



Bayley, J., Messenger, C. and Woan, G. (2019) Generalized application of the Viterbi algorithm to searches for continuous gravitational-wave signals. *Physical Review D*, 100(2), 023006.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<http://eprints.gla.ac.uk/222706/>

Deposited on: 31 August 2020

Enlighten – Research publications by members of the University of Glasgow
<http://eprints.gla.ac.uk>

SOAP: A generalised application of the Viterbi algorithm to searches for continuous gravitational-wave signals

Joe Bayley, Chris Messenger, and Graham Woan
Institute of Gravitational research, University of Glasgow

All-sky and wide parameter space searches for continuous gravitational waves are generally template-matching schemes which test a bank of signal waveforms against data from a gravitational wave detector. Such searches can offer optimal sensitivity for a given computing cost and signal model, but are highly-tuned to specific signal types and are computationally expensive, even for semi-coherent searches. We have developed a search method based on the well-known Viterbi algorithm which is model-agnostic and has a computational cost several orders of magnitude lower than template methods, with a modest reduction in sensitivity. In particular, this method can search for signals which have an unknown frequency evolution. We test the algorithm on three simulated and real data sets: gapless Gaussian noise, Gaussian noise with gaps and real data from the final run of initial LIGO (S6). We show that at 95% efficiency, with a 1% false alarm rate, the algorithm has a depth sensitivity of ~ 33 , 10 and $13 \text{ Hz}^{-1/2}$ with corresponding SNRs of ~ 60 , 72 and 74 in these datasets. We discuss the use of this algorithm for detecting a wide range of quasi-monochromatic gravitational wave signals and instrumental lines.

I. INTRODUCTION

One of the main targets for current ground based gravitational-wave (GW) detectors, including Laser Interferometer Gravitational-wave Observatory (LIGO) [1, 2] and Virgo [3, 4], 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 [5, 6].

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 [7–11]. 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 [12, 13] and references therein.

The analysis that we present here is known as snakes on a plane (SOAP) [14] and is based on the Viterbi algorithm [15]. 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 [16] has recently been used, amongst other applications, as part of a continuous wave (CW) search to track a pulsar with randomly wandering spin frequency [17–21]. We develop an alternative version which is aimed to be applied more generally to search for any long duration signals using just short Fourier transforms (SFTs).

In the next section we will describe the Viterbi algorithm and the basic SOAP implementation 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.

II. THE 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 $\boldsymbol{\nu} \equiv \{\nu_j\}$, where ν_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 $p(\boldsymbol{\nu} | D)$. Using Bayes theorem, this posterior probability can be written as

$$p(\boldsymbol{\nu} | D) = \frac{p(\boldsymbol{\nu})p(D | \boldsymbol{\nu})}{p(D)}, \quad (1)$$

where $p(\boldsymbol{\nu})$ is the prior probability of the track, $p(D | \boldsymbol{\nu})$ 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

$$\begin{aligned} p(\boldsymbol{\nu}) &= p(\nu_{N-1}, \dots, \nu_1, \nu_0) \\ &= p(\nu_{N-1} | \nu_{N-2})p(\nu_{N-2} | \nu_{N-3}) \dots p(\nu_1 | \nu_0)p(\nu_0) \\ &= p(\nu_0) \prod_{j=1}^{N-1} p(\nu_j | \nu_{j-1}), \end{aligned} \quad (2)$$

where $p(\nu_0)$ is the prior probability that the signal in the first time step has a frequency ν_0 and $p(\nu_j | \nu_{j-1})$ is the prior ‘transition’ probability for ν_j given the frequency at the last step was ν_{j-1} .

The noise in each of the segments can be treated as independent, so the likelihood component in Eq. 1 can be factorised as

$$p(D | \boldsymbol{\nu}) = \prod_{j=0}^{N-1} p(\mathbf{x}_j | \nu_j), \quad (3)$$

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

Using Eqs. 1,2 and 3, the posterior probability is then

$$p(\boldsymbol{\nu} | D) = \frac{p(\nu_0)p(\mathbf{x}_0 | \nu_0) \prod_{j=1}^{N-1} p(\nu_j | \nu_{j-1})p(\mathbf{x}_j | \nu_j)}{\sum_S \left\{ p(\nu_0)p(\mathbf{x}_0 | \nu_0) \prod_{j=1}^{N-1} p(\nu_j | \nu_{j-1})p(\mathbf{x}_j | \nu_j) \right\}}, \quad (4)$$

where in the denominator we must sum over all possible tracks S . We require the specific track, or set of frequencies, $\hat{\boldsymbol{\nu}}$ that maximises the posterior probability, and which therefore maximises the numerator on the right-hand side of Eq. 4, i.e.,

$$p(\hat{\boldsymbol{\nu}} | D) \propto \max_{\boldsymbol{\nu}} \left[p(\nu_0)p(\mathbf{x}_0 | \nu_0) \prod_{j=1}^{N-1} p(\nu_j | \nu_{j-1})p(\mathbf{x}_j | \nu_j) \right]. \quad (5)$$

This track also maximises the log of the probability,

$$\begin{aligned} \log p(\hat{\boldsymbol{\nu}} | D) &= \max_{\boldsymbol{\nu}} \left\{ \log p(\nu_0) + \log p(\mathbf{x}_0 | \nu_0) \right. \\ &\quad \left. \sum_{j=1}^{N-1} \left[\log p(\nu_j | \nu_{j-1}) + \log p(\mathbf{x}_j | \nu_j) \right] \right\} + \text{const.} \end{aligned} \quad (6)$$

The Viterbi algorithm finds the most probable track $\hat{\boldsymbol{\nu}}$ by calculating the quantities in Eq. 6 for each frequency at each time step. In the following sections we explain how this is achieved in practice.

A. The transition matrix

We define the ‘transition matrix’ T as the matrix that stores the prior log-probabilities $\log p(\nu_j | \nu_{j-1})$. These transition probabilities depend only on the size and direction of the transition, and in our case correspond 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 up, centre or down (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., $p(\nu_j | \nu_{j-1}) = p(\nu_{j+k} | \nu_{j+k-1})$. 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.

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.

B. 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, i.e., the transition matrix contains only three numbers referring to the track ‘jumps’. 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 $\nu_j \rightarrow \nu_{j,k} = k(j)\Delta f$ 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. 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. 6: the transition probabilities $p(\nu_j | \nu_{j-1})$ and the likelihoods $p(\mathbf{x}_j | \nu_j)$. The transition probabilities are pre-calculated and stored in a transition matrix according to Sec. II A above, and details of how these are calculated are described in Sec. III. To calculate the likelihood we follow the approach of [22] which gives, under the assumption of a single sinusoidal signal in additive Gaussian noise in data segment \mathbf{x}_j ,

$$p(\mathbf{x}_j | \nu_{j,k}, \sigma_{j,k}, I) \propto \exp[C(\nu_{j,k})]. \quad (7)$$

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

$$C(\nu_{j,k}) \equiv C_{j,k} = \frac{1}{\sigma_{j,k}^2} \frac{1}{N_s} \left| \sum_{r=0}^{N_s-1} x_{j,r} e^{i\nu_{j,k} t_r} \right|^2, \quad (8)$$

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 power spectral density (PSD) in the k th sample and the j th data segment. 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. 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. II F.

The Viterbi algorithm records two quantities for each frequency and time bin: The first, $V_{j,k}$, contains the value defined by Eq. 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. 1 and a simple demonstration of the algorithm at work is shown in Fig. 1.

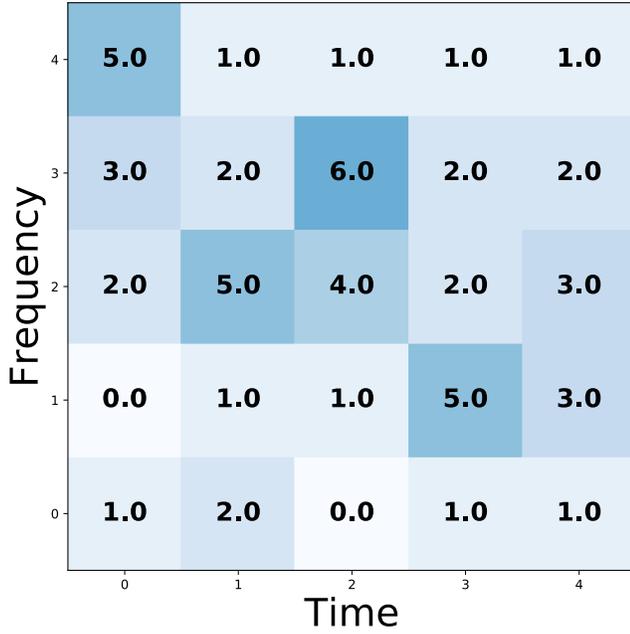
```

1: Input:  $C, T$  {log-likelihood, transition matrix}
2: Output:  $\hat{\nu}, V, A$  {most probable track, track probabilities, jumps}
3:
4: Initialisation
5: for Frequency ( $\nu_{0,k}$ ),  $k = 0 \rightarrow M - 1$  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 ( $\nu_{j,k}$ ),  $k = 0 \rightarrow M - 1$  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:  $\hat{\nu}_{N-1} = \operatorname{argmax}_k (V_{N-1,k})$ 
20: for Segment,  $j = N - 1 \rightarrow 0$  do
21:    $\hat{\nu}_j = \hat{\nu}_{j+1} + A_{j,\nu_{k+1}}$ 
22: end for

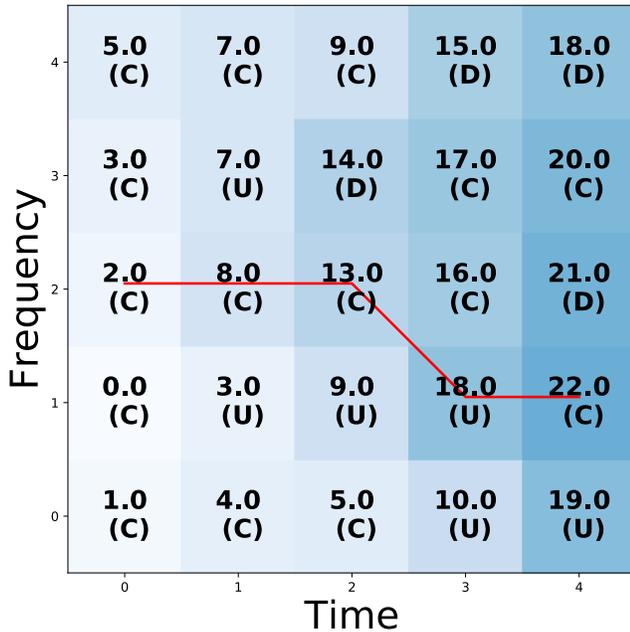
```

ALGORITHM 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. 6 are stored in V , and the jumps are stored in A . The most probable track is denoted by $\hat{\nu}$.

Initialisation: The two parts of Eq. 6, $\log p(\nu_0)$ and $\log p(\mathbf{x}_0 | \nu_0)$, must be computed before the main recursive part of the algorithm can start. Therefore, the initialisation section (lines 5–8) in Alg. 1 calculates the first column in the lower panel of Fig. 1. A priori, there is no preferred initial frequency, so we take the log-prior $\log p(\nu_{0,k})$ 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



(a) The input data



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

FIG. 1: Fig. 1a shows the observed data, i.e the log-likelihood values $C_{j,k}$. Fig. 1b 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 un-normalised prior log-probabilities of these jumps occurring.

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. 6. Lines 11–16 in Alg. 1 calculate the most probable tracks that end at each frequency bin for each segment by using

$$V_{j,k} = \max_i (C_{j,k} + T_i + V_{j-1,k+i}), \quad (9)$$

where i is the size and direction of the jump. For example, in Fig. 1 columns 1–4 are calculated in order using Eq. 9, where it maximises over three possible previous 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. 1 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. 1). In the lower panel of Fig. 1 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 $A_{j,k}$ are followed backwards from this position (lines 20–21). For example, in Fig. 1 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. 1.

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.

C. 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, this is explained in greater detail in Sec. III. As well as differences in signal frequency, due to antenna patterns and other effects, the measured signal amplitude may differ between the detectors [23]. In the following example we assume that the signal has the same amplitude in each detector, however, in Sec. IIF we discuss the case where they differ.

We will now show how the algorithm in Sec. IIB 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, $\nu^{(1)}$ and $\nu^{(2)}$, and the geocentric track ν , given the data, is

$$p(\nu, \nu^{(1)}, \nu^{(2)} | D^{(1)}, D^{(2)}) \propto p(\nu) p(\nu^{(1)}, \nu^{(2)} | \nu) p(D^{(1)} | \nu^{(1)}) p(D^{(2)} | \nu^{(2)}), \quad (10)$$

where $D^{(1)}$ and $D^{(2)}$ represent the data from the two detectors. The main difference between this and that described in Sec. IIB 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. 9) now becomes

$$V_{j,k} = \max_{i,l,m} (C_{j,k+l}^{(1)} + C_{j,k+m}^{(2)} + T_{i,l,m} + V_{j-1,k+i}), \quad (11)$$

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 $p(\nu)$ and $p(\nu^{(1)}, \nu^{(2)} | \nu)$. 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. IIB. 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. IIB, 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. 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. 1 are modified to account for two (or more) detectors. Line 13 of Alg. 1 is changed to calculate Eq. 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. 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.

D. 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. 11 then is then modified to,

$$V_{j,k,s} = \max_h (C_{j,k} + T_{s,h} + V_{j-1,k+s,k+s+h}), \quad (12)$$

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. II B. 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. 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. 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 $3^{m+1} \times N \times M$. Despite the increase in computational cost, this method could increase the sensitivity for well modelled sources such as isolated neutron stars, however, this would not be suitable for an un-modelled source.

E. Summed input data

In this section we outline a method of incoherently-summing a set of SFTs to increase the signal-to-noise-ratio (SNR) of a signal in a segment. To be more precise, we actually sum the log-likelihoods, i.e., the quantity in Eq. 8. We can write the new summed set of data F_j as,

$$F_j = \sum_i^{N_s} C_{i,k} \quad (13)$$

where N_s is the number of SFTs to sum together and the log-likelihood $C(\nu_{i,k})$ is defined in Eq. 8. We can see this is possible by looking at Eq. 7, where we can use the

product of likelihoods,

$$\begin{aligned} p(D | \nu) &\propto p(x_1, x_2 \dots x_n | \nu) \\ &\propto p(x_1 | \nu) \dots p(x_n | \nu) \\ &\propto \exp\left(\sum_i C_{j,k}\right). \end{aligned} \quad (14)$$

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 than 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. II C.

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.

F. Line-aware statistic

The single-detector algorithm described in Sec. II B returns the most probable track of the loudest signal assumed to be in Gaussian noise. However, an astrophysical signal is not expected to have an amplitude which is orders of magnitude above the noise floor, but have an amplitude more similar to the noise. Therefore, a signal with a large amplitude is more likely to be of instrumental origin rather than astrophysical [24–26].

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,

$$p(F_j | \nu_j, M_N, I) = \frac{1}{2^{d/2} \Gamma(d/2)} F_j^{d/2-1} \exp\left\{-\frac{F_j}{2}\right\} \quad (15)$$

where F_j is the frequency domain power summed over sub-segments within a single day, as described in Sec. II E 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.,

$$p(F_j | \nu_j, \lambda, M_S, I) = \frac{1}{2} \exp\left\{-\frac{F_j + \lambda}{2}\right\} \left(\frac{F_j}{\lambda}\right)^{d/4-1/2} I_{d/2-1}\left(\sqrt{\lambda F_j}\right). \quad (16)$$

If a signal is present we therefore expect the SFT powers in the detector to follow Eq. 16. We can then determine the evidence for model M_S by marginalising over λ ,

$$p(F_j^{(1)} | \nu_j, M_S, I) = \int_0^\infty p(\lambda, w) p(F_j^{(1)} | \nu_j, \lambda, M_S, I) d\lambda. \quad (17)$$

Here 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,

$$p(\lambda, w) = \exp\left(\frac{-\lambda}{w}\right). \quad (18)$$

In this single-detector case, we expect an astrophysical signal to look very similar to that of a line other than its amplitude (or SNR). Therefore, we set the evidence for an astrophysical signal and an instrumental signal to follow Eq. 17, where the width w different between the two models.

We then have three models, one for an astrophysical signal, one for an instrumental line and one for Gaussian noise.

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

$$\begin{aligned} p(M_{GL} | F_j^{(1)}, \nu_j, I) &= p(M_G | F_j^{(1)}, \nu_j, I) \\ &+ p(M_L | F_j^{(1)}, \nu_j, I). \end{aligned} \quad (19)$$

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

$$\begin{aligned}
O_{\text{S/GL}}^{(1)}(F_j^{(1)} | \nu_j) &= \frac{p(M_S | F_j^{(1)}, \nu_j)}{p(M_{\text{GL}} | F_j^{(1)}, \nu_j)} = \frac{p(M_S | F_j^{(1)}, \nu_j)}{p(M_G | F_j^{(1)}, \nu_j) + p(M_L | F_j^{(1)}, \nu_j)} \\
&= \frac{p(M_S)p(F_j^{(1)} | M_S, \nu_j)}{p(M_G)p(F_j^{(1)} | M_G, \nu_j) + p(M_L)p(F_j^{(1)} | M_L, \nu_j)} \\
&= \frac{p(F_j^{(1)} | M_S, \nu_j)p(M_S)/p(M_G)}{p(F_j^{(1)} | M_G, \nu_j) + p(F_j^{(1)} | M_L, \nu_j)p(M_L)/p(M_G)}
\end{aligned} \tag{20}$$

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

$$\begin{aligned}
\log \left[O_{\text{S/GL}}^{(1)}(F_j^{(1)}) \right] &= \log \left[p(F_j^{(1)} | M_S) \right] \\
&\quad - \left[\log \left(p(F_j^{(1)} | M_G) \right) \right. \\
&\quad \left. + p(F_j^{(1)} | M_L)p(M_L)/p(M_G) \right]
\end{aligned} \tag{21}$$

As we are only interested in the maximum of $\log \left[O_{\text{S/GL}}^{(1)}(F_j^{(1)}) \right]$, 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. II C, here we store a value proportional to the log-odds ratio. Here we take the log-odds ratio defined in Eq. 21 and add the log-prior odds $p(\boldsymbol{\nu} | M_S)/p(\boldsymbol{\nu} | M_N) + p(\boldsymbol{\nu} | M_L)$ 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. 9 is modified to,

$$\begin{aligned}
\hat{V}_{i,j} &= \max_{k,l,m} (T_{k,l,m} + b + V_{i-1,j+k} \\
&\quad + \log \left[O_{\text{S/GL}}^{(1)}(F_j^{(1)}) \right]),
\end{aligned} \tag{22}$$

where \hat{V} refers to a log-odds ratio. The maximised statistic now has three tuneable parameters: the width (w_S) in Eq. 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 [27]. The multiple-detector algorithm described in Sec. II C 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. II E). 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. 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 ,

$$\begin{aligned}
p(F_j^{(1)}, F_j^{(2)} | \nu_j, M_S, I) &= \int_0^\infty p(\lambda, w_s) \\
p(F_j^{(1)} | \nu_j, \lambda, M_S, I) p(F_j^{(2)} | \nu_j, \lambda, M_S, I) d\lambda.
\end{aligned} \tag{23}$$

We set the prior on λ the same as in the single detector case in Eq. 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

$$\begin{aligned}
p(F_j^{(1)}, F_j^{(2)} | \nu_j, M_L, I) &= \int_0^\infty p(\lambda, w_L) \\
&\quad \left[p(F_j^{(1)} | \nu_j, M_N, I) p(F_j^{(2)} | \nu_j, \lambda, M_S, I) \right. \\
&\quad \left. + p(F_j^{(1)} | \nu_j, \lambda, M_S, I) p(F_j^{(2)} | \nu_j, M_N, I) \right] d\lambda,
\end{aligned} \tag{24}$$

The third option is the simple case of approximately

Gaussian noise in both of the detectors,

$$p(F_j^{(1)}, F_j^{(2)} | \nu_j, \lambda, M_G, I) = \frac{p(F_j^{(1)} | \nu_j, M_G, I)}{p(F_j^{(2)} | \nu_j, M_G, I)}. \quad (25)$$

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. 20. Once again we write this as a log-odds ratio,

$$\begin{aligned} \log \left[O_{S/GL}^{(2)}(F_j^{(1)}, F_j^{(2)}) \right] &= \log \left[p(F_j^{(1)}, F_j^{(2)} | M_S) \right] \\ &\quad - \left[\log \left(p(F_j^{(1)}, F_j^{(2)} | M_G) \right) \right. \\ &\quad \left. + p(F_j^{(1)}, F_j^{(2)} | M_L) p(M_L) / p(M_G) \right] \end{aligned} \quad (26)$$

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. 11 to,

$$\begin{aligned} \hat{V}_{i,j} &= \max_{k,l,m} (T_{k,l,m} + b + V_{i-1,j+k} \\ &\quad + \log \left[O_{S/GL}^{(2)}(F_j^{(1)}, F_j^{(2)}) \right]), \end{aligned} \quad (27)$$

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

III. 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 mock data challenge (MDC) [28]). 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. III A, III B and III C, but several common pre-processing steps are performed before running these datasets through the Viterbi algorithm:

1. We read SFTs generated from 1800s 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 LAL-Suite code `XLALSFTtoRmed` [29], 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. II E. 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

$$\Delta f_{\text{rot}}^{(1,2)} = \frac{(\mathbf{v}^{(1)} - \mathbf{v}^{(2)}) \cdot \hat{\mathbf{s}}}{c} f_0, \quad (28)$$

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

$$\Delta f_{\text{rot}} \approx 6.5 \times 10^{-7} f_0, \quad (29)$$

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 1800s 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. 28 we can calculate the maximum shift in frequency due to the earths orbit as,

$$\Delta f_{\text{orbit}} = \frac{2\pi R_o}{T_o} \frac{1}{c} f_0 \approx 9.9 \times 10^{-5} f_0, \quad (30)$$

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 $f\text{-dot}$ of,

$$\frac{df}{dt} > \left| \frac{-0.1}{4.05 \times 10^7} \right| = 2.4 \times 10^{-9} \text{ Hz/s.} \quad (31)$$

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. IIF). 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. IIF.
6. The algorithm then returns the most probable track through 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. 22, which is then our detection statistic.

As an example of what the algorithm returns, Fig. 2 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. 22. 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. These are sensitivity depth \mathcal{D} and the optimal SNR ρ . The sensitivity depth, \mathcal{D} , is defined in [30] as,

$$\mathcal{D}(f) = \frac{\sqrt{S_h(f)}}{h_0}, \quad (32)$$

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

$$\rho^2 = \sum_X 4\Re \int_0^\infty \frac{\tilde{h}^X(f)\tilde{h}^{X*}(f)}{S^X(f)} df, \quad (33)$$

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 [31] for a double-sided PSD and we have defined it for the more common single-sided case.

A. 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 [28].

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. IIF and III. 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 follows,

$$s(x; x_0, k) = \frac{1}{1 - \exp(-k(x - x_0))}. \quad (34)$$

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

$$p(x_0, k | b) \propto p(x_0, k)p(x | x_0, k), \quad (35)$$

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,

$$p(\bar{x} | x_0, k) = \prod_{j=0}^n \frac{n!}{k!(n-k)!} s(x_j | x_0, k)^k (1 - s(x_j | x_0, k))^{n-k}. \quad (36)$$

To plot the efficiency curves and lower and upper error bounds, we sample Eq. 35 using Markov-Chain Monte Carlo (MCMC) and then take the mean and the 5th and 95th percentiles respectively for each point in SNR or depth and plot these. Fig. 3a and 3c 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. I. Where we set the prior on the line model to 0 as this part

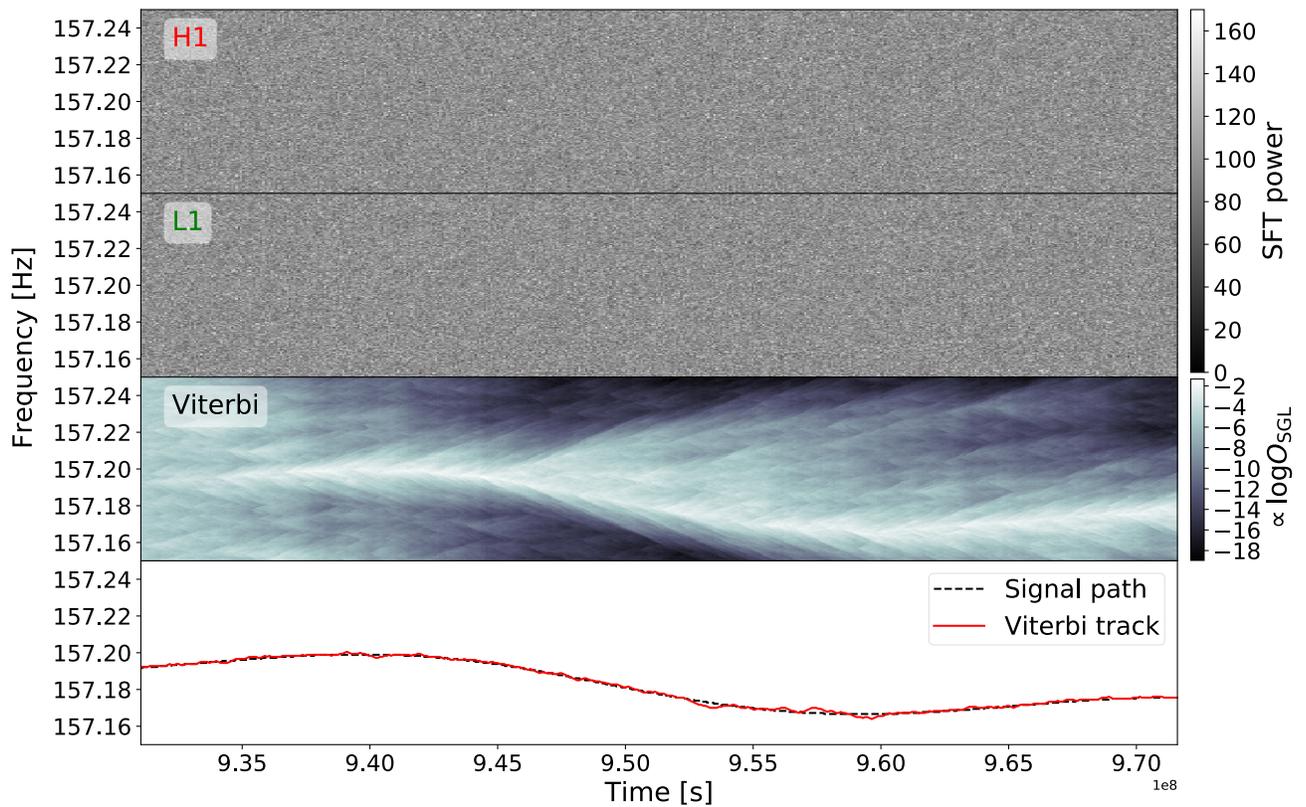


FIG. 2: The results that the SOAP algorithm returns from an injection with an optimal SNR of 90, i.e., the SNR in H1 is 64 and the SNR in L1 is 62. The signal is injected into Gaussian noise, where the 1800 s SFTs have been summed over 1 day. The top panel shows a simulation of summed SFTs from H1, the second panel shows the same for L1, the third panel shows the values proportional to the log-odds ratios in Eq. 22. 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.

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. 3b and 3d, 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.

B. S6 injections into Gaussian noise with gaps

In the second test, we attempt to more closely mirror the S6 MDC [28] in two stages. The first uses the same injection method as Sec. III A 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 $\sim 50\%$ [32], 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

TABLE I: 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

to the single-detector statistic using the same parameters, these are both defined in defined in Sec. IIF.

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

We set a 1% false alarm rate and generate an efficiency curve for SNR and depth in Fig. 3a and Fig. 3c 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. I.

In Fig. 3b and 3d 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).

C. 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 LIGOs 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. III, whereafter normalisation, the data is split into 0.1 Hz wide sub-bands overlapping by 0.05 Hz.

The two detector SOAP algorithm using the line-aware statistic in Sec. IIF 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. I.

As in Sec. IIIB, 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. IIF 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 [24]. 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. IIIB and IIIA.

Fig. 3c and Fig. 3a 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 [28]. Whilst we only search in the 100 - 200 Hz range, the closest comparison in [28] is the test in the 40 - 500 Hz range, such as in Fig. 4 in [28]. Here our algorithm sits roughly in the middle of all other searches in terms of sensitivity.

D. 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,

$$N_{\text{calcs}}^{(1)} = n_1^m NM, \quad (37)$$

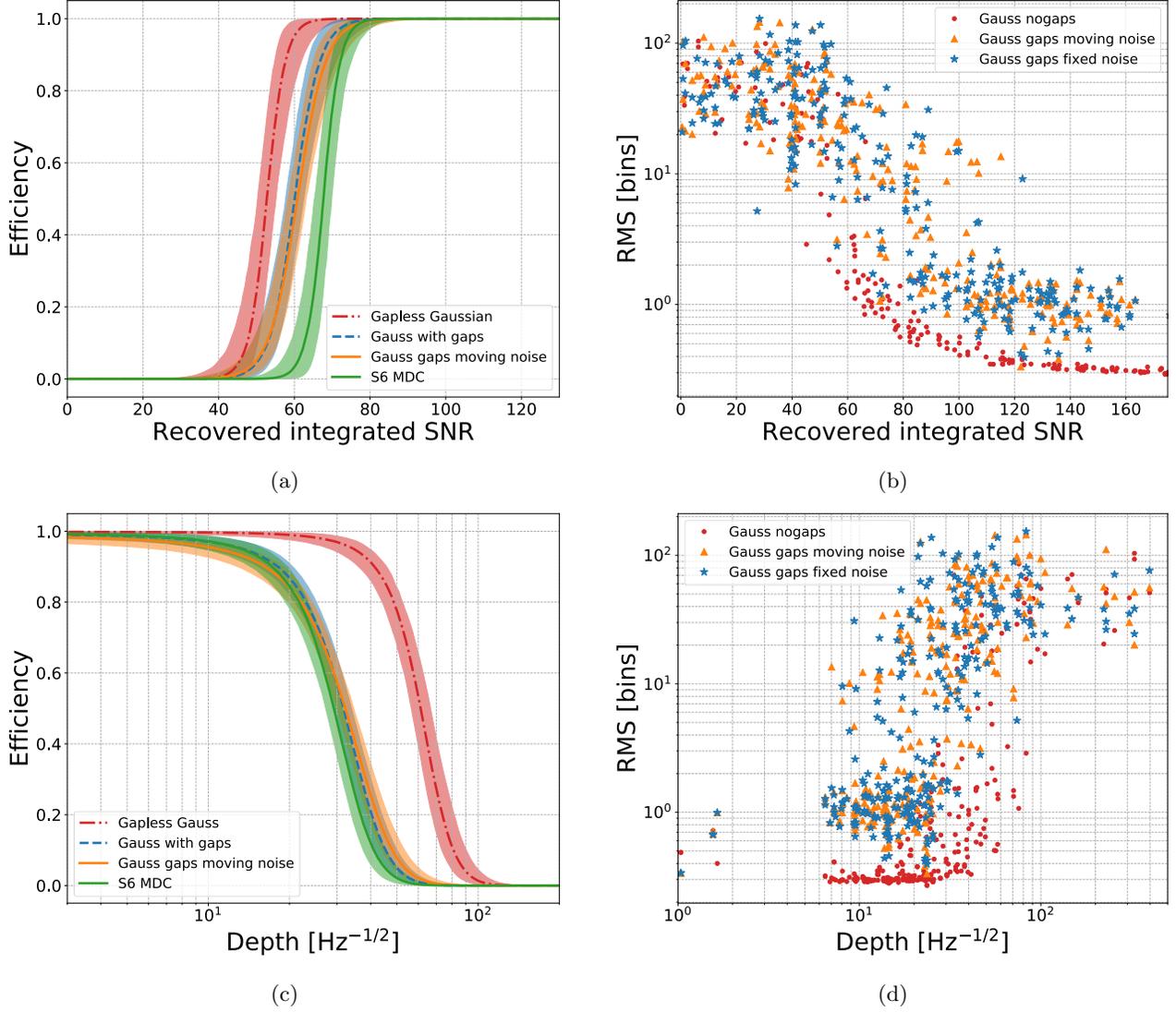


FIG. 3: Panels 3a and 3c show the detection efficiency as a function of SNR and depth respectively. Here SNR is the the integrated SNR which we would expect to recover from the available data. The four curves refer to injections into gapless Gaussian noise (red), Gaussian noise with gaps in data, where 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. 32 to binomial detection data with a 1% false alarm rate, as explained in Sec. III A, 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 3b and 3d 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.

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. II D. 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 detec-

tor 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 write the time taken

, T , as,

$$T = 0.56 \text{ sec} \left(\frac{N}{22538} \right) \left(\frac{M}{180} \right) \left(\frac{N_{\text{bands}}}{1} \right), \quad (38)$$

where N_{bands} is the number of different frequency bands. For the multiple, Q , detector case, we can then generalise Eq. 37 and write the number of calculations $N_{\text{calcs}}^{(Q)}$ as,

$$N_{\text{calcs}}^{(Q)} = NMn_1^m \prod_{q=1}^Q n_{q+1}, \quad (39)$$

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,

$$T = 1.35 \text{ s} \left(\frac{N}{22538} \right) \left(\frac{M}{180} \right) \left(\frac{N_{\text{bands}}}{1} \right). \quad (40)$$

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

$$T = 25.7 \text{ s} \left(\frac{N}{22538} \right) \left(\frac{M}{180} \right) \left(\frac{N_{\text{bands}}}{1} \right). \quad (41)$$

Other searches, excluding Einstein@home which takes on the order of months to run (> 100 million core-hours [28]), take 1 – 10 million core-hours [28]. This search should take $\mathcal{O}(10^3)$ core-hours.

IV. 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 [28], 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. 39 in Sec. III D.

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.

V. ACKNOWLEDGEMENTS

We would like to acknowledge Dr Matt Pitkin for his help during this project, as well as the continuous wave working group of LIGO-Virgo Collaboration. This research is supported by the Science and Technology Facilities Council, G.W. and C.M. are supported by the Science and Technology Research Council (grant No. ST/L000946/1). The authors are grateful for computational resources provided by the LIGO Laboratory and Cardiff University and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459 and STFC grant ST/I006285/1.

-
- [1] B. P. Abbott *et al.*, Reports on Progress in Physics **72**, 076901 (2009).
 - [2] J. Aasi *et al.*, Classical and Quantum Gravity **32**, 074001 (2015).
 - [3] F. Acernese *et al.*, Classical and Quantum Gravity **25**, 114045 (2008).
 - [4] F. Acernese *et al.*, Classical and Quantum Gravity **32**, 024001 (2014).
 - [5] R. Prix, “Gravitational waves from spinning neutron stars,” in *Neutron Stars and Pulsars*, edited by W. Becker (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009) pp. 651–685.
 - [6] B. J. Owen, (2009), arXiv:0903.2603 [astro-ph.SR].
 - [7] R. J. Dupuis and G. Woan, Physical Review D - Particles, Fields, Gravitation and Cosmology **72** (2005), 10.1103/PhysRevD.72.102002, arXiv:0508096 [gr-qc].
 - [8] P. Astone, S. D’Antonio, S. Frasca, and C. Palomba, Classical and Quantum Gravity **27**, 194016 (2010).
 - [9] P. Jaranowski, A. Królak, and B. F. Schutz, Phys. Rev. D **58**, 063001 (1998).
 - [10] L. S. Collaboration, V. Collaboration, S. Buchner, I. Cognard, A. Corongiu, P. C. C. Freire, L. Guillemot, G. B. Hobbs, M. Kerr, A. G. Lyne, A. Possenti, A. Ridolfi, R. M. Shannon, B. W. Stappers, and P. Weltevrede, The Astrophysical Journal **839**, 12 (2017).
 - [11] B. P. Abbott *et al.* (LIGO Scientific, Virgo), (2019), arXiv:1902.08507 [astro-ph.HE].
 - [12] P. R. Brady and T. Creighton, Phys. Rev. D **61**, 082001 (2000).
 - [13] B. P. Abbott *et al.* (LIGO Scientific, Virgo), (2019), arXiv:1903.01901 [astro-ph.HE].
 - [14] D. R. Ellis, J. Heffernan, S. Gutierrez, and D. Dalessandro, “Snakes on a plane,” (2006).
 - [15] A. J. Viterbi, IEEE Transactions on Information Theory **13**, 260 (1967).
 - [16] S. Suvorova, L. Sun, A. Melatos, W. Moran, and R. J. Evans, Physical Review D - Particles, Fields, Gravitation and Cosmology **93**, 1 (2016), arXiv:1606.02412.
 - [17] L. Sun, A. Melatos, S. Suvorova, W. Moran, and R. J. Evans, Phys. Rev. D **97**, 043013 (2018).
 - [18] S. Suvorova, P. Clearwater, A. Melatos, L. Sun, W. Moran, and R. J. Evans, Phys. Rev. D **96**, 102006 (2017).
 - [19] B. P. Abbott *et al.* (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. D **95**, 122003 (2017).
 - [20] B. P. Abbott *et al.* (LIGO Scientific, Virgo), (2018), arXiv:1810.02581 [gr-qc].
 - [21] L. Sun and A. Melatos, arXiv e-prints, arXiv:1810.03577 (2018), arXiv:1810.03577 [astro-ph.IM].
 - [22] G. L. Bretthorst, *Springer-Verlag* (1988) p. 220.
 - [23] P. Jaranowski, A. Królak, and B. Schutz, Physical Review D **58**, 38 (1998), arXiv:9804014 [gr-qc].
 - [24] M. Coughlin, the Ligo Scientific Collaboration, and the Virgo Collaboration, Journal of Physics: Conference Series **243**, 012010 (2010).
 - [25] J. Aasi *et al.*, Classical and Quantum Gravity **32**, 115012 (2015).
 - [26] P. B. Covas *et al.* (LSC Instrument Authors), Phys. Rev. D **97**, 082002 (2018).
 - [27] D. Keitel, R. Prix, M. A. Papa, P. Leaci, and M. Siddiqi, Phys. Rev. D **89**, 064023 (2014).
 - [28] S. Walsh, M. Pitkin, M. Oliver, S. D’Antonio, V. Dergachev, A. Królak, P. Astone, M. Bejger, M. Di Giovanni, O. Dorosh, S. Frasca, P. Leaci, S. Mastrogiovanni, A. Miller, C. Palomba, M. A. Papa, O. J. Piccinni, K. Riles, O. Sauter, and A. M. Sintes, Physical Review D **94** (2016), 10.1103/PhysRevD.94.124010, arXiv:1606.00660.
 - [29] LIGO Scientific Collaboration, “LIGO Algorithm Library - LALSuite,” free software (GPL) (2018).
 - [30] B. Behnke, M. A. Papa, and R. Prix, Physical Review D - Particles, Fields, Gravitation and Cosmology **91**, 1 (2015), arXiv:1410.5997.
 - [31] R. Prix, Physical Review D - Particles, Fields, Gravitation and Cosmology **75** (2007), 10.1103/PhysRevD.75.023004, arXiv:0606088 [gr-qc].
 - [32] J. Aasi *et al.*, Classical and Quantum Gravity **32**, 115012 (2015).