IOP PUBLISHING CLASSICAL AND QUANTUM GRAVITY

Class. Quantum Grav. **25** (2008) 105024 (13pp) doi:10.1088/0264-9381/25/10/105024

A Bayesian coincidence test for noise rejection in a gravitational-wave burst search*

Kipp C Cannon[1]

LIGO Laboratory 18-34, California Institute of Technology, 1200 E California Blvd, Pasadena, CA 91125, USA

E-mail: kcannon@ligo.caltech.edu

Received 13 January 2008, in final form 5 April 2008
Published 7 May 2008
Online at stacks.iop.org/CQG/25/105024

Abstract

In searches for gravitational-wave bursts, a standard technique used to reject noise is to discard burst event candidates that are not seen in coincidence in multiple detectors. A coincidence test in which Bayesian inference is used to measure how noise-like a tuple of events appears is presented here. This technique is shown to yield higher detection efficiencies for a given false alarm rate than do techniques based on per-parameter thresholds when applied to a toy model covering a broad class of event candidate populations. Also presented is the real-world example of a use of the technique for noise rejection in a time–frequency burst search conducted on simulated gravitational-wave detector data. Besides achieving a higher detection efficiency, the technique is significantly less challenging to implement well than is a per-parameter threshold method.

PACS numbers: 95.75.Pq, 95.85.Sz

1. Introduction

In searches for gravitational-wave bursts in the output of gravitational-wave antennae like those in LIGO [1], and also in searches for other discrete gravitational-wave signals like black hole ring downs, compact binary inspirals, etc, a standard technique used to reject noise is to discard events that are not seen in coincidence in multiple detectors. See, for example, the searches for gravitational waves described in [2–14], which all employ multi-instrument coincidence tests to suppress the false alarm rate. In coincidence tests such as those used in the searches listed above, each event found in the output of a detector is analyzed and some number of its physical properties measured and recorded. The coincidence criterion is to demand agreement

[*] LIGO-P070085-00-Z.

[1] Research conducted while at the Department of Physics, University of Wisconsin–Milwaukee, PO Box 413, Milwaukee, WI 53201-0413, USA.

in the physical properties of the events collected from different instruments. For example, in the procedures used in the searches listed above it was demanded that the times of the events not differ from one another by more than some amount. The rationale behind the use of a coincidence test to reduce the false alarm rate is the expectation that the physical properties of events that are the result of noise in the environment or in the instrument will be uncorrelated across instruments, while the physical properties of events resulting from genuine gravitational waves will be correlated across instruments.

The details of a coincidence test lie in how a 'match' is defined. We can state the problem of how to select the coincidence criteria in the following way. Let us consider that we have before us a tuple of events, collected from some number of instruments. For example, from a three-detector network we might have a triple comprised of one event from each detector. How many instruments and how many events from each instrument is not important, but we assume we are able to measure the physical properties of the events in the tuple, and from those properties we assign to the tuple the n parameter values x_1 through x_n. Let us consider these n parameters to describe a point in an n-dimensional space,

$$\vec{x} = (x_1, x_2, \ldots, x_n). \tag{1}$$

The question we wish a coincidence test to answer for us is 'is the tuple of events described by the parameters $\vec{x}$ the result of a gravitational wave?' In the gravitational-wave searches listed above, this question was answered by thresholding on parameter differences, for example thresholding on the difference in the arrival times estimated from multiple instruments, or thresholding on the difference in the estimated waveform energies and so on.

This paper presents an alternative technique, introduced in section 2, based on statistical inference. Instead of defining thresholds on various parameters, the parameters' distribution densities will be measured and used to rank tuples from most to least like a gravitational wave. In section 3, a toy model of a gravitational-wave search is introduced and used to analyze the statistical-inference-based coincidence test, comparing its performance to the traditional process of setting thresholds on the parameters used in the test. This is followed in section 4 with an example of the technique in a real burst search pipeline.

2. A Bayesian coincidence test

If we can collect examples of tuples of events that we know for certain to be the results of noise, and also examples of tuples of events that we know for certain to be gravitational waves, then we can make use of statistical inference techniques to estimate the probability that the tuple of events whose properties are described by $\vec{x}$ belongs to one set or the other. We can use Bayes' theorem to do this. If we denote the tuple of events as T, the set of tuples that are the results of gravitational-wave bursts as S, and the set of tuples that are the results of noise as N, then Bayes' theorem [15] states that

$$P(T \in S|\vec{x}) = \frac{P(\vec{x}|T \in S)P(T \in S)}{P(\vec{x})}. \tag{2}$$

To be clear about the notation, on the left-hand side we have the probability that the tuple T is in the set S of real gravitational-wave events given the measured parameters of the tuple $\vec{x}$. On the right-hand side, in the numerator, we have the probability of observing the parameters $\vec{x}$ in a tuple of events known to be the result of a gravitational wave multiplying the probability that any tuple chosen at random is a gravitational wave. In the denominator, we have the probability of observing the parameters $\vec{x}$ at all, in any kind of event. If the parameters are continuous, then both $P(\vec{x}|T \in S)$ and $P(\vec{x})$ are distribution densities but the same equation holds.

Because each tuple of events is either the result of noise or of a gravitational wave,

$$P(T \in N) = 1 - P(T \in S), \tag{3}$$

and also

$$\begin{aligned} P(\vec{x}) &= P(\vec{x}|T \in S)P(T \in S) + P(\vec{x}|T \in N)P(T \in N) \\ &= P(\vec{x}|T \in N) + [P(\vec{x}|T \in S) - P(\vec{x}|T \in N)]P(T \in S). \end{aligned} \tag{4}$$

Therefore,

$$P(T \in S|\vec{x}) = \frac{P(\vec{x}|T \in S)P(T \in S)}{P(\vec{x}|T \in N) + [P(\vec{x}|T \in S) - P(\vec{x}|T \in N)]P(T \in S)}. \tag{5}$$

From this expression, it can be shown that $P(T \in S|\vec{x})$ is a monotonically increasing function of

$$\Lambda(\vec{x}) = \frac{P(\vec{x}|T \in S)}{P(\vec{x}|T \in N)}, \tag{6}$$

known as the likelihood ratio. Evaluation of the likelihood ratio $\Lambda(\vec{x})$ does not require knowledge of $P(T \in S)$, the *a priori* probability that a tuple chosen at random is the result of a gravitational wave, something equivalent to knowing how many of the tuples (just not which ones) are gravitational waves, which is information we do not have.

Computing the likelihood ratio for each tuple allows us to rank them from the tuple most likely to be the result of a gravitational wave to the tuple that is least likely to be a gravitational wave, and this forms the basis of the Bayesian coincidence test.

The procedure is the following. We begin by choosing a parameter space with which to describe a multi-instrument tuple of events. How to do so for a particular gravitational-wave burst search algorithm is beyond the scope of this paper. We then need two populations of event tuples: a population of tuples known to be the result of gravitational waves, and a population of tuples known to be the result of noise. We cannot obtain such populations, indeed if we possessed the ability to do so then there would be no point to this current exercise. Instead, if we apply to the events from each instrument a large, random, per-instrument time offset, we can use the event tuples collected from this 'time-shifted' data set as a surrogate for tuples we know to be the result of noise. Then, we can use software simulations to inject synthetic gravitational-wave signals into real instrument data streams, and use the tuples that are collected from these injections as surrogates for those resulting from genuine gravitational waves. Software injections and time shifts are standard techniques in trigger-based searches for gravitational-wave signals. See, for example, the searches cited in the introduction.

For each tuple we measure its parameters $\vec{x}$. From the parameters of the tuples obtained from software injections we measure the distribution density $P(\vec{x}|T \in S)$, and from the parameters of the tuples obtained from time-shifted data we measure the distribution density $P(\vec{x}|T \in N)$. We now assign a likelihood ratio, $\Lambda(\vec{x})$, to every tuple of events by measuring its parameters $\vec{x}$ and computing the ratio in (6) using the two distributions we have just measured. We can do this for all of the injection tuples, and all of the time slide tuples. We can also, now, proceed to collect tuples of events from the 'zero lag', or the data with no time offsets applied, and compute and record the likelihood ratio for each of these tuples as well.

The value of the likelihood ratio assigned to each tuple is a measure of how the injection-like tuple appears to be, and there are a number of possible coincidence tests that can be implemented with this information. One easy possibility is to sort the zero-lag tuples from highest to lowest value of their likelihood ratios, choose the number of them we wish to retain, keep that many from the high end of the list and discard the rest. This is easy to implement, but has the disadvantage that it is a relative measure of quality: how much like a software injection a zero-lag event needs to appear in order to survive the cut depends on what other zero-lag events are in the list.

An absolute quality scale can be established using the time slide, or 'background', tuples. To do so, the total observation time analyzed in the background, t_{b}, is computed, as is the observation time analyzed at zero lag, t_{f}. The desired number of zero-lag events, $\langle N_{\mathrm{f}}\rangle$, is multiplied by the ratio of the background-to-zero-lag observation times, and the likelihood ratio threshold is found for which that many time slide tuples are retained. Discarding any zero-lag tuple whose likelihood ratio is below this threshold will leave some unknown number of survivors. If the zero lag consists exclusively of noise events then on average there will be $\langle N_{\mathrm{f}}\rangle$ events surviving this coincidence cut, but there could be any number at all more than this if the zero-lag tuples contain a population of genuine gravitational-wave events.

3. Justification

The justification for the use of Bayesian inference as a coincidence test for noise rejection follows from the analysis of the behavior of the method as the number of parameters used for event comparison increases. Let us consider a simple model in which we have n dimensionless parameters x_i, all restricted to the same domain

$$|x_i| \leqslant X, \tag{7}$$

where $X \gg 1$. In this model, let x_i from injection tuples be found to be n independently-distributed Gaussian (within their domains) random variables with means of 0 and unit variance so that

$$P(\vec{x}|T \in S) \approx \frac{1}{(2\pi)^{n/2}} \mathrm{e}^{-\frac{1}{2}\vec{x}\cdot\vec{x}/n^2} \tag{8}$$

(assuming X is sufficiently large that the normalization error is irrelevant). From the noise tuples, let x_i be found to be n independently-distributed random variables with uniform density over their domains so that

$$P(\vec{x}|T \in N) = \frac{1}{(2X)^n}. \tag{9}$$

The assumption that x_i are confined to finite domains is almost certain to be true in any real application since the mechanism by which potential gravitational-wave tuples are identified for consideration by the coincidence test must involve the requirement of at least some sort of loose agreement among the constituent events, for otherwise the combinatorics become prohibitive. The assumption that the parameters are independently distributed is justified because if this was not the case, if one of the parameters was strongly correlated with another then including it in the coincidence test would not be adding additional information about the tuple. It is reasonable to assume the researcher has sought out parameters for use in the coincidence that are independent of one another. The assumption of Gaussianity in the injection tuples can presumably be made approximately true in any real application through a straightforward transformation, although this is unlikely to leave the noise tuples with uniformly-distributed parameters. In the end, however, the coincidence test will reject all but the 'best' tuples which tend to lie near $\vec{x} = 0$, where it is probably the case that the distributions are approximately Gaussian and flat respectively anyway. Note that I am only asserting the relevance of this simplified model that has been introduced to perform the analysis below. It is not necessary for these assumptions to hold in order to use the technique in a real application.

Let us now compare the Bayesian coincidence test to a coincidence test in which a set of thresholds

$$|x_i| \leqslant \Delta x_i \tag{10}$$

is imposed on the parameters. Imposing per-parameter thresholds is typical of the coincidence tests in use in many searches for gravitational waves. Let the up-stream event generator yield tuples in noise at a rate R, and we wish our coincidence test to sieve the tuples down to the final false alarm rate $r < R$. For the noise tuples with their independent uniformly-distributed parameters, the probability that all n parameters are within the allowed ranges is

$$P(\text{noise survives}) = X^{-n} \prod_{i=1}^{n} \Delta x_i. \tag{11}$$

To achieve the target false alarm rate, the probability that a tuple survives coincidence must be r/R, so

$$\prod_{i=1}^{n} \Delta x_i = \frac{r}{R} X^n. \tag{12}$$

The probability of a software injection surviving the same coincidence test, the coincidence test's efficiency, is

$$\varepsilon = \prod_{i=1}^{n} \operatorname{erf} \frac{\Delta x_i}{\sqrt{2}}, \tag{13}$$

which is maximized (the search is given the highest detection efficiency) by choosing the same threshold for all parameters

$$\Delta x_i = \Delta x = X \left(\frac{r}{R} \right)^{\frac{1}{n}}. \tag{14}$$

Therefore, for fixed false alarm rate r, the detection efficiency achieved using per-parameter thresholds is

$$\varepsilon = \left[\operatorname{erf} \frac{X}{\sqrt{2}} \left(\frac{r}{R} \right)^{\frac{1}{n}} \right]^n. \tag{15}$$

We can invert this to express the size of the parameter space X in terms of the detection efficiency ε and the false alarm survival rate r/R,

$$X = \sqrt{2} \left(\frac{r}{R} \right)^{-\frac{1}{n}} \operatorname{erf}^{-1} \varepsilon^{\frac{1}{n}}. \tag{16}$$

This will be used below to relate the detection efficiency of the likelihood-ratio-based coincidence test to that of the per-parameter-threshold-based test.

Ignoring an irrelevant proportionality constant, the likelihood ratio function can be written as

$$\Lambda(\vec{x}) = \mathrm{e}^{-\frac{1}{2} \vec{x} \cdot \vec{x} / n^2}. \tag{17}$$

In the $\vec{x}$ space, the surfaces of constant $\Lambda(\vec{x})$ are $(n-1)$-spheres centered on the origin whose enclosed volumes are [16]

$$\frac{\pi^{n/2}}{\Gamma(n/2+1)} (\vec{x} \cdot \vec{x})^{n/2}. \tag{18}$$

The values of $\vec{x}$ from noise tuples are uniformly distributed over their domain, the volume of which is $(2X)^n$. Therefore, the radius squared that encloses a fraction r/R of the noise tuples is

$$\vec{x} \cdot \vec{x} = \frac{4X^2}{\pi} \left[\frac{r}{R} \Gamma(n/2+1) \right]^{2/n}. \tag{19}$$

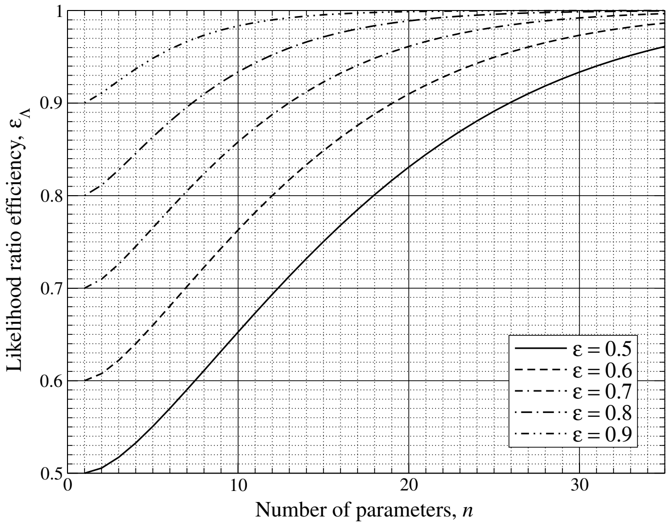

Figure 1. Comparison of the detection efficiencies of the likelihood-ratio-based and parameter-threshold-based coincidence tests. The plot is of ε_Λ, the probability that an injection survives the likelihood-based coincidence test, as a function of the number of parameters used in the coincidence comparison for a variety of values of ε, the probability of the same injection surviving the parameter-threshold-based coincidence test.

In the injection tuples, $\vec{x} \cdot \vec{x}$ is the sum of the squares of n-independent Gaussian-distributed random variables of unit variance, and so is a χ^2-distributed random variable with n degrees of freedom whose cumulative distribution function is

$$F(\vec{x} \cdot \vec{x}; n) = \frac{\gamma\left(n/2, \frac{1}{2}\vec{x} \cdot \vec{x}\right)}{\Gamma(n/2)}, \tag{20}$$

where γ is the lower incomplete gamma function. Therefore, the probability of an injection surviving the Bayesian coincidence test is

$$\varepsilon_\Lambda = \frac{1}{\Gamma(n/2)} \gamma\left(\frac{n}{2}, \frac{2X^2}{\pi}\left[\frac{r}{R}\Gamma(n/2+1)\right]^{2/n}\right). \tag{21}$$

We can use (16) to relate this to the efficiency that is achieved by thresholding on each parameter individually,

$$\varepsilon_\Lambda = \frac{1}{\Gamma(n/2)} \gamma\left(\frac{n}{2}, \frac{4}{\pi}\left[\operatorname{erf}^{-1} \varepsilon^{\frac{1}{n}}\right]^2 \Gamma^{\frac{2}{n}}(n/2+1)\right). \tag{22}$$

Note that the false alarm rate does not appear in this relationship. In this toy model, the number of parameters used in comparing events is the only free parameter in the relationship between the efficiencies that are achieved by thresholding on individual parameters and those achieved by thresholding on the likelihood ratio.

A comparison of the performance of the likelihood-ratio-based coincidence test to the parameter-threshold-based coincidence test is shown in figure 1. This figure is a plot of ε_Λ, the detection efficiency achieved by the likelihood-ratio-based coincidence test in (21), as a function of the number of parameters used in the test for several values of ε, the detection efficiency achieved by the parameter threshold coincidence test. When a small number of parameters is used for event comparison the two techniques are essentially equivalent, but as the number of parameters is increased thresholding on the likelihood becomes an increasingly more significant improvement over single-parameter thresholds.

Despite the evidence this toy model provides in support of the likelihood-based coincidence test's greater sensitivity over a parameter-threshold-based coincidence test, it should be remarked that this toy model is, in many ways, actually favoring the parameter-threshold-based approach. Assuming the software injections result in parameters $\vec{x}$ that are nicely clustered around the origin, as this toy model does, makes setting simple thresholds on those parameters a sensible approach to coincidence analysis. In practice, it can be found that the software injections are not so nicely clustered around a single point in parameter space, indeed it can be found that software injections are found in disjoint regions of the parameter space. As the software injection parameter distribution becomes more complex, and in particular as it becomes multi-modal, one should expect that the likelihood ratio approach to event tuple selection will become a yet greater improvement over a set of per-parameter thresholds. The following section illustrates a more realistic example.

4. A practical example

To illustrate the method of noise rejection via Bayesian coincidence filtering described above, an example implementation has been constructed and applied to the event tuples recorded by a time–frequency 'excess power' burst search [17, 18] run on simulated strain data from the three LIGO instruments. The strain noise was modeled as a stationary white Gaussian process. The excess power burst search was able to measure a number of physical properties of each of the events it identified in the strain data. These include: t and f, estimates of the time and frequency at which the greatest part of the event's strain energy could be found; Δt and Δf, estimates of the event's duration and bandwidth; and h_{rss}, an estimate of the square root of the time integral of the event's strain-squared time series (a quantity related to the energy in the gravitational wave). The search was used to collect event triples for consideration by the Bayesian coincidence test. Each triple contained exactly one event from each of the three simulated LIGO instruments (H1, H2 and L1), and it was required that in these triples the time–frequency tiles representing the events all mutually intersect.

A population of software injections consisting of 214 485 linearly-polarized sine-Gaussian waveforms was used as a surrogate for real gravitational-wave bursts. These waveforms are given by

$$h(t) = h_{\mathrm{max}} \sin(2\pi f t) \exp\left[-\frac{(2\pi f t)^2}{2Q^2}\right], \tag{23}$$

where f is the injection's center frequency, h_{max} is the peak strain amplitude of the Gaussian envelope and Q sets the width of the envelope. The injections had Q held fixed at 8.89 to reproduce the injections used in a number of burst searches, such as [14]. The injections had random center frequencies uniformly distributed in $\log f/1\,\mathrm{Hz}$, and random amplitudes uniformly distributed in $\log h_{\mathrm{max}}$. The range for each parameter was selected to be appropriate for the search, in particular the amplitudes extended to values too small to be detected and sufficiently large as to be nearly guaranteed to be detected. For the time slides, the data from the simulated H1 and H2 instruments which are co-located at the LIGO Hanford Observatory in Hanford, Washington have no time offsets applied but the data from the simulated L1 instrument at the LIGO Livingston Observatory in Livingston, Louisiana were shifted with respect to the simulated H1 and H2 data by 200 different (nonzero) time offsets to synthesize a population of approximately 2×10^8 noise triple surrogates. The time offsets were integer multiples of $\sqrt{80}\,\mathrm{s}$.

In each event triple, the three events were taken pair-wise and five parameters extracted from each pair:

(1) the difference in their durations as a fraction of the average of the two durations,
(2) the difference in their bandwidths as a fraction of the average of the two bandwidths,
(3) the difference in their h_{rss} as a fraction of the average of the two,
(4) the difference in their peak frequencies as a fraction of the average of the two, and
(5) the difference in their peak times.

This resulted in a total of 15 physical quantities being extracted from each tuple of three events. In this example, it is important that the parameters all be 'differences', because the object is to turn this information into a coincidence test to reject triples whose constituent events differ too much from one another. Mechanically, however, nothing in the procedure requires the parameters to be differences. The particular choice of parameters given above was made empirically through observation of the output of the excess power burst search code.

That the parameter space has 15 dimensions creates a number of practical implementation problems. The first is that if the distribution densities are measured by binning the parameter space and counting the number of noise and software injection triples that are found in each bin, the number of bins required for this is very large. For example, placing just 20 bins along each coordinate axis requires a total of 3.3×10^{19} bins to span the entire volume of the parameter space, a number beyond the storage capabilities of present-day computers. Even if a computer could be found that could store that many numbers in memory or on disk, given the speed of present-day computers it would not be possible to collect enough noise events or perform enough software injections to measure the probability density in each bin with acceptable accuracy.

Luckily, investigation shows that the correlation matrix for the 15 parameters described above is nearly diagonal, meaning the parameters are nearly uncorrelated. The ratio of each of the off-diagonal elements in the normalized covariance matrix to the diagonal elements is $\sim$10%. Being uncorrelated is necessary but not sufficient to prove statistical independence, but since testing for statistical independence is difficult with so many parameters, let us treat the parameters as though they are statistically independent. This allows us to approximate the 15-dimensional likelihood ratio as

$$\Lambda(\vec{x}) \approx \prod_{i=1}^{n} \Lambda(x_i) = \prod_{i=1}^{n} \frac{P(x_i | T \in S)}{P(x_i | T \in N)}, \tag{24}$$

where $\Lambda(x_i)$ are the likelihood ratio functions measured for each of the $n = 15$ parameters individually (for notational simplicity, the different functions are identified by their arguments). It should be remarked that, mechanically, the replacement in (24) can be performed whether or not it is a good approximation. The quality of the coincidence test will be improved if this substitution is a good approximation, but machine limitations can require this substitution to be made regardless. In the event that the desired parameters are found to be significantly correlated, and machine limitations prevent the measurement of the parameter distributions throughout the full volume of parameter space, it might be possible to (at least approximately) diagonalize the correlation matrix through the construction of a coordinate transformation to a basis whose coordinates are not as correlated. Since a straightforward approximation scheme was available in this example application the problem has not been investigated further.

The probability densities appearing in the numerator and denominator of (24) are shown in figure 2. Some interesting features are visible in these plots. First, several show families of spikes of high event density, especially in the event triples associated with noise. These have their origin in internal discreteness in the search algorithm used to identify burst events in the

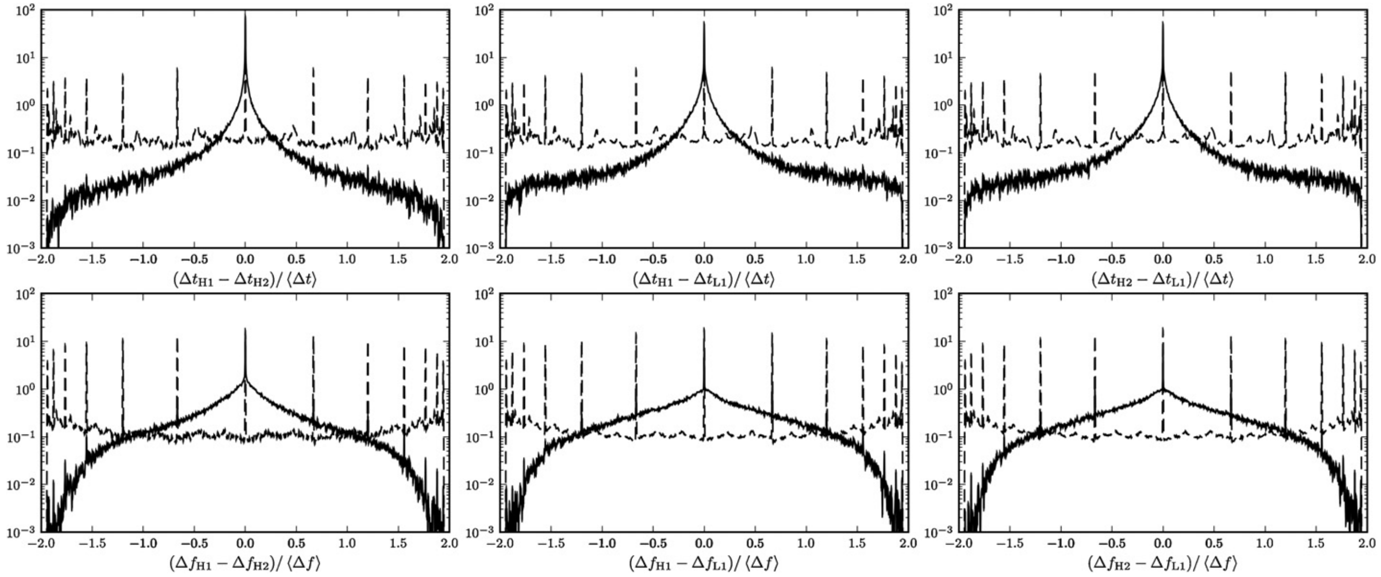

Figure 2. Distribution densities for each of the 15 parameters used for coincidence testing in event triples recorded by an excess power burst search analyzing simulated versions of the outputs of the three LIGO instruments, H1, H2 and L1. The dashed curves show the distribution densities as observed in noise, and the solid show them as observed in software injections. In all plots the vertical axis is probability density. The 15 parameters are found to be approximately uncorrelated, and using that to justify treating them as statistically independent the full joint distributions are given by the products of these 15 functions. The symbols have the following meanings. Δt is the duration of an event, Δf is the bandwidth of an event, h_{rss} is the square root of its integrated strain squared (a quantity related to the energy of the event), f is the event's peak frequency, and t its peak time. Visible in these plots is the entire volume of the parameter space (note in particular the non-linear binning used for the peak time differences).

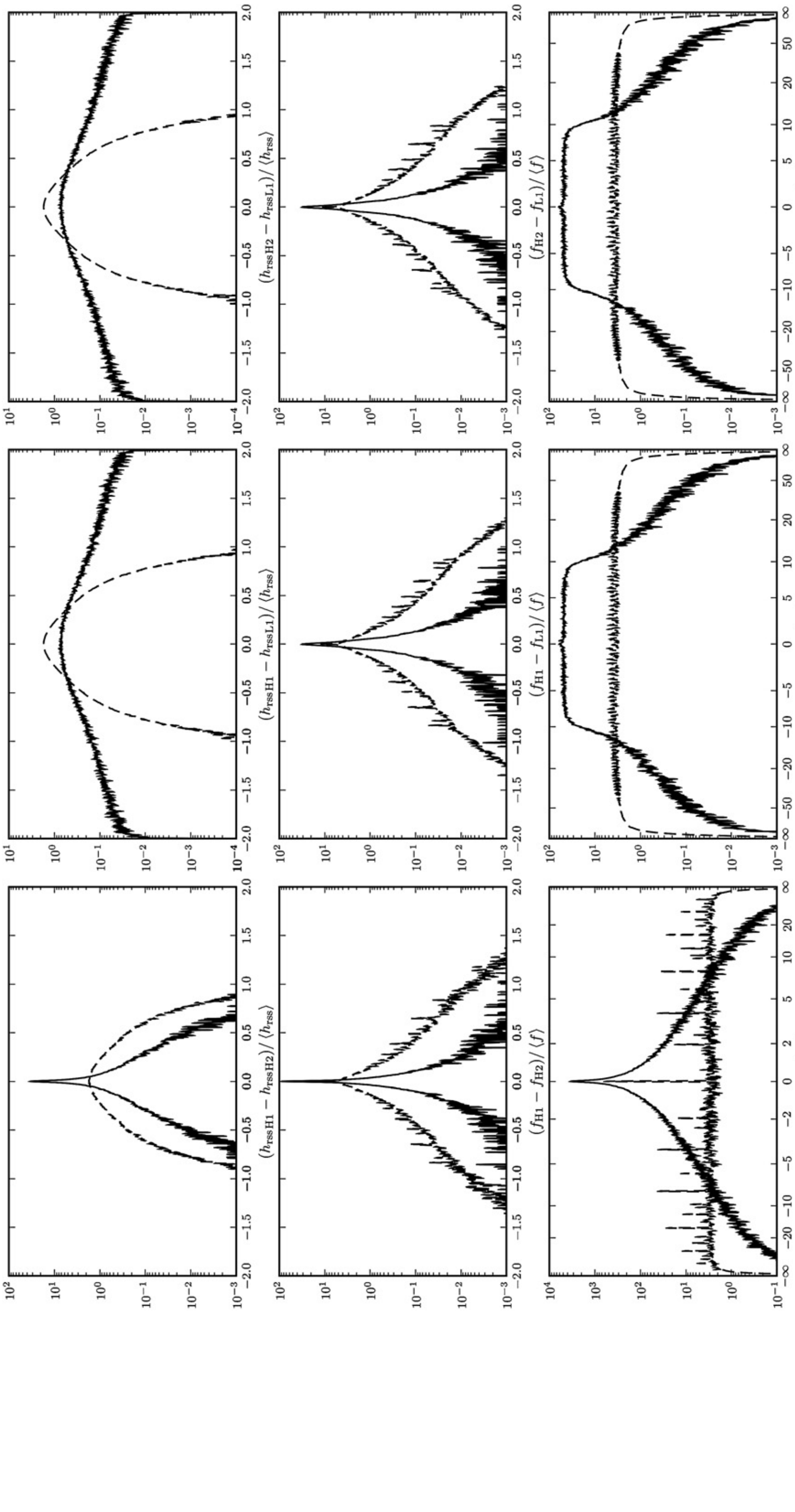

Figure 2. (Continued.)

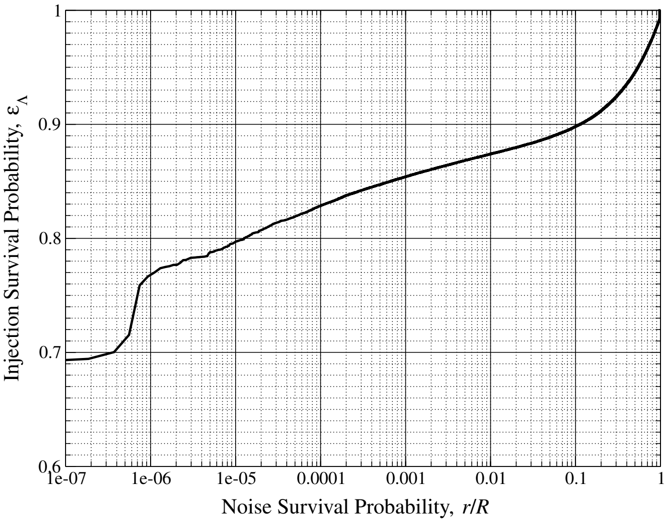

Figure 3. ROC curve for the example application. The left edge of the graph corresponds to the highest threshold, the strictest cut which only the best of the best noise and injections can survive. See the text for discussion.

time series. The algorithm tests the time series against a family of template waveforms, the discreteness of the template bank appears in the distribution of parameters seen in the final event candidates, and for a number of reasons this phenomenon is observed more strongly in noise events than software injections. The suppression of event triples whose parameters occur at these special, bad, values is a characteristic of thresholding on the likelihood ratio that is not easily reproduced by thresholding on individual parameters.

A second feature of note in these distributions is that the h_{rss} differences for the H1–L1 and H2–L1 pairs show that there are volumes of the parameter space in which injections can be found but no noise at all. The likelihood ratio assigned to injections in these regions of parameter space is ∞, and no threshold will ever cut them—they could be interpreted as guaranteed detections of gravitational waves. Unfortunately, this arises here from the simple noise model used in this demonstration, and is unlikely to be seen in a real application. Real instruments exhibit glitches in their outputs that will likely result in the black curves filling the parameter space.

Note also how the 15 parameters are not all equally useful in discriminating noise from injections. For some parameters the distribution densities for the noise and injection tuples are more similar to one another than they are for others. This is one of the strengths of this technique, which it does not require all parameters used for event comparison to be equally good at discriminating noise from injections. Weaker parameters, whose noise and injection distributions are similar, will contribute factors close to 1 in (24) and so contribute less weight to the final ranking of the tuples. In fact, even in the worst-case scenario in which one of the parameters proves to be no better than a random number generator it will simply be disregarded and not harm the fidelity of the ranking.

Having measured the 15 likelihood ratio functions, (24) can now be evaluated for any event triple. In particular, we can collect more noise event triples and do more software injections, and rank the two sets from the highest to the lowest value of their likelihood ratios. We can then apply a likelihood ratio cut to each of the two sets, and ask with what probability a noise triple and an injection triple survive the cut. By plotting the one probability against the other as the likelihood ratio threshold is varied we obtain the ROC curve. This is shown in figure 3. Because some injections are found in regions of parameter space where there are no noise event triples, the curve becomes horizontal at the left edge: there are injections that survive any cut, no matter how high the threshold is set. Similarly, some noise triples are found in regions of parameter space where there are no injection triples or arbitrarily few, so there are noise triples assigned likelihood ratios lower than that assigned to any injection triple. Therefore, the curve becomes vertical at high probabilities (although the logarithmic horizontal scale compresses the feature, making it difficult to see on this graph).

The step feature at the left of the graph is likely meaningless. At $r/R = 10^{-7}$ only $\sim$10 of the highest-ranked noise triples remain following the cut, which is likely too small a sample to conclude anything meaningful about the distribution of likelihood ratios assigned to events in this extreme end of the tail.

5. Concluding remarks

This paper has presented an approach to noise rejection in a multi-instrument gravitational-wave burst search in the form of a multi-event coincidence test based on statistical inference. The technique has the advantages over standard per-parameter threshold based coincidence tests of achieving higher detection probability for a given false alarm rate, being easier to implement, and also being self-tuning and thus less laborious. A toy model has been introduced to understand the technique's behavior, and a practical implementation has been tested using populations of noise and software injection events recovered from a real gravitational-wave burst search pipeline run on stationary white Gaussian noise.

Bayesian inference has been used previously in constructing burst search detection statistics, for example in [19]. There the authors describe designing the waveform basis onto which the data are projected at the event identification stage of a pipeline in a manner that incorporates realistic priors on both the signals being searched for and the noise in the detector network. The technique presented here is closely related to that technique. Instead of using detailed noise and signal models to control the construction of a waveform basis for the event identification stage of the pipeline, in the technique presented here a naive waveform basis is used for event identification (typically having been derived from the assumption that the noise is stationary and Gaussian) and afterwards realistic noise and signal priors are used to re-rank the list of events generated by the pipeline.

The generation of the event lists, particularly the hundreds of time-shifted event lists used as a noise surrogate, can require considerable computer time and disk resources. The use of a naive waveform basis for initial event identification followed by a noise and signal model-dependant re-ranking allows multiple signal models to be considered without the need to regenerate the noise tuple list. Applying detailed noise and signal models at an earlier stage of the pipeline might yield higher detection efficiencies for a given false alarm rate, but the savings in computation time and storage from applying it later can be significant and might produce different kinds of improvements such as the ability to collect higher number statistics on injections and noise in order to measure rate upper limits more accurately.

Acknowledgments

I wish to thank Dr Ruslan Vaulin and Professor Jolien Creighton, University of Wisconsin-Milwaukee, for useful discussions and inspiration. This work was supported in part by the National Science Foundation under grant numbers PHY-0200852 and PHY-0421416.

References

[1] S J Waldman (for the LIGO Science Collaboration) 2006 Status of LIGO at the start of the fifth science run *Class. Quantum Grav.* **23** S653–60

[2] The International Gravitational Event Collaboration 2000 Initial operation of the International Gravitational Event Collaboration *Int. J. Mod. Phys.* D **9** 237–45 (*Preprint* astro-ph/0003106v1)

[3] International Gravitational Event Collaboration (IGEC) 2000 First search for gravitational wave bursts with a network of detectors *Phys. Rev. Lett.* **85** 5046–50 (*Preprint* astro-ph/0007308v2)

[4] International Gravitational Event Collaboration 2003 Methods and results of the IGEC search for burst gravitational waves in the years 1997–2000 *Phys. Rev.* D **68** 022001 (*Preprint* astro-ph/0302482v1)

[5] LIGO Scientific Collaboration 2004 First upper limits from LIGO on gravitational wave bursts *Phys. Rev.* D **69** 102001 (*Preprint* gr-qc/0312056v4)

[6] LIGO Scientific Collaboration 2004 Analysis of LIGO data for gravitational waves from binary neutron stars *Phys. Rev.* D **69** 122001 (*Preprint* gr-qc/0308069v1)

[7] LIGO Scientific Collaboration 2005 Upper limits on gravitational wave bursts in LIGO's second science run *Phys. Rev.* D **72** 062001 (*Preprint* gr-qc/0505029v1)

[8] LIGO Scientific Collaboration 2005 Search for gravitational waves from galactic and extra-galactic binary neutron stars *Phys. Rev.* D **72** 082001 (*Preprint* gr-qc/0505041v2)

[9] LIGO Scientific Collaboration 2005 Search for gravitational waves from primordial black hole binary coalescences in the galactic halo *Phys. Rev.* D **72** 082002 (*Preprint* gr-qc/0505042v1)

[10] LIGO Scientific Collaboration 2005 Upper limits from the LIGO and TAMA detectors on the rate of gravitational-wave bursts *Phys. Rev.* D **72** 122004 (*Preprint* gr-qc/0507081v3)

[11] LIGO Scientific Collaboration 2006 Search for gravitational waves from binary black hole inspirals in LIGO data *Phys. Rev.* D **73** 062001 (*Preprint* gr-qc/0509129v1)

[12] LIGO Scientific Collaboration 2006 Search for gravitational-wave bursts in LIGO's third science run *Class. Quantum Grav.* **23** S29–39 (*Preprint* gr-qc/0511146v1)

[13] LIGO Scientific Collaboration 2006 Joint LIGO and TAMA300 search for gravitational waves from inspiralling neutron star binaries *Phys. Rev.* D **73** 102002 (*Preprint* gr-qc/0512078v2)

[14] LIGO Scientific Collaboration 2007 Search for gravitational-wave bursts in LIGO data from the fourth science run *Class. Quantum Grav.* **24** 5343–69 (*Preprint* 0704.0943)

[15] Sivia D S 1996 *Data Analysis—A Bayesian Tutorial* (Oxford: Oxford University Press)

[16] Zeidler E (ed) 2004 *Oxford Users' Guide to Mathematics* (New York: Oxford University Press)

[17] Anderson W G, Brady P R, Creighton J D E and Flanagan É É 2001 Excess power statistic for detection of burst sources of gravitational radiation *Phys. Rev.* D **63** 042003 (*Preprint* gr-qc/0008066)

[18] Allan B *et al* (LSC Algorithm Library (LAL), LIGO Scientific Collaboration) 2007 CVS version tag LIGO_P070085_00_Z was used for this analysis

[19] Searle A C, Sutton P J, Tinto M and Woan G 2008 Robust Bayesian detection of unmodelled bursts *Preprint* 0712.0196 (to appear in *Class. Quantum Grav.*)