

Data Analysis Techniques for Continuous Gravitational Wave Searches

Joseph Bayley

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Physics and Astronomy
College of Science and Engineering
University of Glasgow



University
of Glasgow

February 2017

Abstract

Contents

Abstract	i
Acknowledgements	viii
Declaration	ix
1 Introduction	1
1.1 Gravitational waves	2
1.2 Sources and signals	5
1.2.1 Transient	5
1.2.2 Stochastic	7
1.2.3 Continuous waves	7
1.3 Detectors	10
1.3.1 Laser Interferometers	11
1.4 Probability and Bayes Theorem	14
1.4.1 Basic probability	14
1.4.2 Bayesian Inference	15
2 Searching for continuous gravitational waves	18
2.0.1 Continuous signal model	18
2.0.2 Signals in data	20
2.0.3 Fully-coherent searches	20
2.0.4 Semi-coherent searches	21
3 Searching for continuous gravitational waves using SOAP	22
3.1 Introduction	24
3.2 Viterbi algorithm	25
3.3 The transition matrix	26
3.4 Single detector	27
3.5 Multiple detectors	31
3.6 Memory	33

3.7	Summed input data	35
3.8	Line-aware statistic	36
3.9	Line aware statistic for consistent amplitude	40
3.10	Testing the algorithm	44
3.10.1	S6 injections into gapless Gaussian noise	47
3.10.2	S6 injections into Gaussian noise with gaps	49
3.10.3	Tests on the S6 MDC	49
3.10.4	Optimisation of Line-aware statistic.	51
3.10.5	Computational cost	53
3.11	Discussion	55
4	Machine learning for continuous wave searches	58
4.1	Neural networks	58
4.1.1	Neurons	59
4.1.2	Network structure	60
4.1.3	Activation functions	60
4.1.4	Loss functions	60
4.1.5	Training	60
4.2	Convolutional Neural Networks	60
4.2.1	Convolutional layers	61
4.2.2	Max pooling layers	63
4.2.3	CNN structure	63
4.3	CW search	63
4.3.1	Network structure	63
4.3.2	Network Visualisation	63
4.4	Data and simulations	63
5	Detector Characterisation with SOAP	66
5.1	Instrumental lines	66
5.2	Identifying and monitoring instrumental lines	67
5.3	Identifying and cleaning lines with SOAP	68
5.4	Summary pages	68
6	Summary	69
A	Continuous gravitational wave injections	70
A.0.1	CW Signal	70
A.0.2	Time series and complex fast Fourier transform (FFT) injections	70
A.0.3	Spectrogram injections	70

List of Tables

3.1	Table shows the ranges of the search parameters and their optimised values for injections into gapless Gaussian noise, Gaussian noise with gaps and the S6 mock data challenge (MDC). For gapless Gaussian noise and Gaussian noise with gaps, there are 10 parameter values spaced linearly between the limits. For the S6 MDC the parameters, τ , w_L and w_S were distributed in log space between the limits and $p(M_L)/p(M_N)$ is distributed uniformly.	48
-----	---	----

List of Figures

1.1	Shows how the plus and cross polarisation's affect a ring of test particles. This assumes the wave is travelling out of the page and the effects have been greatly exaggerated. This also shows an example of how this effects the test masses of an interferometer. This will be described in more detail in Sec. 1.3.	4
1.2	Each gravitational-wave (GW) signal type can be categorised based on its signal length and how well the signal is modelled. Transient signals which are short duration, include both well modelled compact binary coalescence (CBC) signals and unknown Burst signals. Long duration signals include well modelled continuous wave (CW) signals and unknown Stochastic signals.	5
1.3	star mountain	8
1.4	Neutron star rotation CFS	10
1.5	This figure shows a basic interferometer.ETMY and ETMX refer to the external test masses, which are just mirrors at the end of the interferometer arms. ITMY and ITMX refer to the internal test masses, these create a cavity in the interferometers arms which can build up laser power. BS is the beam splitter which splits the Laser beam equally to each arm, this then recombines the beams back to the detector.	12
1.6	The antenna response is shown as in [romano2019SearchesStochastic] for the plus, cross and average polarisation's. The detectors arms lie on the x and y axis in the above plots.	13
1.7	The different noise sources affect the sensitivity of the Laser Interferometer Gravitational-wave Observatory (LIGO) detectors at different frequencies. This shows the various sources how they affect the noise curve [aasi2015AdvancedLIGO].	14

3.1	This figure shows an example of a spectrogram which is typical of LIGO data which is searched through. Here an instrumental line has been injected at 100.006 Hz. The white track shows a random walk track through this spectrogram whereas the red line shows the track which gives the highest sum of detector power. The second panel shows a histogram of a subset of all paths which can be found through the given spectrogram from start to finish. This is a subset as the total number of paths is too large to calculate. The value of the statistic which comes from the optimal path is ~ 1700 . This is much larger than any of the random tracks in our subset and much larger than the mean of all tracks.	23
3.2	Fig. 3.4c shows the observed data, i.e the log-likelihood values $C_{j,k}$. Fig. 3.2b shows the calculated log-probabilities $V_{j,k}$. $A_{j,k}$ is shown in parentheses, where the up, centre or down (UCD) components correspond to $i = [-1, 0, 1]$ respectively. The red line shows the path that gives the maximum probability. The transition matrix for the UCD jumps is $[0, 1, 0]$ and corresponds to the un-normalised prior log-probabilities of these jumps occurring.	30
3.3	Lookup tables using the line aware statistic in Eq. 3.29.	41
3.4	Lookup tables using the line aware statistic for consistent amplitude as in Sec. 3.9. Each of these use the parameters $p_s(\lambda) = 4, p_l(\lambda) = 5$ and $p(M_L)/p(M_G) = 0.03$	43
3.5	The results that the SOAP algorithm returns from an injection with an optimal signal-to-noise-ratio (SNR) of 90, i.e., the SNR in H1 is 64 and the SNR in L1 is 62. The signal is injected into Gaussian noise, where the 1800 s short Fourier transforms (SFTs) have been summed over 1 day. The top panel shows a simulation of summed SFTs from H1, the second panel shows the same for L1, the third panel shows the values proportional to the log-odds ratios in Eq. 3.24. The log-odds have been normalised such that the sum of all the odds ratios in every time bin are equal to 1. The bottom panel shows the injected signal track (black dotted) and the track found in the ‘imaginary’ detector by the two-detector SOAP search with the line-aware statistic (red), both of these tracks are at the geo-centre. In this case the root median square (RMS) of the difference between the Viterbi track and injected signal track was ~ 1 bin, where 1 bin is 0.00056 Hz wide.	46

3.6	Panels 3.6a and 3.6c show the detection efficiency as a function of SNR and depth respectively. Here SNR is the the integrated SNR which we would expect to recover from the available data. The four curves refer to injections into gapless Gaussian noise (red), Gaussian noise with gaps in data, where the noise floor is either fixed (blue-dashed) or it is moving with time (orange) in the same way as the S6 MDC and injections into real data i.e., the S6 MDC. In the gapless Gaussian noise case, the recovered integrated SNR refers to the SNR the injection would have if it had the same amount of data as in the cases with gaps. The curves are made by fitting a sigmoid Eq. 3.39 to binomial detection data with a 1% false alarm rate, as explained in Sec. 3.10.1, the error bounds are the 5% and 95% intervals. At 95% efficiency and a 1% false alarm rate, this shows we can detect to an SNR of ~ 60 and a sensitivity depth of $\sim 34 \text{ Hz}^{-1/2}$ for gapless Gaussian noise and an SNR of ~ 69 and 72 and a sensitivity depth of $\sim 13 \text{ Hz}^{-1/2}$ and $\sim 10 \text{ Hz}^{-1/2}$ for the Gaussian with gaps case with fixed noise floor and moving noise floor respectively. For the S6 MDC we can detect an SNR of ~ 74 and a sensitivity depth of $\sim 13 \text{ Hz}^{-1/2}$. Panels 3.6b and 3.6d show the RMS of the difference between the injected signal track and the track found by SOAP as a function of SNR and sensitivity depth respectively. This is shown in units of bins where each bin is 0.00056 Hz wide.	50
3.7	This figure shows the optimisation of the statistic for the Gaussian noise case.	52
3.8	This figure shows the optimisation of the line-aware statistic for the S6 MDC case.	53
3.9	This figure shows an example of when the line aware statistic is used compared to a version when it is not.	54
4.1	Basic neuron	59
4.2	Simple neural network	61
4.4	Max pooling layer	65
5.1	Caption	67

Acknowledgements

Declaration

Chapter 1

Introduction

Gravitational waves were first predicted in 1915 as a consequence of Einstein's theory of general relativity []. They were theorised as ripples in the fabric of space-time. The first observational evidence that GW exist came from observations of the Hulse-Taylor binary [weisberg1981Gravitational, weisberg2004RelativisticBinary]. This observation showed the binary pulsar system was inspiraling and therefore losing energy. This loss in energy matched the general relativity (GR) prediction which assumed the energy was lost to GW. The first direct detection of gravitational waves was not made until 2015 when the two LIGO detectors in the US [abbott2016ObservationGravitational] identified a signal from a binary black hole (BBH) system. This was not only the first observation of GW but gave information on a yet unobserved astrophysical system. This has since been followed by many more detections of BBH signals involving LIGO and Virgo including [abbott2017GW170814ThreeDetector, theligoscientificcollaboration2020GW190425Observation]. In 2017 the LIGO detectors observed the first binary neutron star (BNS) system [abbott2017GW170817] which had a corresponding electromagnetic counterpart. These detections opened up the field of gravitational wave astronomy, where many more detections are expected. As well as searching for BBH and BNS signals, there are many efforts to detect other types of GW signals.

This thesis focuses on efforts to search for a particular type of GW which are thought to originate from rapidly rotating neutron stars. In this chapter, I will review introductory material for the following chapters. This includes a general introduction to the generation of gravitational waves in Sec. 1.1 and their sources in Sec. 1.2. I will then introduce instruments used to detect GW in Sec. 1.3. In Chapter 2 I will introduce the general model for CWs and current methods used to detect them. Chapters 3, 4 and 5 will go into detail about techniques developed by the author to search for CW signals. Finally I will summarise this work and discuss future developments in chapter 6.

1.1 Gravitational waves

In general relativity, gravity is thought of as the curvature of space-time and it is this curvature that instructs matter how to move. The matter in the universe also has an effect on the curvature of the space-time. The larger the mass of the matter the more the space-time is distorted. Space-time can generally be described by Einstein's field equations,

$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}. \quad (1.1)$$

where $G_{\mu\nu}$ is the Einstein tensor and $T_{\mu\nu}$ is the stress-energy tensor. The stress energy tensor describes the distribution of mass and energy in the universe. The Einstein tensor contains information on the curvature of the universe. This can be derived directly from the metric tensor $g_{\mu\nu}$ which describes the geometry of the universe. In empty space one can assume that space-time is flat, i.e. there is no curvature to space-time. The metric tensor for this can then be defined as,

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

Each index of this matrix refers to a space-time dimension, i.e. $x^0 = t$, $x^1 = x$, $x^2 = y$ and $x^3 = z$. Measuring a distance dx in space-time can be different for different observers, therefore, one needs a measure which is invariant for every observer. This is the space-time interval ds , also known as the line element, between two events. This is defined as,

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

As in Einstein's notation this is a sum over the indices μ and ν . Eq. 1.3 is essentially Pythagoras's theorem, therefore, can be thought to describe the space-time 'distance' between the two events. For flat space-time, $\eta_{\mu\nu}$, this can then be written as,

$$ds^2 = -c^2 dt^2 + dx^2 + dy^2 + dz^2. \quad (1.4)$$

The Einstein equations Eq. 1.1 then demonstrate how the curvature of space-time $G_{\mu\nu}$ depends on the matter and energy distribution $T_{\mu\nu}$ within it.

A gravitational wave can be described as a ripple in this space time. The simplest way to visualise this is just a small time dependent change to the flat space-time metric $\eta_{\mu\nu}$. In linearised theory of gravity, the space-time metric $g_{\mu\nu}$ can then be defined as,

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

where $\eta_{\mu\nu}$ is the metric for flat space-time and $h_{\mu\nu}$ is some perturbation, where $|h_{\mu\nu}| \ll 1$ [flanagan2005BasicsGravitational]. In this linearised theory the perturbations to the metric tensor are assumed to be small, therefore, Einstein's field equations can be solved such that the solution is a plane wave. Whilst I skip over lots of the calculation here, more information on this derivation can be found in [flanagan2005BasicsGravitational] By using $g_{\mu\nu}$ from Eq. 1.5, we can write the linearised Einstein equations as,

$$\square h_{\mu\nu} = -16\pi T_{\mu\nu}. \quad (1.6)$$

In empty space there is no matter, therefore the stress energy tensor $T_{\mu\nu} = 0$. This allows Eq. 1.6 to be reduced to,

$$\square h_{\mu\nu} = 0. \quad (1.7)$$

This is a wave equation which follows the same form as in electrodynamics, therefore, the general solutions can be written down as,

$$h_{\mu\nu} = A_{\mu\nu} e^{ik_\alpha x^\alpha}, \quad (1.8)$$

where each component of $h_{\mu\nu}$ is a sinusoid travelling along vector k^α with amplitude $A_{\mu\nu}$ [capano2011SearchingGravitational]. At this point the set of equations are not simple, the symmetric tensor $A_{\mu\mu}$ has 10 independent components. This can be greatly simplified by choosing a different gauge (coordinate system) where the metric perturbation is both transverse and traceless (TT) [flanagan2005BasicsGravitational]. This is just a choice of coordinate system which does not change any current assumptions. A traceless metric is one where the sum of the diagonal elements are 0 and a transverse metric is when the oscillations are perpendicular to the direction of travel. This gauge imposes two conditions: one is that $h_{\mu\nu}$ is traceless, i.e. that the sum of the diagonal elements are 0 and the other is that $h_{\mu\nu}$ is transverse. The transverse element means that the oscillations of the wave happen perpendicular to the direction of travel. At this point we can choose that the wave is travelling in the z direction which means that $k = (\omega, 0, 0, k)$. By then adopting the TT gauge there are only two unique components to the metric which is defined as,

$$h_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+ & h_\times & 0 \\ 0 & h_\times & -h_+ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} e^{i(kt - \omega t)}. \quad (1.9)$$

This shows the two polarisations of gravitational waves, h_+ and h_\times . The affect of each of the polarisations on a ring of test particles can be seen in Fig. 1.1 where the gravitational wave is travelling out of the page.

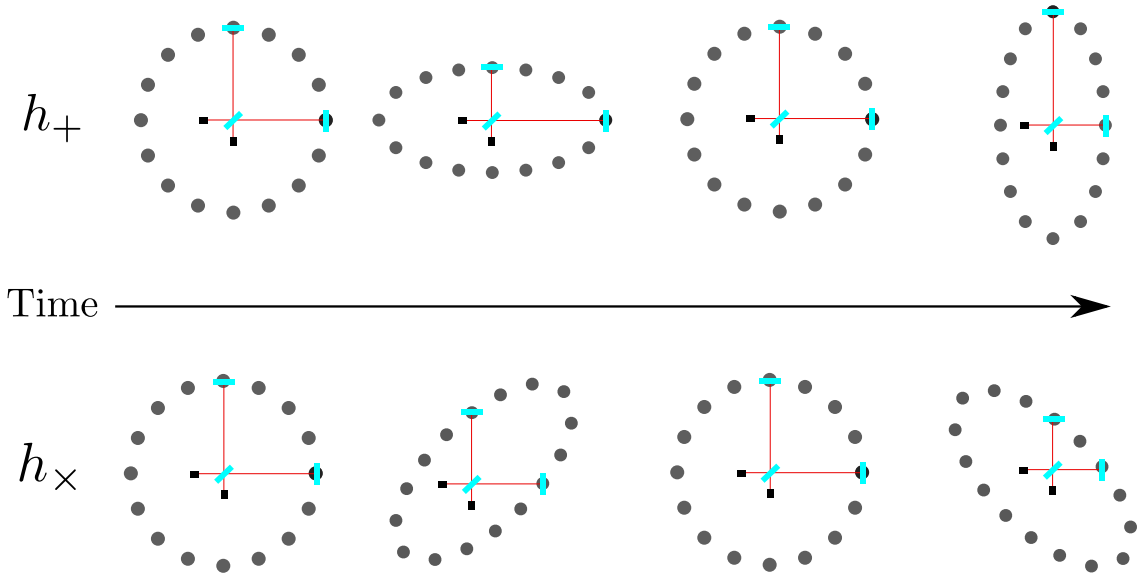


Figure 1.1: Shows how the plus and cross polarisation's affect a ring of test particles. This assumes the wave is travelling out of the page and the effects have been greatly exaggerated. This also shows an example of how this effects the test masses of an interferometer. This will be described in more detail in Sec. 1.3.

Generating gravitational waves

To generate gravitational waves we go back to Eq. 1.6 where we include the stress-energy term on the right hand side. Following the derivation in [flanagan2005BasicsGravitational], one can find that the gravitational wave amplitude is related to the second moment of the mass distribution. The second moment of the mass distribution $I_{\mu\nu}$ is defined as,

$$I_{\mu\nu}(t) = \int \rho(t, \mathbf{x}) x^\mu x^\nu d^3x, \quad (1.10)$$

where ρ is the mass density, and x_i and x_j are the coordinates [flanagan2005BasicsGravitational]. This is the quadrupole moment tensor without the trace subtracted. The gravitational wave amplitude is then defined as,

$$h_{\mu\nu} = \frac{2}{r} \frac{d^2 I_{\mu\nu}(t-r)}{dt^2}. \quad (1.11)$$

This has a slight modification in the TT gauge, see [flanagan2005BasicsGravitational], however, has the same relationship between the mass quadrupole and the GW amplitude. This shows that for a GW to be generated, the second derivative of the mass quadrupole moment is needed. A mass quadrupole moment only exists when the mass distribution is not spherically symmetric. Therefore, a mass which is asymmetric and accelerating will produce a GW.

Systems which will produce detectable GWs are generally rapidly rotating high mass systems which have some asymmetry around their rotation axis. The sources of these GW will be

described in the following section.

1.2 Sources and signals

There are many potential sources for GW. The expected sources can be split into 3 general categories based on their signal type: Transient, Stochastic and CWs. These categories are chosen based on the length of the signal and how well modelled the signal is. Fig. 1.2 shows an example of each of the signals as what category they are a part of. In the sections that follow,

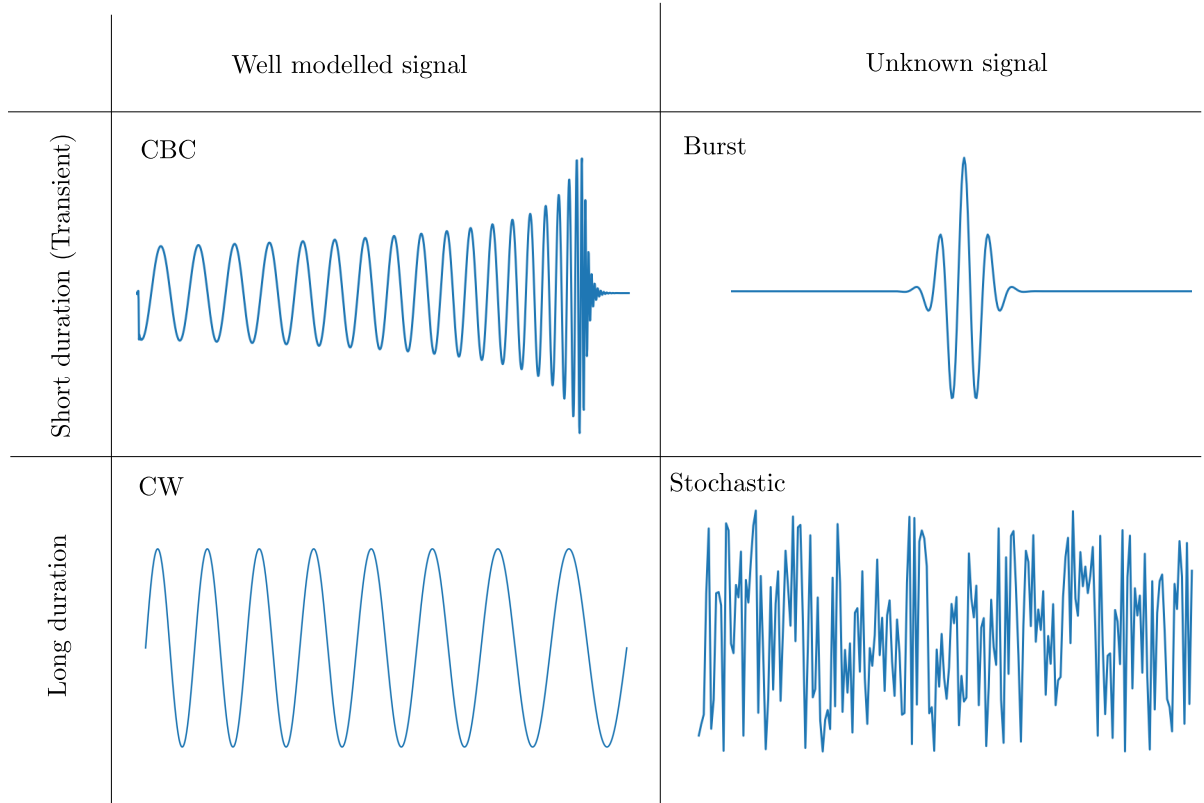


Figure 1.2: Each GW signal type can be categorised based on its signal length and how well the signal is modelled. Transient signals which are short duration, include both well modelled CBC signals and unknown Burst signals. Long duration signals include well modelled CW signals and unknown Stochastic signals.

I will give an overview of the potential sources of each of these signal categories and their wave-forms.

1.2.1 Transient

Transient sources of gravitational waves give short duration signals which are observable from milliseconds to tens of seconds depending on the source. Some of these sources will emit signals for a much longer time, however these are at a lower frequency and lower amplitude and not

observable by current ground based detectors. Transient signals can be well modelled as in CBCs or can have an unknown signal type such as burst signals.

Compact Binary Coalescence

CBCs originate from two compact objects which are gravitationally bound and eventually collide. Dependent on the masses of the two objects, the gravitational waves generated by the system can be detected by ground based detector such as LIGO [aasi2015AdvancedLIGO] and Virgo [acernese2015AdvancedVirgo]. In fact, the only detections to date have been originated from this source which include [abbott2016ObservationGravitational, abbott2017GW170814ThreeD, abbott2017GW170817Observation].

The compact objects referred to here refer to either black holes or neutron stars. There are generally 3 types of CBC source: BBH, BNS and neutron star black hole (NSBH). The general structure of the waveform follows a ‘chirp’ where the GW frequency increases with time until merger. An example of this is shown in Fig. ?? . For higher mass systems such as BBH these signals are detectable by ground based detectors for < 1 s. For lower mass systems such as BNS they can be detected for longer periods $\mathcal{O}(10)$ s.

The wave-form of a CBC signal is generally split into three separate components: the inspiral, the merger and ring-down []. **JOE: rewrite this paragraph** The inspiral is when the two compact objects are orbiting each other. As they lose energy to gravitational waves, the radius of the orbit decreases and therefore the frequency increases. The merger is the period when the two objects begin to join to become a single object. The ring-down is the GW emitted of the merged object. The joint object can oscillate whilst it settles into its final state.

In systems which have a neutron star, during the inspiral when the objects are close, the neutron star will begin to deform due to the strong gravity []. This becomes useful as it will affect the generated waveform and can help determine the equation of state (EOS) for the dense matter in a neutron star []. BNS systems also offer a way to observe objects in multiple different channels, or what is known as multi-messenger astronomy. This is where the object can be viewed in the electromagnetic (EM) spectrum as well as in gravitational waves. This offers much in the field of astronomy as it can aid in the calculation of the Hubble constant [].

Black hole binaries have various formation channels, i.e. there are different ways in which a compact binary system can form []. These formation channels include: two high mass binary stars which collapse, two separate objects capture each other in their gravitational fields and begin to orbit etc.. However, exactly how common these are or if there are dominant ways in which they form is unknown. With many observations of CBC it should be possible to discover which of these, if any, is dominant.

Burst

Burst sources are also short duration however, are un-modelled or difficult to model. This means that the exact wave-form of the signal is unknown. There are a few possible reasons for the lack in knowledge of the waveform: the physics of the system is too complicated to model in a reasonable amount of time or there is no model of the source. As there is no model to generate wave-forms, burst searches cannot use matched filtering as in CBC searches. Rather burst searches look for short bursts in power which is coherent between multiple detectors [cornish2015BayeswaveBayesian, klimenko2008CoherentMethod].

There are a number of systems which could potentially emit a short duration burst signals. These include core collapse supernovae [ott2008GravitationalWave], gamma ray bursts (GRBs) [aasi2014SearchGravitational], cosmic strings [damour2005GravitationalRadiation] and other unknown sources. Detecting GW from one of these sources could offer more insight into the processes inside hostile environments. Burst searches are sensitive to almost any signal which is coherent between detectors. This allows them to be used for searches for CBC also.

1.2.2 Stochastic

The stochastic background has no model for any source, however, is expected to be a persistent source of GW in the background of the detector. The stochastic background is the incoherent sum of many unresolved GW signals. The source of these signals can be anything from cosmological sources such as cosmic strings to CBC signals. These signals can be thought of as the GW analog of the cosmic microwave background (CMB). Where the signal is assumed to be isotropic such that it can be observed at any point on the sky [christensen2018StochasticGravitational]. As the stochastic background is essentially just noise, it is not possible to detect with a single detector [christensen2018StochasticGravitational]. Rather, searches for the stochastic background correlate signals between multiple detectors [romano2019SearchesStochastic, christensen2018]. When detected, these signals may be able to offer insights into the early universe and its formation.

1.2.3 Continuous waves

CWs are long duration signals which can be well modelled. The signals last for times greater than the observation runs of the detector and in general have a fixed or slowly varying frequency. There are a number of potential sources of CWs including CBC long before their merger. Long before a CBC merger, the two compact objects are orbiting each other whilst slowly in-spiralling. This will emit a long duration, almost fixed frequency sinusoid as its signal. This signal however, is at lower frequency than ground based detectors can detect, therefore space based detectors such as laser interferometer space antenna (LISA) [] are expected to observe this type of CW.

The primary source for many CWs searches is for rapidly rotating neutron stars. Neutron stars originate when the remnant of a massive star collapses, they are objects with incredibly high density and are highly magnetised. They have masses around $1.4 - 2 M_{\odot}$ contained in a star with radius of ~ 10 km with magnetic fields ranging from $10^8 - 10^{15}$ G [konar2017MagneticFields]. Despite many observations in the electromagnetic spectrum and a large amount of research, these objects are not well understood. A key part of neutron stars which is not understood is the EOS where a review of the current understanding can be found in [lattimer2016EquationState]. The EOS relates quantities such as the pressure, temperature and volume of a neutron star and dictates how the neutron star matter behaves. Observations of GW from neutron stars can place limits on the EOS of this matter. These observations have already been made in the form of BNS mergers [abbott2017GW170817Observation]. However, independent observations of rapidly rotating neutron stars can add to this understanding.

For a neutron star to emit a gravitational wave it needs to have some asymmetry in its mass distribution around its rotation axis, this follows from Eq. 1.11. There are a number of different mechanisms which could cause this and emit GW, some of these are reviewed in [glampedakis2017GravitationalWaves, riles2017RecentSearches, haskell2015DetectingGravitationalWaves, lasky2015GravitationalWaves]. Here I will summarise two main theories: Neutron star mountains and r-mode instabilities.

Mountains

One of more likely mechanisms for detectable GW emission from neutron stars is from ‘mountains’ on the surface of the star. These are permanent deformations on the surface which are non axisymmetric, i.e. the deformation is not symmetric around the rotation axis. Fig. 1.3 shows an example of a deformation.

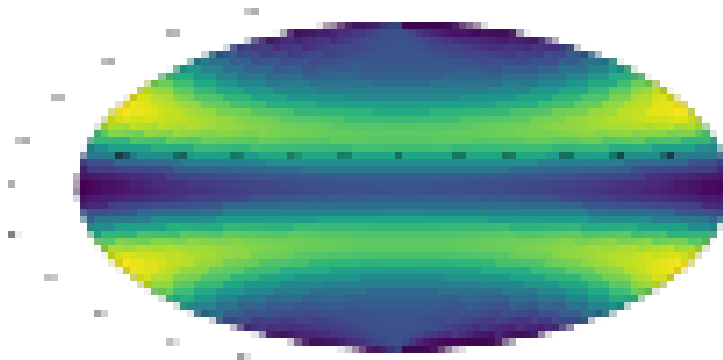


Figure 1.3: star mountain

This deformation or asymmetry can be quantified by the ellipticity ϵ of the neutron star. This

is defined using the principal moments of inertia,

$$\varepsilon = \frac{I_{xx} - I_{yy}}{I_{zz}}, \quad (1.12)$$

where I_{zz}, I_{xx}, I_{yy} are the principal moment of inertia. This is when the star is rotating around the z axis so I_{zz} is along the rotation axis.

There are a number of theories which describe the origin of this axisymmetry. If the pulsar is in a binary system and accreting material from its companion star, the material can be funnelled towards the magnetic poles by the magnetic field, thereby causing a hot spot [haskell2015DetectingGravitational]. This ‘hot spot’ could cause a deformation on the surface of the star which is not axi-symmetric. The magnetic stresses from strong magnetic fields within the star, could potentially also cause non axi-symmetric deformations to the star. Finally the spin down of the pulsar itself could cause stresses in the crust of the star until the point of breaking, its then after this break which could leave a distortion in the crust [becker2009NeutronStars]. More details on the signal waveform of this type of GW and methods to search for it will be explained in Sec. 2.

Neutron star oscillations

There are a number of oscillation modes within a star such as f-modes, p-modes and r-modes [becker2009NeutronStars]. Each of these waves are oscillation in the star similar to oscillations in the earth which are used for seismology. The difference between the different modes are the restoring force bringing the perturbed state back to equilibrium. For example, f-modes use gravity as the restoring force where the oscillations happen in the crust of the star. The more promising of these for gravitational wave emission and detection is the r-mode [lasky2015GravitationalWaves]. These are oscillations in the superfluid neutron part of the star, where the restoring force is the Coriolis effect from the rotation of the star. Fig. ?? shows an highly exaggerated view of a neutron star with an oscillation mode travelling in each direction. If these modes are excited, then they will emit gravitational radiation which carries angular momentum []. For the mode travelling in a clockwise direction, this angular momentum is positive and the mode travelling anti-clockwise the angular momentum is negative. This damps the modes and the magnitude of the perturbation decreases making them extremely difficult to detect. Now if the neutron star is rotating, this can lead to an effect called the Chandrasekhar, Friedman and Schutz (CFS) instability [chanrasakar, Freidmann]. As the rotation speed of the neutron star increases, there are two different effects on the modes travelling in opposite directions. For the mode travelling clockwise with the stars rotation, the mode will appear to be travelling faster, therefore, will emit more GW taking away angular momentum. This means that this mode will be damped more rapidly. The interesting affect is for the mode travelling anti-clockwise, opposite to the stars rotation. At a certain rotation rate, the mode will be ‘frozen’

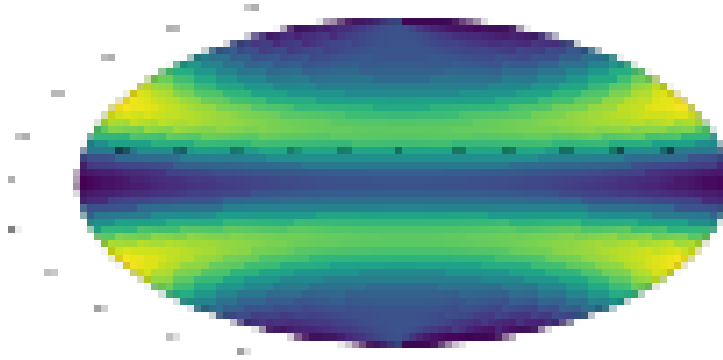


Figure 1.4: Neutron star rotation CFS

from the observers perspective and not GW will be emitted. As the rotation rate increases further, the mode will appear to travel clockwise. Here it is key to remember that this mode had negative angular momentum. As the mode rotates it emits positive angular momentum, which is then subtracted from the modes negative angular momentum. This effect causes the amplitude of the oscillation to grow and therefore become unstable. Therefore, a neutron star is unstable to GW emission if it is rotating sufficiently fast [lasky2015GravitationalWaves]. For a more detailed view on how r-modes generate GW see [owen1998GravitationalWaves]

1.3 Detectors

The theory mentioned above and the indirect detection of gravitational waves from the Hulse-Taylor binary pulsar system left little doubt as to whether GW existed. The real challenge was to design an instrument which could directly detect gravitational waves. There were a number of different methods for the design of the instrument which includes: resonant bar detectors, both ground based and space based interferometers, pulsar timing arrays and Cosmic microwave background detectors. Resonant bar detectors were initially designed and built by Joseph Weber [weber1966ObservationThermal]. These are large cylinders of metal which should resonate as a gravitational wave passes by. There are a few different designs of this type of detector, including an omni-directional design [dewaard2003MiniGRAILFirst]. The majority of these detectors are no longer operational. Pulsar timing arrays aim to use the accurate arrival time of pulses from millisecond pulsars to measure GW [hobbs2017GravitationalWave]. As a GW passes between the pulsar and the observer, the arrival time of the pulses should change. Whilst a detection has not been made with this method, these methods are still in use. Cosmic microwave background detectors aimed to look for evidence of gravitational waves in the polarisation's of the CMB [ade2018ConstraintsPrimordial]. These use a range of detectors to look at the CMB however, are yet to confirm a detection of a GW signal. The most commonly known design of a GW detector is the ground based interferometer, these

made the first detection of GW in 2015 [[abbott2016ObservationGravitational](#)]. These are the focus of this section as the analysis that will follow uses data from the LIGO detectors in the USA [[abbott2009LIGOLaser](#), [aasi2015AdvancedLIGO](#)] and Virgo detector in Italy [[acernese2015AdvancedVirgo](#), [acernese2008StatusVirgo](#)].

1.3.1 Laser Interferometers

Laser interferometers use the interference of light to measure a length with high precision. The majority of this section will focus on ground based interferometers such as LIGO and Virgo [[aasi2015AdvancedLIGO](#), [acernese2015AdvancedVirgo](#)]. A simple design of an interferometer is shown in Fig. 1.5. Here it shows how the laser is equally split into two, each of these beams is sent down perpendicular arms of the detector which are Fabry-Perot cavities []. The light then returns to the beam splitter where the two beams are combined and sent to a photo-detector. At the output, there is an interference pattern between the two beams. If the length of one of the arms is changed then the interference pattern will change as the phase of one beam changes with respect to the other. The Fabry-Perot cavity increases the amount of time light stays within the arms and therefore increases the detectors effective arm length. This helps to increase the sensitivity of the detector.

This can be used in gravitational wave detection as the mirrors at the end of each arm of the interferometer can be treated as ‘free’ test masses. Fig. 1.5 shows the effect of a GW in free test masses. In the interferometer, this affect essentially changes the relative lengths of the two arms. The change in the interference pattern with time is then related to the GW. Actual ground based GW detectors such as LIGO [[abbott2009LIGOLaser](#)] and Virgo [[acernese2015AdvancedVirgo](#)] are much more complicated than described above. They use many techniques to reduce affects on the detector which are not astrophysical. Some of these effects are listed in Sec. 1.3.1

Detector response

An important factor to know when using detector data to search for astrophysical signals is the detectors response. This measures how sensitive the detector is to different locations on the sky. An example of the antenna response for LIGO is in Fig. 1.6 where the detectors arms lie on the x and y axis of the image. This is clear when thinking about how a gravitational wave affects the test masses. As the GW is transverse to its propagation, when the detector is face on to the source, there will be a maximum change in the arm lengths and therefore a maximum sensitivity. In the same way the sensitivity will be at a minimum when edge to the source.

Noise sources

To increase the sensitivity of the LIGO detectors, a good understanding of the noise is needed. This involves understanding what causes certain features in the detector, and how to reduce the

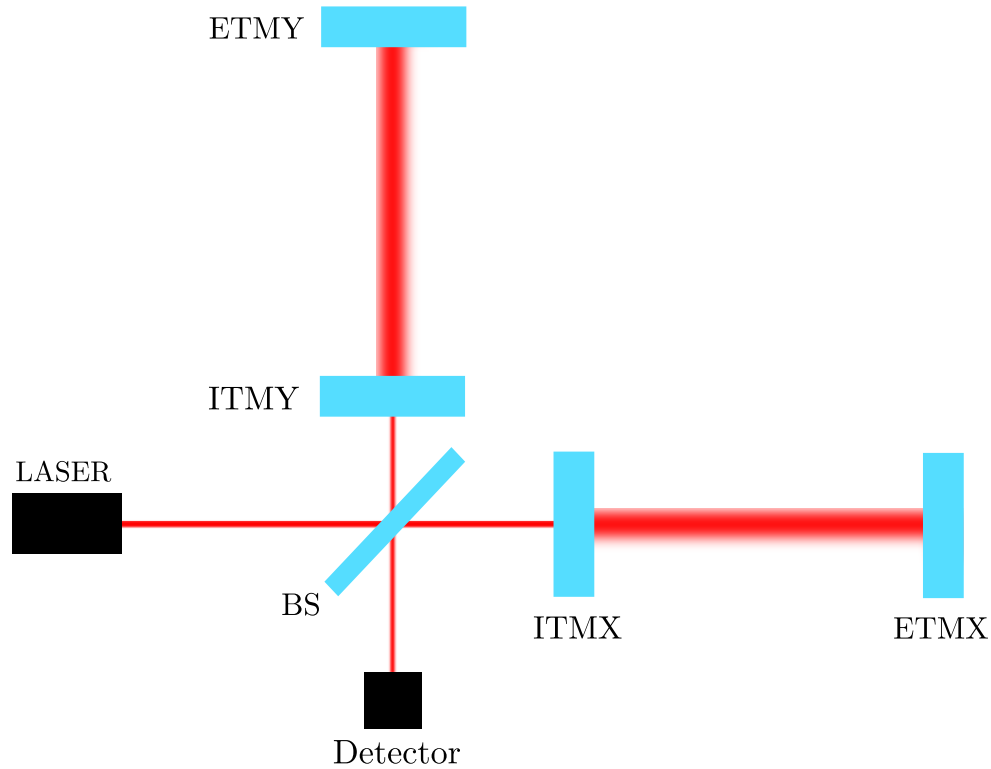


Figure 1.5: This figure shows a basic interferometer. ETMY and ETMX refer to the external test masses, which are just mirrors at the end of the interferometer arms. ITMY and ITMX refer to the internal test masses, these create a cavity in the interferometers arms which can build up laser power. BS is the beam splitter which splits the Laser beam equally to each arm, this then recombines the beams back to the detector.

affect of those. There are many sources of noise in the detector which need to be reduced to be able to detect GW. Some of these noise sources and their affect on the detectors sensitivity are all shown in Fig. 1.7 from [aasi2015AdvancedLIGO]. Here I will summarise some of the limiting sources and also sources which become useful for understanding later sections.

Seismic noise This originates from the seismic activity of the earth and effects the lower frequencies. This can be earthquakes or ocean waves. Seismic waves cause the mirrors to oscillate and induce a change in the length of the arm. This is reduced by having multi stage suspensions in the detectors.

Coating noise This is in general due to two main factors, the thermal noise of the coating and brownian noise. The Brownian noise is from the mechanical dissipation in the coating and the thermal noise is due to thermal dissipation. The Brownian noise is the dominant factor

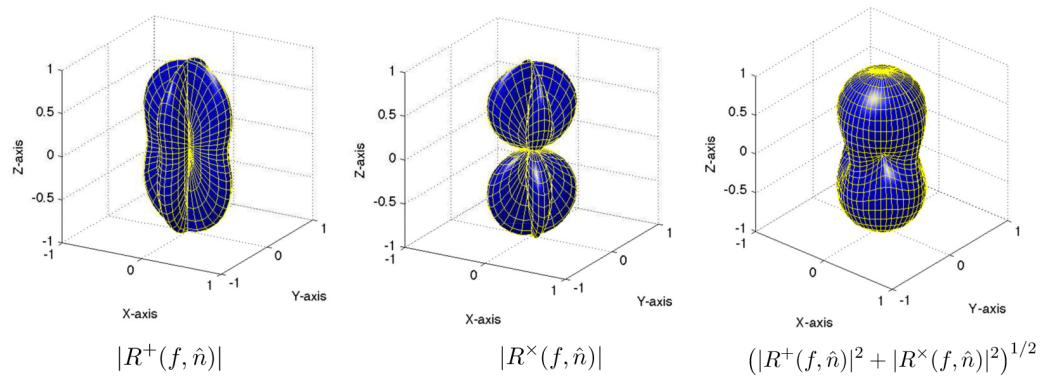


Figure 1.6: The antenna response is shown as in [romano2019SearchesStochastic] for the plus, cross and average polarisation's. The detectors arms lie on the x and y axis in the above plots.

as shown in Fig. 1.7. This is reduced by using different coatings on the mirrors.

Quantum noise Quantum noise is a fundamental limit due to the statistical uncertainty of counting photons. This limits the sensitivity at many frequencies. There are methods to reduce this include squeezing of the light [].

Electronics noise Whilst this is not shown in Fig. 1.7 as it does not affect the noise power spectral density (PSD). These have a different effect on the detector. This is generated by the digital and analog electronics that are used to measure the signal.

There are also many other sources of noise in the detector which I have not listed.

1. Laser interferometers
2. located US (LIGOs) Virgo
3. detect the tidal deformations
4. mention instrumental lines for later section
5. sensitivity curves
6. future detectors
7. antenna response
- 8.

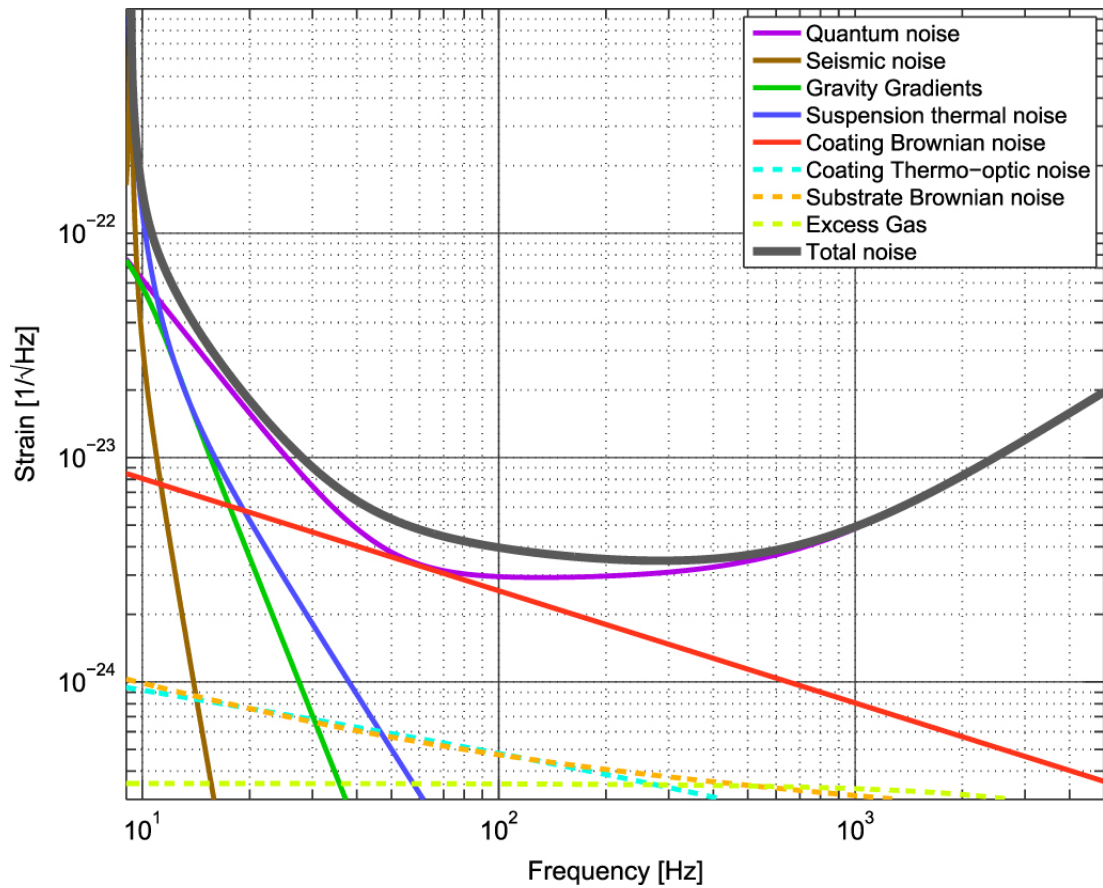


Figure 1.7: The different noise sources affect the sensitivity of the LIGO detectors at different frequencies. This shows the various sources how they affect the noise curve [aasi2015AdvancedLIGO].

1.4 Probability and Bayes Theorem

A key part in understanding the different methods to search for GW or any data analysis, is understanding probability and statistics. This gives understanding of the random processes underlying all measured quantities. Whilst there are generally two approaches to statistics: Frequentist and Bayesian, here I will focus on the Bayesian approach.

1.4.1 Basic probability

Initially I will define some basic concepts of probability. We can define the probability of some event A as $p(A)$ where probabilities follow $0 \leq p(A) \leq 1$ and some other event B which has a probability $p(B)$ and follows $0 \leq p(B) \leq 1$.

Union A union is the probability of either event A happening or event B happening. This is written as, $p(A \cup B)$.

Intersection An intersection is then the probability that both event A and an event B happens. This is written as $p(A \cap B)$.

Independent and dependent Events If the events A and B are independent, i.e. the event A does not affect the outcome of event B , then,

$$p(A \cap B) = p(A)p(B). \quad (1.13)$$

However, if the event A is dependent on event B , i.e. the event A affects event B or vice versa, then the joint probability of both events is,

$$p(A \cap B) = p(A)p(B | A) = p(B)p(A | B). \quad (1.14)$$

Here $p(B | A)$ means the probability of event B happening given that event A has happened.

Conditional probability Conditional probability arises from situations where the outcome of one event will affect the outcome of future events. The definition of this arises from the the dependent events defined above in Eq. 1.14,

$$p(A | B) = \frac{p(A \cap B)}{p(B)}. \quad (1.15)$$

Bayes Theorem Bayes theorem can then be defined using conditional probabilities. i.e we can use

$$p(A | B) = \frac{p(A \cap B)}{p(B)} \quad \text{and} \quad p(B | A) = \frac{p(A \cap B)}{p(A)} \quad (1.16)$$

such that then,

$$p(B)p(A | B) = p(A)p(B | A) \quad (1.17)$$

and this is rearranged to Bayes theorem,

$$p(A | B) = \frac{p(A)p(B | A)}{p(B)} \quad (1.18)$$

1.4.2 Bayesian Inference

We can take Bayes theorem from Sec. 1.4.1 and apply it to a problem which involves inferring some parameters from some model. Here we can relabel the events A and B with the data \mathbf{d} and the parameters $\boldsymbol{\theta}$ of some model I . Eq. 1.4.2 then becomes,

$$p(\boldsymbol{\theta} | \mathbf{d}, I) = \frac{p(\boldsymbol{\theta}, I)p(\mathbf{d} | \boldsymbol{\theta}, I)}{p(\mathbf{d} | I)} \quad (1.19)$$

where each of the components are assigned names: $p(\boldsymbol{\theta} | \mathbf{d})$ is the posterior distribution, $p(\boldsymbol{\theta})$ is the prior distribution, $p(\mathbf{d} | \boldsymbol{\theta})$ is the likelihood and $p(\mathbf{d})$ is the evidence.

Posterior The posterior distribution describes the probability of a parameter $\boldsymbol{\theta}$ in some model

I given some data d . For many problems this is the distribution which is most useful as it informs you the most likely set of parameters of your model given some observation.

Prior The Prior distribution is a key part of Bayesian statistics. This distribution describes any information which you have prior to the observation. This is a distribution defined by the user, where you define a distribution of the parameters based on what you expect to be true.

Likelihood The likelihood is where the observation is included in the calculation. This tells you how probable it is to get the observed data d given the model I with the set of parameters θ .

Evidence The evidence is the probability of the data itself given the choice of model. This is found by integrating over all possible values of θ weighting them by our prior belief of that value of θ .

$$p(d | I) = \int p(\theta, I) p(d | \theta, I) d\theta \quad (1.20)$$

Bayes theorem then gives a description of the probability distribution of some parameters given some observation. Often when using Bayesian statistics the aim is to find posterior distribution of parameters. There are very few cases where this can be calculated analytically, therefore, numerical methods are almost always used to find the posterior. This can be difficult to calculate numerically as in many problems the parameters space has many dimensions. Calculating the evidence in Eq. 1.20 is especially difficult as it is an integral over all possible parameters. There is however, a way around having to calculate this. For any given model I , the evidence $p(d | I)$ is the same for any set of parameters θ in Eq. 1.19. The evidence is then just a normalisation factor for the posterior distribution. When different models are not being compared, and we assume the model I to be true, we no longer need to calculate the evidence. The posterior distribution can then be found by sampling,

$$p(\theta | d, I) \propto p(\theta, I) p(d | \theta, I). \quad (1.21)$$

To find this posterior you could then calculate the value for every point in parameter space. This however, is very computationally expensive and often the posterior distribution is located in a small area in parameter space. Therefore, the majority of the time is sampling a space where the posterior is close to zero, and this is not particularly useful. A different method titled Markov-Chain Monte Carlo (MCMC) was then proposed [metropolis1953EquationState], more information on this can be found in [vanravenswaaij2018SimpleIntroduction, sharma2017MarkovChain]. This randomly jumps around the parameter space and accepts jumps which have a higher posterior value. If the posterior value is lower than the previous step then the jump is accepted with some probability. This means that there is a random chance that a value lower than the current is

accepted. This does not waste time calculating areas in parameter space of low posterior values and therefore, efficiently builds an estimate of the posterior distribution.

In certain situations it can be useful to calculate the evidence in Eq.1.20. For example, if there are two different models which could represent the data, the evidence can be used to determine which of the two models is more likely. This is known as a bayes factor where two models I_1 and I_2 are compares and is defined as,

$$B = \frac{p(\mathbf{d} | I_1)}{p(\mathbf{d} | I_2)} \quad (1.22)$$

This then requires the calculation of the evidence. To estimate the evidence efficiently a method known as Nested sampling [**skilling2006NestedSampling**, **speagle2019DynestyDynamic**] is used.

The methods described above then provide a way to estimate parameters of a model given some data. Also this provides a way to compare different models given some observation. In following sections the methods described above are used to estimate various parameters.

Chapter 2

Searching for continuous gravitational waves

Continuous gravitational waves have particular challenges when it comes to their detection. Whilst I have described the potential sources of the signal and its signal type in Sec. ??, I will go into more detail on the signal and search algorithms here.

2.0.1 Continuous signal model

The model of a GW signal from a pulsar is relatively simple, it is a quasi-sinusoidal signal. This means that the signal is a sinusoid with a slowly varying frequency. One reason for the slow variance in the frequency is due to the energy loss to GW as the pulsar spins down. Here the signal is modelled to originate from an isolated triaxial neutron star rotating around a principal axis. The parameters of each pulsar can be split into two sections: the Doppler components (α, δ, f) and its amplitude components $(\psi, \phi_0, \iota, h_0, \theta)$. This ignores any orbital parameters which would be present if the star was in a binary systems and higher order frequency derivatives. They are defined as follows: the sky positions α and δ refer to the right ascension and declination. f refers to the source frequency and its derivatives. ψ and ϕ_0 and h_0 are the GW polarisation, initial phase and amplitude respectively. ι is the inclination angle which is how much the source is tilted relative to the observer. θ is the ‘wobble angle’ or the angle between the rotation axis and the symmetry axis of the neutron star.

The definition of the GW from a neutron star here follows that in [Schutz1998DataDetection]. The amplitude of the GW can be defined as,

$$h(t) = F_+(t)(h_{1+}(t) + h_{2+}(t)) + F_\times(t)(h_{1\times}(t) + h_{2\times}(t)), \quad (2.1)$$

where h_{1+} and h_{2+} are the two components of the plus polarisation and $h_{1\times}$ and $h_{2\times}$ are the two

components of the cross polarisation functions. These are defined by,

$$\begin{aligned}
 h_{1+}(t) &= \frac{1}{8}h_0 \sin 2\theta \sin 2\iota \cos \Phi(t) \\
 h_{2+}(t) &= \frac{1}{2}h_0 \sin^2 \theta (1 + \cos^2 \iota) \cos 2\Phi(t) \\
 h_{1\times}(t) &= \frac{1}{4}h_0 \sin 2\theta \sin \iota \sin \Phi(t) \\
 h_{2\times}(t) &= h_0 \sin^2 \theta \cos \iota \sin 2\Phi(t).
 \end{aligned} \tag{2.2}$$

The plus and cross polarised components then depend on the GW amplitude h_0 , the wobble angle θ and the inclination angle of the source ι . The phase of the wave $\Phi(t)$ can be defined as,

$$\Phi(t) = \phi_0 + \phi_f(t, f) + \phi_{\text{sky}}(t, \alpha, \delta). \tag{2.3}$$

This consists of an initial phase ϕ_0 , a component of the phase which describes how the frequency of the source changes with time $\phi_f(t)$ and a component which accounts for the Doppler shift of the detectors as the earth rotates and orbits the sun $\phi_{\text{sky}}(t, \alpha, \delta)$. A full derivation of this can be found in [Schutz1998DataDetection] where each of these terms are expanded. The amplitudes h_0 in Eq. 2.2 are defined by,

$$h_0 = \frac{16\pi^2 G \epsilon I f^2}{c^4 r}, \tag{2.4}$$

where G is the gravitational constant, c is the speed of light, ϵ is the ellipticity of the star, f is the sum of the frequency of rotation of the star and the frequency of precession, r is the distance to the star and I_{zz} is the moment of inertia with respect to the rotation axis z . The ellipticity of the star ϵ is a measure of the distortion of the star around its rotation axis and is defined by,

$$\epsilon = \frac{I_{xx} - I_{yy}}{I_{zz}}, \tag{2.5}$$

where I_{xx} , I_{yy} and I_{zz} are the moments of inertia for each axis.

In Eq. 2.1, $F_+(t)$ and $F_\times(t)$ are the antenna pattern functions of the detector. These describe how sensitive a detector is to a particular location on the sky at any given time. The amplitude of the signal will vary dependent on the orientation and location of the detector relative to the source. This is described in Sec. 1.3 and the response to sky location is shown in Fig. 1.6. These components are defined in [Schutz1998DataDetection] as,

$$\begin{aligned}
 F_+(t) &= \sin \zeta [a(t) \cos(2\psi) + b(t) \sin(2\psi)], \\
 F_\times(t) &= \sin \zeta [b(t) \cos(2\psi) - a(t) \sin(2\psi)],
 \end{aligned} \tag{2.6}$$

where ζ is the angle between the arms of the detectors, ψ is the polarisation angle of the GW and $a(t)$ and $b(t)$ are defined in [Schutz1998DataDetection] and relate the sky location to the

orientation of the detector at a given time.

Eqs. 2.1 - 2.6 then describe the amplitude and phase evolution of a signal at a given detector location and orientation.

2.0.2 Signals in data

The data recorded from a detector, $x(t)$, will include the signal model described in Eq. ?? above, but it will be buried in the noise of the detector. If we assume the noise is Gaussian distributed and the noise and signal add linearly, then,

$$x(t) = n(t) + h(t; \mathcal{A}, \lambda), \quad (2.7)$$

where $n(t)$ is the noise, $h(t)$ is the signal and \mathcal{A} and λ refer to the amplitude and Doppler parameters respectively. The optimal signal to noise ratio (SNR) squared of this signal is defined as the scalar product of the signal with itself,

$$\rho^2(0) = (\mathbf{h} | \mathbf{h}) = \sum_X (h^X | h^X), \quad (2.8)$$

where if there is more than one detector the SNR squared for each detector X can be summed [Prix2007]. The scalar product of two time series, $x(t)$ and $y(t)$, is defined by,

$$(x | y) = 4\Re \int_0^\infty \frac{\tilde{x}(f)\tilde{y}^*(f)}{S_n(f)} df, \quad (2.9)$$

where $\tilde{x}(f)$ is the Fourier transform of $x(t)$, $\tilde{y}^*(f)$ is the complex conjugate of the Fourier transform of $y(t)$ and $S_n(f)$ is the single sided noise power spectral density [Prix2007].

2.0.3 Fully-coherent searches

Fully coherent searches are currently the most sensitive searches for continuous sources of gravitational waves. These searches are based on matched filtering which coherently correlate pre-generated waveforms with the data, this is used in [Dupuis2005, Jaranowski1998]. In a simple form the matched filter filters the data h with a template w using the inner product defined in Eq. 2.9.

Continuous wave signals need a large amount of integration time, $\mathcal{O}(\text{years})$, for the signal to become clear within the data, therefore, given that the LIGO detectors sample at 16 kHz, the amount of data to search over is large. Performing the coherent matched filter on this data can take a large amount of time, therefore, as the majority of sources which are searched for have a narrow bandwidth, many searches use techniques to reduce the amount of data. For example, in [Dupuis2005] the data is heterodyned (mixed with some local oscillator), low passed filtered and then down-sampled. This reduced the amount of data to be searched over while not losing

any source information.

Whilst the fully coherent matched filter searches have methods to reduce the computational time for known sources, in all sky searches no parameters of the source are known, therefore, enough templates need to be made to sufficiently cover the large parameter space. This task quickly becomes impossible for coherent matched filtering for an entire observing run due to the amount of time needed. This problem led to the development of semi-coherent searches which will be introduced in the next section.

2.0.4 Semi-coherent searches

Semi-coherent searches offered a solution to searching over large parameters spaces and large amounts of data. The data is split into smaller segments and the analysis is run separately on each of those segments, then each result is combined incoherently. This can greatly reduce the time taken for the analysis depending on the segment length, however, will always come with some loss in sensitivity.

There are many different types of semi-coherent search which use various methods to incoherently combine the coherently analysed results. Many of these use 1800s (30 mins) SFTs as the input to their search as they can be calculated efficiently.

Chapter 3

Searching for continuous gravitational waves using SOAP

The SOAP search is another semi-coherent CW search algorithm that aims to reduce the computational time needed to find a potential signal. The algorithm looks through narrow-banded spectrograms of data to find the most probable track in frequency through it. The basic idea of the search is simple, if we looked at a frequency band in a spectrogram as in Fig. 3.1, we could find every possible randomly wandering track from a starting frequency bin to an end frequency bin. For each of these tracks the sum of the spectrogram power along the track can be found such that for each track there is a single value, Fig. 3.1 shows a histogram of a subset of these values. If there is a signal present in this band then it could be assumed that the signal which gives the highest sum of spectrogram power is the track which is most likely to follow the signals frequency track. In Fig. 3.1 the optimum track in red shows gives a statistic value of ~ 1700 which is far outside the main distribution of summed powers. This follows an injected signal track which demonstrates the idea. Therefore, it can be assumed that if the frequency track with the highest sum of spectrogram power is found, then the corresponding track is most likely for follow a signal. Now as calculating all possible tracks is computationally expensive. Given that in the example in Fig. 3.1, the spectrogram is 180 bins wide M and 400 segments long N and after each segment the track has T possible options to jump to, The total number of possible tracks is T^{M*N} . For this spectrogram this value is 180^{72000} , this is an unreasonable number of tracks to possibly calculate. This is where the Viterbi algorithm [viterbi1967ErrorBounds] is useful as it can efficiently find the track which gives the maximum sum of power. This is explained in the following sections.

The majority of this chapter that follows has been reviewed and published as in [bayley2019Generaliz excluding Sec. 3.10.4 and Sec. 3.9 which is supplementary material.

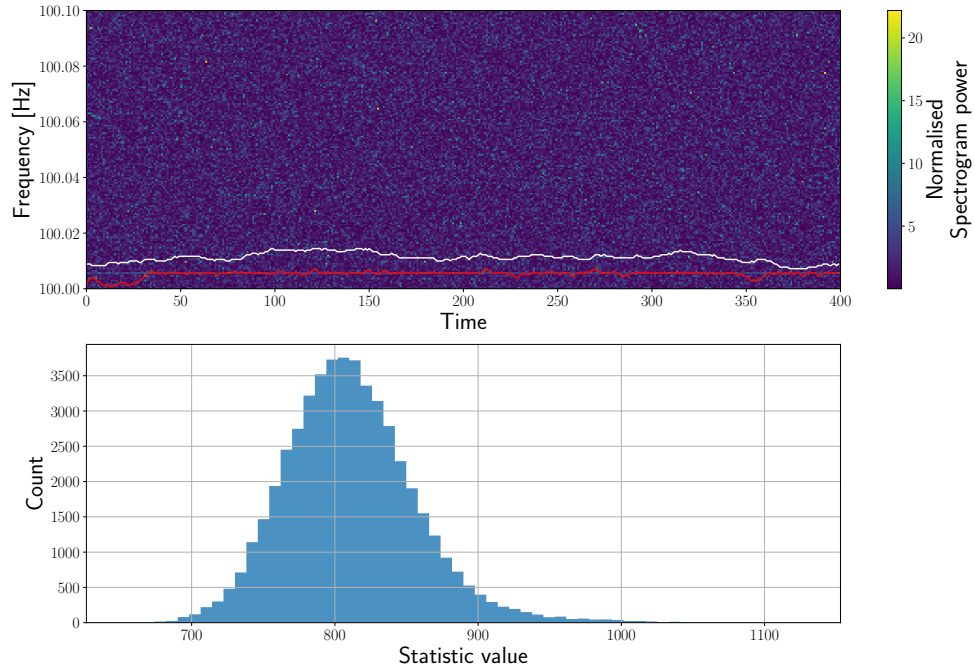


Figure 3.1: This figure shows an example of a spectrogram which is typical of LIGO data which is searched through. Here an instrumental line has been injected at 100.006 Hz. The white track shows a random walk track through this spectrogram whereas the red line shows the track which gives the highest sum of detector power. The second panel shows a histogram of a subset of all paths which can be found through the given spectrogram from start to finish. This is a subset as the total number of paths is too large to calculate. The value of the statistic which comes from the optimal path is ~ 1700 . This is much larger than any of the random tracks in our subset and much larger than the mean of all tracks.

3.1 Introduction

One of the main targets for current ground based GW detectors, including LIGO [**abbott2009LIGO**, **Laser**, **aasi2015AdvancedLIGO**] and Virgo [**acernese2008StatusVirgo**, **acernese2015AdvancedVirgo**], are sources of continuous gravitational waves. These are long-duration, quasi-monochromatic sinusoidal signals that are well-modelled by a Taylor series expansion in the signal phase. A likely source of such signals are rapidly spinning non axisymmetric neutron stars. A number of possible emission mechanisms are outlined in [**prix2009GravitationalWaves**, **owen2009ProbingNeutron**].

These types of GWs are expected to give strain amplitudes that are significantly below the detector’s noise spectral density, and need sensitive search algorithms for detection. The most sensitive method is to use a coherent matched filter which requires knowledge of the waveform beforehand such that it can be coherently correlated with the data. This approach is used in searches for gravitational signals from known pulsars such as [**dupuis2005BayesianEstimation**, **astone2010MethodDetection**, **schutz1998DataAnalysis**, **collaboration2017FirstSearch**, **abbott2019S**]. For broad parameter space searches, where the parameters of the signal are unknown, a large number of template waveforms must be used to sufficiently cover the parameter space. This approach rapidly becomes computationally impractical as the search space grows, so semi-coherent search methods have been developed to deliver the maximum overall sensitivity for a given computational cost. Semi-coherent searches break the data up into sections of either time or frequency and perform a coherent analysis on these sections separately. These intermediate results can then be recombined incoherently in a number of different ways to form the final search result outlined in [**creighton2000SearchingPeriodic**, **abbott2019AllskySearch**] and references therein.

The analysis that we present here is known as SOAP [**ellis2006SnakesPlanea**] and is based on the Viterbi algorithm [**viterbi1967ErrorBounds**]. The algorithm models a process that has a discrete number of states at discrete time steps, and computes the set of states which gives the highest probability (suitably defined) given the data. Our implementation of SOAP is intended as a stand-alone search which is naturally non-parametric and has broad applications to both searches for known signal types and signals which have an unknown frequency evolution. The algorithm works in time-frequency plane, where our ‘states’ are represented by the time and frequency coordinates of a potential signal. We can then find the most probable set of frequencies a possible signal could have, i.e., we can find the most probable track in frequency as a function of time. This is not the first application of the Viterbi algorithm to GW data. Another variant of the algorithm [**suvorova2016HiddenMarkova**] has recently been used, amongst other applications, as part of a CW search to track a pulsar with randomly wandering spin frequency [**sun2018HiddenMarkov**, **suvorova2017HiddenMarkov**, **abbott2017SearchGravitational**, **Abbott:2018hgk**, **sun2018ApplicationHidden**]. We develop an alternative version which is aimed to be applied more generally to search for any long duration signals using just SFTs.

In the next section we will describe the Viterbi algorithm and the basic SOAP implementa-

tion to GW time-frequency data. We then describe additional features to the algorithm, including the use of data from multiple detectors. As well as this we describe methods used to ignore instrumental effects in the data, such as incoherently summing data and a ‘line aware’ statistic. In the final section as well as a test of the computational cost of the search, we show results of a search performed on datasets of increasing complexity: Gaussian noise with no gaps (i.e., contiguous in time), Gaussian noise with gaps simulating real data more accurately, and finally real LIGO data taken during the sixth science run.

3.2 Viterbi algorithm

The Viterbi algorithm is an efficient method for determining the most probable set of states (a single ‘track’ of steps on the time-frequency plane) in a Markov model dependent on data, where the model has a discrete number of states at each step. Rather than computing the probability of every possible track and selecting the most probable, the algorithm maximises this probability after every discrete step. As a result, a partial track which cannot ultimately be the most probable is rejected before the next step is calculated, and only a fraction of all possible tracks need to be computed to find the one that is most probable.

In this work we apply the Viterbi algorithm to a GW strain time-series to find the most probable track of a single variable-frequency signal in the noisy data. We divide the time series into N equal-length and contiguous segments \mathbf{x}_j , defining the set $D \equiv \{\mathbf{x}_j\}$. The ‘states’ in the model correspond to the frequencies a signal could have in each segment. A ‘track’ is a list of such frequencies $\mathbf{v} \equiv \{v_j\}$, where v_j is the frequency in the segment \mathbf{x}_j .

Our objective is to calculate the most probable track given the data, i.e., the track that maximises $p(\mathbf{v} | D)$. Using Bayes theorem, this posterior probability can be written as

$$p(\mathbf{v} | D) = \frac{p(\mathbf{v})p(D | \mathbf{v})}{p(D)}, \quad (3.1)$$

where $p(\mathbf{v})$ is the prior probability of the track, $p(D | \mathbf{v})$ is the likelihood of the track (i.e., the probability of the data given the track) and $p(D)$ is the model evidence (or marginalised likelihood).

The Viterbi algorithm treats the track as the result of a Markovian process, such that the current state depends only on the previous state. It is therefore useful to split the track’s prior into a set of transition probabilities such that

$$\begin{aligned} p(\mathbf{v}) &= p(v_{N-1}, \dots, v_1, v_0) \\ &= p(v_{N-1} | v_{N-2})p(v_{N-2} | v_{N-3}) \dots p(v_1 | v_0)p(v_0) \\ &= p(v_0) \prod_{j=1}^{N-1} p(v_j | v_{j-1}), \end{aligned} \quad (3.2)$$

where $p(v_0)$ is the prior probability that the signal in the first time step has a frequency v_0 and $p(v_j | v_{j-1})$ is the prior ‘transition’ probability for v_j given the frequency at the last step was v_{j-1} .

The noise in each of the segments can be treated as independent, so the likelihood component in Eq. 3.1 can be factorised as

$$p(D | \mathbf{v}) = \prod_{j=0}^{N-1} p(\mathbf{x}_j | v_j), \quad (3.3)$$

where $p(\mathbf{x}_j | v_j)$ is the likelihood of our signal having a frequency v_j in the j th segment.

Using Eqs. 3.1, 3.2 and 3.3, the posterior probability is then

$$p(\mathbf{v} | D) = \frac{p(v_0)p(\mathbf{x}_0 | v_0) \prod_{j=1}^{N-1} p(v_j | v_{j-1})p(\mathbf{x}_j | v_j)}{\sum_S \left\{ p(v_0)p(\mathbf{x}_0 | v_0) \prod_{j=1}^{N-1} p(v_j | v_{j-1})p(\mathbf{x}_j | v_j) \right\}}, \quad (3.4)$$

where in the denominator we must sum over all possible tracks S [Slade2014]. We require the specific track, or set of frequencies, $\hat{\mathbf{v}}$ that maximises the posterior probability. Therefore, as the denominator in Eq. 3.4 is a sum over all possible tracks, the track which maximises the posterior is the same track which maximises the numerator on the right-hand side of Eq. 3.4, i.e.,

$$p(\hat{\mathbf{v}} | D) \propto \max_{\mathbf{v}} \left[p(v_0)p(\mathbf{x}_0 | v_0) \prod_{j=1}^{N-1} p(v_j | v_{j-1})p(\mathbf{x}_j | v_j) \right]. \quad (3.5)$$

This track also maximises the log of the probability and can be written as,

$$\begin{aligned} \log p(\hat{\mathbf{v}} | D) = \max_{\mathbf{v}} & \left\{ \log p(v_0) + \log p(\mathbf{x}_0 | v_0) \right. \\ & \left. \sum_{j=1}^{N-1} \left[\log p(v_j | v_{j-1}) + \log p(\mathbf{x}_j | v_j) \right] \right\} + \text{const.} \end{aligned} \quad (3.6)$$

The Viterbi algorithm finds the most probable track $\hat{\mathbf{v}}$ by calculating the quantities in Eq. 3.6 for each frequency at each time step. In the following sections we explain how this is achieved in practice.

3.3 The transition matrix

An important concept when using the Viterbi algorithm is the ‘transition matrix’ T , which is defined as the matrix that stores the prior log-probabilities $\log p(v_j | v_{j-1})$. These transition probabilities depend only on the size and direction of the transition, and in our case corre-

spond to a jump in frequency when moving from the $(j - 1)$ th to the j th state. It is within the transition matrix that we impose some loose model constraints. For example it is usual in the time-frequency plane for frequencies to only have discrete values (frequency bins) and a track might only be allowed to move by one bin in each time step, restricting it to a UCD transition or ‘jump’ or equivalently setting the size of the first dimension of the transition matrix $n_1 = 3$. We can also impose that the transition probabilities are independent of the current track location in frequency, i.e. $p(v_j | v_{j-1}) = p(v_{j+k} | v_{j+k-1})$. This leads to the transition matrix containing only three numbers, corresponding to the three prior log-probabilities that the track was in the corresponding UCD frequency bin at the previous time step. These numbers are chosen to reflect the prior probability of a frequency deviation in the track and depend on the class of signals that one wishes to detect. For the majority of examples that follow, a symmetric transition matrix is used, i.e. the probability of a transition up a frequency bin is equal to the probability of a transition down a frequency bin. This allows us to parameterise the one dimensional transition matrix with a single value, this value is the ratio of the probability of a transition to the same frequency bin, to either up or down a frequency bin.

In later sections we will consider more complex situations in which the transition matrix describes the prior probability associated with sequences of even earlier transitions (‘memory’) and the case where there are multiple detectors. In these cases the number of dimensions of the transition matrix can grow substantially to account for the extra complexity of the problem.

3.4 Single detector

We will first consider the simple case of a single dataset D , generated by a single gravitational wave detector, and consider only a one-dimensional transition matrix. We will make use of discrete Fourier transforms so that frequencies, and hence the track frequencies, are also discrete. These frequencies will be indexed by k and therefore $v_j \rightarrow v_{j,k} = k(j)\Delta f$ where $\Delta f = 1/T$ is the frequency bin width for a segment of duration T .

The Viterbi algorithm determines the most probable track on the time-frequency plane by calculating the value of Eq. 3.6 for every discrete Fourier frequency, incrementally in time. In other words, at each time segment it finds the most probable earlier track which ends at each particular frequency. On reaching the final segment it can look back to identify the most probable track connecting segment 1 to segment N .

There are two main components to Eq. 3.6: the transition probabilities $p(v_j | v_{j-1})$ and the likelihoods $p(\mathbf{x}_j | v_j)$. The transition probabilities are pre-calculated and stored in a transition matrix according to Sec. 3.3 above. To calculate the likelihood we follow the approach of [bretthorst1988BayesianSpectrum] which gives, under the assumption of a single sinusoidal signal in additive Gaussian noise in data segment \mathbf{x}_j ,

JOE: look at this is more detail, write more derivation, derive for log-odds as well as

likelihood

$$p(\mathbf{x}_j \mid \mathbf{v}_{j,k}, \sigma_{j,k}, I) \propto \exp [C(\mathbf{v}_{j,k})]. \quad (3.7)$$

where $C_{j,k}(\mathbf{v}_{j,k})$ is the Schuster periodogram normalised to the noise variance at frequency $\mathbf{v}_{j,k}$ of segment j . This is equivalent to the log-likelihood, and is defined as

$$C(\mathbf{v}_{j,k}) \equiv C_{j,k} = \frac{1}{\sigma_{j,k}^2} \frac{1}{N_s} \left| \sum_{r=0}^{N_s-1} x_{j,r} e^{i\mathbf{v}_{j,k} t_r} \right|^2, \quad (3.8)$$

where N_s is the number of data points in each segment and t_r is the time corresponding to $x_{j,r}$, the r th sample in the j th data segment. $\sigma_{j,k}^2$ is the noise variance and is calculated as an estimate of the noise PSD in the k th sample and the j th data segment. It is worth noting at this point that it is also possible to write this as a likelihood ratio, and therefore write out detection statistic as a log-odds ratio, however, we will discuss this in more depth in Sec. 3.8. The log-likelihoods of each segment can be calculated at discrete frequencies before running the algorithm by computing the power spectra for each segment from discrete Fourier transforms of the data. In the GW field these standard data forms are known as SFTs.

The Viterbi algorithm records two quantities for each frequency and time bin: The first, $V_{j,k}$, contains the value defined by Eq. 3.6, which is the log-probability of the most probable path ending in position j, k . The second, $A_{j,k}$, is the transition, or ‘jump’, used to achieve the most probable path. The algorithm can be divided into three main sections: initialisation, iteration and identification. These three sections are described in pseudo-code in Alg. 3.1 and a simple demonstration of the algorithm at work is shown in Fig. 3.2.

Initialisation The two parts of Eq. 3.6, $\log p(\mathbf{v}_0)$ and $\log p(\mathbf{x}_0 \mid \mathbf{v}_0)$, must be computed before the main recursive part of the algorithm can start. Therefore, the initialisation section (lines 5–8) in Alg. 3.1 calculates the first column in the lower panel of Fig. 3.2. A priori, there is no preferred initial frequency, so we take the log-prior $\log p(\mathbf{v}_{0,k})$ to be uniform over the complete frequency range. As a result, this does not affect the maximisation for any jump, therefore, can be omitted from the calculation. We then use the pre-calculated log-likelihood values $C_{0,k}$ to fill the track probabilities $V_{0,k}$. There is no previous position to jump from in this case, so the transition probabilities are irrelevant and $A_{0,k}$ are set to zero.

Iteration The main part of the calculation is the sum in Eq. 3.6. Lines 11–16 in Alg. 3.1 calculate the most probable tracks that end at each frequency bin for each segment by using

$$V_{j,k} = \max_i (C_{j,k} + T_i + V_{j-1,k+i}), \quad (3.9)$$

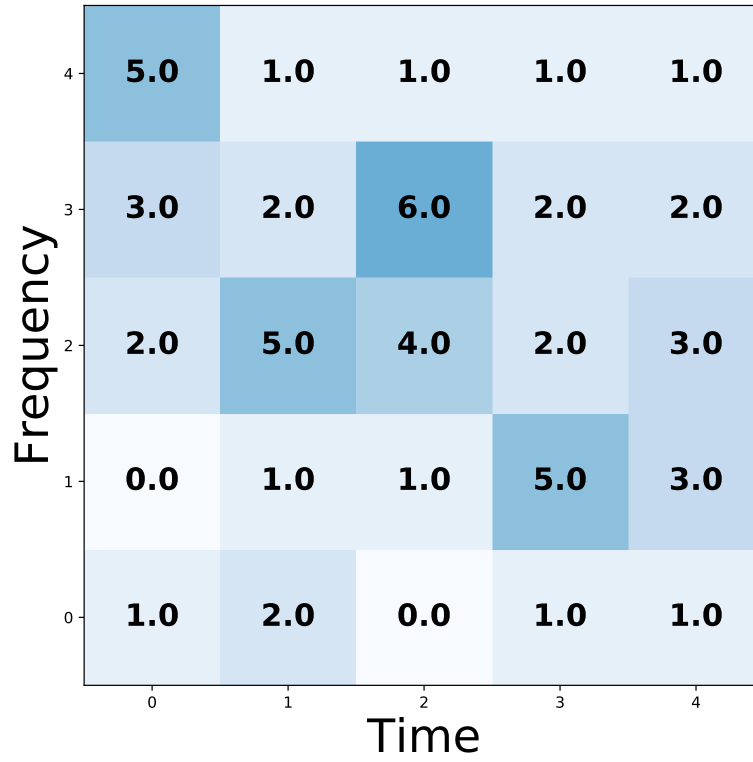
where i is the size and direction of the jump. For example, in Fig. 3.2 columns 1–4 are calculated in order using Eq. 3.9, where it maximises over three possible previous

```

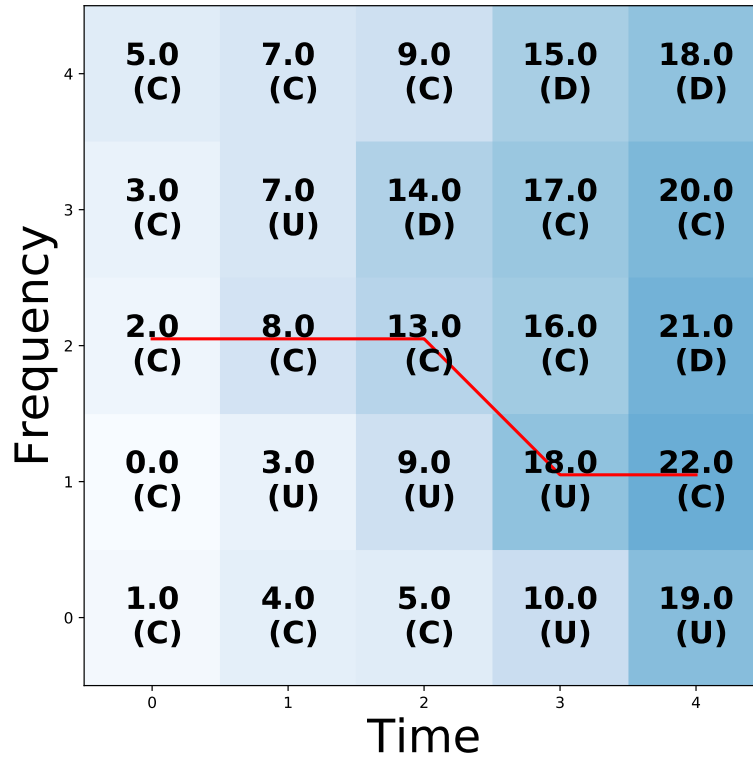
1: Input:  $C, T$  {log-likelihood, transition matrix}
2: Output:  $\hat{\mathbf{v}}, V, A$  {most probable track, track probabilities, jumps}
3:
4: Initialisation
5: for Frequency ( $v_{0,k}$ ),  $k = 0 \rightarrow M - 1$  do
6:    $V_{0,k} = C_{0k}$ 
7:    $A_{0,k} = 0$ 
8: end for
9:
10: Iteration
11: for Segment,  $j = 0 \rightarrow N - 1$  do
12:   for Frequency ( $v_{j,k}$ ),  $k = 0 \rightarrow M - 1$  do
13:      $V_{j,k} = \max_i (C_{j,k} + T_i + V_{j-1,j+i})$ 
14:      $A_{j,k} = \operatorname{argmax}_i (C_{j,k} + T_i + V_{j-1,j+i})$ 
15:   end for
16: end for
17:
18: Identification
19:  $\hat{v}_{N-1} = \operatorname{argmax}_k (V_{N-1,k})$ 
20: for Segment,  $j = N - 1 \rightarrow 0$  do
21:    $\hat{v}_j = \hat{v}_{j+1} + A_{j, v_{j+1}}$ 
22: end for

```

ALGORITHM 3.1: The Viterbi algorithm in pseudo-code. N is the number of segments, M is the number of frequency bins in each segment. Here the maximisations over i run between $\pm(n_1 - 1)/2$ where n_1 is the size of the transition matrix. The values from Eq. 3.6 are stored in V , and the jumps are stored in A . The most probable track is denoted by $\hat{\mathbf{v}}$.



(a) The input data



(b) The log-probabilities, jumps, and most probable path

Figure 3.2: Fig. 3.4c shows the observed data, i.e the log-likelihood values $C_{j,k}$. Fig. 3.2b shows the calculated log-probabilities $V_{j,k}$. $A_{j,k}$ is shown in parentheses, where the UCD components correspond to $i = [-1, 0, 1]$ respectively. The red line shows the path that gives the maximum probability. The transition matrix for the UCD jumps is $[0, 1, 0]$ and corresponds to the unnormalised prior log-probabilities of these jumps occurring.

positions in frequency. These positions are the frequency bins UCD of the current position. The size and direction of the jump, i , which gives the maximum probability is then saved to $A_{j,k}$. These are shown in parentheses below the log-probabilities in Fig. 3.2 where UCD correspond to values of $i = [-1, 0, 1]$ respectively.

Identification The final stage of the algorithm identifies the most probable track. This is done by initially finding the highest log-probability values in the final time segment, $\max_k(V_{N-1,k})$ (line 19 in Alg. 3.1). In the lower panel of Fig. 3.2 this is located at position $j, k = 4, 1$ with $V_{4,1} = 22$. To find the track which corresponds to this, the values in A_{jk} are followed backwards from this position (lines 20–21). For example, in Fig. 3.2 the final position is $j, k = 4, 1$ and $A_{j,k} = \text{Center} = 0$, this means that at the previous segment the most probable track was at position $j, k = 4 - 1, 1 + 0 = 3, 1$. At this time $A_{3,1} = R = 1$, therefore, the next track element is at $j, k = 3 - 1, 1 + 1 = 2, 2$. This then continues until $j = 0$ whereupon these retraced positions constitute the most probable track, highlighted in red in Fig. 3.2.

The most probable track is the one traced backwards from the highest probability final segment frequency position. However, tracks can also be traced back from any of the end-frequency positions, returning the most probable track conditional on a given final position. Such tracks should not be confused with the being equal to the second, third, fourth, etc. most probable tracks. Information regarding the rankings and properties of all possible tracks (excluding the most probable and conditionally most probable tracks) is lost during the maximisation procedures computed at each stage in the algorithm – a necessary consequence of the algorithm’s speed and efficiency.

3.5 Multiple detectors

If there are Q detectors operating simultaneously we have Q sets of data which can be combined appropriately to provide input to the Viterbi search described above. We must also modify the allowed transitions encoded within the transition matrix to take account of the extra prior constraints that are now available.

The received instantaneous frequency of a given astrophysical signal will be nearly the same for all ground-based GW detectors, and our algorithm should be sensitive to tracks that show this consistency in frequency. However there *will* be small differences between the frequencies measured at detectors that are not co-located, due to differential Doppler shifts caused by Earth rotation. As a result the signal could fall in different frequency bins at each detector.

To account for these small differences in signal tracks in each detector, we reference the observed tracks to a third (pseudo) detector located at the centre of the Earth which would be insensitive to Earth spin. The signal frequencies in each real detector are then allowed to

vary within a certain number of frequency bins from the track in the reference detector. In the examples that follow, we only consider the possibilities that the track in each real detector is no more than one frequency bin away from the reference track. We can tune the length of the SFTs to ensure this is a valid assumption. As well as differences in signal frequency, due to antenna patterns and other effects, the measured signal amplitude may differ between the detectors. In the following example we assume that the signal has the same amplitude in each detector, however, in Sec. 3.8 we discuss the case where they differ.

We will now show how the algorithm in Sec. 3.4 can be modified to handle a two-detector network (i.e., $Q = 2$), however any number of detectors can easily be accommodated. In the two detector case the joint probability of two (real) tracks, $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$, and the geocentric track \mathbf{v} , given the data, is

$$p(\mathbf{v}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)} | D^{(1)}, D^{(2)}) \propto p(\mathbf{v}) p(\mathbf{v}^{(1)}, \mathbf{v}^{(2)} | \mathbf{v}) p(D^{(1)} | \mathbf{v}^{(1)}) p(D^{(2)} | \mathbf{v}^{(2)}), \quad (3.10)$$

where $D^{(1)}$ and $D^{(2)}$ represent the data from the two detectors. The main difference between this and that described in Sec. 3.4 is that the track probabilities $V_{j,k}$ are stored for the geocentric pseudo-detector. The main iterative calculation (defined for the single detector case in Eq. 3.9) now becomes

$$V_{j,k} = \max_{i,l,m} (C_{j,k+l}^{(1)} + C_{j,k+m}^{(2)} + T_{i,l,m} + V_{j-1,k+i}), \quad (3.11)$$

where $C^{(1)}$ and $C^{(2)}$ refer to the log-likelihoods in detectors 1 and 2 respectively and the transition matrix T is an $n_1 \times n_2 \times n_3$ matrix, where n_1 dimension refers to the jump from the previous time step, n_2 and n_3 refer to the relative frequency positions in each real detector. The transition matrix is now three-dimensional and holds the prior log-probabilities of $p(\mathbf{v})$ and $p(\mathbf{v}^{(1)}, \mathbf{v}^{(2)} | \mathbf{v})$. We now need to maximise over three indices: i, l and m . The index i refers to the size and direction of the jump at the geocentre (as before). The indices l and m refer to the number of frequency bins by which the two real tracks deviate from the geocentre track. For example, if the most probable track in the geocentred detector is in bin $j, k = 5, 12$ and the values of $i, l, m = 0, -1, 1$, then detector 1 is in position $j, k = 5, 11$ and detector 2 is in position $j, k = 5, 13$ and the geocentred track was in the position $j, k = 4, 12$ at the previous time step. As a result, the track at the geocentre is only affected by Doppler modulations from the Earth's orbit whereas the tracks in the real detectors include Doppler modulations from the Earth's spin.

At every time step the frequency bin position for each real detector is forced to be within n_l or n_m bins of the track in the geocentred detector, where n_l and n_m depend on how much each detector could possibly be Doppler shifted. As mentioned previously, we only consider the case where $n_l = 1$ and $n_m = 1$, allowing the track from each real detector to be at most one frequency bin away from the geocentred track position. While we tune the SFT length to keep this condition for different frequencies, it is also possible to tune the values of n_l and n_m

to get a similar effect. The implementation of the multi-detector algorithm is similar to the single detector case described in Sec. 3.4. However in the single detector case there is only a single variable to be maximised over for each time-frequency bin. This variable is the frequency jump from the position in the previous segment. For the multi-detector case there are at least three variables to be maximised over: the probability of the jump, i , at the geo-centre and the probability of the signal being in the surrounding positions in each of Q real detectors, l, m, \dots . The values of i, l, m, \dots are then saved to $A_{j,k}$ and are ultimately used to reconstruct the most probable consistent tracks in each real detector.

As in Sec. 3.4, there are three main sections: Initialisation, iteration, and the identification. For the multi-detector case each element is modified as follows.

Initialisation The first-row calculation (lines 5–8) in Alg. 3.1, are now modified to additionally maximise over the real detector track positions l and m . For each time-frequency bin the maximum sum of the log-likelihoods is saved together with the frequency locations of the corresponding tracks in the real detectors. The index $i = 0$ is kept constant as there is no previous position.

Iteration To process the subsequent time segments, lines 13–14 in Alg. 3.1 are modified to account for two (or more) detectors. Line 13 of Alg. 3.1 is changed to calculate Eq. 3.11, the log-probability of a track at the geocentre ending in bin j, k given that signal is in the real detector positions of $j, k + l$ and $j, k + m$. Line 14 is then modified so that $A_{j,k}$ stores the jump values, i , and the real detector positions, l and m , which returned the highest probability.

Identification The most probable track is identified in the same way as for the single detector case, first by finding the maximum value in the final time step of $V_{j,k}$ (line 19 in Alg. 3.1). The track at the geocentre can then be found by iteratively following the jump values stored in $A_{j,k}$ back from this position. The track in each of the real detectors is determined by using the values of l and m indices also stored in $A_{j,k}$ to find the relative position of the track in each real detector compared to the geocentre.

This method can be extended to more than two detectors by including additional datasets and expanding the corresponding number dimensions of the maximisation procedures in the iterative steps.

3.6 Memory

In this section we extend the basic Viterbi algorithm to improve its sensitivity to non-stochastic signals where there is some knowledge of its frequency evolution. We do this by including a form of ‘memory’ and this extension applies to both the single and multiple-detector cases.

Rather than considering only the previous step in our decision-making process, we now include the previous $m + 1$ steps and expand the transition matrix to include these values. A memory of $m = 0$ therefore corresponds to the methods described in previous sections. With a non-zero memory the transition matrix can a-priori make certain sequences of jumps more probable and assign different prior probabilities for these jump sequences e.g., ‘up then centre’ may be less preferable to ‘centre then centre’. As a result we can increase the chance of the most probable track matching an expected astrophysical signal. In a single detector search with a memory of $m = 1$, if we only allow UCD transitions, then for every frequency bin we save 3 values. These are proportional to the log-probabilities of a track coming from a UCD bin in the previous time step, where the maximisation is over the corresponding UCD bins two time steps back. Eq. 3.11 then is then modified to,

$$V_{j,k,s} = \max_h (C_{j,k} + T_{s,h} + V_{j-1,k+s,k+s+h}), \quad (3.12)$$

where s and h refer to the UCD jumps at the time step $j - 1$ and $j - 2$ respectively. Similar to the previous two sections, the algorithm is split into three parts: initialisation, iteration, and the track identification:

Initialisation The initialisation process needs to populate the first $m + 1$ steps before the main iteration can start. At the first time step, the elements $V_{0,k,s}$ are set to the log-likelihoods $C_{0,k}$ as in Sec. 3.4. There is no previous time step, so the element s is not relevant. At the second time step, $V_{1,k,s}$ is calculated using Eq. 3.12, where there is no maximisation over h , it is assumed to be 0, or a center jump. As there is no data before $j = 0$, the maximisation at this point will always return the jump which has the largest prior probability, which in this case is a center jump. Therefore, the maximisation returns the same value for all frequency bins and can be set to a center jump.

Iteration For all following time steps the values for each element of $V_{j,k,s}$ in Eq. 3.12 are calculated. This quantity is proportional to the log-probability of the track ending in time-frequency bin j, k , which was in the previous position of $j - 1, k + s$. The corresponding value of h that maximised the log-probability of the track is recorded in $A_{j,k,s}$.

Identification The most probable track is identified in a similar way to the non-memory cases, by finding the highest-valued last element, $V_{N-1,k,s}$. The values of s and h are then followed back to find the most probable track. As an example, let us assume the most probable track finishes in bin $j, k, s = 10, 5, 0$, where the value of m is $A_{10,5,0} = 1 = \text{up}$. The previous position is then $j, k, s = 10 - 1, 5 + s, m = 10 - 1, 5 + 0, 1 = 9, 5, 1$ with a value $A_{9,5,1} = 0 = \text{Center}$, and the next track position is $j, k, s = 9 - 1, 5 + 1, 0 = 8, 6, 0$ etc. The values of j, k along this track describes most probable path.

The number of elements over which one must search increases rapidly with memory length,

and has a strong impact on the computational cost of the analysis. For the single detector Viterbi approach the number of calculations made is $3 \times N \times M$ if we only allow UCD jumps, where N and M are the number of time and frequency bins respectively. When memory is included this increases to $3^{m+1} \times N \times M$.

3.7 Summed input data

In this section a method of incoherently-summing a set of SFTs to increase the SNR of a signal in a segment is outlined. To be more precise, it is actually the log-likelihoods which are summed, i.e. the quantity in Eq. 3.8. We can write the new summed set of data F_j as,

$$F_j = \sum_i^{N_s} C_{i,k} \quad (3.13)$$

where N_s is the number of SFTs to sum together and the log-likelihood $C(v_{i,k})$ is defined in Eq. 3.8. We can see this is possible by looking at Eq. 3.7, where we can use the product of likelihoods,

$$\begin{aligned} p(D | v) &\propto p(x_1, x_2 \dots x_n | v) \\ &\propto p(x_1 | v) \dots p(x_n | v) \\ &\propto \exp \left(\sum_i C_{j,k} \right). \end{aligned} \quad (3.14)$$

If the data contains gaps where the detector was not observing, then we fill the gaps in the power spectrum with a constant value which is the expectation value of the log-likelihood. The procedure of filling in the gaps of the data is completed before any summing. Therefore, the data should have the same mean regardless of how much real data is in each sum. In the examples that follow, we sum the SFTs over the length of one day.

The main motivation for summing the data is to increase the SNR of a signal in the segments. The risk is that a signal can move between adjacent frequency bins during a day. To reduce this risk, we choose the frequency bin width such that it is more likely that a signal will be contained within a single frequency bin that cross a bin edge. In practise, to ensure that this is true, the segment or SFT length and the number of segments which are summed can be tuned for each search. As well as increasing the SNR, summing over one day should average out the antenna pattern. This means that the log-likelihood value in any bin should be more similar between detectors, however, there is still some variation due to the sky localisation and polarisation.

This also has two main effects on the transition matrix, the first is that as each segment of data is now one day long, a jump between frequency bins is far more likely, therefore, the transition matrix elements are modified to account for this. The second is that as the data is averaged over

one day, the signal should remain in the same frequency bin between detectors, therefore, there is no longer a need for the multi-dimensional transition matrix described in Sec. 3.5.

The volume of the data is also reduced by a factor of $1/N_s$, therefore, the time taken for the algorithm to run is also reduced by the same factor.

3.8 Line-aware statistic

The multiple-detector algorithm described in Sec. 3.5 returns the most probable track of a common signal assumed to be in Gaussian noise. As a consequence the algorithm will return large values of the log-likelihood even if there are inconsistent values of SFT power between the detectors, either from non-Gaussian noise or because the signal is not equally strong in the two detectors. However a signal with unequal power in the two detectors is more likely to be a non-Gaussian instrumental line than an astrophysical signal. The line-aware statistic described in this section is designed to make the search more robust to such instrumental artefacts within realistic non-Gaussian data whilst maintaining sensitivity to astrophysical signals.

For most of the analysis examples presented here we use data which is the incoherent sum of 30-minute normalised SFTs over a day (described in more detail in Sec. 3.7). As a result the effects of the detector antenna patterns and of differential Doppler shifts are significantly reduced, and any signal should have a broadly similar summed log-likelihood in the same frequency bin in each detector. The statistic can then be modified such that we expect a similar log-likelihood in each detector.

We first consider the model of Gaussian noise with no signal present. Within a single summed segment, the likelihood of Gaussian noise at frequency ν is given by a χ^2 distribution,

$$p(F_j|\nu_j, M_N, I) = \frac{1}{2^{d/2}\Gamma(d/2)} F_j^{d/2-1} \exp\left\{-\frac{F_j}{2}\right\} \quad (3.15)$$

where F_j is the frequency domain power summed over sub-segments within a single day, as described in Sec. 3.7 and d is the number of degrees of freedom, equal to twice the total number of summed SFTs. M_N represents the model that the data is simply Gaussian noise. In the presence of a signal (model M_S), the power should follow a non central χ^2 distribution in which the non-centrality parameter λ is the square of the SNR, ($\lambda = \rho_{\text{opt}}^2$), i.e.

$$p(F_j|\nu_j, \lambda, M_S, I) = \frac{1}{2} \exp\left\{-\frac{F_j + \lambda}{2}\right\} \left(\frac{F_j}{\lambda}\right)^{d/4-1/2} I_{d/2-1}\left(\sqrt{\lambda F_j}\right). \quad (3.16)$$

If a signal is present we therefore expect the SFT powers in both detectors to follow Eq. 3.16. Assuming for the moment that the noise variance is the same in both, we can determine the

evidence for model M_S by marginalising over λ ,

$$p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, M_S, I) = \int_0^\infty p(\lambda, w_s) p(F_j^{(1)} | \mathbf{v}_j, \lambda, M_S, I) p(F_j^{(2)} | \mathbf{v}_j, \lambda, M_S, I) d\lambda. \quad (3.17)$$

We set the prior on λ to be an exponential distribution of width w , this is done somewhat arbitrarily as we expect the majority of signals to have a low SNR. This distribution follows,

$$p(\lambda, w) = \exp\left(\frac{-\lambda}{w}\right). \quad (3.18)$$

On the other hand, if an instrumental line is present in one of the detectors we expect to see signal-like power in that detector and noise-like power in the other. The evidence for this ‘line’ model (M_L) is therefore

$$\begin{aligned} p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, M_L, I) &= \int_0^\infty p(\lambda, w_L) \\ &\quad \left[p(F_j^{(1)} | \mathbf{v}_j, M_N, I) p(F_j^{(2)} | \mathbf{v}_j, \lambda, M_S, I) \right. \\ &\quad \left. + p(F_j^{(1)} | \mathbf{v}_j, \lambda, M_S, I) p(F_j^{(2)} | \mathbf{v}_j, M_N, I) \right] d\lambda, \end{aligned} \quad (3.19)$$

The third option to consider is the simple case of approximately Gaussian noise in both of the detectors,

$$\begin{aligned} p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, \lambda, M_G, I) &= p(F_j^{(1)} | \mathbf{v}_j, M_G, I) \\ &\quad p(F_j^{(2)} | \mathbf{v}_j, M_G, I). \end{aligned} \quad (3.20)$$

The posterior probability of model M_{GL} , which contains the probability of Gaussian noise or Gaussian noise with a line in one detector, (taken as mutually exclusive) is

$$\begin{aligned} p(M_{GL} | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j, I) &= p(M_G | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j, I) \\ &\quad + p(M_L | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j, I), \end{aligned} \quad (3.21)$$

where we assume that M_G and M_L are mutually exclusive.

We can now find the posterior odds ratio for the presence of a signal over noise or a line,

$$\begin{aligned}
 O_{\text{SGL}}(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j) &= \frac{p(M_S | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j)}{p(M_{\text{GL}} | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j)} = \frac{p(M_S | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j)}{p(M_G | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j) + p(M_L | F_j^{(1)}, F_j^{(2)}, \mathbf{v}_j)} \\
 &= \frac{p(M_S)p(F_j^{(1)}, F_j^{(2)} | M_S, \mathbf{v}_j)}{p(M_G)p(F_j^{(1)}, F_j^{(2)} | M_G, \mathbf{v}_j) + p(M_L)p(F_j^{(1)}, F_j^{(2)} | M_L, \mathbf{v}_j)} \\
 &= \frac{p(F_j^{(1)}, F_j^{(2)} | M_S, \mathbf{v}_j)p(M_S)/p(M_G)}{p(F_j^{(1)}, F_j^{(2)} | M_G, \mathbf{v}_j) + p(F_j^{(1)}, F_j^{(2)} | M_L, \mathbf{v}_j)p(M_L)/p(M_N)}
 \end{aligned} \tag{3.22}$$

In practice it is convenient to use the log odds ratio,

$$\begin{aligned}
 \log [O_{\text{SGL}}(F_j^{(1)}, F_j^{(2)})] &= \log [p(F_j^{(1)}, F_j^{(2)} | M_S)] \\
 &\quad - \left[\log (p(F_j^{(1)}, F_j^{(2)} | M_G) \right. \\
 &\quad \left. + p(F_j^{(1)}, F_j^{(2)} | M_L)p(M_L)/p(M_G)) \right]
 \end{aligned} \tag{3.23}$$

As we are only interested in the maximum of $\log [O_{\text{SGL}}(F_j^{(1)}, F_j^{(2)})]$, the factor $\log [p(M_S)/p(M_G)]$ can be dropped from the expression.

In this version of the Viterbi algorithm, rather than storing a value proportional to the log-probabilities as in Sec. 3.5, here we store a value proportional to the log-odds ratio. Here we take the log-odds ratio defined in Eq. 3.23, which is the log-odds of a signal having a similar power in each detector, and add the log-prior odds $p(\mathbf{v} | M_S)/(p(\mathbf{v} | M_N) + p(\mathbf{v} | M_L))$ which is the log-prior or any particular track. By assuming that the track transitions for the line and noise model are equally probable for any jump, we set the denominator of the prior-odds is a constant b . This then means Eq. 3.11 is modified to,

$$\begin{aligned}
 \hat{V}_{i,j} &= \max_{k,l,m} (T_{k,l,m} + b + V_{i-1,j+k} \\
 &\quad + \log [O_{\text{SGL}}(F_j^{(1)}, F_j^{(2)})]) ,
 \end{aligned} \tag{3.24}$$

where \hat{V} refers to a log-odds ratio. The maximised statistic now has three tuneable parameters: the width, w_S in Eq. 3.18, on the prior for a signal SNR squared, $p_S(\lambda)$, the width, w_L of the prior in the case of a line, $p_L(\lambda)$, and the ratio of the prior on the line and noise models, $p(M_L)/p(M_G)$. These parameters are optimised for each search, where we initially estimate the SNR of a signal we hope to be sensitive to in each time slice, then use this as a guide for the width of the signal prior. This is then repeated for an expected line SNR and this is used for the width of the line prior. The ratio of line and noise models runs in the range 0 to 1, we set this limit as we do not expect an instrumental line to be as likely as Gaussian noise in any particular

frequency bin.

This line-aware statistic can be applied in a more powerful way when we use multiple detectors and is similar to the approach in [keitel2014SearchContinuousa]. The multiple-detector algorithm described in Sec. 3.5 returns the most probable track of a common signal assumed to be in Gaussian noise. As a consequence the algorithm will return large values of the log-likelihood even if there are inconsistent values of SFT power between the detectors, either from non-Gaussian noise or because the signal is not equally strong in the two detectors. However a signal with unequal power in the two detectors is more likely to be a non-Gaussian instrumental line than an astrophysical signal. The line-aware statistic described in this section is designed to make the search more robust to such instrumental artefacts within realistic non-Gaussian data whilst maintaining sensitivity to astrophysical signals.

For most of the analysis examples presented here we use data which is the incoherent sum of 30-minute normalised SFTs over a day (described in more detail in Sec. 3.7). As a result the effects of the detector antenna patterns and of differential Doppler shifts are significantly reduced, and any signal should have a broadly similar summed log-likelihood in the same frequency bin in each detector. The statistic can then be modified such that we expect a similar log-likelihood in each detector.

In a similar way to the single-detector case, we can write out the evidence for each of the three models as follows. If a signal is present we therefore expect the SFT powers in both detectors to follow Eq. 3.16. Assuming for the moment that the noise variance is the same in both, we can determine the evidence for model M_S by marginalising over λ ,

$$p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, M_S, I) = \int_0^\infty p(\lambda, w_s) p(F_j^{(1)} | \mathbf{v}_j, \lambda, M_S, I) p(F_j^{(2)} | \mathbf{v}_j, \lambda, M_S, I) d\lambda. \quad (3.25)$$

We set the prior on λ the same as in the single detector case in Eq. 3.18. In this case, if an instrumental line is present in one of the detectors we expect to see signal-like power in that detector and noise-like power in the other. The evidence for this ‘line’ model (M_L) is therefore

$$p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, M_L, I) = \int_0^\infty p(\lambda, w_L) \left[p(F_j^{(1)} | \mathbf{v}_j, M_N, I) p(F_j^{(2)} | \mathbf{v}_j, \lambda, M_S, I) + p(F_j^{(1)} | \mathbf{v}_j, \lambda, M_S, I) p(F_j^{(2)} | \mathbf{v}_j, M_N, I) \right] d\lambda, \quad (3.26)$$

The third option is the simple case of approximately Gaussian noise in both of the detectors,

$$p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, \lambda, M_G, I) = p(F_j^{(1)} | \mathbf{v}_j, M_G, I) p(F_j^{(2)} | \mathbf{v}_j, M_G, I). \quad (3.27)$$

We can now find the posterior odds ratio for the presence of a signal over noise or a line by following the same steps as in Eq. ???. Once again we write this as a log-odds ratio,

$$\begin{aligned} \log \left[O_{S/GL}^{(2)}(F_j^{(1)}, F_j^{(2)}) \right] &= \log \left[p(F_j^{(1)}, F_j^{(2)} | M_S) \right] \\ &- \left[\log \left(p(F_j^{(1)}, F_j^{(2)} | M_G) \right) \right. \\ &\quad \left. + p(F_j^{(1)}, F_j^{(2)} | M_L) p(M_L) / p(M_G) \right] \end{aligned} \quad (3.28)$$

The factor $\log [p(M_S)/p(M_G)]$ can again be dropped from the expression.

For the multi-detector case we then modify Eq. 3.11 to,

$$\begin{aligned} \hat{V}_{i,j} &= \max_{k,l,m} (T_{k,l,m} + b + V_{i-1,j+k} \\ &\quad + \log \left[O_{S/GL}^{(2)}(F_j^{(1)}, F_j^{(2)}) \right]), \end{aligned} \quad (3.29)$$

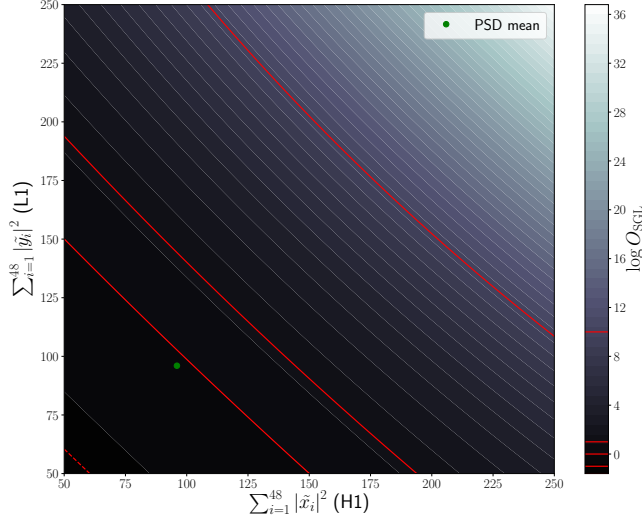
where \hat{V} refers to a log-odds ratio. This is then optimised over the same three parameters as the single detector case.

Fig. 3.3 shows an example of the output of the statistic in Eq. 3.28 for different FFT powers F .

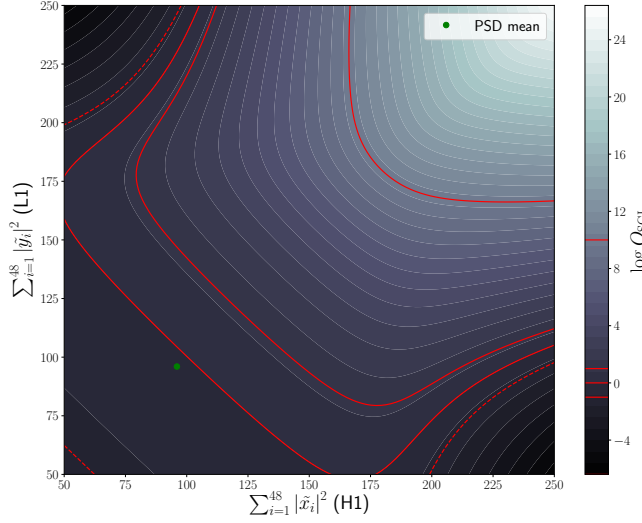
3.9 Line aware statistic for consistent amplitude

In Sec. 3.8 the ‘line aware’ statistic was designed to penalise high SFT powers in a single detector and reward powers which have a similar SNR. This is often a useful statistic to use when the detectors have similar sensitivities, however, this is not always the case. During an observing run of a gravitational wave detector, their sensitivity will vary with time due fluctuating or new noise sources or potentially upgrades which can increase the sensitivity. A change in the sensitivity, or noise floor, affects the SNR of a possible signal in the data, i.e. a lower noise floor results in a higher SNR. In this section the above ‘line aware’ statistic is modified to account for the difference in sensitivities of the detectors, and therefore search for a consistent amplitude between detectors as opposed to SNR.

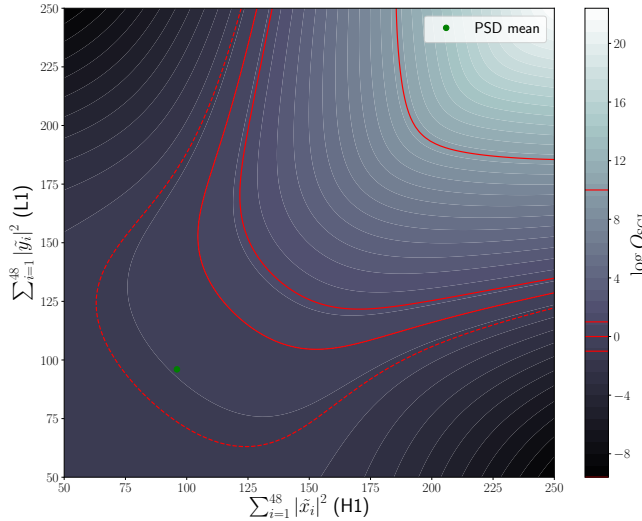
There are two main factors which are taken into account when determining how sensitive a detector is in a particular time interval: the PSD of detector and the duty cycle. The PSD of the detector is essentially how sensitive the detector is at that time and the duty cycle is the fraction of time in a given interval that the detector was collecting data. A decrease in the duty cycle and an increase in the PSD will decrease the SNR and vice-versa. To search for consistent amplitude Eq. 3.24 is modified by weighting each detector by its PSD and duty cycle.



(a) This shows the distribution of the lines aware statistic plotted against the FFT power in each detector. This example is for parameters $p_s(\lambda) = 4, p_l(\lambda) = 0$ and $p(M_L)/p(M_G) = 0$. So the line part of the statistic is not operating.



(b) This shows the distribution of the lines aware statistic plotted against the FFT power in each detector. This example is for parameters $p_s(\lambda) = 4, p_l(\lambda) = 5$ and $p(M_L)/p(M_G) = 0.03$. Here we include the line part of the statistic.



(c) This shows the distribution of the lines aware statistic plotted against the FFT power in each detector. This example is for parameters $p_s(\lambda) = 4, p_l(\lambda) = 5$ and $p(M_L)/p(M_G) = 1$. Here the effect of lines is expected to be larger than the previous panel on the search. Therefore, the statistic forces the two detectors to have more similar power.

Figure 3.3: Lookup tables using the line aware statistic in Eq. 3.29.

The definition of SNR is taken from [1] as,

$$\rho_0^2 = \frac{h_0^2 T}{S} (\alpha_1 A + \alpha_2 B + \alpha_3 C), \quad (3.30)$$

where ρ_0 is the optimal SNR, h_0 is the signal amplitude, T is the time of observation, S is the noise PSD and the term in brackets include the antenna pattern of the detector. The signal with amplitude h_0 will be the same amplitude at both detectors (H and L), therefore we can relate the SNR in each detector by,

$$\rho_L^2 = \frac{\rho_H^2 S_H T_L}{S_L T_H} \frac{(\alpha_1 A_L + \alpha_2 B_L + \alpha_3 C_L)}{(\alpha_1 A_H + \alpha_2 B_H + \alpha_3 C_H)}. \quad (3.31)$$

For the majority of the analysis that follows the SFTs are summed over one day, this will be explained in greater detail in Sec. 3.7. The components in the above equation which have the form $(\alpha_1 A + \alpha_2 B + \alpha_3 C)$, account for the antenna pattern of the earth as it rotates. These can be approximated to be the same for the two detectors H1 and L1 as we will essentially be averaging out the daily modulation by summing SFTs. Therefore we can simplify the above Eq. 3.31 to,

$$\rho_L^2 = \frac{\rho_H^2 S_H T_L}{S_L T_H} = l \rho_H^2. \quad (3.32)$$

This then gives a factor $l = S_H T_L / S_L T_H$ which relates the SNR of each detector, where S and T are values that are known for a given data-set prior to running the search.

This ratio of SNRs can be included in the integral over SNR for the signal model in Eq. 3.25 as follows,

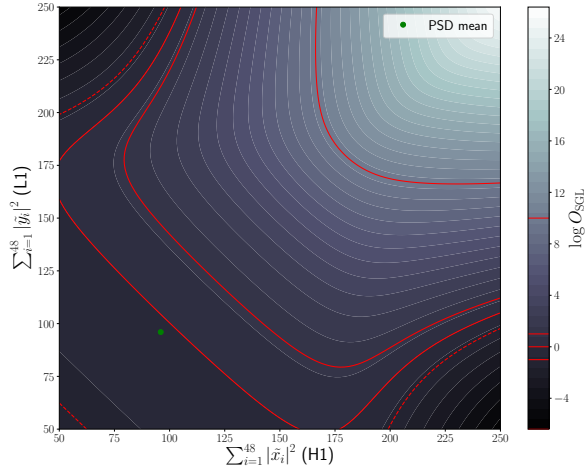
$$p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, M_S, I) = \int_0^\infty p(\lambda, w_s) p(F_j^{(1)} | \mathbf{v}_j, \lambda, M_S, I) p(F_j^{(2)} | \mathbf{v}_j, l\lambda, M_S, I) d\lambda. \quad (3.33)$$

Similarly, the line model in Eq. 3.26 can be modified as,

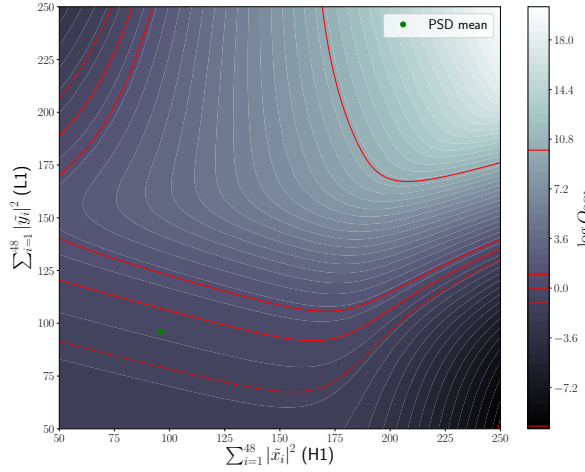
$$p(F_j^{(1)}, F_j^{(2)} | \mathbf{v}_j, M_L, I) = \int_0^\infty p(\lambda, w_L) \left[p(F_j^{(1)} | \mathbf{v}_j, M_N, I) p(F_j^{(2)} | \mathbf{v}_j, l\lambda, M_S, I) \right. \\ \left. + p(F_j^{(1)} | \mathbf{v}_j, \lambda, M_S, I) p(F_j^{(2)} | \mathbf{v}_j, M_N, I) \right] d\lambda. \quad (3.34)$$

Fig. 3.4 shows an example of the values of the statistic described in Eq. 3.34 plotted against a range of FFT powers from each detector. This demonstrated how the statistic accounts for a difference in sensitivity on detectors by allowing the FFT power or effectively SNR to vary more.

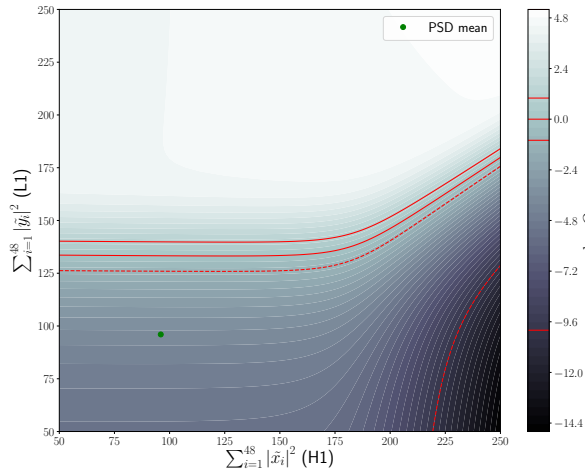
In Fig. 3.4 we show an example of two detectors with large differences in sensitivity, and how the statistic which takes this into account can improve the search sensitivity in this case.



(a) This shows the distribution of the lines aware statistic plotted against the FFT power in each detector. This example is for parameters $p_s(\lambda) = 2, p_l(\lambda) = 0$ and $p(M_L)/p(M_G) = 0$. So the line part of the statistic is not operating.



(b) This shows the distribution of the lines aware statistic plotted against the FFT power in each detector. This example is for parameters $p_s(\lambda) = 2, p_l(\lambda) = 2$ and $p(M_L)/p(M_G) = 1$. Here the line part of the statistic has the same SNR and the signal part, i.e. we expect the SNR of a signal to be similar to that of a line.



(c) This shows the distribution of the lines aware statistic plotted against the FFT power in each detector. This example is for parameters $p_s(\lambda) = 2, p_l(\lambda) = 10$ and $p(M_L)/p(M_G) = 1$. Here we expect the SNR of a line to be larger than a signal.

Figure 3.4: Lookup tables using the line aware statistic for consistent amplitude as in Sec. 3.9. Each of these use the parameters $p_s(\lambda) = 4, p_l(\lambda) = 5$ and $p(M_L)/p(M_G) = 0.03$.

3.10 Testing the algorithm

The sensitivity of the algorithm was tested by searching for artificial signals from isolated pulsars added to three types of noise-like data: continuous Gaussian noise, Gaussian noise but with periods of missing data, and real detector data (the S6 MDC [[walsh2016ComparisonMethods](#)]). The S6 MDC refers to a standardised set of simulated signals which are injected into real data, this set is also what is used for the injections into the two Gaussian noise cases. We describe each of the tests in more detail in Sec. [3.10.1](#), [3.10.2](#) and [3.10.3](#), but several common pre-processing steps are performed before running these datasets through the Viterbi algorithm:

1. We read SFTs generated from 1800 s stretches of data in 2 Hz bands between 100 and 200 Hz. The SFTs length is chosen to ensure that any signal is likely to be contained within the width of a single frequency bin during the length of one day, rather than being split across the bin edges (see below).
2. We estimate the noise PSD for each SFT by calculating a running median over frequency using LALSuite code `XLALSFTtoRnmed` [[ligoscientificcollaboration2018LIGOAlgorithm](#)], this includes a bias factor to convert this to the mean and has a width of 100 bins. We then normalise the SFT by dividing it by its running median, giving the noise-like parts of the spectrum a mean power of approximately one.
3. The SFTs are then summed over one day, as described in Sec. [3.7](#). The signal parameters are chosen so that within the frequencies of the search, the signal will not fall in more than two frequency bins over this period.

The differential Doppler shift of a signal seen at two detector sites due to the Earth's rotation $\Delta f_{\text{rot}}^{(1,2)}$ is simply

$$\Delta f_{\text{rot}}^{(1,2)} = \frac{(\mathbf{v}^{(1)} - \mathbf{v}^{(2)}) \cdot \hat{\mathbf{s}}}{c} f_0, \quad (3.35)$$

where $\mathbf{v}^{(1,2)}$ is the velocity of detector 1,2 in an inertial reference frame, f_0 is the instantaneous signal frequency in the frame, $\hat{\mathbf{s}}$ is the unit vector in the direction of the source and c is the speed of light. The maximum difference in frequency seen by the two LIGO detectors is

$$\Delta f_{\text{rot}} \approx 6.5 \times 10^{-7} f_0, \quad (3.36)$$

so the frequency measured from a source in the equatorial plane with $f_0 = 200$ Hz will differ by up to 1.3×10^{-4} Hz in the two detectors. This is ~ 4 times smaller than the frequency bin width of 1800 s SFTs (5.6×10^{-4} Hz), so signals at frequencies lower than this are likely to appear in the same frequency bin in the two detectors. Therefore, whilst at higher frequencies we still allow the signal to be in different frequency bins between the detectors, in the following searches, we do not allow this.

4. The data is then split into 0.1 Hz sub-bands which are overlapping by 0.05 Hz. These were chosen to ensure that signals are contained within a sub-band over the year. On these timescales the important contributions to the frequency evolution are the spin-down rate of the pulsar and the Doppler shift due to the earth orbit. To investigate the doppler shift, we can look at a signal at 200 Hz, using Eq. 3.35 we can calculate the maximum shift in frequency due to the earths orbit as,

$$\Delta f_{\text{orbit}} = \frac{2\pi R_o}{T_o} \frac{1}{c} f_0 \approx 9.9 \times 10^{-5} f_0, \quad (3.37)$$

where T_o and R_o are the earth orbit time and radius. This gives a maximum doppler shift of 0.019 Hz, this is a $\sim 1/5$ of the width of a sub-band, therefore, is more likely to be totally contained within a sub-band than crossing over the edge. To account for the cases where the signal frequency crosses over the edge of a sub-band, the sub-bands overlap by 0.05 Hz so that the majority of the signals should be completely contained within at least one of the sub-bands. To investigate the spin-down of the pulsar, we look at the length of data, $T = 4.05 \times 10^7$ s and we choose a sub-band width of 0.1 Hz. For a signal to drift over the width of a whole sub-band we would need \dot{f} of,

$$\frac{df}{dt} > \left| \frac{-0.1}{4.05 \times 10^7} \right| = 2.4 \times 10^{-9} \text{ Hz/s}. \quad (3.38)$$

The majority of the injections that follow satisfy this condition, signals which are greater than this, and therefore drift over multiple bands, are vetoed from the search.

5. The two detector Viterbi algorithm is then run using the line aware statistic (see Sec. 3.8). There are 4 parameters which we optimise in this search. The transition probabilities, where we have one parameter τ which is the ratio of the probability of going straight to the probability of going either up or down. Due to the averaging procedure, the signals received at each detector are forced to follow a common track which is equal to the ‘imaginary’ detectors track. The other three parameters, w_S, w_L and $p(M_L)/p(M_N)$, are described in Sec. 3.8.
6. The algorithm then returns the most probable track though the data, and the value \propto the log-odds in the final time step, i.e., the maximum final value, $\max_j(V_{N,j})$, in Eq. 3.24, which is then our detection statistic.

As an example of what the algorithm returns, Fig. 3.5 shows the tracks in the two detectors, H1 and L1. This also shows the log-odds ratio of ending in any frequency bin, i.e., all the elements in Eq. 3.24. In this figure, each time segment of the odds ratios have been normalised such that the sum of the odds ratios is 1.

In the following tests there are two main quantities which we use to determine the sensitivity.

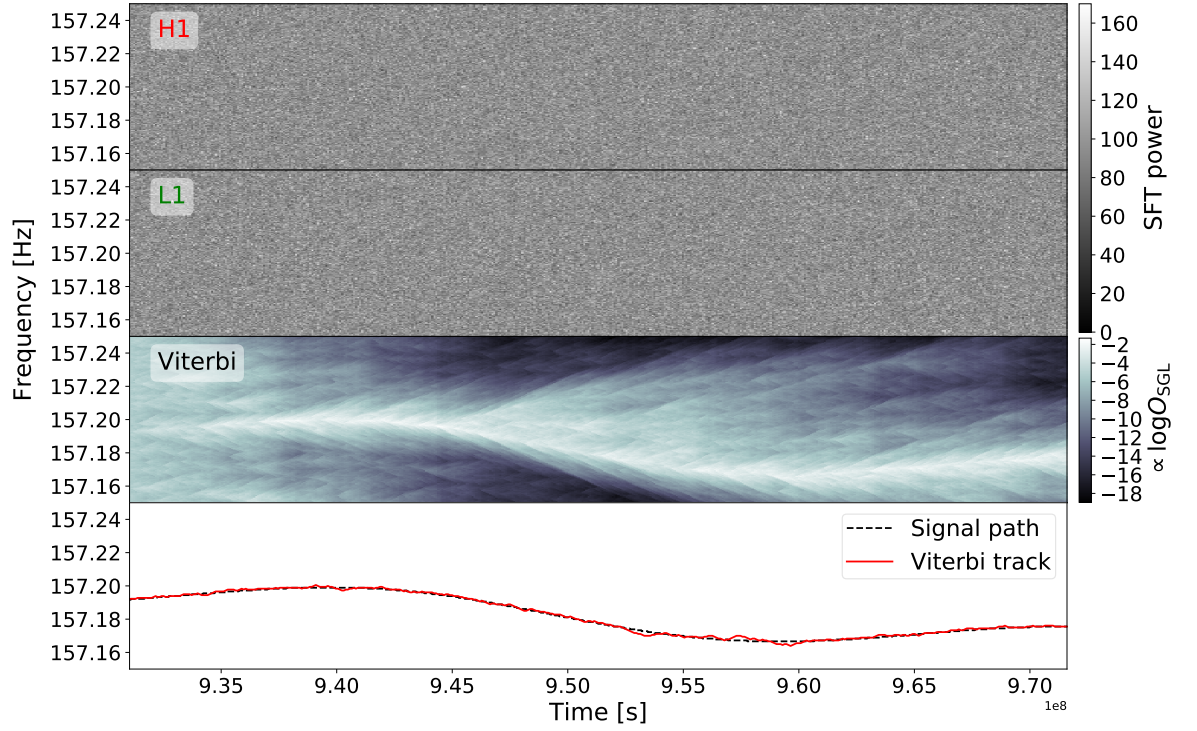


Figure 3.5: The results that the SOAP algorithm returns from an injection with an optimal SNR of 90, i.e., the SNR in H1 is 64 and the SNR in L1 is 62. The signal is injected into Gaussian noise, where the 1800 s SFTs have been summed over 1 day. The top panel shows a simulation of summed SFTs from H1, the second panel shows the same for L1, the third panel shows the values proportional to the log-odds ratios in Eq. 3.24. The log-odds have been normalised such that the sum of all the odds ratios in every time bin are equal to 1. The bottom panel shows the injected signal track (black dotted) and the track found in the ‘imaginary’ detector by the two-detector SOAP search with the line-aware statistic (red), both of these tracks are at the geo-centre. In this case the RMS of the difference between the Viterbi track and injected signal track was ~ 1 bin, where 1 bin is 0.00056 Hz wide.

These are sensitivity depth \mathcal{D} and the optimal SNR ρ . The sensitivity depth, \mathcal{D} , is defined in [behnke2015PostprocessingMethods] as,

$$\mathcal{D}(f) = \frac{\sqrt{S_h(f)}}{h_0}, \quad (3.39)$$

where $S_h(f)$ is the single-sided noise PSD and h_0 is the GW amplitude. The optimal SNR is defined as,

$$\rho^2 = \sum_X 4\Re \int_0^\infty \frac{\tilde{h}^X(f) \tilde{h}^{X*}(f)}{S^X(f)} df, \quad (3.40)$$

where X indexes the detectors and $\tilde{h}(f)$ is the Fourier transform of the time series of the signal $h(t)$. This expression is defined in [prix2007SearchContinuous] for a double-sided PSD and we have defined it for the more common single-sided case.

3.10.1 S6 injections into gapless Gaussian noise

The first test involves injecting signals into Gaussian noise. The power spectrum of a Gaussian noise time-series follows a χ^2 distribution with two degrees of freedom, therefore, as we search through the power spectrum, we generate spectrograms which follow a χ^2 distribution. These spectrograms are 0.1 Hz wide and are set at 0.05 Hz intervals between 100 Hz and 200 Hz. The bins are 1./1800 Hz wide and 1800s long, where the total length of data is the same as S6, i.e., ~ 1.3 years. We then generate the signals, where the pulsars parameters are fixed to the same values as the injections in the S6 MDC in this band, these values are outlined in [walsh2016ComparisonMethods].

The values of f_0 for the injections were not always centred in a sub-band, therefore a number of sub-bands contained only part of the injected signal. These sub-bands were ignored as they contaminated the signal statistics and only the sub-band which contained the whole signal was accepted. This reduced the number of sub-bands from 2000 to 1762 with the removal of 238 sub-bands containing only part of a signal. This set also includes signals that drift across multiple sub-bands due to their high spin-down rate. Only two signals were removed due to their spin-down values, which were $> 5 \times 10^{-9}$ Hz/s, these were the two hardware injections in the 100-200 Hz band.

For each injection the SOAP algorithm returns the detection statistic described in Sec. 3.8 and 3.10. We calculate a false alarm rate, which is the fraction of bands that have no injection that do exceed a given threshold. This is set to 1% and is used as a detection threshold. We then take all of the bands and if they pass the threshold we set them as detected, i.e., 1, and if they do not they are set as not detected, i.e., 0. This then leaves us with a set of binomial data, where the efficiency curves later in the paper are sigmoids which have been fitted to this. The sigmoid

Table 3.1: Table shows the ranges of the search parameters and their optimised values for injections into gapless Gaussian noise, Gaussian noise with gaps and the S6 MDC. For gapless Gaussian noise and Gaussian noise with gaps, there are 10 parameter values spaced linearly between the limits. For the S6 MDC the parameters, τ , w_L and w_S were distributed in log space between the limits and $p(M_L)/p(M_N)$ is distributed uniformly.

	τ	w_S	w_L	$p(M_L)/p(M_N)$
Gapless Gaussian				
limits	[1.0,1.3]	[0.1,5.0]	None	0.0
optimised	1.1	2.06	None	0.0
Gaussian with gaps				
limits	[1.0,1.3]	[0.1,5.0]	None	0.0
optimised	1.1	2.06	None	0.0
S6 MDC				
limits	[1.0,1.1]	[0.1,5.0]	[0.1,6.0]	[0.0,1.0]
optimised	1.00000001	4.0	5.0	0.0387

follows,

$$s(x; x_0, k) = \frac{1}{1 - \exp(-k(x - x_0))}. \quad (3.41)$$

The fit is done by sampling the posterior, i.e.,

$$p(x_0, k | b) \propto p(x_0, k) p(x | x_0, k), \quad (3.42)$$

where $p(x_0, k)$ is the prior and we set to a flat prior and $p(x | x_0, k)$ is the likelihood function which is defined by,

$$p(\bar{x} | x_0, k) = \prod_{j=0}^n \frac{n!}{k!(n-k)!} s(x_j | x_0, k)^k (1 - s(x_j | x_0, k))^{n-k}. \quad (3.43)$$

To plot the efficiency curves and lower and upper error bounds, we sample Eq. 3.42 using MCMC and then take the mean and the 5th and 95th percentiles respectively for each point in SNR or depth and plot these. Fig. 3.6a and 3.6c then show the efficiency curves for the analyses plotted against the signals optimal SNR and depth respectively. The parameters of the search and their optimised values are shown in Tab. 3.1. Where we set the prior on the line model to 0 as this part is irrelevant to this search due to the lack of lines in the data.

From this we can determine that in Gaussian noise without gaps, the Viterbi algorithm can detect to an SNR of ~ 60 and a depth of $\sim 33 \text{ Hz}^{-1/2}$ with 95% efficiency at a 1% false alarm.

Fig. 3.6b and 3.6d, show the RMS of the difference between the injected signal track and the track found by Viterbi for SNR and sensitivity depth respectively. This shows that at SNR of 60, where we are detecting signals with a 95% efficiency, the signals have a mean RMS of ~ 2 frequency bins. Here one bin width is 0.00056 Hz therefore, we have an RMS of ~ 0.0012 Hz.

3.10.2 S6 injections into Gaussian noise with gaps

In the second test, we attempt to more closely mirror the S6 MDC [walsh2016ComparisonMethods] in two stages. The first uses the same injection method as Sec. 3.10.1 however, removes the SFTs where there are gaps in S6. The second uses the same injection method again including gaps, however, uses a different value for the noise floor for each SFT, this is calculated for each band and SFT from S6 data.

Both detectors in S6 had a duty cycle of $\sim 50\%$ [aasi2015CharacterizationLIGO], which means that there are sections of time where there is no data in either one or both detectors. In the sections where one detector is observing but the other is not, the multi detector statistic will not behave correctly as it only has access to data from a single detector. In these sections we switch from using the multi-detector statistic to the single-detector statistic using the same parameters, these are both defined in defined in Sec. 3.8.

The process of removing sub-bands and generating efficiency curves is the same as in Sec. 3.10.1.

We set a 1% false alarm rate and generate an efficiency curve for SNR and depth in Fig. 3.6a and Fig. 3.6c respectively. From these efficiency plots we can see to an SNR of ~ 72 or a depth of $\sim 13 \text{ Hz}^{-1/2}$ at a 95% confidence with a false alarm of 1%.

The parameters of the search which were optimised and their optimised values are shown in Tab. 3.1.

In Fig. 3.6b and 3.6d show the RMS of the difference between the injected signal track and the track found by Viterbi for SNR and sensitivity depth respectively. This shows that at SNR of 72, where we are detecting signals with a 95% efficiency, the signals have a mean RMS of ~ 10 frequency bins (0.0056 Hz).

3.10.3 Tests on the S6 MDC

For a more direct comparison to other CW searches and to see how the algorithm performs with real data, we test the two detector SOAP algorithm using the S6 MDC. We focus this search on the 100-200 Hz band, there are two main reasons for this, one being that this is LIGO's most sensitive band and the other is that for much higher frequencies the signal will drift over larger frequency ranges, therefore, our SFT length will have to be changed. Here the 1800 s SFTs are split as in Sec. 3.10, whereafter normalisation, the data is split into 0.1 Hz wide sub-bands

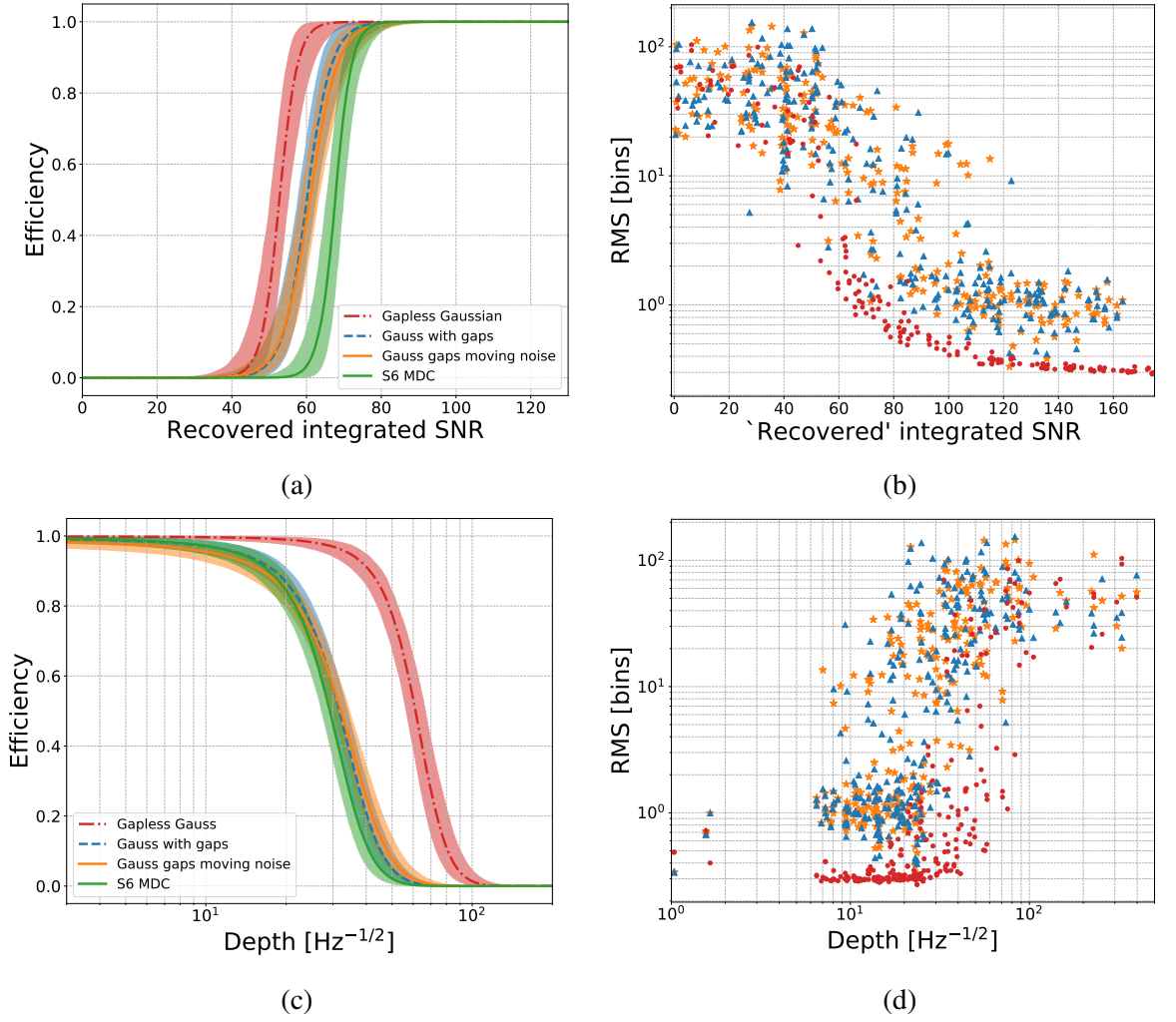


Figure 3.6: Panels 3.6a and 3.6c show the detection efficiency as a function of SNR and depth respectively. Here SNR is the the integrated SNR which we would expect to recover from the available data. The four curves refer to injections into gapless Gaussian noise (red), Gaussian noise with gaps in data, where the noise floor is either fixed (blue-dashed) or it is moving with time (orange) in the same way as the S6 MDC and injections into real data i.e., the S6 MDC. In the gapless Gaussian noise case, the recovered integrated SNR refers to the SNR the injection would have if it had the same amount of data as in the cases with gaps. The curves are made by fitting a sigmoid Eq. 3.39 to binomial detection data with a 1% false alarm rate, as explained in Sec. 3.10.1, the error bounds are the 5% and 95% intervals. At 95% efficiency and a 1% false alarm rate, this shows we can detect to an SNR of ~ 60 and a sensitivity depth of $\sim 34 \text{ Hz}^{-1/2}$ for gapless Gaussian noise and an SNR of ~ 69 and 72 and a sensitivity depth of $\sim 13 \text{ Hz}^{-1/2}$ and $\sim 10 \text{ Hz}^{-1/2}$ for the Gaussian with gaps case with fixed noise floor and moving noise floor respectively. For the S6 MDC we can detect an SNR of ~ 74 and a sensitivity depth of $\sim 13 \text{ Hz}^{-1/2}$. Panels 3.6b and 3.6d show the RMS of the difference between the injected signal track and the track found by SOAP as a function of SNR and sensitivity depth respectively. This is shown in units of bins where each bin is 0.00056 Hz wide.

overlapping by 0.05 Hz.

The two detector SOAP algorithm using the line-aware statistic in Sec. 3.8 is then run on each sub-band under the assumption that the detectors have the same sensitivity. For this search we have four parameters which we optimise, the ranges and optimised values are shown in Tab. 3.1.

As in Sec. 3.10.2, only the sub-bands which contained the entire frequency evolution of the signal were selected. Out of the 2000 sub-bands, 238 were removed due the sub-band only containing part of the signals frequency evolution. The main difference between the analysis for Gaussian noise and real data is that the real data is contaminated with instrumental lines. This means that whilst the techniques described in Sec. 3.8 reduce the number of contaminated bands with a high statistic value, there are still instrumental lines which are coincident between the detectors and which could not be removed with these techniques. Within the data there are large number of lines at integer Hertz, which are seen in coincidence between the two detectors, these are thought to originate from digital electronics [coughlin2010NoiseLine]. Therefore the frequency bins ± 1 bin of each integer frequency in Hertz were removed and filled with the expectation value of the noise. To remove instrumental effects at other frequencies, the sub-bands which gave values of our statistic above a chosen threshold were investigated by eye. In this case 344 sub-bands were investigated, and any which were contaminated were vetoed. From these 344 sub-bands, 193 were removed from the analysis. The predominant feature in the bands which were removed were broad spectral features which lasted the whole run. Therefore, out of the 2000 sub-bands which are searched over, a total number of 431 sub-bands were removed.

The process to calculate the efficiency curves is the same as in Sec. 3.10.2 and 3.10.1.

Fig. 3.6c and Fig. 3.6a show the efficiency curves for SNR and depth respectively. These show that we can detect and SNR of ~ 74 and a sensitivity depth of $\sim 13 \text{ Hz}^{-1/2}$ with an efficiency of 95% at a false alarm of 1%. These results can then be compared to other searches in the S6 MDC comparison paper [walsh2016ComparisonMethods]. Whilst we only search in the 100 - 200 Hz range, the closest comparison in [walsh2016ComparisonMethods] is the test in the 40 - 500 Hz range, such as in Fig. 4 in [walsh2016ComparisonMethods]. Here our algorithm sits roughly in the middle of all other searches in terms of sensitivity.

3.10.4 Optimisation of Line-aware statistic.

For the above searches we used optimised versions on the line aware statistic, however, we have yet to explain how this was optimised. The aim is to find the best parameters for any given search; the four parameters are $\tau, p_S(\lambda), p_L(\lambda)$ and $p(M_L)/p(M_G)$. We find the optimum values empirically by running the entire search for each parameter value that need to be tested. This is possible as the search is relatively fast, this will be explained in Sec. 3.10.5. The line aware statistic is time consuming to calculate, therefore, to reduce the computational time, it is pre-calculated and places into lookup tables such that it is calculated once and called many times.

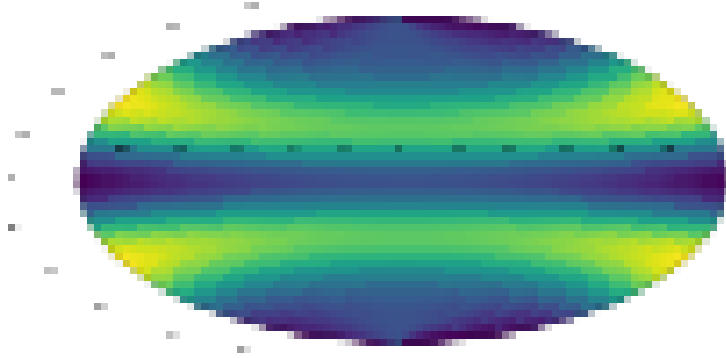


Figure 3.7: This figure shows the optimisation of the statistic for the Gaussian noise case.

These lookup tables were calculated for values of FFT power F in the range 1 to 400 in each of the detectors as shown in Fig. 3.3. For each parameter the ranges in which they were optimised were chosen based on expected SNRs of each of the injections.

We can use a measure of the sensitivity of that search and pick the lookup table which gives the highest sensitivity. We measure the sensitivity by taking the value of SNR which is at 90% efficiency at 1% false alarm.

S6 injections into gapless Gaussian noise and Gaussian noise with gaps

For injections into Gaussian noise, we know that there are no instrumental lines, therefore, we do not need to optimise over the ‘lines’ part of the statistic and can set the parameter $p(M_L)/p(M_G)$ to zero which renders the parameter $p_L(\lambda)$ redundant. This then reduces the complexity of the problem by leaving us with only two parameters to optimise over, τ and $p_S(\lambda)$. Tab. 3.1 shows the ranges which the two parameters are optimised over.

Fig. 3.7 shows the values for the SNR at 90% efficiency for each of the parameter values which were tested, the optimum which is **JOE: run test and fill in**

S6 MDC injections

As the S6 MDC data-set is real detector data, there are many examples of instrumental lines. This is where we expect the line-aware statistic to have the greatest effect in rejecting lines. Here the all of the four parameters are optimised over in the ranges described in Tab. 3.1. This greatly increases the number of lookup tables which need to be generated and therefore the number of time the search needs to be run. This optimisation is then carried over to all future searches. Fig. 3.8 shows the slices through parameters space for each of the parameters where the values are the SNRs at 90% efficiency.

Now the parameters have been optimised for both the Gaussian noise case and the real S6 data case, we can test the effect of the statistic when used on a band which has an instrumental

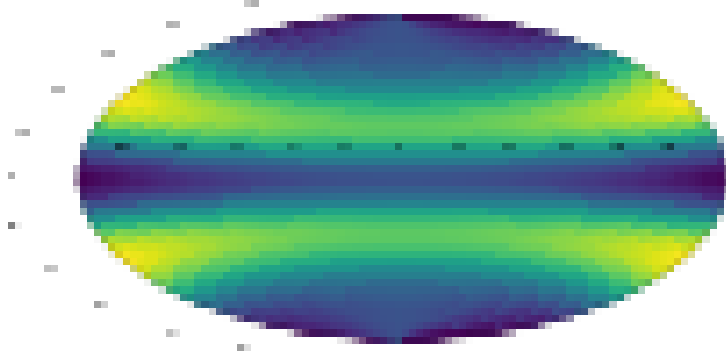


Figure 3.8: This figure shows the optimisation of the line-aware statistic for the S6 MDC case.

line injected. In Fig. 3.9 an example a CW signal injected into Gaussian noise in both detectors. This also shows a simple example of an instrumental line injected into one detector. In the Gaussian noise optimised case the search is looking for high power in both detectors which the strong instrumental line satisfied when the astrophysical signal is weak. The S6 optimised statistic looks for more consistent SNR in each of the detectors. The two different lines show the statistic optimised on the Gaussian noise injections and the S6 MDC injections. This demonstrates how the ‘line aware’ statistic improves the robustness of the algorithm against non astrophysical signals.

3.10.5 Computational cost

One of the main strengths of this search is the drastically reduced computational cost when compared to other current CW searches. The scaling of the computing cost can be estimated for a single detector by looking at the number of calculations that need to be made. The number of calculations for a single detector search, $N_{\text{calcs}}^{(1)}$ is,

$$N_{\text{calcs}}^{(1)} = n_1^m N M, \quad (3.44)$$

where n_1 is the size of the transition matrix, N is the number of SFTs, M is the number of frequency bins and m is the amount of memory described in Sec. 3.6. Where the computing cost scales linearly with the number of frequency bins and SFTs. In the following test we ignore ‘memory’ and look at the time taken for the single detector search where the time taken to read and save data is ignored. Here the data is the same size as the S6 MDC for a single detector search and the search is over a 0.1 Hz band, where we set $n_1 = 3$. This test, and the following test, was run locally on a MacBook Air with a 1.3 GHz Intel Core i5 processor. We can then

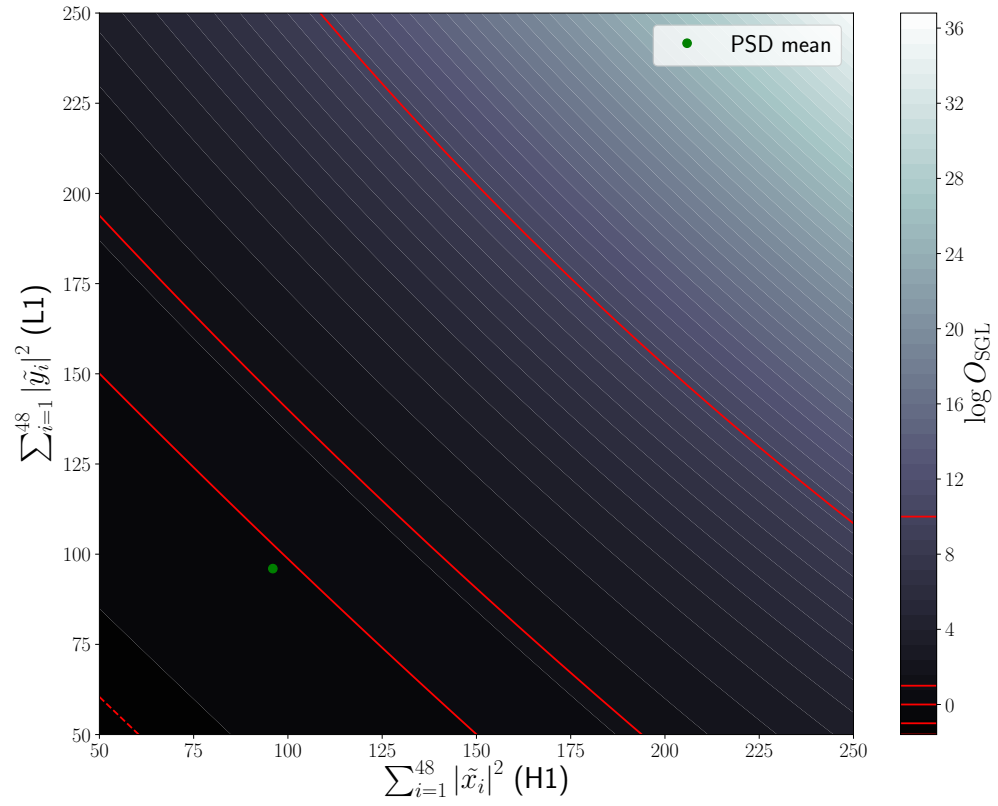


Figure 3.9: This figure shows an example of when the line aware statistic is used compared to a version when it is not.

write the time taken T , as,

$$T = 0.56 \text{ sec} \left(\frac{N}{22538} \right) \left(\frac{M}{180} \right) \left(\frac{N_{\text{bands}}}{1} \right), \quad (3.45)$$

where N_{bands} is the number of different frequency bands. For the multiple, Q , detector case, we can then generalise Eq. 3.44 and write the number of calculations $N_{\text{calcs}}^{(Q)}$ as,

$$N_{\text{calcs}}^{(Q)} = NMn_1^m \prod_{q=1}^Q n_{q+1}, \quad (3.46)$$

where n_1 is the first dimension of the transition matrix, Q is the number of detectors and n_{q+1} is the size of the transition matrix element which refers to detector q . For our tests we set $n_1 = n_{q+1} = 3$ and use 2 detectors i.e., $Q = 2$ which each have the same size data as the previous test. The actual time taken to run however, depends on the version of the algorithm which is run. For example, including the line aware statistic slows the search slightly. For the two detector where two SFT powers are summed,

$$T = 1.35 \text{ s} \left(\frac{N}{22538} \right) \left(\frac{M}{180} \right) \left(\frac{N_{\text{bands}}}{1} \right). \quad (3.47)$$

The same search now including the line aware statistic, which is implemented using a lookup table, changes this to,

$$T = 25.7 \text{ s} \left(\frac{N}{22538} \right) \left(\frac{M}{180} \right) \left(\frac{N_{\text{bands}}}{1} \right). \quad (3.48)$$

Other searches, excluding Einstein@home which takes on the order of months to run (> 100 million core-hours [walsh2016ComparisonMethods]), take 1 – 10 million core-hours [walsh2016ComparisonMethods]. This search should take $\mathcal{O}(10^3)$ core-hours.

3.11 Discussion

In this paper we describe an application of the Viterbi algorithm, called SOAP, to search for continuous sources of gravitational waves. This paper outlines the method and derives the statistics behind the method in a consistent Bayesian formalism. It then presents the results from the first set of tests of sensitivity for the SOAP algorithm on three separate datasets.

We tested SOAP on a set of fake isolated pulsar signals in the 100 – 200 Hz range, based on 1800s SFTs summed over one day. The three datasets that included these signals comprised continuous Gaussian noise, Gaussian noise but with temporal gaps corresponding to LIGO dead times in the S6 data run, and real data, i.e., the S6 MDC. Although a major attraction of SOAP is its sensitivity to a wide range of signal types, in the tests above it was optimised to detect isolated pulsar signals below 100 Hz with low spin-down to offer a comparison with other CW searches.

From these tests, by setting a 95% efficiency and a false alarm of 1%, we found that in the case of continuous Gaussian data we could detect a signal with an optimal SNR of ~ 60 and a depth of $\sim 33 \text{ Hz}^{-1/2}$ with an RMS of the difference between the injected and Viterbi track being ~ 2 frequency bins (0.0012 Hz). When gaps were introduced into the data to simulate S6 we could detect a signal with an SNR ~ 72 and a depth of $\sim 10 \text{ Hz}^{-1/2}$, with an RMS of ~ 10 bins (0.0056 Hz). The drop in sensitivity here is simply because there is $\sim 50\%$ less data compared to the previous case. Finally, in the S6 MDC we could detect a signal with an SNR ~ 74 and a depth of $\sim 13 \text{ Hz}^{-1/2}$. These real data contain non-Gaussian artefacts such as instrumental lines and this causes a further drop in sensitivity. Whilst not a full comparison to other searches in the S6 MDC [walsh2016ComparisonMethods], as we only tested on a subset of the bands, this search has a sensitivity which is comparable to some other CW searches, however offers a massive increase in speed.

We chose the specific frequency band to search over as the data which we used, i.e., the summed data, becomes less effective at frequencies much higher than 200 Hz, and using the parameters of our simulations, signals can spread over many frequency bins in a day, reducing sensitivity further, however this can be mitigated by using shorter SFTs or performing their summation over 12 (rather than 24) hours.

The methods described in this paper present a basic approach for gravitational-wave signal searches using SOAP. However there are several further developments that could increase its sensitivity. Some of these are outlined below:

One of the main features which reduces the sensitivity of the search is non-gaussianities within the data, namely instrumental lines. Although we have a statistic which penalises these features, in some cases it will also penalise a strong signal. For example, when the amplitude of the noise floor is high for one detector or the duty factor is lower, the signal will appear more like an instrumental line to this statistic. We hope to improve the search statistic in the future by searching for consistent amplitudes as opposed to consistent SNR, i.e, the statistic will take the amplitude of the noise floor and the duty factor into account.

One variation of this method which has been described in this paper is ‘memory’, which is where the tracks jump in frequency is determined by the previous n jumps. This has yet to be fully tested, however, we expect that this will increase our sensitivity to signals where have a better idea of their frequency evolution. This however, comes at a cost in computational time which we can estimate given Eq. 3.46 in Sec. 3.10.5.

Further additions to the search include using the Fourier transform of the SFT power along the Viterbi track as a detection statistic. If the Viterbi track follows that from an astrophysical signal, then we should see the effects of the antenna pattern in this Fourier transform as a peak at half a sidereal day. If the track follows something which is not astrophysical then this should not be seen this peak in this Fourier transform. This only applies to the search directly on the SFTs not the summed data, as the antenna pattern variations will have been averaged out in the

summing.

As well as searching for astrophysical signals, SOAP can also be used to search for and identify instrumental lines. Here we use single detector data, or multiple channels from a single detector, to identify quasi-monochromatic features on the data for further study.

Whilst this paper presents initial tests on sensitivity, further tests will be needed for a full comparison to other CW search methods. This search, however, aims to look for signals which may not follow the standard frequency evolution and is intended to return potentially interesting candidates for a more sensitive followup.

Chapter 4

Machine learning for continuous wave searches

Machine learning is a term which was used by Arthur Samuel in 1959. He described it as a "Field of study that gives computers the ability to learn without being explicitly programmed" []. This can be thought of as a subset of artificial intelligence.

With the development in computing in recent years, including GPUs and the languages used to program with them, machine learning has become more accessible.

4.1 Neural networks

Throughout this section I will summarise one machine learning technique which are known as Neural networks. Neural networks, as the name may suggest, was developed as a way for a computer to mimic the neurons in the brain. To understand why this would be useful, I will give short example which will be followed through this section. A good example used extensively in explanations of neural networks is the ability to identify hand written digits. This is a simple task which the brain can complete with ease. However, writing a traditional algorithm to perform this same task is very difficult. The algorithm would have to identify a particular shape, which has a huge amount of variation. For example, any number could be written at a slightly different angle or the number 9 could have a much longer tail etc. This is where a neural network can help. These can be trained in a similar way to a human brain. This is where in a lifetime many examples of different symbols are seen and each time a new one is seen the brain 'updates' itself based on what is observed. This process is essentially replicated for a neural network, where the algorithm can be updated such that it can correctly identify each digit.

To explain how a neural network can be updated, we need to start with the operation of a single neuron.

- introduce machine learning as a technique with previous examples

- general idea, train a NN on lots of data
- GPU development allowed these to train fast and be useful
- General intended application to CW searches
- reduce affect of instrumental artefacts

4.1.1 Neurons

Neurons are the building blocks of any neural network. They perform simple operations on any number of input values and then output a single value. The output o of a neuron is defined by the equation,

$$o = f \left(b + \sum_{i=1}^N w_i x_i \right), \quad (4.1)$$

where b is the bias, x is the input, w is the weights, f is the activation function and o is the output. Here the input x represents either the data which is input, i.e the pixels of the image which contains the digit in the example above, or the output of another neuron. The weights w then represents how important each of those data points are to this problem, or specifically this neuron. The bias b is then just an extra factor which can shift the data by a fixed value. The activation function f is then a function which can have many forms, in the simplest case in a neuron known as a ‘perceptron’, it provides a cut where any value above a given threshold is 1 and any below is 0. However, there are many different types of activation functions which can be applied to different situations. This will be explained in more detail in Sec. 4.1.3.

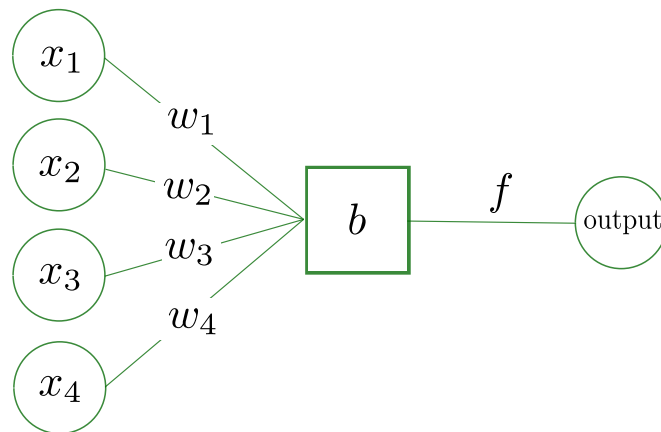


Figure 4.1: Basic neuron

In the example in Fig. 4.1 I have shows a neuron which has 4 input variables, or 4 input data points. When a network is trained, or when it learns, the weights applied to each of the inputs

and the bias are updated to better represent the input data. This training procedure is explained in more detail in Sec. 4.1.5 Many neurons are then used in combination with each other to develop a neural network which can be applied to more complex problems.

4.1.2 Network structure

The structure of the network is defined by the user and there is not set way to design the network. However, in general the complexity of the network reflect the complexity and size of the input data. If there is a small number of training examples and a large and complex network, it may be able to learn the input data set as opposed the the general information that they represent.

Neural networks are structured in features called layers, sometimes known as fully connected layers. These are rows of N neurons which all take the same input such that there is N output values. An example of a simple neural network is shown in Fig. 4.2. The first layer is the input layer, this is just the input data. In the example of hand drawn digits, this would be the pixels from the image of the digit. The final layer represents the information that you intend the network to extract from the input data. In the hand drawn digit example, this could have 10 output values corresponding to each digit 0-9. Each of these outputs is then a value which is related to the probability of that digit being present in the image.

4.1.3 Activation functions

The activation function is how the output of a neuron is transformed. The most simple activation function is a cut as described in Sec. 4.1.1, however, this type of activation does not perform well. The activation function has been shown to be effective when it is a non-linear function.

4.1.4 Loss functions

4.1.5 Training

For example, imagine you are playing darts and have the aim of trying to hit bullseye. After each dart that misses you would adjust where you aim the dart based on how far away from the bullseye it landed.

4.2 Convolutional Neural Networks

Convolutional neural networks are based on the same principles as the neural networks described above. The difference being that convolutional neural networks (CNNs) use additional types of layers which help to contain information of pixels relative to each other. In a standard neural network, it does not have any information on the location of any input value. For example, in

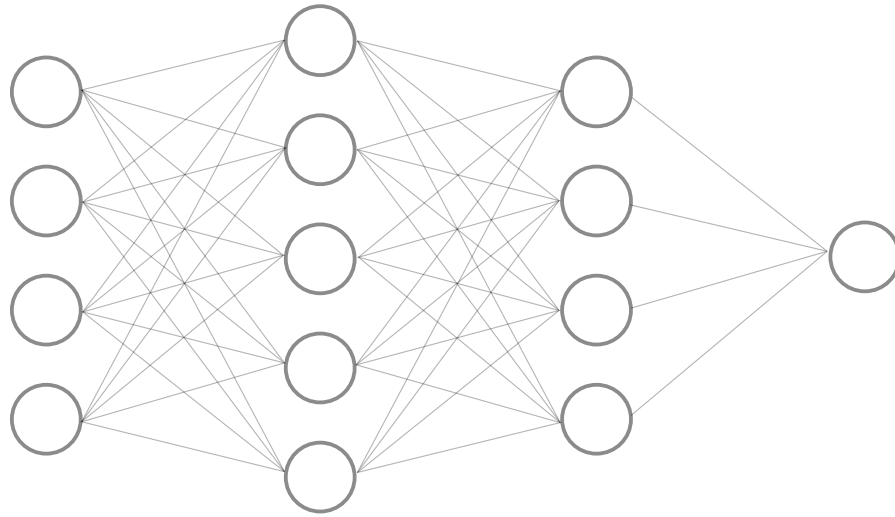


Figure 4.2: Simple neural network

an input image every pixel is flattened into a single dimensional array before being input to the network. CNNs try to retain some location information for each of the pixels. Whilst CNNs can be used for single dimensional inputs, they were generally developed for images. As well as retaining some location information, they reduce the size of the networks parameters.

4.2.1 Convolutional layers

Convolutional layers have some similarities to standard fully connected layers as described in Sec. 4.1.2. The main difference being how the weights are applied to the inputs. To show the difference I have an example of a simple input image which has 9 pixels, this is shown in Fig. ??.

A fully connected neural network would flatten this image and apply Eq. 4.1 to the inputs, such that the output is,

$$\begin{aligned}
O_k &= f \left(\sum_i \sum_j I_{i,j} w_{i,j,k} + b_k \right) \\
O_1 &= f (I_1 w_{1,1} + I_2 w_{2,1} + I_3 w_{3,1} + I_4 w_{4,1} + I_5 w_{5,1} + I_6 w_{6,1} + I_7 w_{7,1} + I_8 w_{8,1} + I_9 w_{9,1} + b_1) \\
&\dots
\end{aligned} \tag{4.2}$$

If the size of the convolutional filter was 2×2 , then the convolutional layer only has 5 parameters to vary as opposed to 10 in this case. This would then use the equation,

$$O_{i,j} = \sum_m \sum_n K_{m,n} I_{i-m,j-n} \tag{4.3}$$

$$\begin{aligned}
O_1 &= f (I_1 w_1 + I_2 w_2 + I_4 w_3 + I_5 w_4 + b) \\
O_2 &= f (I_2 w_1 + I_3 w_2 + I_5 w_3 + I_5 w_4 + b) \\
O_3 &= f (I_4 w_1 + I_5 w_2 + I_7 w_3 + I_8 w_4 + b) \\
O_4 &= f (I_5 w_1 + I_6 w_2 + I_8 w_3 + I_9 w_4 + b).
\end{aligned} \tag{4.4}$$

Whilst this is hard to picture mathematically, it can be easier seen as a 2×2 filter which is convolved with the input image. Fig. ?? shows how the convolutional filter is applied. The output of a convolutional layer is then an image which is half the filter length less length in each dimension. In practice, the input images are often padded with zeros such that the output image is the same size as the input image.

A convolutional layer has a number of different parameters which can be varied when setting up a model. Below I list each of the adaptable parameters and what they do.

Filter size The filter size is the size and shape of the convolutional filter. The filter does not have to be square, however must be less than the dimensions of the image.

Number of filters The number of filters can be any value. If you have K filter kernels, then the convolutional layer will output K filtered images.

Activation function The activation function is generally kept the same for each of the layers, however this can be set here. The different types have been explained in Sec. 4.1.3.

Stride The stride is when the convolution is applied to every other pixel. This method reduces the size of the output by the same factor. This is not used in the rest of this work.

The convolutional layers will reduce the number of parameters used in each network or model, which can speed up the training procedure. In a normal neural network the image is flattened, therefore, any information which related the location of a pixel to another is lost. Convolutions can keep hold of this information and look for similar patterns within an image.

4.2.2 Max pooling layers

Max pooling layers are designed to reduce the size of the problem whilst holding on to as much important information as possible. the idea of this layer is relatively simple, it reduces the image size by taking the maximum value in a reigon of a given size. For example, in Fig. ??, the image is reduced by a 2x2 max pooling layer.

4.2.3 CNN structure

CNNs are usually structured such that they can extract larger features from an input image, then the outputs from this are passed on to be classified. To extract the larger features any number of different convolutional layers and max-pooling layers used. Once this then outputs K images, the values are flattened into a one dimensional array an passed to a set of fully connected layers. This is essentially the same type of network described in Sec. 4.1. This can then classify the network into the chosen output.

CNNs are then trained in the same way as neural networks as described in Sec. 4.1.5.

4.3 CW search

4.3.1 Network structure

Viterbi statistic

Viterbi Maps

Spectrograms

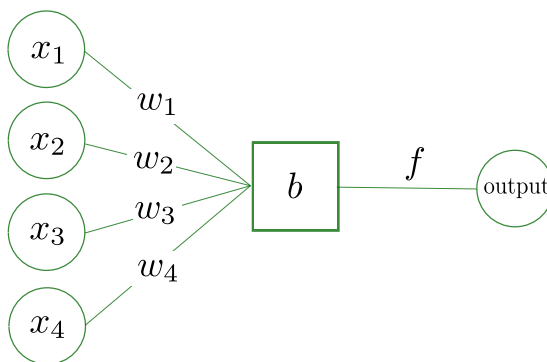
Viterbi map and spectrograms

Viterbi map and Viterbi statistic

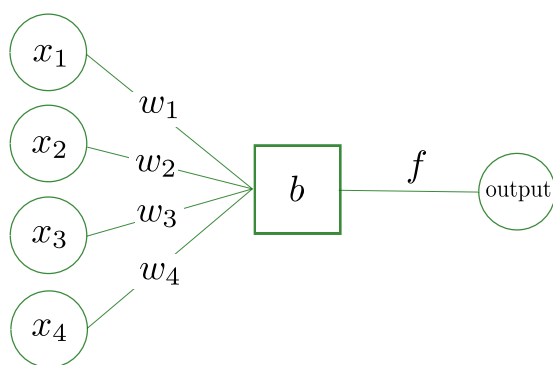
Viterbi map, Viterbi statistic and spectrograms

4.3.2 Network Visualisation

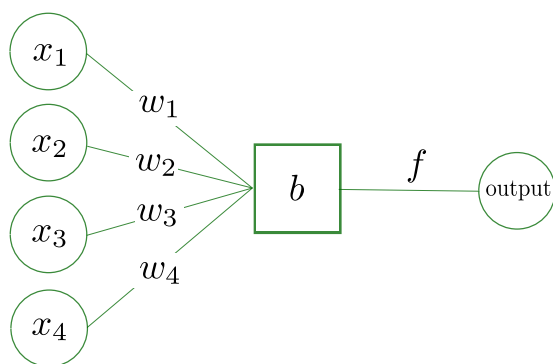
4.4 Data and simulations



(a) Simple neural network



(b) Simple neural network



(c) Simple neural network

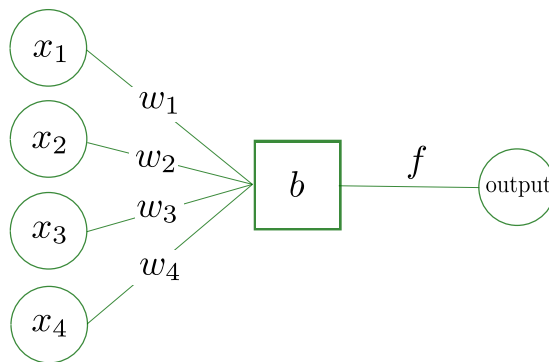


Figure 4.4: Max pooling layer

Chapter 5

Detector Characterisation with SOAP

When searching for GW signals, it is important to understand the origins of noise in the detectors data which does not originate from an astrophysical source. A large fraction of GW search algorithms, including SOAP above, assume that the detectors noise follows a Gaussian distribution. However, the detectors contain artefacts which are not distributed as a Gaussian. These artefacts can negatively affect many searches for GW as they can be easily mistaken for a real GW signal. Some of the potential sources of these artefacts have been mentioned in Sec. 1.3.1. There are many different classes of artefact such as: glitches, **JOE: fill**, instrumental lines and . To conduct a reliable search there are two main tasks which are necessary for detector characterisation. The first is identifying the artefact such that any search knows that part of data is contaminated. The search can then remove that section of data, or use more sophisticated techniques to deal with the artefact []. The second task is to find the source of the artefact. If the source of the artefact is found, it can potentially be removed or limited for future data runs. The majority of this section will focus on a particular class of artefact called instrumental lines and how they affect CW searches.

5.1 Instrumental lines

Instrumental lines have the general structure that they are persistent noise artefacts. There are many classes of instrumental line including: narrow, fixed frequency spectral artefacts or broad features which wander in frequency. A few examples of these are shown in Fig. ??.

As described in Sec. 2, CW are long duration signals with a slowly varying frequency. In the case of an isolated neutron star, the signal which is searched for is narrow-band and a fixed frequency which is Doppler modulated by the earths rotation and orbit. For certain areas of parameter space, the astrophysical signal of an isolated neutron star can appear very similar to an instrumental line. Many of these lines can be ignored if they appear in a single detector, or a statistic such as in Sec. 3.8 can be used to limit their effect. However, there are many examples of instrumental line which appear at the same or similar frequencies in multiple detectors. These

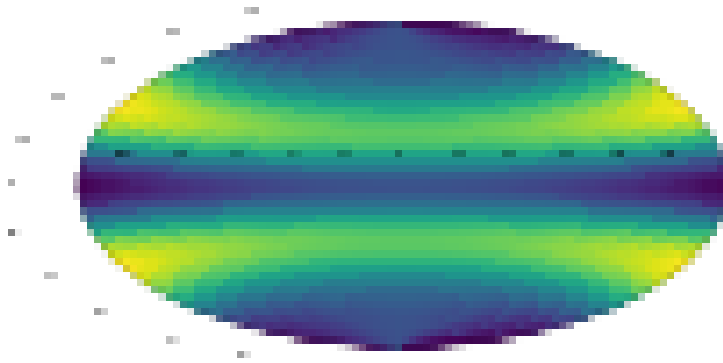


Figure 5.1: Caption

pose a real challenge to some CW searches, and require a lot of investigation to limit their affect.

The origin of many instrumental lines have been identified. Many of these can and have been removed, however, some cannot be removed. An example of a line which cannot be removed is the violin modes of the suspensions of the detectors mirrors. These areas of the frequency spectrum are generally assumed to be contaminated and are ignored in any search.

- instrumental lines are long or short duration detector artifacts
- can be narrow and short or broad and long duration
- many examples from known sources
- many searches currently exist which look through data
- . wandering lines are a large problem as hard to track especially when weak

5.2 Identifying and monitoring instrumental lines

When a detector is running, it is very important to identify instrumental lines and monitor them. This can then lead to either locating the origin of the line such that it can be removed, or allowing it to be flagged for other search algorithms. The lines are generally identified in the GW channel, this is the output of the detector which GW are observed and is the data using is previous chapters.

As well and the GW channel the detector records many different channels known as auxiliary channels. These channels monitor many components of the detector and anything which may affect the GW channel. For example, the seismometer which are located near the corner stations and end stations are all channels which are monitored.

Some of the channels which are useful for monitoring and locating lines include: physical environment monitor (PEM) , PEM channels include instrumental such as magnetometers

and These are useful to investigate alongside the main GW channel. The main goal is to reduce the number of artefacts in the GW such that it is as close to Gaussian noise as possible. If an artefact shows up in the GW channel in coincidence with one of the PEM or other channels, then this is an indicator that the artefact originates from something related to that PEM. For example,

5.3 Identifying and cleaning lines with SOAP

- overview of soap search and what type is used for line searches
- how this is applied and how to interpret the output for lines searches
- SOAP can identify weak lines when they are wandering or fixed frequency
-

5.4 Summary pages

- why summary pages are useful
- how to read them and what they mean
- how to use the information with other searches

Chapter 6

Summary

Appendix A

Continuous gravitational wave injections

In this section I outline how we inject a CW signal into data. This can generally be done in two different ways: simulating a signal in the time domain and injecting into time domain noise or simulating the power spectrum of a signal and injecting the signal into a PSD.

A.0.1 CW Signal

This section has been covered in Sec. ??, The CW signal is generated by using a lalsuite package where

A.0.2 Time series and complex FFT injections

Injections into time-series data is relatively simple. Given a set of parameters for the source the signal can be generated in the time-series, this is then just summed with the time-series which it is injected into. Similarly with the FFT, the time-series of the signal at the correct time and for the correct duration is generated, the complex FFTs are then summed.

A.0.3 Spectrogram injections

To inject into a spectrogram the power spectrum of the signal will need to be simulated. In our injection we do not have access to a time-series, therefore, we do not simulate the signal in the same way, rather we use the signals estimated SNR

It can be shown that the PSD of Gaussian noise with zero mean and unit variance is a χ^2 distribution with 2 degrees of freedom. Therefore, if we want to generate a spectrogram for Gaussian noise, we just generate a two dimensional array of values distributed as χ^2 with two degrees of freedom. Assuming that there is some sinusoidal signal with a given SNR within a Gaussian noise time-series with zero mean and unit variance, the FFT power in a particular frequency bin can be estimated using a non-central χ^2 distribution with 2 degrees of freedom, where the non centrality parameter is the square of the SNR. To calculate the SNR in a given

frequency bin the equation in [Prix2007] for optimal SNR was used,

$$\rho(0)^2 = \frac{1}{2} h_0^2 T S^{-1} [\alpha_1 A + \alpha_2 B + \alpha_3 C], \quad (\text{A.1})$$

where h_0 is the GW amplitude, T is the total observing time in seconds, S^{-1} is the mean PSD noise floor. The values of α are then defined by,

$$\begin{aligned} \alpha_1 &= (\mathcal{A}^1)^2 + (\mathcal{A}^3)^2 \\ \alpha_2 &= (\mathcal{A}^2)^2 + (\mathcal{A}^4)^2 \\ \alpha_3 &= \mathcal{A}^1 \mathcal{A}^2 + \mathcal{A}^3 \mathcal{A}^4 \end{aligned} \quad (\text{A.2})$$

where,

$$\begin{aligned} \mathcal{A}^1 &= A_+ \cos(2\psi_0) \cos(2\phi) - A_\times \sin(2\psi_0) \sin(2\phi) \\ \mathcal{A}^2 &= A_+ \cos(2\psi_0) \sin(2\phi) + A_\times \sin(2\psi_0) \cos(2\phi) \\ \mathcal{A}^3 &= A_+ \sin(2\psi_0) \cos(2\phi) - A_\times \cos(2\psi_0) \sin(2\phi) \\ \mathcal{A}^4 &= A_+ \sin(2\psi_0) \sin(2\phi) + A_\times \cos(2\psi_0) \cos(2\phi) \end{aligned} \quad (\text{A.3})$$

The signals frequency varies with time and will not always be located at the center of a frequency bin, therefore, when taking the FFT some of the power is spread over multiple frequency bins. In our injections into the power spectrum we need to account for this effect. For a given frequency bin width