but three trials have input intermittencies within the top five as ranked by posterior.

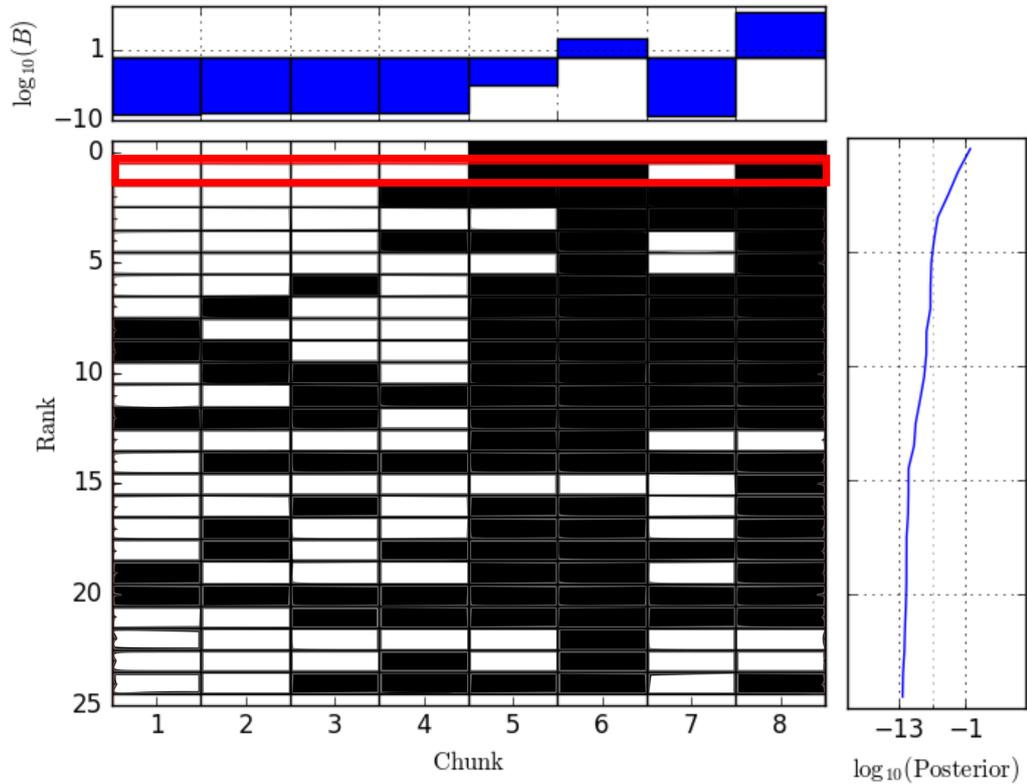


FIGURE 6.6: The barcode plot of trial number 234 with $\text{cSNR} = 5$ and $\Gamma_{\text{inj}} = \Gamma_{13}$.

In figure 6.5, we look at the finer detail at the top of the ranked lists. We show the histogram of the same data, with finer bins on the vertical axis, and focussed only on the top ten ranks. Trials where the input intermittency were in the top ten are included in this plot, and all others are omitted.

We see in this finer structure, when combined with the barcode plots for a given intermittency, a key feature of RBB. Even if RBB is not able to recover the *exact* input intermittency, it is able to recover much of the coarse structure in the intermittency.

To demonstrate this, we will examine that barcode plot of a trial with a $\text{cSNR} = 5$, input intermittency of $\Gamma_{13} = \square\square\square\square\square\square\square\square$, which was incorrectly recovered. We show this barcode plot in figure 6.6.

In particular, we are interested in the central and right hand panels of figure 6.6,

which show the ranked intermittencies, and their posteriors. In this case, we see that the input intermittency has the second highest posterior (it is highlighted by a red box). However, the recovered intermittency is only different in one position – the \square in the seventh position was swapped for a \blacksquare . This has the effect of reducing the number of changepoints by two. Further, the posterior probability of the highest ranked intermittency was $P(\Gamma_{15}) = 0.879$, where the second highest posterior was $P(\Gamma_{13}) = 0.0974$. The posterior of Γ_{15} is inconsistent with the posterior distribution we find in only correctly recovered intermittencies (see figure 6.3), and the posterior of Γ_{13} is not negligible relative to $P(\Gamma_{15})$, which signals that RBB is unable to strongly discriminate between these two intermittencies.

Further, if we continue to look down the ranked list in figure 6.6, we see that many of the same features present in Γ_{13} and Γ_{15} are present in several of highest ranked intermittencies. The third and fourth highest ranked intermittencies can be considered modifications of Γ_{15} , with the length of the tCW signal increased and decreased by a single chunk respectively. Similarly, the fifth and sixth highest ranked intermittencies show similar intermittencies, with the seventh chunk swapped out for noise instead.

With this in mind, we can again examine figure 6.5. We see that, up to input cSNR of five, there is a tendency for the input intermittency to be the second, or third in the ranked list. Combining this with the tendency for RBB to cluster similar intermittencies together, we can expect that, in many trials which were not correctly recovered, their position being relatively high on the list correlates with having very few mismatched intermittency bits. We also see this in figure 6.1, where, trials with an input cSNR of four have a much larger proportion of trials recovered with only one mismatched bit (gold coloured bar) than those with cSNR of 3, despite the number of correctly recovered intermittencies (white coloured bar) being very similar.

In this sense, RBB is effective at indicating whether or not a given signal exhibits transient behaviour, and coarsely outlining the transient nature.

Having tested RBB against well controlled and understood data, injected on top of Gaussian noise, and found the resulting performance to be sensible, it is important to test RBB against known intermittent signals in more realistic data. To do this, we can use actual data from the aLIGO detectors. In section 6.4, we will detail how the hardware injections at LLO throughout O2 are good candidates to test RBB against, and review the performance of RBB against these injected signals.

6.4 RBB on hardware injections

In each observing run of iLIGO and aLIGO, a number of “hardware injections” are induced in the detector, by physically moving the test masses at each site, in a way that is consistent with an astrophysical signal [226]. Recovery of these hardware injections allows for the the data collection and analysis methods to be tested holistically.

6.4.1 Continuous wave hardware injections

Hardware injections allow us to test the detection and analysis framework for GWs. A hardware injection is designed to be consistent with an astrophysical signal, which is injected into the GW detector’s data by moving the test masses, in a way consistent with a source. For CW injections, we begin with a parameter (`.par`) file as in the simulated datasets discussed in section 6.3. As the aLIGO detectors are in different locations, and oriented differently, we expect to see slightly differing signals in each detector. As such, a template is generated for each detector, based on the source parameters in the `.par` file. The test mass displacements required to generate the signal template are then calculated, and the displacements are then applied to the test masses.

In order to induce a signal in the test mass, we must have very fine control of the forces applied to the test mass. To do this, we use the same hardware used to control the

test masses, and damp their motion. Specifically, we use the electro-static driver [145] and the photon calibrator (PCAL) [227]. The electrostatic driver is formed of a pair of capacitor plates positioned either side of the test mass. Built up electrostatic charge within the test mass causes electrostatic interaction with the capacitor plates, applying a force towards the midpoint of the capacitor plates [228]. Alternatively, the PCAL is an auxiliary laser system, which also manipulates the test masses. Photons from this laser reflect off of the test masses, and in doing so, impart momentum. Modulation of the PCAL intensity is used to generate the calibration lines in the aLIGO data, but can equally be used to inject the templates of the injection signals.

6.4.1.1 Why have hardware injections?

Hardware injection are distributed throughout the GW spectrum, and allow us to test our understanding of the GW detector's control system, its data acquisition system, and the data analysis tools used to recover CW signals.

An accurately injected signal requires precise control of the test masses, and accurate reading of the IFO's output. The tests of the data analysis tools lies in the recovery of the injected signals. We are able to test targeted, directed and all-sky searches (as well as follow-up procedures) on these injections, simply by presenting the searches with given subsets of the injection parameters. For example, we can give a targeted search the full set of parameters for each hardware injection, or give a directed search only the sky position of an injection.

Similarly to the simulated datasets discussed previously in section 6.1.1, having precise knowledge of the injections allows for us to measure the accuracy or sensitivity of our searches against a signal in real detector data. This is considered to be a more stringent test than a software injection in Gaussian noise, as the detector data is not necessarily stationary, and the data can be non-Gaussian. As such, it is common that

analysis pipelines will attempt to recover hardware injections in order to demonstrate sensitivity and accuracy.

6.4.1.2 Hardware injections in O2

The CW hardware injections in O2 are an interesting example to consider for a tCW search such as RBB. The hardware injections were not functioning in LLO between June 5th 2017 16:00 UTC and June 13th 2017 18:00 UTC (more precisely, the GPS times between 1180713618 and 1181412018 did not contain the injected signal at LLO). Despite this, the injected signals were consistently injected throughout the observation at LHO. This allows for an interesting opportunity to test RBB on detector data. Additionally, a signal drop-out for eight sidereal days is consistent with several models of the relaxation period following a glitch.

As such, the hardware injections in LLO during O2 have pre-existing intermittency profile. We can use RBB to try to recover this intermittency in each of the hardware injections. The parameters of each of the hardware injections are given in table 6.6

Additionally, we can consider how loud each injection is in the spectrum. The detector's sensitivity varied both with the frequency (see the sensitivity curve in figure 2.1) and with time, over the course of O2. As such, an expected signal-to-noise ratio (SNR) (and by extension cSNR) value is likely to be incorrect. Instead, we can use calculated values of ρ , defined as $h_0/\sqrt{S_h}$, the injected strain divided into the square root of the PSD of the detector's noise floor at the source's frequency. In table 6.7, we give the empirical values of ρ found by coherently analysing all of O2 in LLO [229]. We also present values of cSNR_1 and cSNR_2 , the cSNR values in each of the two chunking regimes outlined in the next section.

TABLE 6.6: Here, we present the source parameters (frequency evolution and sky position) of 15 hardware injections injected into O2. The parameters presented here are rounded for convenience.

Injection	f_0^{rot} (Hz)	\dot{f} (Hz s $^{-1}$)	α	δ
PULSAR00	132.79	-2.075×10^{-12}	04:46:12.46	-56:13:02.95
PULSAR01	424.54	-1.50×10^{-10}	02:29:34.52	-29:27:08.86
PULSAR02	287.58	-6.85×10^{-14}	14:21:01.48	03:26:38.36
PULSAR03	54.43	-7.3×10^{-18}	11:53:29.42	-33:26:11.77
PULSAR04	701.58	-1.27×10^{-08}	18:39:57.04	-12:27:59.85
PULSAR05	26.40	-2.015×10^{-18}	20:10:30.39	-83:50:20.90
PULSAR06	74.36	-3.365×10^{-09}	23:55:00.23	-65:25:21.45
PULSAR07	610.49	-5.60×10^{-10}	14:53:42.15	-20:27:02.27
PULSAR08	97.15	-4.325×10^{-09}	14:53:42.5	-33:25:06.66
PULSAR09	381.92	-7.25×10^{-18}	13:15:32.54	75:41:22.52
PULSAR10	13.18	-4.25×10^{-11}	14:46:13.35	42:52:38.29
PULSAR11	15.71	-2.535×10^{-13}	19:00:23.36	-58:16:19.54
PULSAR12	19.86	-3.125×10^{-9}	22:07:24.64	-16:58:22.37
PULSAR13	6.22	-5.0×10^{-12}	00:57:17.75	14:19:26.20
PULSAR14	995.55	-5×10^{-13}	20:03:12.68	-14:19:26.20

6.4.1.3 Description of search set-up

O2 began on December 1st 2016, and ran until August 25th 2017, for nearly nine months of data collection. Given that the drop-out of the hardware injections was for a period of only eight days, in order to match the RBB chunk lengths, we would require around 30 chunks. However, this would mean 465 different blocks to analyse, which is impractical. This leaves two alternatives. We can either analyse a subset of O2, for example, the final 96 days, which we can chunk into 12 chunks of eight days each. Alternatively, we can analyse the entire span of O2, but chunk the data so that the length of each chunk is longer than the ideal eight days. This mismatched chunk length will lead to less sensitivity to the underlying intermittency of the data. We show a schematic of both of these regimes in figure 6.7

In order to test the sensitivity of the method we implemented RBB over both chunking regimes. In each case, we only examine the data from LLO, which was intermittent in its injection, whereas the injections at LHO were constant over the observation.

TABLE 6.7: The injected SNR of 15 hardware injections in O2, and their corresponding cSNR values in each of the two chunking regimes tested.

Injection	ρ	cSNR ₁	cSNR ₂
PULSAR00	223.5	74.5	37.8
PULSAR01	382.2	127.4	64.6
PULSAR02	323.9	108.0	54.8
PULSAR03	80.3	26.8	13.6
PULSAR04	189.2	63.1	32.0
PULSAR05	137.3	45.8	23.2
PULSAR06	83.7	27.9	14.2
PULSAR07	362.6	120.9	61.3
PULSAR08	109.3	36.4	18.5
PULSAR09	290.6	96.9	49.2
PULSAR10	229.6	76.5	38.8
PULSAR11	227.4	75.8	38.5
PULSAR12	178.3	59.4	30.2
PULSAR13	117.0	39.0	19.8
PULSAR14	84.3	28.1	14.3

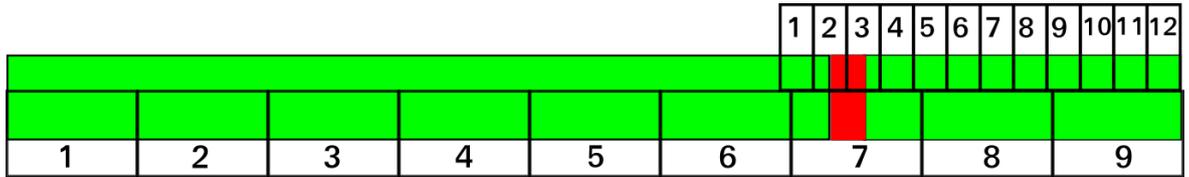


FIGURE 6.7: This time-line shows O2 as the coloured bar. The injection drop-out occurs during the red period. Above the time-line are 12 equally distributed chunks of the second regime. Below are the 9 equally distributed chunks of the second regime. In practise, the chunks vary with the injection due to the data cleaning that occurs prior to analysis.

Additionally, as discussed in chapter 3, the data quality at LHO was inconsistent over the observation, which may become problematic for recovering the intermittency. In this case, I only used science ready data: data which has not been flagged for poor quality. Additionally, the data used had undergone a feed-forward line cleaning process.

Often when recovering hardware injections, the focus is on recovering the signal parameters. In this case, as RBB is intended as a late stage follow-up, we will assume that the template parameters are known. RBB allows for a varying amplitude in each block, and so we retain the search over the amplitude parameters, however in this case,

we are not interested in the recovery of the correct amplitude parameters. Instead, we are only concerned with the recovery of the injected intermittency.

6.4.2 Regime 1: Subset of O2, shorter chunks

Firstly, we will discuss the analysis of the latter part of O2 only. In this configuration, we used data only from the end of O2, between the 26th of May 2017 and the end of O2 on the 25th of August 2017 (GPS times 1179814169 and 1187733585). This start time was chosen as it was the nearest data sample to exactly 96 days prior to the end of O2.

6.4.2.1 Input intermittency

In order to determine whether or not RBB was able to correctly recover the intermittency, we must first work out the true intermittency. To do this, we must find the times about which the data will be chunked, and compare those boundaries to the times during which there were no injections.

As we are splitting this dataset into 12 chunks, we find that the chunk division times are those given in table 6.8.

We see that the drop-out in the hardware injections in this case is spread between chunks 2 and 3. This means that the drop-out is not aligned with the chunk boundaries. This means that the injected intermittency in each case, $\Gamma_{inj} = \Gamma_{2559}$, which can be expressed chunk-wise as $\blacksquare \square \square \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare \blacksquare$.

6.4.2.2 Recovery efficiency

An overview of the recovered results is shown in table 6.9. We can see that out of the 15 injected signals, RBB was able to exactly recover none of the injected intermittencies.

TABLE 6.8: In this table, we show the start and end times for each of the 12 chunks in this configuration of RBB. In LLO, the O2 hardware injections dropped out between 1180713618 and 1181412018, which falls across both chunks 2 and 3.

Chunk number	Start time	End time
1	1179814169	1180474120
2	1180474121	1181134071
3	1181134072	1181794023
4	1181794024	1182453974
5	1182453975	1183113925
6	1183113926	1183773877
7	1183773878	1184433828
8	1184433829	1185093779
9	1185093780	1185753731
10	1185753732	1186413682
11	1186413683	1187073633
12	1187073634	1187733585

However, we list the rank of the injected intermittency in each case. We see that the injected intermittency is never lower than rank position six.

As noted, the drop-out of signal falls between two chunks, and so no chunk is entirely without injected signal. As the injected signals have such high cSNR (as seen in table 6.6, it's very likely that in each of chunks containing the drop-out, the loud signal contributed enough to the evidence that the block more likely resembles signal than noise.

In these injections, we are using real detector data, which has drop-outs in its data, meaning that not each chunk of data has the same number of samples, despite spanning the same time. This happens when the GW detector is disrupted, for example by an earthquake, and the IFO is no longer able to observe without intervention to bring it back online. This results in having less integration time for the injected signal, and thus lower signal evidence. Similarly, fewer data samples results in a lower noise evidence.

Further, as this is real detector data, we can no longer always trust that the data are perfectly Gaussian distributed at all times and all frequencies. As we discussed

in this regime, as the chunks are longer, we can expect that RBB is more sensitive to the injected signals.

As in the first regime, before we analyse the data with RBB, we should find the injected intermittency.

6.4.3.1 Input intermittency

We can find the input intermittency, similarly, by finding the times around which the observation is chunked, and seeing which bin the drop-out falls into. In this case, as the bins are wider, we do not expect that the drop-out falls across two bins as it did in the first regime.

Further, in this case, as we are analysing all of O2, there is an unexpected consequence. Only data not flagged for bad data quality is analysed, and these flags are applied in the time-frequency domain. As such, the first sample used, and thus the start time of the analysis, is different in different injections, as their source frequencies differ. This leads to the drop-out falling into different chunks for different injections. In table 6.10, we list the input intermittencies found for each injection.

6.4.3.2 Recovery efficiency

We present the overview of the recovered intermittencies in table 6.10. Of the 15 injections analysed, only two input intermittencies were correctly recovered. Those were PULSAR05 AND PULSAR10. The remainder of the injections were recovered as $\Gamma_{511} = \blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare\blacksquare$. We might expect this: the injection drops out within a single chunk, and for only eight days of a ≈ 30 day chunk. The high-cSNR injection in this regime likely outweighs the drop-out contribution. In most cases where the intermittency was not correctly recovered, it was found in the next position in the ranked list.

TABLE 6.10: The input intermittency and recovered intermittency for in the datasets for each of the fifteen different hardware injection for all of O2.

Injection	Injection intermittency	Recovered intermittency	Rank
00	■ ■ ■ ■ ■ ■ ■ □ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
01	■ ■ ■ ■ ■ ■ ■ □ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
02	■ ■ ■ ■ ■ ■ ■ □ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
03	■ ■ ■ ■ ■ ■ □ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
04	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
05	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	0
06	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
07	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	2
08	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	2
09	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	2
10	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	0
11	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	2
12	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1
13	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	4
14	■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	1

In the cases where Γ_{inj} is not the second ranked in terms of posterior, we see that this is generally associated with the duty factor of the detector. The duty factor during a given chunk determines how many data samples in a chunk, and therefore the size of the evidences. These lower evidences tend to lead to smaller Bayes factors for the chunk. This Bayes factor is roughly the difference between Chunks with fewer samples are more likely to be changed from ■ to □ (or vice-versa), as the difference between two similar intermittencies (such as ■ ■ ■ ■ ■ □ ■ ■ ■ ■ ■ and ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■). Thus, these chunks with fewer samples are often more likely to change bit-value than those containing more samples.

6.4.3.3 Is RBB behaving as it did with software injections?

In the performance of RBB on these hardware injections, we see some differences in behaviour when compared to the simulated datasets discussed in section 6.2 and 6.3.

Firstly, we see the recovery rate and the types of mistakes made by RBB are similar

to those made in very high cSNR injections. We see that RBB often prefers to find a signal in each chunk. In each case, there *is* signal in each chunk, even if the signal does not last for the entire chunk.

We see that in the first chunking regime that RBB more often picks up on the signal drop-out than in the second chunking regime. This is somewhat in line with what we might expect. As the chunks in regime 1 are shorter (≈ 8 days in regime 1, and ≈ 29 days in regime 2), we expect a lower cSNR in the first regime, as we see in table 6.7.

Additionally, in the first chunking regime, each of the two chunks containing the drop-out, the drop-out lasts roughly half of the chunk. This effectively makes the cSNR in those chunks $\text{cSNR}_1(\sqrt{1 - 1/2})$. Conversely, in the second regime, the drop out lasts for 8 or the ≈ 29 days, meaning that the effective cSNR in the chunk containing the drop-out in the second regime is roughly $\text{cSNR}_2(\sqrt{1 - 8/29})$.

Additionally, we see in most of the injections in both of the chunking regimes, that the injected intermittency places very high on the ranked list of intermittencies. We see, looking down the barcode plots, that the chunks containing the drop-outs are usually the first to be switched to \square .

Even though RBB did not correctly recover the intermittency correctly for all of the hardware injections in O2, it did not show unexpected results, given the very loud injected SNR, and the chunks used.

6.5 RBB on O1 all-sky signal candidates

Having validated RBB on simulated data, and re-affirmed its behaviour by examining transient hardware injections in O2 at LLO, we can now look to examining candidate CW signals from prior observations. As yet, there has been no unambiguous detection of CWs in either the initial or advanced detector era, however all-sky searches for CW

signals in O1 returned four candidates which could not be dismissed. In this section, we apply the RBB algorithm to these outliers from the O1 all-sky Einstein@Home search.

6.5.1 O1 all-sky search and outlying results

The first observation of aLIGO was the first in the advanced detector era, and brought about the best prospects yet for detecting GWs. Detections of several short transient events [79–84] cemented the dawning of GW astronomy. Following the end of the observation, searches for CWs in the O1 dataset began. In the subsections within (6.5.1.1, we outline the O1 all sky search procedure, and the stages of follow-up applied in order to determine whether they are viable astrophysical sources, or the result of Gaussian noise.

6.5.1.1 Describing the O1 low frequency All-sky search

The first all-sky search to be completed for O1 focussed on only the low frequencies, between $f = 20$ Hz and 100 Hz, and with spin-down rates between $\dot{f} = -2.65 \times 10^{-9}$ and -2.65×10^{-10} . Due to the large population of lines and combs in O1, the data in contaminated frequency bins were replaced with Gaussian noise, with average power in these bins scaled to be consistent with the power in nearby, uncontaminated bins. Further, as the detectors often fell out of lock, the duty factor for O1 was roughly 0.5 in each detector.

This search was a semi-coherent analysis, which was comprised of $N_{\text{seg}} = 12$ segments of $T_{\text{coh}} = 210$ hours. The frequency and spin down resolutions in the search were 8.3×10^{-7} Hz and 1.3×10^{-11} Hz/s respectively. Within each segment, the search analyses the data coherently for all possible templates, calculating the detection statistic $2\mathcal{F}$ (as given in [47]) and $\hat{\beta}_{\text{S/GLtL}}$. The values of $2\mathcal{F}$ are averaged, to generate the $\bar{\mathcal{F}}$ -statistic value for the semi-coherent search. The statistic $\hat{\beta}_{\text{S/GLtL}}$ is the line and transient robust

statistic, and is the log-10 value of $\hat{B}_{S/GLtL}$, the Bayes factor for signal (for a given set of templates) against Gaussian noise (G), lines (L), and transient lines (tL), as defined in [222]. This statistic is used to create a ranked list of the templates of most likely candidates. The $\bar{\mathcal{F}}$ -statistic then gives the significance of each candidate.

The total number of templates searched was 3×10^{17} , which is only feasible by the distributed computing power available to Einstein@Home. Each distributed node in the Einstein@Home framework analysed 1.5×10^{11} templates. Out of these only the top 10 000 were returned to create a “toplist” of most significant templates. The final length of the toplist prior to the follow-up stages was 1 900 000.

6.5.1.2 Following up significant candidates

As described in section 1.3.3, in order to reduce the number of candidates, several stages of follow-up investigations are used to discriminate between astrophysical signals and detector artefacts. In the case of the O1 low frequency all-sky search, three rounds of follow-up search were applied in order to reduce the number of candidate signals, and perform more stringent (and thus more expensive) searches on the remaining candidates at each stage.

Following every stage of follow-up, four candidate signals could not be ruled out. There were three stages of follow-up search, the most stringent round of follow-ups consisted of a fully coherent analysis ($N_{\text{seg}} = 1$) with frequency and spin-down resolution of $\delta f = 6.7 \times 10^{-8}$ and $\delta \dot{f} = 9.3 \times 10^{-14}$ respectively. 6 349 candidates passed all three stages of this following. Additionally, these candidates were subject to a “Doppler modulation off” (DM-off) veto [230], in which the candidate template’s Doppler modulation due to the motion of the GW detectors around the Earth and solar system barycentre was removed. This would *reduce* the significance of an astrophysical source, but may *increase* the significance of a local source. Several further searches were run for the

candidates with the DM-off regime. Only four candidates survived this round of follow-up, and could not be ruled out as candidates in O1.

As an additional measure, the templates for these four outstanding candidates were extended into O2, and a follow-up search was conducted using three months of O2 data. None of the four candidates remained significant in O2, and so all four were dismissed as candidate CW sources. However, these four candidates show promise as sources of tCWs.

6.5.1.3 Four all-sky outliers

The O1 all-sky search returned four candidates that were not able to be dismissed through several rounds of follow-up. These four outliers were named PSR J0709-2945, PSR J1051-2654, PSR J1901+3333 and PSR J2228-4359, which we will call J0709, J1051, J1901 and J2228 respectively. The source parameters, as far as they have been constrained, are given in table 6.11.

TABLE 6.11: In this table, we show the source parameters for the four candidates in the all-sky searched which could not be ruled out during O1.

Candidate	α	δ	f (Hz)	\dot{f} ($\times 10^{-10}$ Hz/s)
J0709	1.87245	-0.51971	58.970435900	-10.81102
J1051	2.842702	-0.469603	99.220728369	-24.98113
J1901	4.98020	0.58542	62.081409292	-23.26246
J2228	5.88374	-0.76773	97.197674733	+2.28614

The values stated in table 6.11 necessarily have uncertainties on them, which can be found by examining the most stringent round of follow-up. Each of the four candidate sources were recovered within

$$\Delta f = f_{\text{cand}} \pm 7.5 \times 10^{-6} \text{ Hz}, \quad (6.15)$$

$$\Delta \dot{f} = \dot{f}_{\text{cand}} \pm 7 \times 10^{-12} \text{ Hz}. \quad (6.16)$$

We can sensibly use these ranges as uncertainties on the frequency and spin-down of the candidate sources. The resolution and uncertainties on the sky localisation of each candidate are addressed in [211], as the sky tiling was approximately uniform over the sky, but the number of templates scale with frequency.

6.5.2 RBB configuration

In order to search for tCWs from the four all-sky candidates, we decided to run RBB on each of the four candidates using O1 data. As RBB is based on a targeted method, we selected the template which was recovered with the most significance in the final round of follow-up. This corresponds to the values reported in table 6.11.

O1 lasted for approximately 129 sidereal days, which we can chunk into eight chunks of approximately 16 days. We chose this as several of the tCW sources discussed in section 4.2 had a characteristic duration of around 10 days. We saw in section 6.4.3 that in the second chunking regime, RBB was still sensitive to a transient signal even when the drop-out time was small relative to the chunk duration. This motivates a chunk time longer than a characteristic emission time for tCWs.

In setting up the nested sampling routine for each chunk, we chose to use a value of N_{live} , the number of live points in the nested sampler, of 512. As discussed earlier, in [52] details the effect of different values of the number of live points, with a lower value of N_{live} resulting in a slightly higher variation in the estimated values of the signal evidence. However, this lower value of N_{live} is also associated with a significant speed-up in the run time of the nested sampler. As we require the nested sampler to run $N_b(N_b + 1)/2$ times for each candidate, we have favoured the faster choice of fewer live points. If the ranked list of intermittencies shows random behaviour which can not be justified, this may be due to the uncertainty introduced here, in which case, the process can be repeated with a higher value of N_{live} .

Data from both detectors, heterodyned to the source frequency, were used in this follow-up search.

6.5.3 Recovered intermittencies

Below, in figures 6.8,6.9,6.10 and 6.11, we present four barcode plots, one for each of the signal candidates.

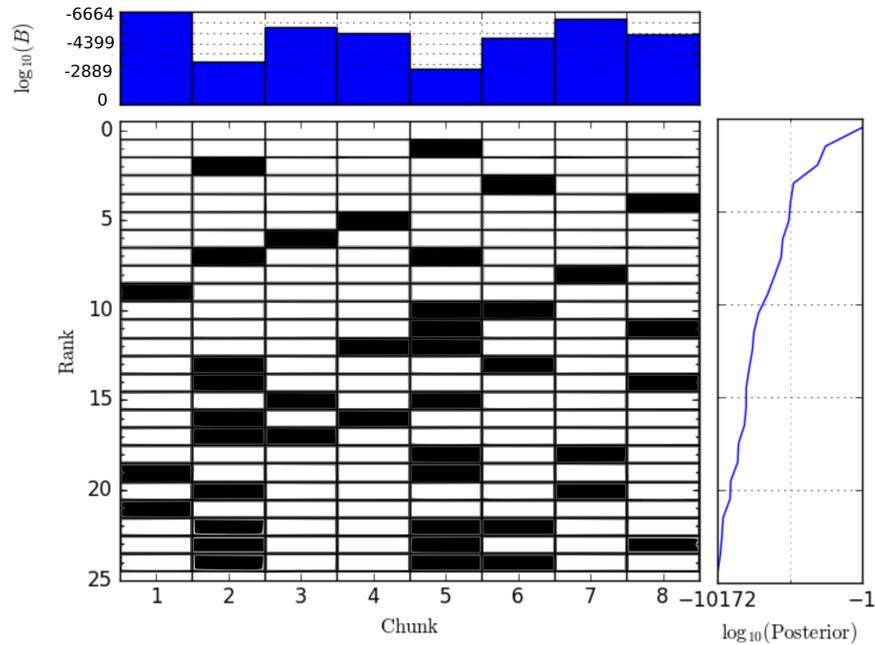


FIGURE 6.8: The barcode plot for the tCW candidate J0709.

We can see that in each of the four barcode plots, the recovered intermittency is Γ_0 , which represents only noise. Further, we see in the right-hand panel of each barcode plot, that the posterior probability sharply drops by several orders of magnitude with the inclusion of a single chunks of \blacksquare . This would seem to rule out each of these four candidates with these parameters.

One interesting feature common among all four candidates is in the second and third ranked intermittencies. In all four cases, these are Γ_8 and Γ_{64} . These correspond to the fifth and second chunks respectively being \blacksquare , containing signal. Naively, we may

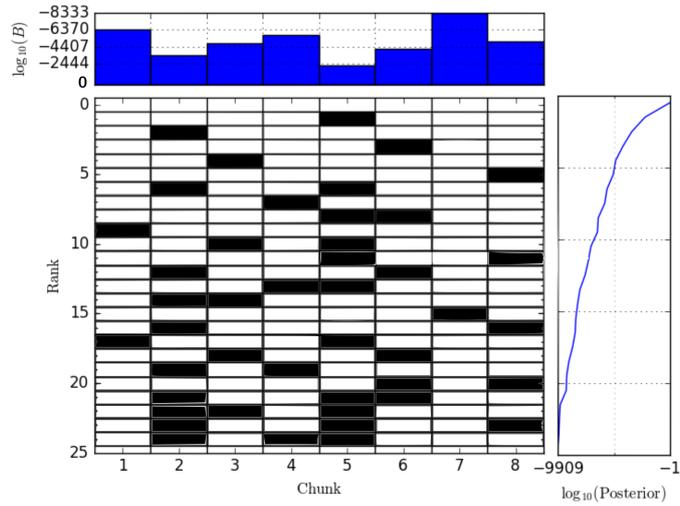


FIGURE 6.9: The barcode plot for the tCW candidate J1051.

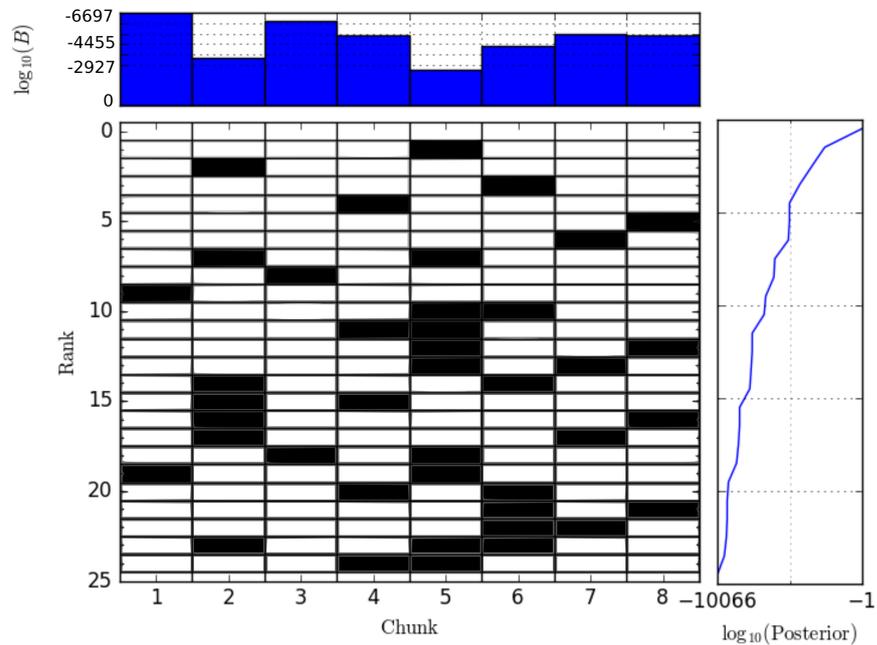


FIGURE 6.10: The barcode plot for the tCW candidate J1901.

consider that this could be attributed to having fewer data samples in the second and fifth chunks due to the duty factor of the detector. This results in lower evidence values

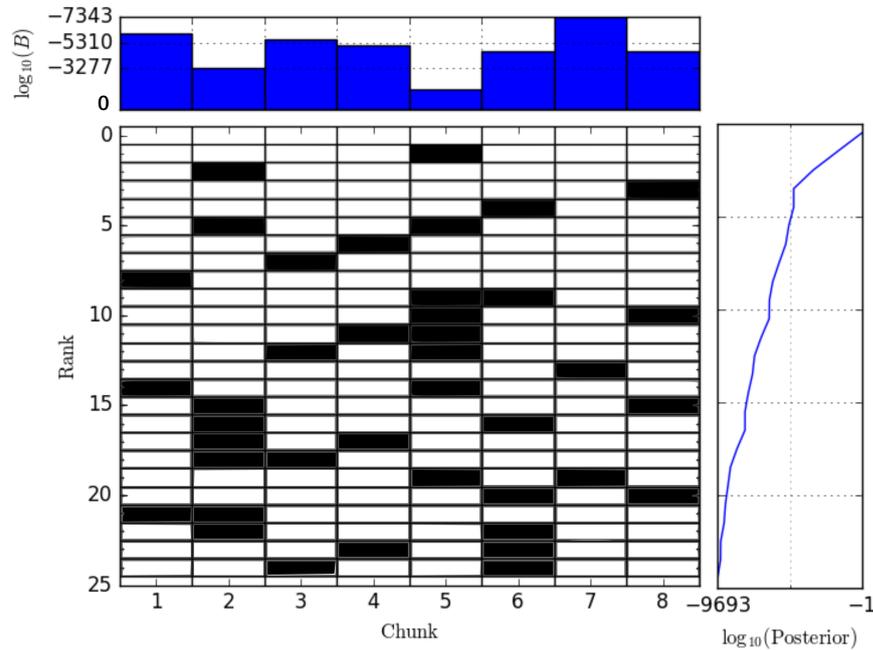


FIGURE 6.11: The barcode plot for the tCW candidate J2228.

for both noise and signal, and thus the least costly bit to change value. However, we do not see this. In fact, the fifth and second chunks are the first and third most well sampled chunks in LLO and the second and third most well sampled chunks in LHO. We see that this also corresponds to relatively high values of the evidences for both signal and noise, as we would expect, as can be seen in tables 6.12 and 6.13.

TABLE 6.12: This table shows the signal-to-noise ln-Bayes factor values for each individual chunk for both J0709 and J1051.

Chunk	J0709	J1051
1	-6.66×10^3	-3.75×10^3
2	-3.12×10^3	-3.37×10^3
3	-5.58×10^3	-4.8×10^3
4	-5.2×10^3	-5.77×10^3
5	-2.58×10^3	-2.20×10^3
6	-4.83×10^3	-4.15×10^3
7	-6.17×10^3	-8.33×10^3
8	-5.03×10^3	-5.02×10^3

We see that, in fact, this is due to the signal-versus-noise Bayes factor (top panel of the barcode plot) is at its least negative in these two chunks. Once the possibility of only

TABLE 6.13: This table shows the signal-to-noise ln-Bayes factor values for each individual chunk for both J1901 and J2228.

Chunk	J1901	J2228
1	-6.7×10^3	-5.98×10^3
2	-3.47×10^3	-3.27×10^3
3	-6.13×10^3	-5.56×10^3
4	-5.05×10^3	-5.12×10^3
5	-2.59×10^3	-1.64×10^3
6	-4.33×10^3	-4.57×10^3
7	-5.12×10^3	-7.34×10^3
8	-5.05×10^3	-4.58×10^3

noise has been checked, we necessarily must consider at least one chunk with signal. The second highest ranking intermittency is one whose evidences differ the least from the first, thus the preference for the least negative log-Bayes factor.

The underlying reason behind this minimised log-Bayes factor in the second and fifth chunks may be one of any number, ranging from the hints of a transient signal in the data, to local noise sources being reduced in the source band at either of two detectors. However, though the log-Bayes factors are the least negative in these two chunks, the nested sampler still recovers a noise evidence a few thousand orders of magnitude higher than the evidence for signal. This seems to rule out the idea of hints of signal beneath the noise.

Instead, it seems far more likely that the commonalities between the second and third most likely intermittencies are due to variations in the performance of the detectors over the course of O1. As all four candidates are in the low frequency regions between 20 Hz and 100 Hz, a single broadband transient noise in the detector may affect the noise levels around each of the candidates.

Additionally, the drop-off in the posterior probabilities for any intermittency besides Γ_0 in each of the four candidates is large enough that the posterior probability of the second ranked intermittency is negligible.

Thus, we can conclude that RBB was able to decisively recover the intermittency Γ_0 in each of the four candidate signals from the O1 all-sky low frequency search for GWs using the most significant template recovered in the search.

Chapter 7

Conclusions and future work

In this work, we have explored methods and examined results over two key aspects of GW detection for tCWs. Firstly, we discussed detector characterisation, data quality and noise mitigation. Secondly, we explored the astrophysical sources of tCWs, and the local effects which can cause a CW signal to resemble a tCW. We also discussed the emerging prospects for detecting tCWs, and developed a new technique for determining whether a CW candidate may have transient behaviour.

7.1 Conclusions

In the first half of the work, we examined aspects of data quality and detector characterisation throughout S5 and S6 in iLIGO, and O1 and O2 in aLIGO.

7.1.1 Data quality and detector characteristics

First of all, we explored work done previously into characterising a broadening of the 60 Hz line at LLO. The original investigation, performed by site staff at LLO and Robert Schofield involved determining a problem (the broadening of the 60 Hz line at LLO), finding a likely witness channel (by taking the coherence between the magnetometer channels at LLO and the GW data), finding a coupling source (by investigating with a

hand-held magnetometer), characterising the magnetometer glitches (determining the diurnal nature of the glitch rate) and tracking the noise back to its source (the local substation and subterranean pipes). This was all to characterise one noise source.

Then, in this work, we attempted to mitigate the effect of the noise, by designing and implementing a non-linear filter. However, when this filter was not able to remove the noise from around the 60 Hz line, we decided to look deeper into the coupling. We examined the correlation between strain samples and magnetometer samples, we examined the ratio of power (in the GW data) coincident with the magnetometer glitches, and at times where there were no glitches. We found that the power ratio at glitch times was no different than a background rate. By examining this, we were able to determine that such a non-linear filter to remove the impact of the magnetometer glitches would be unable to reduce the broadening of the 60 Hz line in iLIGO data. We then showed that the coupling mechanism leading to the broadening of the 60 Hz line at LLO had been mitigated between the initial and advanced detector eras.

Though we were not able to salvage the S5 and S6 data, through the efforts of several on-site staff, the issue was identified and eventually mitigated for future observations.

In the second case study presented in chapter 3, we detailed how we (myself, Pep Covas, Ansel Neunzert, and coordinated by Evan Goetz) analysed the spectrum of a “clean” segment of O2 for lines and combs. By meticulously examining the the spectra of the two detectors, we were able to compile lists of lines and combs, each with its own witness channel. This work, though time consuming, does not necessarily lead directly to the mitigation of any line or comb. Nevertheless, without this work, beginning an investigation of a line or comb can be far more difficult. Such a line list serves two key functions:

- Finding loud lines and associated witness channels allows a focus and starting point for any investigation in order to increase data quality for persistent GW searches.

- Listing the lines which can be attributed only to local noise allows searches to veto frequency bands which are dominated by non-Gaussian noise of terrestrial origin.

Finally, we detailed one such investigation of a comb identified in the O1 line list by Keith Riles. This investigation was conducted by myself, Vincent Roma and Ansel Neunzert at LHO. To begin, we examined the correlation between the strain channel and each of the PEM magnetometers around the site. We found that the correlation was highest in the electronics bays distributed around the site. Following this, we examined the Y-arm EBAY and corner station EBAY with a portable magnetometer. We found that the correlations were highest around the timing equipment at each location. Further, in data folding studies, Keith Riles found a strong periodic behaviour in the portable magnetometer channel, corresponding to hard edges exactly at the second boundaries. This led us to investigate the timing infrastructure as a coupling method. We tested coupling by implementing temporary power supplies in the I/O chassis through which the strain data are handled. This did not mitigate the comb, which ruled out the coupling mechanism. In a mitigation attempt led by Ansel Neunzert, the firmware governing the flashing LEDs in each of the timing cards were altered. This was found to reduce the size of the comb substantially.

From start to finish, the investigation spanned around six months, and involved the work of several researchers, site staff scientists and engineers, and advice from the larger continuous wave detection group in the LIGO and Virgo collaborations. However, the changes made at both LLO and LHO greatly reduced the influence of a comb prevalent in the low-frequency bands (up to about 120 Hz) in O1.

Additionally, in the investigation of this comb, several analysis tools were created or updated for the advanced detector era. This will greatly aid future line and comb investigations, and contributed to the creation of the O2 lines list.

In each of these three cases studies, we see that several people have been involved

in time consuming studies which do not necessarily directly increase the chances of detecting GWs from persistent sources. However, the role played by these detector characterisation studies is to provide a greater knowledge of the data produced by the GW detectors, and applying that knowledge in reducing the influence of environmental and instrumental noise in narrow bands (or sets of narrow bands) of data. It also allows for GW searches to more confidently and deeply explore the GW data, avoiding the lines, combs and wandering lines in the data.

7.1.2 Transient continuous waves and reduced Bayesian blocks

In the second half of this work, we shift focus from the detectors, to a subset of sources – tCWs. We began by defining tCWs as GWs with morphology similar to a CW, that is, quasi-monochromatic GW radiation, that can be shown to begin and end at some times. We then explore several tCW sources, such as pulsar glitches, transiently accreting pulsars, and even wandering lines (though these are not truly tCWs, but may appear transient to template driven CW searches).

We then discussed several search methods for tCWs, including the long-duration STAMP search and the transient \mathcal{F} -statistic method. We outlined the assumptions underlying these methods, and the characteristic times for which these are most sensitive. Finally, we turned our attention to an optimally flexible search method, such as BB, but conclude that a full BB implementation for a tCW search is prohibitively costly.

In chapter 5, we outline a method to identify whether or not a transient signal matching a given template is in the data, and to loosely constrain the beginning and ending times of any transient signal. We showed how this method was consistent with a BB configuration with constraints on the allowed positions and numbers of changepoints.

Then, in chapter 6, we examined the performance of RBB. First, we examined in detail how RBB obtains and combines evidences for signal and noise in a small simulated

dataset. Then, in a larger simulated dataset, we examined the distributions of a number of recovery statistics, such as the posterior probability of the recovered intermittency, the distribution of mismatched intermittency bits by cSNR of the injected signal, and the spread in the position of the injected intermittency in the list of all intermittencies, ranked by posterior probability. We found that RBB behaved as we expected it to on the simulated data, preferring Gaussian noise in the weakest signals, and recovering the majority of the stronger signals.

In section 6.4, we turned RBB onto data from the aLIGO detectors, attempting to recover 15 hardware injections. The injections were transient in LLO, dropping out for approximately eight days late in O2. We ran RBB with two different chunking regimes. In the first regime, we matched the chunk length to the characteristic time of the transient signal, with chunks eight days long. We used a subset of O2 data, taking only the last 96 days of observations. In this regime, RBB estimated that the signal was transient in 9 of the 15 injections, however, due in part to the loud injected signals, RBB was unable to recover the exact input intermittency.

In a second chunking regime, we analysed the entirety of O2, using nine chunks of roughly 29 sidereal days each. In this case, the length of the chunk was not matched up to the characteristic time scale of the transient injection, as would likely be the case in a real transient signal. In this regime, only two injections were recovered with the correct intermittency. RBB was more often sensitive to the strong injection throughout most of the chunk containing the drop-out. In this regime, we found that in the first or second highest ranked intermittencies, RBB recovered the injected intermittency. This is likely because the drop-out lasted less than one third the duration of each chunk.

Finally, in section 6.5, we turned RBB onto four outlying candidates from O1. These candidates could not be ruled out in O1 alone, and were good candidate signals for tCWs. We found in each of the four candidates that the preferred intermittency was Γ_0 , Gaussian noise throughout the observation.

7.2 Future work

Detector characterisation and the monitoring of the detectors' data quality is an ongoing pursuit. With each upgrade to the detectors, and each new dataset, we must seek to understand and characterise the data. It can be hard to predict exactly which issues are to be the highest priority to tackle, or new noise artefacts that may arise. Detector characterisation requires constant monitoring of the state of the detectors and data. This will remain a field for constant exploration so long as there is new data produced. We discussed briefly how the data quality of the detectors in O1 may have affected the recovered intermittencies in section 6.5.

However, an analysis method should also be relatively robust to the varying noise levels of the detector. One of several future changes that could be made to RBB is an adaptive chunking procedure. We could tune the length of each chunk to have roughly the same sensitivity to a tCW. This may involve tuning the duration of each chunk to achieve equal integration time, or to minimise the difference in the noise evidences between chunks. This is an avenue of investigation for which simulated data is not sufficient, but signals injected onto detector data (favouring software injections, for the ability to control the injection amplitude and intermittency easily) are preferred.

Further, we can envisage one further round of tCW follow-up to more tightly constrain the changepoints in the data. Using RBB as a guide, we could set up a BB implementation with N_τ , the number of changepoints, dictated by the number found by RBB, and their location constrained to two chunks, the chunks preceding and following each changepoint recovered by RBB. This follow-up stage may be able to more tightly constrain the times of the true changepoints in the data, whilst being far more computationally efficient than a full BB implementation of the data. This may take the form of a finer chunking regime within the two chunks surrounding a recovered changepoint, or a full BB treatment, with Markov-chain Monte-Carlo methods used to

estimate the position of the changepoints.

During O2, both the Crab and Vela pulsars were observed to glitch [231]. This makes them both good candidates for tCW sources. As high priority targets, their source parameters (besides the four amplitude parameters) are very well constrained, providing optimal targets for RBB. Further, we know the times of the glitches in both sources, and observations of NS relaxation periods, combined with predictions from pulsar glitch models give reasonable constraints on the characteristic time scales for these two targets.

Finally, the evidence estimator used in RBB is a targeted method, and not suited to exploring a range of frequency and spin-down parameters, or sky positions, due to computational expense. However, a spectral interpolation method, SplInter [232], is able to greatly increase the speed of such a search over these parameters. By incorporating SplInter into the RBB method, we can perform a more comprehensive follow-up on candidates such as the four candidates examined in section 6.5.

Appendix A

Bayesian and RBB glossary

Throughout chapters 5 and 6, I defined and used several terms to describe the RBB method. Here, I have included a glossary of terms used for easy reference.

A.1 Bayesian terms: Glossary

Prior probability $P(X)$ A priori belief in the statement X .

Prior volume The N-dimensional parameter space defined by the individual prior probability distributions.

Likelihood $P(D|X)$ The probability of generating the data, given the parameters.

Signal likelihood The likelihood evaluated with a given set (or range) signal parameters.

Noise likelihood The likelihood of achieving the data given the absence of any GW signal.

Evidence for X $P(X) \times P(D|X)$ - the prior probability of X multiplied by the likelihood of X .

Posterior probability, $P(X_i) \frac{P(X_i) \times P(D|X_i)}{\sum_i P(X_i) \times P(D|X_i)}$. The evidence for X divided into the evidence for all possible variations of X .

Bayes factor ($B_{i,j} = P(X_i)/P(X_j)$) The ratio of two posteriors. Useful as the denominator of $P(X_i)$ cancels. Often calculated as $\frac{P(X_i) \times P(D|X_i)}{P(X_j) \times P(D|X_j)}$.

Signal-to-noise Bayes factor The evidence for signal divided into the evidence for noise. A handy metric for determining the relative probability of a signal in the data.

Overfitting Achieving the best description of data by introducing too many new variables, making the model either very computationally expensive, or unable to generalise to a more general problem.

Occam factor The tendency to prefer models with fewer variables. This effect counteracts the tendency to overfit.

A.2 RBB terms: Glossary

Intermittency (Γ_p) A bitwise description of a transient data hypothesis, composed of N_b bits.

Number of bits (N_b) The length of the intermittency.

Intermittency bits (γ_j) Data hypothesis at each point in the intermittency.

$\gamma_j = \blacksquare$ The data hypothesis in the j^{th} position is signal.

$\gamma_j = \square$ The data hypothesis in the j^{th} position is noise.

D The full dataset.

Chunk The full dataset is divided into N_b equal-sized datasets, called chunks.

d_j The data in the j^{th} chunk. $\{d_j\} = D$

Block (α_i) A set of contiguous bits with the same data hypothesis.

Block of data (β_i) Concatenated adjacent blocks of data, corresponding to a block α_i .

Changepoint The divider between two adjacent blocks.

Block data hypothesis ($b(\alpha_i)$) The data hypothesis corresponding to the value of γ_j throughout a block (■ or □).

Bayesian heterodyne method Nested sampler evidence evaluation outlined in section 1.3.2.1.

Appendix B

Lines list

In this appendix, I list the lines I found in spectra of data from LLO during O2, above 100 Hz. The list is unvetted, meaning that includes both lines coincident with noise in witness channels, and unidentified lines. A vetted list is used in searches to rule out candidates and notch out frequency bins.

For combs (type 1), the Frequency (spacing) column gives the spacing between adjacent teeth, and the frequency offset gives the distance from the first tooth (extrapolated backwards towards 0 Hz) and 0 Hz.

Frequency (spacing) [Hz]	Type (0:line; 1:comb)	Frequency offset [Hz]	First visible harmonic	Last visible harmonic	Left width [Hz]	Right width [Hz]
99.999256	1	0	4	20	0.0001	0.0001
306.0596	0	0	1	1	0.0125	0.0109
306.2186	0	0	1	1	0.0725	0.0751
307.3476	0	0	1	1	0.0089	0.0122
307.5051	0	0	1	1	0.0457	0.0400
315.0471	0	0	1	1	0.2427	0.2529
331.3000	0	0	1	1	0.2500	0.3200
499.6042	0	0	1	1	0.2042	0.2958

Frequency (spacing) [Hz]	Type (0:line; 1:comb)	Frequency offset [Hz]	First visible harmonic	Last visible harmonic	Left width [Hz]	Right width [Hz]
503.1825	0	0	1	1	0.2325	0.1175
513.332	0	0	1	1	5.332	3.768
612.4740	0	0	1	1	0.1223	0.1153
615.0496	0	0	1	1	0.0469	0.0431
630.1251	0	0	1	1	0.3508	0.3263
918.8029	0	0	1	1	0.0850	0.0650
922.6639	0	0	1	1	0.0242	0.0232
945.2852	0	0	1	1	0.4866	0.4662
998.6922	0	0	1	1	0.0172	0.0128
1083.1000	0	0	1	1	0.0001	0.0001
1225.2557	0	0	1	1	0.0295	0.0258
1230.3970	0	0	1	1	0.0110	0.01
1457.5980	0	0	1	1	0.208	0.3224
1470.7466	0	0	1	1	0.1850	0.1560
1496.1183	0	0	1	1	3.8884	3.0117
1504.4270	0	0	1	1	1.2683	1.4446
1510.6650	0	0	1	1	0.16	0.4194
1922.3545	0	0	1	1	0.0813	0.0716
1941.0383	0	0	1	1	0.0576	0.0485
1953.8899	0	0	1	1	0.02	0.01
1958.1737	0	0	1	1	0.0137	0.0149
1962.2153	0	0	1	1	0.0407	0.0367
1966.1029	0	0	1	1	0.0579	0.0565
1966.6835	0	0	1	1	0.0520	0.0782

Frequency (spacing) [Hz]	Type (0:line; 1:comb)	Frequency offset [Hz]	First visible harmonic	Last visible harmonic	Left width [Hz]	Right width [Hz]
1967.0721	0	0	1	1	0.0538	0.0591
1967.3337	0	0	1	1	0.0058	0.0091
1967.5790	0	0	1	1	0.0045	0.0055
1973.2363	0	0	1	1	0.0428	0.0396
1977.9272	0	0	1	1	0.0329	0.0555
1982.5292	0	0	1	1	0.0856	0.0879
1985.4741	0	0	1	1	0.0377	0.0523
1986.2230	0	0	1	1	0.0801	0.0731
1988.1592	0	0	1	1	0.032	0.0325
1988.8929	0	0	1	1	0.0009	0.0037
1988.3729	0	0	1	1	0.0234	0.0280
1989.8495	0	0	1	1	0.0287	0.0279
1	1	0.1	1238	1416	0.00014	0.00014
1	1	0	101	140	0.00014	0.00014
0.6	1	0.569	742	745	0.00014	0.00014
0.6	1	0.069	769	773	0.00014	0.00014
0.6	1	0.169	795	801	0.00014	0.00014
0.6	1	0.3042	804	806	0.00014	0.00014
0.6	1	0.269	822	828	0.00014	0.00014
0.6	1	0.469	876	882	0.00014	0.00014
0.6	1	0.569	903	908	0.00014	0.00014
0.6	1	0.069	932	935	0.00014	0.00014
0.6	1	0.169	960	962	0.00014	0.00014
0.6	1	0.3382	1596	1597	0.00014	0.00014

Frequency (spacing) [Hz]	Type (0:line; 1:comb)	Frequency offset [Hz]	First visible harmonic	Last visible harmonic	Left width [Hz]	Right width [Hz]
0.6	1	0.4381	1623	1627	0.00014	0.00014
0.6	1	0.5381	1648	1651	0.00014	0.00014
0.6	1	0.0381	1675	1681	0.00014	0.00014
0.6	1	0.1381	1702	1708	0.00014	0.00014
0.6	1	0.2381	1729	1735	0.00014	0.00014
0.6	1	0.3381	1756	1762	0.00014	0.00014
0.6	1	0.438	1784	1788	0.00014	0.00014
0.6	1	0.53	1812	1814	0.00014	0.00014
0.6	1	0.0979	2400	2403	0.00014	0.00014
0.6	1	0.2073	2475	2479	0.00014	0.00014
0.6	1	0.3071	2502	2504	0.00014	0.00014
0.6	1	0.4072	2527	2533	0.00014	0.00014
0.6	1	0.5071	2555	2560	0.00014	0.00014
0.6	1	0.0073	2582	2588	0.00014	0.00014
0.6	1	0.1073	2610	2613	0.00014	0.00014
0.6	1	0.2071	2638	2640	0.00014	0.00014
0.996789	1	0	23	695	0.00028	0.00028
0.996794	1	0	21	685	0.00014	0.00014

TABLE B.2: This table contains the portion of the O2 lines list that I contributed: Lines and combs in LLO above 100 Hz.

Appendix C

Representative results for eight-chunk injections

In chapter 6, I presented an overview of results from simulated data with eight-bit intermittencies. These injections had integer cSNR values between 1 and 10, with 250 simulations for each value of cSNR. Rather than present the full 2500 results, here I present a representative sample: ten simulations for each value of cSNR.

The data are presented as an overview: the input and recovered intermittencies, whether RBB correctly recovered the injection, the rank of the injected intermittency, the posterior of the recovered intermittency, and the number of mismatched bits.

TABLE C.1: Representative results from the eight-chunk simulations. Here I present 10 examples from each value of injected cSNR.

cSNR	Γ_{inj}	Γ_{rec}	Correct	List position	$P(\Gamma_{\text{rec}})$	n_{mis}
1	□■□■□■□■	□□□□□□□□	No	247	0.959	4
1	□□□■□■□■	□□□□□□□□	No	67	0.984	4
1	□□□■□□■□	□□□□□□□□	No	42	0.993	2
1	□■□□□■□■	□□□□□□□□	No	39	0.985	4
1	■□□□□□□□	□□□□□□□□	No	5	0.985	1
1	□□□□□□□□	□□□□□□□□	Yes	0	0.984	0

TABLE C.1: Representative results from the eight-chunk simulations.
Here I present 10 examples from each value of injected cSNR.

cSNR	Γ_{inj}	Γ_{rec}	Correct	List position	$P(\Gamma_{\text{rec}})$	n_{mis}
1	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	247	0.989	4
1	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	183	0.987	4
1	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	Yes	0	0.990	0
1	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	5	0.930	6
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	243	0.947	4
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	37	0.910	4
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	164	0.961	4
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	41	0.978	2
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	247	0.891	4
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	1	0.241	4
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	1	0.970	3
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	163	0.602	5
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	2	0.489	1
2	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	51	0.963	4
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	1	0.626	1
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	2	0.924	2
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	32	0.897	3
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	156	0.669	3
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	99	0.881	2
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	Yes	0	0.901	0
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	Yes	0	0.962	0
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	15	0.793	1
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	247	0.691	4
3	▣▣▣▣▣▣▣▣	▣▣▣▣▣▣▣▣	No	37	0.960	2

TABLE C.1: Representative results from the eight-chunk simulations.
Here I present 10 examples from each value of injected cSNR.

cSNR	Γ_{inj}	Γ_{rec}	Correct	List position	$P(\Gamma_{\text{rec}})$	n_{mis}
4	□□□■□■□■	□□□□□□□□	No	96	0.476	3
4	□■□□■□■□	□■□□□□□□	No	104	0.794	2
4	□■□■□■□□	□■□□□□□□	No	16	0.965	1
4	□□■□□□□□	□□□□□□□□	No	1	0.834	1
4	□□□□■□□■	□□□□■□□■	Yes	0	0.788	0
4	□■□■□■□■	□■□□□□□□	No	83	0.873	2
4	□□□□□■□■	□□□□□■□■	Yes	0	0.998	0
4	□□■□□■□□	□□■□□■□□	Yes	0	0.944	0
4	□□□□■□□■	□□□□□□■□	No	2	0.813	1
4	□■□□■□□■	□□□□□□□□	No	45	0.433	3
5	□■□□■□□■	□■□□□□□□	No	7	0.763	3
5	□□□□■□□■	□□□□■□□□	No	1	0.978	1
5	□□■□□■□■	□□■□□□□■	No	5	0.813	2
5	□■□■□■□■	□■□□□□□■	No	99	0.910	3
5	□□■□□■□□	□□■□□□□□	No	18	0.673	2
5	□■□■□■□■	□■□□□□■□	No	2	0.696	1
5	□■□■□■□■	□■□□□□□□	No	2	0.639	2
5	□■□■□■□■	□■□□□□■□	Yes	0	0.999	0
5	□□■□□■□■	□□■□□□□■	No	11	0.451	2
5	□□■□□■□■	□□■□□□□■	No	2	0.951	1
6	□■□■□■□■	□■□□□□□■	No	3	0.300	3
6	□□□■□□■□	□□□■□□■□	Yes	0	0.989	0
6	□■□□□□■□	□■□□□□■□	Yes	0	0.899	0
6	□■□□■□□■	□■□□■□□■	No	1	0.750	1

TABLE C.1: Representative results from the eight-chunk simulations.
Here I present 10 examples from each value of injected cSNR.

cSNR	Γ_{inj}	Γ_{rec}	Correct	List position	$P(\Gamma_{\text{rec}})$	n_{mis}
6	□□■□□□■□	□□■□□□■□	Yes	0	0.998	0
6	□■□■□□□■	□■□■□□□■	Yes	0	0.999	0
6	□□■□□□■□	□□■□□□■□	Yes	0	0.999	0
6	□□□□■□□■	□□□□■□□■	Yes	0	0.729	0
6	□□□■□■□■	□□□■□■□■	Yes	0	0.998	0
6	□□□□■□■□	□□□□■□■□	Yes	0	0.978	0
7	□□■□■□■□	□□■□■□■□	Yes	0	0.994	0
7	□■□■□■□■	□■□■□■□■	Yes	0	0.906	0
7	□■□■□■□■	□■□■□■□■	No	1	0.677	1
7	□■□■□■□■	□■□■□■□■	Yes	0	0.811	0
7	□■□■□■□■	□■□■□■□■	Yes	0	0.993	0
7	□□□□□□□□	□□□□□□□□	Yes	0	0.997	0
7	□■□□□□□□	□■□□□□□□	Yes	0	0.996	0
7	□■□■□■□■	□■□■□■□■	No	1	0.830	1
7	□□□■□■□■	□□□■□■□■	Yes	0	0.999	0
7	□□■□■□■□	□□■□■□■□	Yes	0	0.999	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.999	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.999	0
8	□□■□■□■□	□□■□■□■□	Yes	0	0.999	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.976	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.999	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.628	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.999	0
8	□■□■□■□■	□■□■□■□■	Yes	0	0.447	0

TABLE C.1: Representative results from the eight-chunk simulations.
Here I present 10 examples from each value of injected cSNR.

cSNR	Γ_{inj}	Γ_{rec}	Correct	List position	$P(\Gamma_{\text{rec}})$	n_{mis}
8	□■□■□■□■□	□■□■□■□■□	Yes	0	0.981	0
8	□■□■□■□■□	□■□■□■□■□	Yes	0	0.988	0
9	□□□□□□□□	□□□□□□□□	Yes	0	0.977	0
9	□■□■□■□■□	□■□■□■□■□	Yes	0	0.999	0
9	□□□□□□□□	□□□□□□□□	Yes	0	0.971	0
9	□□□■□□□□	□□□■□□□□	Yes	0	0.999	0
9	□□■□■□■□	□□■□■□■□	Yes	0	0.995	0
9	□□□■□■□■	□□□■□■□■	Yes	0	0.998	0
9	□■□■□■□■□	□■□■□■□■□	Yes	0	0.999	0
9	□□□■□■□■	□□□■□■□■	Yes	0	0.999	0
9	□■□■□■□■□	□■□■□■□■□	Yes	0	0.999	0
9	□□□■□■□■	□□□■□■□■	Yes	0	0.999	0
10	□□□□□□□□	□□□□□□□□	Yes	0	0.977	0
10	□■□■□■□■□	□■□■□■□■□	Yes	0	0.999	0
10	□■□□□■□■	□■□□□■□■	Yes	0	0.999	0
10	□□■□■□■□	□□■□■□■□	Yes	0	0.999	0
10	□■□■□■□■□	□■□■□■□■□	Yes	0	0.999	0
10	□□□□□□□□	□□□□□□□□	Yes	0	0.983	0
10	□□□□□□□■	□□□□□□□■	Yes	0	0.994	0
10	□■□■□■□■□	□■□■□■□■□	Yes	0	0.999	0
10	□□■□■□■□	□□■□■□■□	Yes	0	0.999	0
10	□□□□□□■□	□□□□□□■□	Yes	0	0.998	0

Appendix D

Example .par and .priors file

In this appendix, I include an example of a .par and .priors file.

D.1 Priors file

The .priors file determines the prior volume, and prior distributions of the searched variables.

```
H0 uniform 1e-25 1e-20
PHI0 uniform 0 3.14159265359
PSI uniform 0 1.57079632679
COSIOTA uniform -1 1
```

D.2 Par file

The *.par* file includes all of the known parameters of the source. In this case, the *.par* file is for a hardware injection, for which we know all of the parameters.

```
NAME JPULSAR08
PSRJ JPULSAR08
F0 97.15415925
F1 -4.325e-09
RA 23:25:33.4997197871
DEC -33:25:06.6608320859
PEPOCH 52944.0007428703684126958
UNITS TDB
H0 1.1001376015540779e-24
PSI 0.170470927
PHI0 2.945
COSIOTA 0.073902656035643471
HPLUS 5.530730579525436e-25
HCROSS 8.130309075934591e-26
```

Appendix E

Python codes for RBB

The RBB method was defined in chapter 5, and used to generate results in chapter 6. In this appendix, I present the core code for splitting data into chunks, compiling into blocks, calling the Bayesian evidence estimators in parallel, calculating the RBB posteriors, and plotting the barcode plots.

E.1 do_splitting.py

```
#!/usr/bin/env python

# Import dependencies
import os
from argparse import ArgumentParser

parser = ArgumentParser()

#Define inputs

parser.add_argument("-n", "--n_chunks", dest = "n_chunks",
                    help = "Number of chunks", metavar = "INT")
```

```
parser.add_argument("-H", "--H1_input", dest = "H1_input",
                    help = "Full path to H1 input file", metavar =
                        "STRING")
parser.add_argument("-L", "--L1_input", dest = "L1_input",
                    help = "Full path to L1 input file", metavar =
                        "STRING")
parser.add_argument("-o", "--output_path", dest = "output_path",
                    help = "Full path to output of split files", metavar =
                        "STRING")
parser.add_argument("-P", "--Par_path", dest = "par_files",
                    help = "Full path to par files", metavar = "STRING")
parser.add_argument("-p", "--Prior_path", dest = "prior_files",
                    help = "Full path to prior files", metavar =
                        "STRING")
parser.add_argument("-E", "--Execdir", dest = "execdir",
                    help = "Full path to .py files used to do
                        analysis", metavar = "STRING")
parser.add_argument("-D", "--analysis_dir", dest = "analysis_dir",
                    help = "Full path to output of LPPEN files", metavar
                        = "STRING")
parser.add_argument("-l", "--nlive", dest = "nlive",
                    help = "Number of live points used for the nested
                        sampler", metavar = "STRING")

#Parse inputs
args = parser.parse_args()

n_chunks = int(args.n_chunks)
H1_input = str(args.H1_input)
L1_input = str(args.L1_input)
```

```
output_path = str(args.output_path)
par_files = str(args.par_files)
prior_files = str(args.prior_files)
analysis_dir = str(args.analysis_dir)
execdir = str(args.execdir)

# Call timesplitter.py for each dataset
os.system('python ' + execdir + 'timesplitter.py -f ' + H1_input +
          ' -n ' + str(n_chunks) + ' -o ' + output_path + ' -d H1')
os.system('python ' + execdir + 'timesplitter.py -f ' + L1_input +
          ' -n ' + str(n_chunks) + ' -o ' + output_path + ' -d L1')

#Write arguments for each block for the nested sampler
os.system('python ' + execdir + 'write_analysis_args.py -P ' +
          par_files + ' -p ' + prior_files + ' -D ' + analysis_dir + ' -n
          ' + str(n_chunks) + ' -l ' + str(args.nlive))
```

E.2 timesplitter.py

```
# Program to split a given time series into chunks.
# The code will read in a large stretch of data, and a cmdln arg
# for number of chunks n, and it should output, and it should
    output
# 2^n - 1 time series of continuous chunks
import os
from argparse import ArgumentParser

# Define inputs
parser = ArgumentParser()

parser.add_argument("-f", "--file", dest = "filename",
                    help = "Location of data file to unpack", metavar =
                        "STRING")
parser.add_argument("-n", "--n_chunks", dest = "n_chunks",
                    help = "Number of chunks to split the data into", metavar =
                        "INT")
parser.add_argument("-o", "--outfile", dest = "outfile",
                    help = "Outfile directory", metavar = "STRING")
parser.add_argument("-d", "--detector", dest = "detector",
                    help = "Detector Name eg H1, L1, V1", metavar =
                        "STRING")

#Parse inputs
args = parser.parse_args()

n_chunks = int(args.n_chunks)

filename = str(args.filename)
```

```
if not os.path.exists(filename):
    print('Please double check input file')

detector_name = str(args.detector)

outfile = str(args.outfile)
if outfile[-1] != '//':
    outfile = outfile + '/'
if not os.path.exists(outfile):
    os.makedirs(outfile)
elif not os.path.exists(outfile + detector_name + '//'):
    os.makedirs(outfile + detector_name + '//')

#Read data file
lines = []

with open(filename, 'r') as f:
    for line in f:
        lines.append(line[:-1])
    splitline = [line.split() for line in lines]

#Define chunk start and end times
time_length = float(splitline[-1][0]) - float(splitline[0][0])
chunk_time = []
for i in range(n_chunks):
    chunk_time.append(float(splitline[0][0]) + float(i *
        (time_length / n_chunks)))
chunk_time.append(float(splitline[-1][0]))

# Move off-integer to avoid double-counting
```


E.3 write_analysis_args.py

```
# This code generates and writes out the analysis arguments
# for each block of split data, to be read in by the nested
# sampler when run in parallel.

#Imports
import os
from argparse import ArgumentParser
parser = ArgumentParser()

# Define inputs

parser.add_argument("-D", "--Analysis_directory", dest="Directory",
                    help = "Directory in which analysis takes place", metavar =
                        "STRING")
parser.add_argument("-p", "--prior_path", dest = "prior_path",
                    help = "Path to pulsar prior file", metavar = "INT")
parser.add_argument('-P', '--par_path', dest = 'par_path',
                    help = 'Path to pulsar par file', metavar = 'STR')
parser.add_argument('-n', '--n_chunks', dest = 'n_chunks',
                    help = 'Number of cunks used in this analysis',
                    metavar = 'INT')
parser.add_argument('-l', '--nlive', dest = 'nlive',
                    help = 'Number of live points to use for the nested
                        sampler', metavar = 'STR')

# Parse inputs
args = parser.parse_args()
directory = str(args.Directory)
```

```
prior_path = str(args.prior_path)
par_path = str(args.par_path)
n_chunks = int(args.n_chunks)

if not directory[-1]=='//':
    directory = directory + '/'

if not os.path.exists(directory):
    print('Analysis directory does not exist. \nPlease reconsider
          the choices that bought you here.\n')
    quit(0)

# Test the number of chunked time files for each detector
H1_chunks = []
L1_chunks = []

H1dir = directory + 'H1/'
L1dir = directory + 'L1/'

for H1item in os.listdir(H1dir):
    H1_chunks.append(H1item)

for L1item in os.listdir(L1dir):
    L1_chunks.append(L1item)

if len(L1_chunks) != len(H1_chunks):
    print('Something has gone wrong with the chunking process.
          Please review.\n')
    quit(0)
```

```
if len(L1_chunks) != len(H1_chunks):
    print('Number of chunk files in each detector not equal! Please
          address')

if not os.path.exists('output/'):
    os.makedirs('output/')

#Define file where the arguments are to be written

outfile = directory + 'lalapps_args.txt'
outtext = open(outfile, 'w+')

outtext.seek(0)

# Write out the arguments
for i in range(n_chunks):
    for j in range(n_chunks + 1):
        if j>i:
            args_to_write = '--detectors H1,L1 --oldChunks --par-file '
                + par_path + ' --input-files ' + directory +
                'H1/H1-chunk_' + str(i) + '_to_' + str(j) + '.txt,' +
                directory + 'L1/L1-chunk_' + str(i) + '_to_' + str(j) +
                '.txt --outfile ' + directory + 'chunk_' + str(i)
                + '_to_' + str(j) + '.hdf --Nlive ' + str(args.nlive) + '
                --Nmcmcinitial 0 --tolerance 0.1 --prior-file ' +
                prior_path + '\n'
            print(args_to_write)
            outtext.write(args_to_write)
```

E.4 create_analysis_subs.py

```
#!/usr/bin/env python

# This bit of code was adapted from an earlier version by Matt
# Pitkin
# This will create the '.dag' and '.sub' files necessary to run
# the lalapps_pulsar_parameter_estimation_nested script in
# parallel,
# with one job for each block of data.

# Import required dependencies
from __future__ import print_function

import os
import sys
import uuid
import argparse

parser = argparse.ArgumentParser( )

#Define inputs

parser.add_argument("-r", "--rundir", dest="rundir",
                    required=True, help="Set the run directory for outputs")
parser.add_argument("-p", "--exec-path", dest="execpath",
                    required=True, help="Set the path to the required executables")
parser.add_argument("-N", "--N-sims", dest = "N_sims",
                    default=100, type=int, help="Set the number of parallel trials
                    to run")
```

```
parser.add_argument("-C", "--n_chunks", dest = "n_chunks", help =
    "Number of chunks to generate", metavar = "INT")
parser.add_argument("-a", "--analysis_code", dest="run_analysis",
    required=True, help="Set the RBB analysis run script location")
parser.add_argument("-s", "--SNR", dest="SNR", required=True,
    help="SNR of fake data chunks analysed")
parser.add_argument("-S", "--to-split", dest="to_split", help="Set
    to 0 if splitting and running LPPEN is not required. Default
    is 1", metavar="INT", default=0)
parser.add_argument("-o", "--outdir", dest="outdir",
    help="Location for the outputs of each run to go",
    metavar="STR")

# Parse input options
opts = parser.parse_args()

# The base directory
basedir = opts.rundir
if not os.path.isdir(basedir):
    print("Error... base directory '%s' does not exist." % basedir,
        file=sys.stderr)
    sys.exit(1)
if basedir[-1] != '/':
    basedir = basedir + '/'

outdir = opts.outdir
if not os.path.isdir(outdir):
    print("Output directory '%s' does not exist. Creating
        directory" % outdir, file=sys.stderr)
```

```
        os.system('mkdir -p ' + outdir)
if outdir[-1]!='//':
    outdir = outdir + '/'

# Create log directory if it doesn't exist
logdir = os.path.join(basedir, 'log')
if not os.path.isdir(logdir):
    os.mkdir(logdir)

# check executable path is a directory
if not os.path.isdir(opts.execpath):
    print("Error... path for run executables does not exist.",
          file=sys.stderr)
    sys.exit(1)

### -----

# Generate the .sub file to execute data collating and RBB
# analysis,
# using the lalapps_pulsar_parameter_estimation_nested outputs

grandchildsubfile = os.path.join(basedir,
    'run_grandchild_analysis.sub')
fp = open(grandchildsubfile, 'w')

# The contents of the file
grandchildsubtext = """universe = vanilla
executable = %s
```

```

arguments = -r %sPULSAR$(Pulsar_number) / -E %s -n %i -S %i -o
    %sPULSAR$(Pulsar_number) /
getenv = True
log = %s
error = %s
output = %s
notification = never
accounting_group = aluk.dev.01.cw.transient.development
queue 1
""" % (opts.run_analysis, basedir, opts.execpath,
    int(opts.n_chunks), int(opts.SNR), outdir, os.path.join(logdir,
    'run-$(cluster).log'),
    os.path.join(logdir, 'run-$(cluster).err'),
    os.path.join(logdir, 'run-$(cluster).out'))

# Write the sub file
fp.write(grandchildsubtext)
fp.close()

### -----

# setup Condor sub file for each of the nested sampler runs.

# The procedure if we require running the nested sampler
to_split = float(opts.to_split)

if to_split!=0:
    childsubfile = os.path.join(basedir, 'run_child_analysis.sub')
    fp = open(childsubfile, 'w')
    # Contents of the sub file

```

```

childsubtext = """universe = vanilla
executable =
    /software/physics/ligo/spack/000/linux-redhat6-x86_64/gcc-5.4.0/ldg/1
arguments = $(args)
getenv = True
log = %s
error = %s
output = %s
notification = never
accounting_group = aluk.dev.ol.cw.transient.development
queue 1
""" % (os.path.join(logdir, 'run-$(cluster).log'),
      os.path.join(logdir, 'run-$(cluster).err'),
      os.path.join(logdir, 'run-$(cluster).out'))
# Write the sub file
fp.write(childsubtext)
fp.close()

#The .dag file, which determines the order in which to run the
    jobs
dagfile = os.path.join(basedir, 'run_analysis.dag')
fp = open(dagfile, 'w')

# one loop for each target
for i in range(opts.N_sims):
    # Define job IDs for each target
    parent_uippen = '%03d' % (i+1)
    triangle_number = sum(range(int(opts.n_chunks)+1))
    argsfile = '%sPULSAR%s/output/lalapps_args.txt' % (basedir,
        parent_uippen)

```

```

lines = [line.strip('\n') for line in open(argsfile)]
# Job IDs for the nested sampler runs
parent_uippen = '%03d' %(i+1)
# Grandchild (gc) job ID for the RBB analysis
grandchild_uippen = '%03d_gc' %(i+1)

# Write dag line for the RBB job
dagstr = 'JOB %s %s\nRETRY %s 0\nVARS %s
        Pulsar_number=\"%s\" \n\n' % (grandchild_uippen,
        grandchildsubfile, grandchild_uippen,
        grandchild_uippen, parent_uippen)
fp.write(dagstr)

# Write dag lines for the nested sampler jobs
for j in range(triangle_number):
    # Child ID job number = concatenated parent_ID + '_' +
    # child id %03d
    child_uippen = '%03d_%03d' % (i+1, j+1)
    # Read in args file, parse as 'args="contents"'
    args = lines[j]
    # Child ID job number = concatenated parent_ID + '_' +
    # child id %03d
    dagchildstr = 'JOB %s %s\nRETRY %s 0\nVARS %s args=\"%s\"
        \n\n' % (child_uippen, childsubfile, child_uippen,
        child_uippen, args)
    fp.write(dagchildstr)

    dag_spawn_str = 'PARENT %03d_%03d CHILD %s_gc \n\n' %
        (i+1, j+1, parent_uippen)
    fp.write(dag_spawn_str)

```

```
fp.close()

# create dag and sub files if we DON'T need new nested sampler
# results
if to_split==0:
    dagfile2 = os.path.join(basedir, 'run_RBB.dag')
    fd = open(dagfile2, 'w')

# one loop for each target
for i in range(opts.N_sims):
    # Define job IDs for each target
    parent_uippen = '%03d' %(i+1)
    # Grandchild jobs to perform RBB
    grandchild_uippen = '%03d_gc' %(i+1)

    # Write dag line for the first sub file
    dagstr = 'JOB %s %s\nRETRY %s 0\nVARS %s Pulsar_number=\"%s\"
            \n\n' % (grandchild_uippen, grandchildsubfile,
                    grandchild_uippen, grandchild_uippen, parent_uippen)
    fd.write(dagstr)

fd.close()
```

E.5 do_analysis.py

```
#!/usr/bin/env python

# This file calls and manages the RBB protocol.

import os
from argparse import ArgumentParser
parser = ArgumentParser()

# Define inputs

# Directory to run in
parser.add_argument("-r", "--rundir", dest = "rundir",
                    help = "Full location to perform analysis (incl.
                            */Pulsar###/", metavar = "STRING")
# Directory containing executables
parser.add_argument("-E", "--Execdir", dest = "execdir",
                    help = "Full path to executable files used to do
                            analysis", metavar = "STRING")
# Number of chunks
parser.add_argument("-n", "--n_chunks", dest = "n_chunks",
                    help = "Number of chunks", metavar = "INT")
# Injected cSNR, if known (not vital if not known)
parser.add_argument("-S", "--SNR", dest = "snr", default = "1",
                    help = "SNR of data analysed, use 1 if not known",
                    metavar = "STR")
# Directory to write the outputs
parser.add_argument("-o", "--outdir", dest = "outdir",
```

```
        help = "Location for the output of each run",
        metavar = "STR")
# Destination for one-line summary of the run
parser.add_argument("-C", "--collated_stats_out", dest =
    "stats_dir",
        help = "Location and filename for the data from
        each run to be output", metavar = "STR")

args = parser.parse_args()
# Parse inputs
n_chunks = int(args.n_chunks)
rundir = str(args.rundir)
execdir = str(args.execdir)
outdir = str(args.outdir)
statsdir = str(args.stats_dir)

if execdir[-1]!='//':
    execdir = execdir + '/'

if rundir[-1]!='//':
    rundir = rundir + '/'

if outdir[-1]!='//':
    outdir = outdir + '/'

# Make sure output directory exists
os.system('mkdir -p ' + outdir)

# Read the posterior samples from nested sampler run
```

```
os.system('python ' + exedir + 'post_samples_test.py -N ' +
         str(n_chunks) + ' -d ' + rundir + 'output/')

# Extract data from .hdf5 files generated by nested sampler
os.system('python ' + exedir + 'read_data.py -i ' + rundir +
         'output/')

# Write evidences for all \Gamma_p configurations
os.system('python ' + exedir + 'collate_evidences.py -n ' +
         str(n_chunks) + ' -d ' + rundir + 'output/ -o ' + outdir +
         'data.txt')

# Call the RBB code
os.system('python ' + exedir + 'RBB_summing_evidence.py -i ' +
         outdir + 'data.txt -o ' + outdir + 'analysis_out' + ' -b ' +
         rundir + 'input_binary.txt')

# Generate one line summary
os.system('python ' + exedir + 'read_statistics.py -t ' + rundir
         + 'input_binary.txt -s ' + outdir +
         'analysis_out/sorted_binaries.txt -d ' + outdir +
         'analysis_out/output.txt -o ' + statsdir + ' -n ' +
         rundir[-4:-1] + ' -S ' + str(args.snr))
```

E.6 read_data.py

```
# This translates the .h5 or .hdf files output by the nested
    sampler
# into a less compact but easier to handle format.

import h5py
from argparse import ArgumentParser
import os

parser = ArgumentParser()
# Input the location of the .h5/.hdf files
parser.add_argument("-i", "--Input_directory", dest = "directory",
                    help = "Location of .hdf5s file to unpack", metavar
                        = "STRING")

args = parser.parse_args()
# Parse input

directory = str(args.directory)

# Define reading function
def read_hdf5(directory, item, suffix):
    # Open file "item" with h5py
    f = h5py.File(directory + '/' + item, 'r')
    # Read file f in the frameworks 'lalinference' and
        'lalinference_nest'
    data = f['lalinference']['lalinference_nest']
    # start empty arrays for variable names and mean values
    vari = []
```

```
    valu = []

# Populate list of variables with file contents
    for i in data.attrs:
        vari.append(i)
        valu.append(data.attrs[i])

# Create output .txt file
if suffix == 'hdf':
    output = directory + '/' + item[:-3]
elif suffix == 'h5':
    output = directory + '/' + item[:-2]
    if output[-1] != '.':
        outname = output + '.output.txt'
    else:
        outname = output + 'output.txt'

# Write out variables and results
    out_file = open(outname, 'w+')
    out_file.seek(0)
    for i in range(len(vari)):
        out_file.write(str(vari[i]) + ' = ' + str(valu[i]) +
            '\n')
    out_file.close()

# Read every file in the results directory
for item in os.listdir(directory):
    if item[-3:] == 'hdf':
        read_hdf5(directory, item, 'hdf')
    elif item[-2:] == 'h5':
        read_hdf5(directory, item, 'h5')
```

E.7 collate_evidences.py

```
# This code collates the evidences read from h5 and hdf
# files into all possible values of Gamma_p, for use in the
# upcoming RBB script.

# Inputs:
import os
from binary_structure import binary_structure
from argparse import ArgumentParser

parser = ArgumentParser()

# The number of chunks used
parser.add_argument("-n", "--n_chunks", dest = "n_chunks",
                    help = "Number of chunks used in data", metavar =
                        "INT")

# Directory with the evidence values written out previously
parser.add_argument('-d', '--input_directory', dest =
                    'input_directory',
                    help = 'Directory containing all of the outputs from
                            lalapps_pulsar_parameter_estimation_nested',
                    metavar = 'STRING')

# Output destination
parser.add_argument('-o', '--output', dest = 'output',
                    help = 'Location to output data file', metavar = 'STRING')

# Parse inputs
```

```
args = parser.parse_args()

n_chunks = int(args.n_chunks)

# The number of intermittenencies, and empty array ot fill
# with binary numbers for each intermittenency
numbers = range(2**n_chunks)
binary_number = []

# Populate list of binary numbers
for itt,number in enumerate(numbers):
    binary = bin(number)
    x = []
    # Remove first 2 characters (unnecessary)
    for item in binary[2:]:
        x.append(int(item))
    # Pad with a zero
    x.append(0)
    binary_number.append(x)

binary_numbers = []
for binary in binary_number:
    while len(binary) < n_chunks+1:
        binary.insert(0,0) #`Pad end of each number with a zero
    binary_numbers.append(binary)

print(binary_numbers)
```

```
#Import the relevant data for each configuration of chunks
input_directory = str(args.input_directory)

# Function to import signal or noise evidence based on bit
  ('atom') value
def assign_data_value(directory, chunk_start, chunk_end,
  atom_value):
  # .txt file with evidences for the given block
  f = directory + '/chunk_' + str(chunk_start) + '_to_' +
    str(chunk_end) + '.output.txt'
  item_instance = open(f, 'r')
  lines = []
  for line in item_instance:
    lines.append(line)
  # if signal hypothesis, read the signal evidence
  if atom_value == 1:
    atom_line = lines[0].split('=')
    chunks_value = atom_line[1]
  # if noise hypothesis, read noise evidence
  elif atom_value == 0:
    atom_line = lines[3].split('=')
    chunks_value = atom_line[1]
  # Just in case
  else:
    print("I'm sorry Dave, I can't do that for you")
    quit(0)
  return chunks_value

# This section of the code determines where blocks start and
```

```
# end in each intermittency by examining the diff between two
# adjacent bits. This is why the binary numbers are padded with
# zero.

# binns is abbreviation of binaries

data_bins = []
for bin_itt, bin_num in enumerate(binary_number):
    # Initiate data, to be filled with evidence values
    data = []

    # Initiate index, the position within the binary
    index = 0

    while index < len(bin_num)-1:
        # Start with - if the binary values are the same (ie a block
        # longer than a single chunk)
        if bin_num[index] - bin_num[index+1] == 0:
            # Running index looks forward to find the end of the block
            running_index = index + 1
            while running_index < n_chunks+1:
                # If the block continues in the next position
                if bin_num[index] - bin_num[running_index] == 0:
                    # Store the running index value
                    index_adder = running_index

                    # Try the next position
                    running_index += 1
                else:
                    # Exit the loop
                    running_index = n_chunks + 100
            chunk_start = index
```

```
if index_adder == n_chunks:
    chunk_end = index_adder
else:
    chunk_end = index_adder + 1 # Block ends AFTER the chunk
# Use functions above to pick out relevant attributes:
    intermittency value, evidence value (for signal or
    noise), posterior
chunk_value = bin_num[index]
data_value = assign_data_value(input_directory,
    chunk_start, chunk_end, chunk_value)
# Add these values to a data list to be output as the data
    for RBB
data.append(float(data_value[:-1]))
# One value gives the evidence for the plock. Pack the
    evidence values for the rest of the block with 0.
while index+1 < chunk_end:
    index +=1
    data.append(0)
index += 1
# Now what happens if the block is of length 1 chunk, and
    contains noise
elif bin_num[index+1] - bin_num[index] == 1:
    chunk_start = index
    chunk_end = index + 1
    chunk_value = 0
    data_value = assign_data_value(input_directory,
        chunk_start, chunk_end, chunk_value)
    data.append(float(data_value[:-1]))
    index +=1
```

```
# And if the block is of length 1, and is signal
elif bin_num[index+1] - bin_num[index] == -1:
    chunk_start = index
        chunk_end = index + 1
        chunk_value = 1
    data_value = assign_data_value(input_directory,
        chunk_start, chunk_end, chunk_value)
        data.append(float(data_value[:-1]))
        index +=1
else:
    print("I'm sorry Dave, I can't do that for you.")
    quit(0)

# Append selected evidence value(s) for the intermittencies to
    the list for all intermittencies.
data_bins.append(data)

# Write list of lists out
output = str(args.output)
outfile = open(output, 'w+')
outfile.seek(0)
for num,item in enumerate(data_bins):
    outfile.write(str(item) + '\n')
    print(str(num) + ': ' + str(item))
outfile.close()
```

E.8 RBB_summing_evidence.py

```
#!/usr/bin/env python

# This script is to find the probability of obtaining any one of
# the 256
# 8-bit binary unmbers which could constitute a changepoint
# configuration.
# We define \Gamma as the 8 bit number, and \gamma_i as each
# binary element
# of \Gamma.
# We can define the probability of \gamma_i as
# P_gamma =
# (1/(((2*pi)^-0.5)*sigma))*exp((- (D-gamma_i*h)^2)/(2*sigma^2));
# Now we can define some other things. for example, D is fed
# forward from
# the CW analysis. Sigma may be given (look into feed forward
# from CW
# analysis), h fed forward from CW analysis
# We compute the probability for each of the 256 8-bit numbers

from __future__ import division
import numpy as np

import matplotlib as mpl
mpl.use("Agg")
from argparse import ArgumentParser
import inspect
import os
import datetime
```

```
mpl.rcParams['agg.path.chunksize'] = 10000

import matplotlib.pyplot as plt
from binary_structure import binary_structure
from logaddexp import logaddexp
from logaddexpvect import logaddexpvect
import math
import matplotlib as mpl
import plot_functions

# Define a binomial coefficient for the prior probability later
def nCr(n,r):
    f = math.factorial
    return f(n) / f(r) / f(n-r)

# Define inputs
parser = ArgumentParser()

# Location of the collated data file
parser.add_argument("-i", "--input_file", dest = "infile",
                    help = "Location of the data file to be used as input, as a
                             string.", metavar = "STRING")

# Output destination
parser.add_argument("-o", "--Output_path", dest = "outpath",
                    help = "Location of the data file to be used to
                             print output.", metavar = "STRING")

# .txt file containing a binary hypothesis to highlight
parser.add_argument("-b", "--Test-binary", dest = "test_binary",
                    help = "Location of a file containing a hypothesis
                             test binary number.", metavar = "STRING")

# Number of intermittencies to plot on the barcode plot
```

```
parser.add_argument("-p", "--plot_number", dest = "plot_number",
                    help = "The number of ranked intermittencies to plot
                             as a barcode plot", metavar = "STRING", default =
                             25)

# Parse arguments
args = parser.parse_args()

infile = args.infile
if not os.path.exists(infile):
    print('Please double check input file')

output = args.outpath
if output[-1] != '/':
    output += '/'
if not os.path.exists(output):
    os.makedirs(output)
    print(output + ' created.')

# Define subroutine to read in evidences from file

data = []
def read_data(infile):
    data = []
    with open(infile, mode='r') as file:
        content = file.read()
    for line in content.split('\n'):
        # split up and clean each line
        string_line = line.split(',')
```

```
for item in range(len(string_line)):
    # Clean up string
    string_line[item] = string_line[item].strip()
    string_line[item] = string_line[item].strip('\t')
    string_line[item] = string_line[item].strip('[')
    string_line[item] = string_line[item].strip(']')
data_line = []
# populate the list 'data'
for item in string_line:
    if item != '':
        data_line.append(float(item))
if data_line != []:
    data.append(data_line)
return data

# read the data
data = read_data(infile)
all_data = np.asarray(data)

#create a list of binary numbers of length len(data)
numlist = np.linspace(0, (2**len(all_data[0]))-1,
    num=2**len(all_data[0]))

# generate list of binary numbers, start with empty array
bin_list = []

for itt in range(len(numlist)):
    #generate binary number
    number = bin(int(numlist[int(itt)]))
```

```
x = []
for item in number[2:]:
    x.append(int(item))
# pad front with zeros
while len(x) < len(all_data[0]):
    x = [0] + x
# add to list of binary numbers
bin_list.append(x)

bin_array = np.array(bin_list)

# Initialise vars for RBB
l_likelihood = np.zeros(len(bin_list))
l_evidence = []
config = 0
n_changepoints = []
l_norm = []

#Do RBB
for itt_number in range(len(bin_array)):
    config = itt_number
    # Get n_cp for given bin number
    binary_number = bin_list[itt_number]
    binary_struc = binary_structure( binary_number )
    n_changepoints.append(binary_struc["n_changepoints"])
    # select the data for this intermittency
    data = all_data[itt_number]
```

```

# Sum the evidences for signal and the evidences for noise as
    appropriate in each block
l_likelihood[config] = sum(data)

# calculate the prior for the configuration
l_norm.append(np.log((1/(len(binary_number))) * (nCn(len(data)-1, n_changes))))
l_evidence.append(l_likelihood[config])

##### Calculate odds, posteriors etc

# Calculate odds vs all-noise
l_odds = l_evidence - l_evidence[0]

# Convert from natural log to log10
l10_evidence = []
for item in l_evidence:
    print(item)
    l10_item = item / math.log(np.e,10)
    l10_evidence.append(l10_item)

# Calculate posteriors for Gamma_p

# Initialise array
posterior = np.zeros(len(l_evidence))

# log-plus all log-evidence values
for index in range(len(l_evidence)):
    running_denominator_sum = -1 * np.inf # log(0) to start
        running sum
    for ind2 in range(len(l_evidence)):

```

```
running_standin_sum = np.logaddexp(running_denominator_sum,
    (l_evidence[ind2] + l_norm[ind2])) # use logplus function
    to perform sum in logspace
running_denominator_sum = running_standin_sum
posterior_denom = running_denominator_sum
#Posterior calculated here:
posterior[index] = (l_evidence[index] + l_norm[index]) -
    posterior_denom

##### Sorted lists #####
# Most of these sorted lists are useful as sanity checks, and
    diagnostics.
# List of sorted posteriors, binary numbers sorted by posteriors,
    are outputs.

# Generate ranked lists
sorted_evidence = sorted(l_evidence)
sorted_l10_evidence = sorted(l10_evidence)

# index ranked list
evidence_index = [i[0] for i in sorted(enumerate(l_evidence),
    key=lambda x:x[1])]
sorted_n_CP = [n_changepoints[i] for i in evidence_index]

# Sorted by odds vs all noise
sorted_odds = sorted(l_odds)
sort_index = [i[0] for i in sorted(enumerate(l_odds), key=lambda
    x:x[1])]
```

```
# Sorted list of posteriors
sorted_posteriors = sorted(posterior)
posterior_index = [i[0] for i in sorted(enumerate(posterior),
    key=lambda x:x[1])]

# Binary numbers sorted by posterior probability
sorted_binaries = [bin_list[i] for i in posterior_index]

# List of priors sorted by posterior of associated intermittency
sorted_priors = [l_norm[i] for i in posterior_index]

# Evidence list sorted by posterior of corresponding intermittency
sorted_data = [all_data[i] for i in odds_index]

#Compare with binary hypothesis
# Read hypothesis binary
with open(args.test_binary, mode='r') as file:
    content = file.read()
    content_lines = content.split('\n')
    test_bin = content_lines[1].strip('\n')

test_binary = []
# Parse hypothesis into list
for entry in test_bin:
    test_binary.append(int(entry))

# Write out list of sorted binary numbers
bins_out = output + 'sorted_binaries.txt'
b = open(bins_out, 'w+')
b.seek(0)
```

```

for numb,binary_list in enumerate(sorted_binaries):
    b.write(str(binary_list) + '\n')
    # With each entry, test against test binary
    if binary_list[0] == test_binary[0]:
        run_sum = 0
        for entry in range(len(test_binary)):
            # Calculatte diff between entry and test binary
            run_sum = run_sum + np.absolute(int(test_binary[entry]) -
                int(binary_list[entry]))
            # If they match, find position down list of input binary
            if run_sum == 0:
                test_binary_position = 2**len(test_binary) - (numb + 1)

##### Generate results: full dictionary #####

# Write out full results dictionary
out = output + 'output.txt'
o = open(out, 'w+')
o.seek(0)
# Write out position of test binary
o.write('test_binary_position:\n')
o.write(str(test_binary_position) + '\n\n')

# One dictionary line for each Gamma_p, sorted by posterior.
    Dictionary includes:
# Binary number, posterior probability, evidence sum, Log-10
    evidence sum
results_dict = []
for i in range(len(sorted_odds_all)):

```

```
results_dict.append({"sorted_binaries":sorted_binaries[i],
                    "sorted_posteriors":np.exp(sorted_posteriors[i]),
                    "sorted_evidence":sorted_evidence[i] , "Log-10 sorted
                    evidence":sorted_l10_evidence[i]})

# Write out dictionaries
for i in range(len(results_dict)):
    o.write(str(results_dict[i]) + "\n")
o.close()

##### Generate barcode plot #####

# Posteriors to plot in the RHS
post_to_plot = sorted_posteriors[:, :-1]

# List of binaries to plot in the centre
sorted_binaries_to_plot = sorted_binaries[:, :-1]

# Calculate the Bayes factors for the top plot
# Atom = value for a single chunk
# Generate an empty list, to be filled with one entry for each
    chunk
data_signal_atoms = []
j = 0
data_noise_atoms = []
data_atoms = []
# Fill data_atoms array with xeros to fill
for index in range(len(sorted_binaries[0])):
    data_atoms.append(0)
```

```
# From 0 to n_chunks
for i in range(len(sorted_binaries[0])):
    # Placement = chunk number
    placement = (len(sorted_binaries[0]) - 1) - i
    # This line will have only one bit of value 1, in the
        "placement"th position
    signal_line = all_data[2**i]
    signal_atom = signal_line[placement]
    # Last binary in the list
    top_number = 2*(len(sorted_binaries[0])) - 1
    # Find the binary with only one bit of value 0, in the
        "placement"th position
    noise_line = all_data[top_number - 2**i]
    noise_atom = noise_line[placement]
    # Take their difference for the log-bayes factor
    atom_SNR = signal_atom - noise_atom
    # Add to list
    data_atoms[placement] = atom_SNR

# Call barcode plot function
plot_num = int(args.plot_number)
if len(sorted_odds_all) > plot_num:
    plot_functions.barcode_plot(sorted_binaries_to_plot[:plot_num],
        post_to_plot[:plot_num], data_atoms, true_binary_position,
        output)
else:
    plot_functions.barcode_plot(sorted_binaries,
        sorted_odds_all, data_atoms, true_binary_position, output)
```

```
##### Sanity check: Do posteriors sum to one? #####
posterior_sum = 0
for j in posterior:
    posterior_sum += np.exp(j)
print("Sanity check: summing posteriors. Sum = " +
      str(posterior_sum) + ".\n")
```

E.9 binary_structure.py

```
# This function is extracted from collate_data.py, and is designed
    to
# probe a binary number to determine how many changepoints, the
    number of plocks
# and the length of blocks.

def binary_structure(binary_number):
# Variables to be returned: block_length, block_numbers,
    n_breaks, n_changepoints
# This function deconstructs the binary number and determined how
    many
# blocks, how many changepoints, and where each block begins and
    ends.
import numpy as np

is_changepoint = np.abs(np.diff(binary_number))
n_changepoints = np.sum(is_changepoint)
bookend_binary = np.append(binary_number, 0)
bookend_binary = np.insert(bookend_binary, 0, 0)
block_end = (np.diff(bookend_binary)<0)
n_breaks = sum(block_end)
block_start = (np.diff(bookend_binary)>0)

block_number = np.zeros(len(bookend_binary))
n_blocks = np.sum(block_start)
block_length = []
block = 0
itt = 0
```

```
for itt in range(len(binary_number)):
    if block_start[itt] == 1:
        count = 0
        while block_end[itt+count] == 0:
            count = count + 1

        block_length.append(count)
        block = block + 1

    block_number[itt+1] = block_number[itt] + block_start[itt]

block_numbers = block_number[1:-1]
for i in range(len(block_numbers)):
    if binary_number[i] == 0:
        block_numbers[i] = 'NAN'

if block_length == []:
    block_length = 0
retvars = {"block_length":block_length,
           "block_numbers":block_numbers, "n_breaks":n_breaks ,
           "n_changepoints":n_changepoints}
return retvars
```

E.10 plot_functions.py

```
def plot_data(data, output):  
    # Generate the top plot for the barcode  
    # Import all the important stuff  
    import matplotlib as mpl  
    mpl.use("Agg")  
    import matplotlib.pyplot as plt  
    import numpy as np  
  
    # Plot the values as a bar chart  
    plt.bar(range(len(data)), data, width=1.0)  
    plt.grid(True)  
    plt.ylabel(r'$\log_{10}(B)$')  
    ax = plt.gca();  
    # Tidy up axes  
    ax.set_xticklabels([])  
    ax.set_ylim([0, np.amax(data) + (0.05 * (np.amax(data) - np.amin(data)))] )  
    major_ticks = np.arange(np.floor(np.amin(data)),  
                             np.floor(np.amax(data)), np.floor(np.amax(data) -  
                             np.amin(data)/2))  
    ax.set_yticks(major_ticks, minor=False)  
    ax.axhline(linewidth=1, color='k')  
    plt.show()  
  
def plot_odds(var, output):  
    # Plots the preferred statistic alongside the barcode. Used to  
    plot posteriors
```

```

# Import the important stuff
    import matplotlib as mpl
import numpy as np
mpl.use("Agg")
    import matplotlib.pyplot as plt

# Plot the chosen data as a line
    plt.plot(var, range(len(var),0,-1))
# Start tidying plot
    plt.xlabel(r"$\log_{10}(\mathrm{Posterior})$")
    plt.grid(True)
bx = plt.gca()
bx.set_xlim([np.amin(var)-10 , np.amax(var)+10 ])
# Number of ticks for x axis
major_ticks = np.arange(np.floor(np.amin(var)),
    np.floor(np.amax(var)+2), ( np.floor(np.amax(var) -
    np.amin(var))))
minor_ticks = np.arange(np.floor(np.amin(var)),
    np.floor(np.amax(var)+2), 0.5 * ( np.floor(np.amax(var) -
    np.amin(var))))
    bx.set_xticks(major_ticks, minor=False)
bx.set_xticks(minor_ticks, minor=True)
    bx.grid(which = 'minor', alpha=0.5)
bx.set_yticklabels([])
plt.show()

def barcode_plot(sorted_binaries, sorted_post, data,
    true_binary_position, output):
# Generates the barcode plot
import matplotlib as mpl

```

```
mpl.use("Agg")

import matplotlib.pyplot as plt
import numpy as np
import matplotlib.gridspec as gridspec

##### Pre-process, scale the posteriors for grey scale #####
shift_up_post = []
for post in sorted_post:
    shift_up_post.append(post +
        np.abs(np.amin((sorted_post[-1]))))
bin_array = np.array(sorted_binaries)
# Normalise values wrt largest value, so they are all between 0
    and 1
odds_scale = [(x/shift_up_post[0]) for x in shift_up_post]
shifted_array = []
# Between -1 and 1
for item in bin_array:
    shifted_array.append(item - 0.5)
scaled_binaries = []
for [i,item] in enumerate(shifted_array):
    scaled_binaries.append(odds_scale[i] * item)
    scaled_binaries[i] = scaled_binaries[i] + 0.5
##### Initialise layout of barcode plot
gs =
    gridspec.GridSpec(2,2,width_ratios=[4,1],height_ratios=[1,5])
gs.update(wspace=0.05, hspace=0.05)
# Plot the binaries as greyscale grid

ax3 = plt.subplot(gs[1:,:-1])
```

```

y = plt.imshow(scaled_binaries, cmap = 'gray_r',
               interpolation='none', aspect='auto')

# Tidy up axes
plt.xlabel(r'$\mathrm{Chunk}$')
plt.ylabel(r'$\mathrm{Rank}$')
cx = plt.gca();
major_ticks = np.arange(0, len(sorted_post)+1, 5)
minor_ticks = np.arange(0, len(sorted_post), 1)
cx.set_yticklabels(major_ticks)
cx.set_yticks(np.arange(0, len(sorted_post), 1), minor=True)
cx.set_yticks(major_ticks)
cx.set_yticklabels(major_ticks)
cx.set_xticks(np.arange(0, len(sorted_binaries[1]), 1))
cx.set_xticklabels(np.arange(1, len(sorted_binaries[1])+1,
                             1))

# Fill in edges with black
for i in range(len(sorted_post)):
    cx.axhline(i+0.5, linestyle='-', color='k')
    for j in range(len(sorted_binaries[1])):
        cx.axvline(j+0.5, linestyle='-', color='k')

# Put in the red lines around true binary
if true_binary_position<50:
    cx.axhline(true_binary_position-0.5, linestyle='-', color='r')
    cx.axhline(true_binary_position+0.5, linestyle='-', color='r')
    cx.axvline(len(sorted_binaries[1])-0.5, linestyle='-',
               color='r')
    cx.axvline(-0.5, linestyle='-', color='r')
cx.grid(which='min r')

#Generate top plot

```

```
ax1 = plt.subplot(gs[0, :-1])
plot_data(data, output)

# Generate RHS plot
ax4 = plt.subplot(gs[1:, -1])
plot_odds(sorted_post, output)
plt.bbox_inches='tight'

# Save
plt.savefig(output + 'barcode.png')
```

Bibliography

- [1] A. Einstein. *Die Grundlage der allgemeinen Relativitätstheories*, 1916.
- [2] F. W. Dyson, A. S. Eddington, C. Davidson. *A determination of the deflection of light by the sun's gravitational field, from observations made at the total eclipse of May 29, 1919*. *Philos. Trans. Royal Soc. A*, Bd. 220(571-581):S. 291–333, 1920.
- [3] N. S. Hetherington. *Sirius B and the gravitational redshift - an historical review*. *QJRAS*, Bd. 21:S. 246–252, 1980.
- [4] D. H. Frisch, J. H. Smith. *Measurement of the Relativistic Time Dilation Using μ -Mesons*. *Am. J. Phys*, Bd. 31(5):S. 342–355, 1963.
- [5] J. C. Hafele, R. E. Keating. *Around-the-World Atomic Clocks: Observed Relativistic Time Gains*. *Science*, Bd. 177(4044):S. 168–170, 1972.
- [6] T. Damour, J. H. Taylor. *Strong-field tests of relativistic gravity and binary pulsars*. *Phys. Rev. D*, Bd. 45:S. 1840–1868, 1992.
- [7] S. Alam, et al.. *Testing gravity on large scales by combining weak lensing with galaxy clustering using CFHTLenS and BOSS CMASS*. *MNRAS*, Bd. 465(4):S. 4853–4865, 2017.
- [8] I. Ciufolini, E. C. Pavlis. *A confirmation of the general relativistic prediction of the Lense-Thirring effect*. *Nature*, Bd. 431:S. 958–960, 2004.
- [9] J. M. Weisberg, J. H. Taylor, L. A. Fowler. *Gravitational waves from an orbiting pulsar*. *Scientific American*, Bd. 245:S. 74–82, 1981.

-
- [10] C. Misner, K. Thorne, J. Wheeler. *Gravitation*. 2017.
- [11] A. Liddle. *An introduction to modern cosmology*. John Wiley & Sons, 2015.
- [12] W. Rindler. *Relativity: special, general, and cosmological*. Oxford University Press, 2006.
- [13] L. A. Boyle, P. J. Steinhardt. *Probing the early universe with inflationary gravitational waves*. Phys. Rev. D, Bd. 77:S. 063504, 2008.
- [14] S. Chongchitnan, G. Efstathiou. *Prospects for direct detection of primordial gravitational waves*. Phys. Rev. D, Bd. 73, 2006.
- [15] B. S. Sathyaprakash, B. F. Schutz. *Physics, Astrophysics and Cosmology with Gravitational Waves*. Living Rev. Relativ., Bd. 12(1):S. 2, 2009.
- [16] C. Cutler, K. S. Thorne. *An overview of gravitational-wave sources*. In *General Relativity and Gravitation*, S. 72–111. World Scientific, 2002.
- [17] B. F. Schutz. *Gravitational-wave sources*. Class. Quantum Grav., Bd. 13(11A), 1996.
- [18] R. J. Dupuis, G. Woan. *Bayesian estimation of pulsar parameters from gravitational wave data*. Phys. Rev. D, Bd. 72:S. 102002, 2005.
- [19] A. C. Searle, P. J. Sutton, M. Tinto, G. Woan. *Robust Bayesian detection of unmodelled bursts*. Class. Quantum Grav., Bd. 25(11):S. 114038, 2008.
- [20] J. Veitch, et al.. *Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library*. Phys. Rev. D, Bd. 91:S. 042003, 2015.
- [21] B. P. Abbott, et al.. *Observation of Gravitational Waves from a Binary Black Hole Merger*. Phys. Rev. Lett., Bd. 116:S. 061102, 2016.

- [22] J. W. Barrett, et al.. *Accuracy of inference on the physics of binary evolution from gravitational-wave observations*. MNRAS, Bd. 477(4):S. 4685–4695, 2018.
- [23] B. S. Sathyaprakash, S. V. Dhurandhar. *Choice of filters for the detection of gravitational waves from coalescing binaries*. Phys. Rev. D, Bd. 44:S. 3819–3834, 1991.
- [24] S. A. Usman, et al.. *The PyCBC search for gravitational waves from compact binary coalescence*. Class. Quantum Grav., Bd. 33(21), 2016.
- [25] B. P. Abbott, et al.. *Search for gravitational waves from low mass binary coalescences in the first year of LIGO’s S5 data*. Phys. Rev. D, Bd. 79:S. 122001, 2009.
- [26] M. Branchesi. *Electromagnetic follow-up of gravitational wave transient signal candidates*. J. Phys. Conf. Ser, Bd. 375(6):S. 062004, 2012.
- [27] P. A. Evans, et al.. *Swift follow-up of gravitational wave triggers: results from the first aLIGO run and optimization for the future*. MNRAS, Bd. 462(2):S. 1591–1602, 2016.
- [28] B. P. Abbott, et al.. *First targeted search for gravitational-wave bursts from core-collapse supernovae in data of first-generation laser interferometer detectors*. Phys. Rev. D, Bd. 94(10), 2016.
- [29] T. Takiwaki, K. Kotake, Y. Suwa. *A comparison of two- and three-dimensional neutrino hydrodynamics simulations of core collapse supernovae*. Ap.J., Bd. 786(2):S. 83, 2014.
- [30] A. Burrows. *Colloquium: Perspectives on core-collapse supernova theory*. Rev. Mod. Phys., Bd. 85:S. 245–261, 2013.
- [31] R. Diehl, et al.. *Radioactive ^{26}Al from massive stars in the Galaxy*. Nature, Bd. 439(7072):S. 45–47, 2006.

- [32] A. Mezzacappa. *Ascertaining the core collapse supernova mechanism: The State of the Art and the Road Ahead*. ANNU REV NUCL PART S, Bd. 55(1):S. 467–515, 2005.
- [33] B. Müller. *The Status of Multi-Dimensional Core-Collapse Supernova Models*. Publ. Astron. Soc. Aust., Bd. 33, 2016.
- [34] G. Pagliaroli, F. Vissani, E. Coccia, W. Fulgione. *Neutrinos from Supernovae as a Trigger for Gravitational Wave Search*. Phys. Rev. Lett., Bd. 103(3), 2009.
- [35] A. C. Searle, P. J. Sutton, M. Tinto. *Bayesian detection of unmodeled bursts of gravitational waves*. Class. Quantum Grav., Bd. 26(15):S. 155017, 2009.
- [36] S. Klimenko, I. Yakushin, A. Mercer, G. Mitselmakher. *A coherent method for detection of gravitational wave bursts*. Class. Quantum Grav., Bd. 25(11):S. 114029, 2008.
- [37] B. P. Abbott, et al.. *All-sky search for long-duration gravitational wave transients in the first Advanced LIGO observing run*. Class. and Quant. Grav., Bd. 35, 2017.
- [38] B. Allen, A. C. Ottewill. *Detection of anisotropies in the gravitational-wave stochastic background*. Phys. Rev. D., Bd. 56(2):S. 545, 1997.
- [39] Bar-Kana. *Limits on direct detection of gravitational waves*. Phys. Rev. D, Bd. 50 2:S. 1157–1160, 1994.
- [40] B. P. Abbott, et al.. *An upper limit on the stochastic gravitational-wave background of cosmological origin*. Nature, Bd. 460, 2009.
- [41] J. Aasi, et al.. *Searching for stochastic gravitational waves using data from the two colocated LIGO Hanford detectors*. Phys. Rev. D, Bd. 91(2):S. 022003, 2015.
- [42] M. Zimmermann, E. Szedenits. *Gravitational waves from rotating and precessing rigid bodies: Simple models and applications to pulsars*. Phys. Rev. D, Bd. 20:S. 351–355, 1979.

- [43] P. B. Jones. *Comment on “Constraining Hadronic Superfluidity with Neutron Star Precession”*. Phys. Rev. Lett., Bd. 92:S. 149001, 2004.
- [44] J. G. Davies, G. C. Hunt, F. G. Smith. *Changing Periodicities in the Pulsars*. Nature, Bd. 221:S. 27, 1969.
- [45] V. RADHAKRISHNAN, D. J. COOKE, M. M. KOMESAROFF, D. MORRIS. *Evidence in Support of a Rotational Model for the Pulsar PSR 0833-45*. Nature, Bd. 221:S. 443, 1969.
- [46] J. E. Staff, P. Jaikumar, V. Chan, R. Ouyed. *Spindown of isolated neutron stars: Gravitational waves or magnetic braking?* Ap.J., Bd. 751(1):S. 24, 2012.
- [47] P. Jaranowski, A. Królak, B. F. Schutz. *Data analysis of gravitational-wave signals from spinning neutron stars: The signal and its detection*. Phys. Rev. D, Bd. 58:S. 063001, 1998.
- [48] A. Abramovici, et al.. *LIGO: The Laser Interferometer Gravitational-Wave Observatory*. Science, Bd. 256(5055):S. 325–333, 1992.
- [49] P. Jaranowski, A. Krolak, B. F. Schutz. *Data analysis of gravitational-wave signals from spinning neutron stars: The signal and its detection*. Phys. Rev. D, Bd. 58(6):S. 063001, 1998.
- [50] T. Adams, et al.. *Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era*. Class. Quantum Grav., Bd. 33(17):S. 175012, 2016.
- [51] P. Jaranowski, A. Królak. *Searching for gravitational waves from known pulsars using the F and G statistics*. Class. Quantum Grav., Bd. 27(19), 2010.
- [52] M. Pitkin, M. Isi, J. Veitch, G. Woan. *A nested sampling code for targeted searches for continuous gravitational waves from pulsars*. arXiv preprint arXiv:1705.08978, 2017.

- [53] J. Goodman, J. Weare. *Ensemble samplers with affine invariance*. *Comm App Math Com Sc*, Bd. 5:S. 65–80, 2010.
- [54] B. J. Brewer, L. B. Pártay, G. Csányi. *Diffusive nested sampling*. *Comput. Stat.*, Bd. 21(4):S. 649–656, 2011.
- [55] W. J. Handley, M. P. Hobson, A. N. Lasenby. *Polychord: nested sampling for cosmology*. *MNRAS: Letters*, Bd. 450(1):S. L61–L65, 2015.
- [56] G. Ashton, D. Jones, R. Prix. *Effect of timing noise on targeted and narrow-band coherent searches for continuous gravitational waves from pulsars*. *Phys. Rev. D*, Bd. 91(6), 2015.
- [57] J. Aasi, et al.. *Directed search for continuous gravitational waves from the Galactic center*. *Phys. Rev. D*, Bd. 88(10), 2013.
- [58] G. D. Meadors, E. Goetz, K. Riles. *Tuning into Scorpius X-1: adapting a continuous gravitational-wave search for a known binary system*. *Class. Quantum Grav.*, Bd. 33(10):S. 105017, 2016.
- [59] B. P. Abbott, et al.. *Upper Limits on Gravitational Waves from Scorpius X-1 from a Model-based Cross-correlation Search in Advanced LIGO Data*. *Ap.J.*, Bd. 847(1):S. 47, 2017.
- [60] J. Ming, et al.. *Optimally setting up directed searches for continuous gravitational waves in Advanced LIGO O1 data*. *Phys. Rev. D*, Bd. 97(2), 2018.
- [61] S. L. Shapiro, S. A. Teukolsky. *Black holes, white dwarfs, and neutron stars: The physics of compact objects*. Wiley-VCH, 1983.
- [62] *The ATNF Pulsar Database*. URL <http://www.atnf.csiro.au/people/pulsar/psrcat/>.
- [63] *Einstein@Home*. URL <https://einsteinathome.org/>.

-
- [64] M. A. Papa, et al.. *Hierarchical follow-up of subthreshold candidates of an all-sky Einstein@Home search for continuous gravitational waves on LIGO sixth science run data.* Phys. Rev. D, Bd. 94:S. 122006, 2016.
- [65] B. P. Abbott, et al.. *Results of the deepest all-sky survey for continuous gravitational waves on LIGO S6 data running on the Einstein@Home volunteer distributed computing project.* Phys. Rev. D, Bd. 94:S. 102002, 2016.
- [66] A. Singh, et al.. *Results of an all-sky high-frequency Einstein@Home search for continuous gravitational waves in LIGO's fifth science run.* Phys. Rev. D, Bd. 94:S. 064061, 2016.
- [67] J. Aasi, et al.. *Einstein@Home all-sky search for periodic gravitational waves in LIGO S5 data.* Phys. Rev. D, Bd. 87:S. 042001, 2013.
- [68] B. P. Abbott, et al.. *Einstein@Home search for periodic gravitational waves in early S5 LIGO data.* Phys. Rev. D, Bd. 80:S. 042003, 2009.
- [69] B. Abbott, et al.. *Einstein@Home search for periodic gravitational waves in LIGO S4 data.* Phys. Rev. D, Bd. 79:S. 022001, 2009.
- [70] C. Cutler, B. F. Schutz. *Generalized \mathcal{F} -statistic: Multiple detectors and multiple gravitational wave pulsars.* Phys. Rev. D, Bd. 72:S. 063006, 2005.
- [71] E. Goetz, K. Riles. *An all-sky search algorithm for continuous gravitational waves from spinning neutron stars in binary systems.* Class. Quantum Grav., Bd. 28(21):S. 215006, 2011.
- [72] B. Abbott, et al.. *All-sky search for periodic gravitational waves in LIGO S4 data.* Phys. Rev. D, Bd. 77:S. 022001, 2008.
- [73] B. P. Abbott, et al.. *Comprehensive all-sky search for periodic gravitational waves in the sixth science run LIGO data.* Phys. Rev. D, Bd. 94:S. 042002, 2016.

- [74] B. P. Abbott, et al.. *First low-frequency Einstein@Home all-sky search for continuous gravitational waves in Advanced LIGO data*. Phys. Rev. D, Bd. 96(12), 2017.
- [75] M. Shaltev. *Coherent follow-up of Continuous Gravitational-Wave candidates: minimal required observation time*. J. Phys. Conf. Ser, Bd. 363(1):S. 012043, 2012.
- [76] B. P. Abbott, et al.. *First narrow-band search for continuous gravitational waves from known pulsars in advanced detector data*. Phys. Rev. D, Bd. 96(12), 2017.
- [77] M. Pitkin. *Prospects of observing continuous gravitational waves from known pulsars*. MNRAS, Bd. 415(2):S. 1849–1863, 2011.
- [78] J. Aasi, et al.. *Advanced LIGO*. Class. Quantum Grav., Bd. 32(7):S. 074001, 2015.
- [79] B. P. Abbott, et al.. *Observation of Gravitational Waves from a Binary Black Hole Merger*. Phys. Rev. Lett., Bd. 116:S. 061102, 2016.
- [80] B. P. Abbott, et al.. *GW151226: Observation of Gravitational Waves from a 22-Solar-Mass Binary Black Hole Coalescence*. Phys. Rev. Lett., Bd. 116:S. 241103, 2016.
- [81] B. P. Abbott, et al.. *GW170104: Observation of a 50-Solar-Mass Binary Black Hole Coalescence at Redshift 0.2*. Phys. Rev. Lett., Bd. 118:S. 221101, 2017.
- [82] B. P. Abbott, et al.. *GW170608: Observation of a 19 Solar-mass Binary Black Hole Coalescence*. Ap.J.Lett., Bd. 851(2):S. L35, 2017.
- [83] B. P. Abbott, et al.. *GW170814: A Three-Detector Observation of Gravitational Waves from a Binary Black Hole Coalescence*. Phys. Rev. Lett., Bd. 119:S. 141101, 2017.

- [84] B. P. Abbott, et al.. *GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral*. Phys. Rev. Lett., Bd. 119:S. 161101, 2017.
- [85] B. P. Abbott, et al.. *Properties of the Binary Black Hole Merger GW150914*. Phys. Rev. Lett., Bd. 116(24), 2016.
- [86] B. P. Abbott, et al.. *Binary Black Hole Mergers in the First Advanced LIGO Observing Run*. Phys. Rev. X, Bd. 6:S. 041015, 2016.
- [87] F. Acernese, et al.. *Advanced Virgo: a second-generation interferometric gravitational wave detector*. Class. Quantum Grav., Bd. 32(2):S. 024001, 2014.
- [88] B. P. Abbott, et al.. *Estimating the Contribution of Dynamical Ejecta in the Kilonova Associated with GW170817*. Ap.J., Bd. 850(2):S. L39, 2017.
- [89] B. J. Shappee, et al.. *Early spectra of the gravitational wave source GW170817: Evolution of a neutron star merger*. Science, Bd. 358(6370):S. 1574–1578, 2017.
- [90] B. P. Abbot, et al.. *A gravitational-wave standard siren measurement of the Hubble constant*. Nature, Bd. 551(7678):S. 85–88, 2017.
- [91] A. Albert, et al.. *Search for High-energy Neutrinos from Binary Neutron Star Merger GW170817 with ANTARES, IceCube, and the Pierre Auger Observatory*. Ap.J., Bd. 850(2):S. L35, 2017.
- [92] M. Branchesi. *Multi-messenger astronomy: gravitational waves, neutrinos, photons, and cosmic rays*. J. Phys. Conf. Ser, Bd. 718(2):S. 022004, 2016.
- [93] I. Bartos. *Multimessenger Prospects with Gravitational Waves and Neutrinos after LIGO's First Discovery*. J. Phys. Conf. Ser, Bd. 888:S. 012001, 2017.
- [94] S. M. Adams, et al.. *Observing the next galactic supernova*. Ap.J., Bd. 778(2):S. 164, 2013.
- [95] K. Abe, et al.. *Detectability of galactic supernova neutrinos coherently scattered on xenon nuclei in XMASS*. Astropart. Phys., Bd. 89:S. 51–56, 2017.

-
- [96] B. P. Abbott, et al.. *Multi-messenger Observations of a Binary Neutron Star Merger*. Ap.J., Bd. 848(2):S. L12, 2017.
- [97] B. P. Abbott, et al.. *Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A*. Ap.J., Bd. 848(2):S. L13, 2017.
- [98] M. Agathos, et al.. *Constraining the neutron star equation of state with gravitational wave signals from coalescing binary neutron stars*. Phys. Rev. D, Bd. 92(2), 2015.
- [99] K. Takami, L. Rezzolla, L. Baiotti. *Constraining the Equation of State of Neutron Stars from Binary Mergers*. J. Phys. Conf. Ser, Bd. 600:S. 012056, 2015.
- [100] P. Kumar, M. Pürrer, H. P. Pfeiffer. *Measuring neutron star tidal deformability with Advanced LIGO: A Bayesian analysis of neutron star-black hole binary observations*. Phys. Rev. D, Bd. 95:S. 044039, 2017.
- [101] A. Mytidis, M. Coughlin, B. Whiting. *Constraining the R-mode saturation amplitude from a hypothetical detection of R-mode gravitational waves from a newborn neutron star: Sensitivity study*. Ap.J., Bd. 810(1):S. 27, 2015.
- [102] J. Cervantes-Cota, S. Galindo-Uribarri, G. Smoot. *A Brief History of Gravitational Waves*. Universe, Bd. 2(3):S. 22, 2016.
- [103] J. Weber. *Gravitational Radiation*. Phys. Rev. Lett., Bd. 18:S. 498–501, 1967.
- [104] O. D. Aguiar. *Past, present and future of the Resonant-Mass gravitational wave detectors*. Res. Astron. Astrophys., Bd. 11(1):S. 1, 2011.
- [105] J. Weber. *Gravitational-Wave-Detector Events*. Phys. Rev. Lett., Bd. 20:S. 1307–1308, 1968.
- [106] O. D. Aguiar. *Past, present and future of the Resonant-Mass gravitational wave detectors*. Res. Astron. Astrophys., Bd. 11(1):S. 1, 2011.

- [107] P. Linsay, et al.. *A Study of a Long Baseline Gravitational Wave Antenna System*, 1983. URL <https://dcc.ligo.org/T830001/public>.
- [108] S. N. Bagayev, V. P. Chebotayev, A. S. Dychkov, V. G. Goldort. *On the possibility of using lasers as detectors of gravitational waves*. *Appl. Phys.*, Bd. 25(2):S. 161–164, 1981.
- [109] D. I. Robertson, et al.. *The Glasgow 10m prototype laser interferometric gravitational wave detector*. *Rev. Scient. Instr.*, Bd. 66(9):S. 4447–4452, 1995.
- [110] D. Shoemaker, et al.. *Noise behavior of the Garching 30-meter prototype gravitational-wave detector*. *Phys. Rev. D.*, Bd. 38:S. 423–432, 1988.
- [111] M. E. Zucker. *The LIGO 40 m prototype laser interferometer gravitational wave detector*. In F. Satō, T. Nakamura, Herausgeber, *Marcel Grossmann Meeting on General Relativity*. 1992.
- [112] B. T. Lantz. *Quantum limited optical phase detection in a high power suspended interferometer*. Dissertation, Massachusetts Institute of Technology, 1999.
- [113] P. R. Saulson. *Terrestrial gravitational noise on a gravitational wave antenna*. *Phys. Rev. D*, Bd. 30:S. 732–736, 1984.
- [114] A. Abramovici, et al.. *LIGO: The Laser Interferometer Gravitational-Wave Observatory*. *Science*, Bd. 256(5055):S. 325–333, 1992.
- [115] J. Giaime, P. Saha, D. Shoemaker, L. Sievers. *A passive vibration isolation stack for LIGO: Design, modeling, and testing*. *Rev. Scient. Instr.*, Bd. 67(1):S. 208–214, 1996.
- [116] S. Wen, et al.. *Hydraulic external pre-isolator system for LIGO*. *Class. and Quant. Grav.*, Bd. 31(23):S. 235001, 2014.
- [117] F. Matichard, et al.. *Seismic isolation of Advanced LIGO: Review of strategy, instrumentation and performance*. *Class. Quantum Grav.*, Bd. 32(18):S. 185003,

- 2015.
- [118] A. V. Cumming, et al.. *Design and development of the advanced LIGO monolithic fused silica suspension*. *Class. Quantum Grav.*, Bd. 29(3):S. 035003, 2012.
- [119] M. Beker, et al.. *Seismic Attenuation Technology for the Advanced Virgo Gravitational Wave Detector*. *Phys. Procedia*, Bd. 37:S. 1389–1397, 2012.
- [120] K. Somiya. *Detector configuration of KAGRA—the Japanese cryogenic gravitational-wave detector*. *Class. and Quant.Grav.*, Bd. 29(12):S. 124007, 2012.
- [121] M. Punturo, et al.. *The Einstein Telescope: a third-generation gravitational wave observatory*. *Class. and Quant. Grav.*, Bd. 27(19):S. 194002, 2010.
- [122] Y. T. Liu, K. S. Thorne. *Thermoelastic noise and homogeneous thermal noise in finite sized gravitational-wave test masses*. *Phys. Rev. D.*, Bd. 62(12), 2000.
- [123] D. Martynov, et al.. *Sensitivity of the Advanced LIGO detectors at the beginning of gravitational wave astronomy*. *Phys. Rev. D*, Bd. 93(11):S. 112004, 2016.
- [124] G. M. Harry, et al.. *Thermal noise in interferometric gravitational wave detectors due to dielectric optical coatings*. *Class. Quantum Grav.*, Bd. 19(5):S. 897–917, 2002.
- [125] T. Hong, et al.. *Brownian thermal noise in multilayer coated mirrors*. *Phys. Rev. D.*, Bd. 87(8), 2013.
- [126] K. Somiya. *Detector configuration of KAGRA—the Japanese cryogenic gravitational-wave detector*. *Class. Quantum Grav.*, Bd. 29(12):S. 124007, 2012.
- [127] K. Kuroda, et al.. *Large-scale cryogenic gravitational wave telescope*. *Int. J. Mod. Phys. D*, Bd. 08(05):S. 557–579, 1999.
- [128] T. Uchiyama, et al.. *Cryogenic cooling of a sapphire mirror-suspension for interferometric gravitational wave detectors*. *Phys. Lett. A.*, Bd. 242:S. 211–214, 1998.

- [129] A. Buonanno, Y. Chen. *Optical noise correlations and beating the standard quantum limit in advanced gravitational-wave detectors*. *Class. Quantum Grav.*, Bd. 18(15):S. L95, 2001.
- [130] J. Aasi, et al.. *Enhanced sensitivity of the LIGO gravitational wave detector by using squeezed states of light*. *Nat. Photonics*, Bd. 7(8):S. 613, 2013.
- [131] M. G. Beker, J. F. J. van den Brand, E. Hennes, D. S. Rabeling. *Newtonian noise and ambient ground motion for gravitational wave detectors*. *J. Phys.: Conf. Ser.*, Bd. 363:S. 012004, 2012.
- [132] G. M. Harry, the LIGO Scientific Collaboration. *Advanced LIGO: the next generation of gravitational wave detectors*. *Class. Quantum Grav.*, Bd. 27(8):S. 084006, 2010.
- [133] *TAMA ground-based interferometer for the detection of gravitational waves*, 2000.
- [134] B. Willke, et al.. *The GEO 600 gravitational wave detector*. *Class. Quantum Grav.*, Bd. 19(7):S. 1377, 2002.
- [135] T. Accadia, et al.. *Virgo: a laser interferometer to detect gravitational waves*. *J. Instrum.*, Bd. 7(03):S. P03012, 2012.
- [136] A. Abramovici, et al.. *LIGO: The Laser Interferometer Gravitational-Wave Observatory*. *Science*, Bd. 256(5055):S. 325–333, 1992.
- [137] B. Abbott, et al.. *Detector description and performance for the first coincidence observations between LIGO and GEO*. *Nucl. Instrum. Methods Phys. Res. A.*, Bd. 517(1):S. 154 – 179, 2004.
- [138] B. Abbott, et al.. *Detector description and performance for the first coincidence observations between LIGO and GEO*. *Nucl. Instrum. Methods Phys. Res. A.*, Bd. 517(1-3):S. 154–179, 2004.

- [139] J. Aasi, et al.. *Characterization of the LIGO detectors during their sixth science run*. *Class. Quantum Grav.*, Bd. 32(11):S. 115012, 2015.
- [140] J. R. Smith. *The path to the enhanced and advanced LIGO gravitational-wave detectors*. *Class. Quantum Grav.*, Bd. 26(11):S. 114013, 2009.
- [141] B. P. Abbott, et al.. *Search For Gravitational-wave Bursts Associated with Gamma-ray Bursts using Data from LIGO Science Run 5 and Virgo Science Run 1*. *Ap.J.*, Bd. 715(2):S. 1438, 2010.
- [142] J. Aasi, et al.. *Gravitational Waves from Known Pulsars: Results from the Initial Detector Era*. *Ap.J.*, Bd. 785(2):S. 119, 2014.
- [143] *Large optic suspension, LIGO image gallery*. LIGO Scientific Collaboration - The science of LSC research. URL https://www.ligo.org/multimedia/gallery/opt-images/4670_20120918221814_DSC02834_llo_itmy.jpg.
- [144] C. L. Mueller, et al.. *The advanced LIGO input optics*. *Rev. Scient. Instr.*, Bd. 87(1):S. 014502, 2016.
- [145] J. Aasi, et al.. *Advanced LIGO*. *Class. Quantum Grav.*, Bd. 32(7):S. 074001, 2015.
- [146] A. F. Brooks, et al.. *Overview of Advanced LIGO adaptive optics*. *Appl. Opt.*, Bd. 55(29):S. 8256–8265, 2016.
- [147] M. Evans, et al.. *Observation of Parametric Instability in Advanced LIGO*. *Phys. Rev. Lett.*, Bd. 114:S. 161102, 2015.
- [148] C. Blair, et al.. *First Demonstration of Electrostatic Damping of Parametric Instability at Advanced LIGO*. *Phys. Rev. Lett.*, Bd. 118:S. 151102, 2017.
- [149] E. Hirose, et al.. *Update on the development of cryogenic sapphire mirrors and their seismic attenuation system for KAGRA*. *Class. Quantum Grav.*,

- Bd. 31(22):S. 224004, 2014.
- [150] J. Miller, et al.. *Prospects for doubling the range of Advanced LIGO*. Phys. Rev. D, Bd. 91:S. 062005, 2015.
- [151] B. Shapiro, et al.. *Cryogenically cooled ultra low vibration silicon mirrors for gravitational wave observatories*. Cryogenics, Bd. 81:S. 83–92, 2017.
- [152] B. P. Abbott, et al.. *Exploring the sensitivity of next generation gravitational wave detectors*. Class. Quantum Grav., Bd. 34(4):S. 044001, 2017.
- [153] M. Punturo, et al.. *The Einstein Telescope: a third-generation gravitational wave observatory*. Class. Quantum Grav., Bd. 27(19):S. 194002, 2010.
- [154] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 Aufl., 2007.
- [155] R. Schofield. Private communication, private wiki page.
- [156] S. Suvorova, et al.. *Hidden Markov model tracking of continuous gravitational waves from a neutron star with wandering spin*. Phys. Rev. D, Bd. 93:S. 123009, 2016.
- [157] B. P. Abbott, et al.. *First narrow-band search for continuous gravitational waves from known pulsars in advanced detector data*. Phys. Rev. D, Bd. 96:S. 122006, 2017.
- [158] *PEM Channel Info*. URL <http://pem.ligo.org/channelinfo/index.php>.
- [159] J. J. L. Duyvendak. *Further Data Bearing on the Identification of the Crab Nebula with the Supernova of 1054 A.D. Part I. The Ancient Oriental Chronicles*. Publ. Astron. Soc. Pac., Bd. 54(318):S. 91, 1942.

- [160] B. J. Owen, et al.. *Gravitational waves from hot young rapidly rotating neutron stars*. Phys. Rev. D, Bd. 58:S. 084020, 1998.
- [161] B. Shaw, et al.. *The largest glitch observed in the Crab pulsar*. MNRAS, Bd. 478(3):S. 3832–3840, 2018.
- [162] A. Effler, et al.. *Environmental influences on the LIGO gravitational wave detectors during the 6th science run*. Classical and Quantum Gravity, Bd. 32(3):S. 035017, 2015.
- [163] K. Riles. *Narrow lines in H1 DARM in O1 week 1*. URL <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=21982>.
- [164] T. A. et al. *The NoEMi (Noise Frequency Event Miner) framework*. J. Phys. Conf. Ser, Bd. 363(1):S. 012037, 2012.
- [165] L. S. Collaboration. *aLIGO LHO Logbook*. URL <https://alog.ligo-wa.caltech.edu/aLOG/>.
- [166] L. S. Collaboration. *aLIGO LLO Logbook*. URL <https://alog.ligo-la.caltech.edu/aLOG/>.
- [167] P. B. Covas, et al.. *Identification and mitigation of narrow spectral artefacts that degrade searches for persistent gravitational waves in the first two observing runs of Advanced LIGO*. Phys. Rev. D, Bd. 97:S. 082002, 2018.
- [168] K. Riles, W. Liu. *Folding of additional H1 corner station magnetometer channels*, 2016. URL <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=26527>.
- [169] K. Riles. *Narrow lines in ER9 H1 DARM*, 2016. URL <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=28364>.
- [170] D. Barker. *CDS maintenance summary*, 2016. URL <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=28508>.

- [171] *Narrow lines in H1 ER10 data*, 2016. URL <https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=32289>.
- [172] K. Riles. *Narrow L1 DARM lines have worsened at low frequencies following the Nov 15 maintenance*, 2016. URL <https://alog.ligo-la.caltech.edu/aLOG/index.php?callRep=29853>.
- [173] A. Hewish, et al.. *Observation of a Rapidly Pulsating Radio Source*. *Nature*, Bd. 217:S. 709, 1968.
- [174] D. R. Lorimer, M. Bailes, R. J. Dewey, P. A. Harrison. *Pulsar statistics: the birthrate and initial spin periods of radio pulsars*. *MNRAS*, Bd. 263(2):S. 403–415, 1993.
- [175] T. M. Tauris. *Spin-Down of Radio Millisecond Pulsars at Genesis*. *Science*, Bd. 335(6068):S. 561–563, 2012.
- [176] F. Camilo, S. E. Thorsett, S. R. Kulkarni. *The magnetic fields, ages, and original spin periods of millisecond pulsars*. *Ap.J. Lett.*, Bd. 421:S. L15–L18, 1994.
- [177] S. Detweiler. *Pulsar timing measurements and the search for gravitational waves*. *Ap.J.*, Bd. 234:S. 1100–1104, 1979.
- [178] Z. Arzoumanian, et al.. *The NANOGrav 11 Year Data Set: Pulsar-timing Constraints on the Stochastic Gravitational-wave Background*. *Ap.J.*, Bd. 859(1):S. 47, 2018.
- [179] S. L. Shemar, A. G. Lyne. *Observations of pulsar glitches*. *MNRAS*, Bd. 282(2):S. 677–690, 1996.
- [180] A. G. Lyne, S. L. Shemar, F. G. Smith. *Statistical studies of pulsar glitches*. *MNRAS*, Bd. 315:S. 534–542, 2000.
- [181] G. Ashton, R. Prix, D. I. Jones. *Statistical characterization of pulsar glitches and their potential impact on searches for continuous gravitational waves*. *Phys.*

- Rev. D, Bd. 96:S. 063004, 2017.
- [182] B. Haskell, A. Melatos. *Models of pulsar glitches*. Int. J. Mod. Phys. D, Bd. 24(03):S. 1530008, 2015.
- [183] M. A. Alpar, D. Pines, P. W. Anderson, J. Shaham. *Vortex creep and the internal temperature of neutron stars. I - General theory*. Ap.J., Bd. 276:S. 325–334, 1984.
- [184] P. W. Anderson, N. Itoh. *Pulsar glitches and restlessness as a hard superfluidity phenomenon*. Nature, Bd. 256:S. 25, 1975.
- [185] V. M. Lipunov. *The astrophysics of neutron stars*. Springer-Verlag Berlin Heidelberg, 1987.
- [186] M. F. Bennett, C. A. van Eysden, A. Melatos. *Continuous-wave gravitational radiation from pulsar glitch recovery*. MNRAS, Bd. 409(4):S. 1705–1718, 2010.
- [187] C. A. van Eysden, A. Melatos. *Gravitational radiation from pulsar glitches*. Class. Quantum Grav., Bd. 25(22):S. 225020, 2008.
- [188] A. Singh. *Gravitational wave transient signal emission via Ekman pumping in neutron stars during post-glitch relaxation phase*. Phys. Rev. D, Bd. 95:S. 024022, 2017.
- [189] B. Link, R. I. Epstein, J. M. Lattimer. *Pulsar constraints on neutron star structure and equation of state*. Phys. Rev. Lett., Bd. 83(17):S. 3362, 1999.
- [190] R. Manchester. *Pulsar glitches and their impact on neutron-star astrophysics*. arXiv preprint arXiv:1801.04332, 2018.
- [191] B. Shaw, et al.. *The largest glitch observed in the Crab pulsar*. MNRAS, Bd. 478(3):S. 3832–3840, 2018.
- [192] G. Ushomirsky, R. E. Rutledge. *Time-variable emission from transiently accreting neutron stars in quiescence due to deep crustal heating*. MNRAS, Bd. 325(3):S. 1157–1166, 2001.

- [193] B. Haskell, A. Patruno. *Are Gravitational Waves Spinning Down PSR J1023 + 0038?* Phys. Rev. Lett., Bd. 119:S. 161103, 2017.
- [194] T. Shahbaz, et al.. *The binary millisecond pulsar PSRJ1023+0038 during its accretion state – I. Optical variability.* MNRAS, Bd. 453(4):S. 3461–3473, 2015.
- [195] A. Papitto, et al.. *Swings between rotation and accretion power in a binary millisecond pulsar.* Nature, Bd. 501(7468):S. 517, 2013.
- [196] A. M. Archibald, et al.. *A Radio Pulsar/X-ray Binary Link.* Science, Bd. 324(5933):S. 1411–1414, 2009.
- [197] J. R. Thorstensen, E. Armstrong. *Is FIRST J102347.6+003841 Really a Cataclysmic Binary?* Astron. J, Bd. 130(2):S. 759, 2005.
- [198] Z. Arzoumanian, J. M. Cordes, I. Wasserman. *Pulsar Spin Evolution, Kinematics, and the Birthrate of Neutron Star Binaries.* Ap.J., Bd. 520(2):S. 696, 1999.
- [199] B. Abbott, et al.. *Beating the Spin-Down Limit on Gravitational Wave Emission from the Crab Pulsar.* Ap.J. Lett., Bd. 683(1):S. L45, 2008.
- [200] S. Mereghetti, J. A. Pons, A. Melatos. *Magnetars: Properties, Origin and Evolution.* Space Sci. Rev., Bd. 191(1):S. 315–338, 2015.
- [201] R. Turolla, S. Zane, A. L. Watts. *Magnetars: the physics behind observations. A review.* Rep. Prog. Phys, Bd. 78(11):S. 116901, 2015.
- [202] S. Mereghetti. *The strongest cosmic magnets: soft gamma-ray repeaters and anomalous X-ray pulsars.* Astron. Astrophys. Rev., Bd. 15(4):S. 225–287, 2008.
- [203] D. Huppenkothen, et al.. *Quasi-periodic Oscillations in Short Recurring Bursts of the Soft Gamma Repeater J1550–5418.* Ap.J., Bd. 787(2):S. 128, 2014. URL <http://stacks.iop.org/0004-637X/787/i=2/a=128>.

- [204] Y. Levin, M. van Hoven. *On the excitation of f modes and torsional modes by magnetar giant flares*. MNRAS, Bd. 418(1):S. 659–663, 2011. URL <http://dx.doi.org/10.1111/j.1365-2966.2011.19515.x>.
- [205] C. Chirenti, G. H. de Souza, W. Kastaun. *Fundamental oscillation modes of neutron stars: Validity of universal relations*. Phys. Rev. D, Bd. 91:S. 044034, 2015.
- [206] Y. Kojima, T. Okita. *Propagation and Transmission of Alfvén Waves in Rotating Magnetars*. Ap.J., Bd. 614(2):S. 922, 2004.
- [207] R. Quitzow-James, et al.. *Exploring a search for long-duration transient gravitational waves associated with magnetar bursts*. Class. Quantum Grav., Bd. 34(16):S. 164002, 2017. URL <http://stacks.iop.org/0264-9381/34/i=16/a=164002>.
- [208] H. Sotani, K. D. Kokkotas, N. Stergioulas. *Alfvén QPOs in Magnetars*. Journal of Physics: Conference Series, Bd. 189(1):S. 012038, 2009. URL <http://stacks.iop.org/1742-6596/189/i=1/a=012038>.
- [209] T. E. Strohmayer, A. L. Watts. *The 2004 Hyperflare from SGR 1806–20: Further Evidence for Global Torsional Vibrations*. Ap.J., Bd. 653(1):S. 593, 2006.
- [210] L. Bildsten, et al.. *Observations of Accreting Pulsars*. Astrophys. J. Suppl. S., Bd. 113(2):S. 367, 1997.
- [211] B. P. Abbott, et al.. *First low-frequency Einstein@Home all-sky search for continuous gravitational waves in Advanced LIGO data*. Phys. Rev. D, Bd. 96:S. 122004, 2017.
- [212] D. R. Lorimer. *Radio Pulsar Statistics*. In *Neutron Stars and Pulsars*, S. 1–17. Springer, 2009.

-
- [213] B. P. Abbott, et al.. *All-sky search for periodic gravitational waves in the O1 LIGO data*. Phys. Rev. D, Bd. 96:S. 062002, 2017.
- [214] E. Thrane, et al.. *All-sky, narrowband, gravitational-wave radiometry with folded data*. Phys. Rev. D., Bd. 91(12):S. 124012, 2015.
- [215] S. W. Ballmer. *A radiometer for stochastic gravitational waves*. Class. Quantum Grav., Bd. 23(8):S. S179, 2006.
- [216] B. Abbott, et al.. *Upper limit map of a background of gravitational waves*. Phys. Rev. D, Bd. 76:S. 082003, 2007.
- [217] E. Thrane, et al.. *Long gravitational-wave transients and associated detection strategies for a network of terrestrial interferometers*. Phys. Rev. D, Bd. 83:S. 083004, 2011.
- [218] E. Thrane, V. Mandic, N. Christensen. *Detecting very long-lived gravitational-wave transients lasting hours to weeks*. Phys. Rev. D, Bd. 91:S. 104021, 2015.
- [219] B. Abbott, et al.. *Analysis of first LIGO science data for stochastic gravitational waves*. Phys. Rev. D, Bd. 69:S. 122004, 2004.
- [220] E. Thrane, M. Coughlin. *Searching for gravitational-wave transients with a qualitative signal model: Seedless clustering strategies*. Phys. Rev. D, Bd. 88:S. 083010, 2013.
- [221] R. Prix, S. Giampanis, C. Messenger. *Search method for long-duration gravitational-wave transients from neutron stars*. Phys. Rev. D, Bd. 84:S. 023007, 2011.
- [222] D. Keitel. *Robust semicoherent searches for continuous gravitational waves with noise and signal models including hours to days long transients*. Phys. Rev. D, Bd. 93:S. 084024, 2016.

- [223] D. Keitel, G. Ashton. *Faster search for long gravitational-wave transients: GPU implementation of the transient F-statistic*. *Class. Quantum Grav.*, 2018.
- [224] P. Astone, et al.. *Method for all-sky searches of continuous gravitational wave signals using the frequency-Hough transform*. *Phys. Rev. D*, Bd. 90:S. 042002, 2014.
- [225] J. D. Scargle. *Studies in Astronomical Time Series Analysis. V. Bayesian Blocks, a New Method to Analyze Structure in Photon Counting Data*. *Ap.J.*, Bd. 504(1):S. 405, 1998.
- [226] C. Biwer, et al.. *Validating gravitational-wave detections: The Advanced LIGO hardware injection system*. *Phys. Rev. D.*, Bd. 95(6):S. 062002, 2017.
- [227] S. Karki, et al.. *The Advanced LIGO photon calibrators*. *Rev. Scient. Instr.*, Bd. 87(11):S. 114503, 2016.
- [228] H. Wittel, et al.. *New design of electrostatic mirror actuators for application in high-precision interferometry*. *Class. Quantum Grav.*, Bd. 32(17):S. 175021, 2015.
- [229] M. Pitkin. Private communication, SNR of hardware injections.
- [230] S. J. Zhu, M. A. Papa, S. Walsh. *New veto for continuous gravitational wave searches*. *Phys. Rev. D*, Bd. 96:S. 124007, 2017.
- [231] *Jodrell bank: Glitch table*, <http://www.jb.man.ac.uk/pulsar/glitches/gTable.html>.
URL <http://www.jb.man.ac.uk/pulsar/glitches/gTable.html>.
- [232] G. S. Davies, M. Pitkin, G. Woan. *A targeted spectral interpolation algorithm for the detection of continuous gravitational waves*. *Class. Quant. Grav.*, Bd. 34(1):S. 015010, 2016.