Model selection and data quality for core-collapse supernova waveform simulations

Sarah Wesolowski
Department of Physics, Saint Vincent College
300 Fraser Purchase Rd. Latrobe, PA 15650, United States

Dr. Ik Siong Heng
Institute for Gravitational Research, University of Glasgow
Glasgow G12 8QQ, United Kingdom

Abstract

We examine the performance of the Supernova Model Evidence Extractor (SMEE) algorithm on injected supernova waveforms in different types of noise. The algorithm performs well in the identification of injected waveforms added to simulated white Gaussian noise at LIGO S5 levels, ET levels, and ET colored Gaussian noise. However, the algorithm is not equipped for non-Gaussian noise scenarios. A clear bias is present when the algorithm is forced to choose between two models in GEO noise, but it cannot successfully identify injected signals with the current noise model. We conclude that the Gaussian noise model is not adequate in describing real situations such as GEO noise, and that the algorithm must be modified to accurately identify waveforms over noise.

1 Introduction

GEO 600 is the smallest of the gravitational wave detectors in the LSC/Virgo network, and it utilizes many novel techniques in order to reach comparable sensitivity to the other detectors of the network. These advanced techniques will eventually become a part of the upgraded "advanced" versions of the LIGO and Virgo detectors. GEO 600 is currently engaged in a planned science run from 2011 to 2015 while the LIGO and Virgo detectors are being upgraded to their advanced instruments. All of the upgrades necessary for GEO-HF have been implemented, and so GEO 600 is now operating at

its planned sensitivity [3]. The Einstein Telescope (ET) is a project concentrated on the development of a third generation gravitational wave detector. The project focuses on observational capabilities and possibilities that will be present after the projected detection of gravitational waves by the second generation detectors. The purpose of ET is to create an infrastructure that can host several underground gravitational wave detectors that will be continually upgraded as the associated technology improves[8].

The prospect of detection of gravitational waves is becoming a real possibility due to the current sensitivity upgrades of gravitational wave detectors around the world, and one promising candidate source for detection is a core-collapse supernova event. Several waveform catalogs have been produced from numerical simulations for core-collapse supernovae, and in this report we focus on two different numerical simulation catalogs. There are many proposed models for different types of wavebursts that result from supernova events, and these burst events range in duration from milliseconds to one second. Unlike the well-modeled predictions for such events as binary inspiral, core-collapse supernovae are a poorly modeled source of gravitational waves.

Numerical models for supernova waveforms are difficult to predict due to the complexity of the physical processes involved; these include particle physics, general relativity, and the several mechanisms involved in the supernova's lifespan. This also creates difficulty in detection, because even if the templates are correct a template based search is computationally impossible to perform on all data due to the complexity of the signal parameters. Singular Value Decomposition (SVD) allows us to decompose waveforms into a small set of basis vectors that effectively span the parameter space for a type of supernova waveform. We can then search for waveforms by reconstructing real signals with this small set of basis vectors that describe the waveform. We can perform Bayesian inference on the resulting reconstructed waveform to determine whether the signal is a good candidate and which catalog's Principal Component (PC) basis vectors best describe the waveform. Ideally this information will help us determine the physical processes that occurred in the detected waveform. It is projected that gravitational wave observation may be the only way that certain supernova events will be detected in the near future[6].

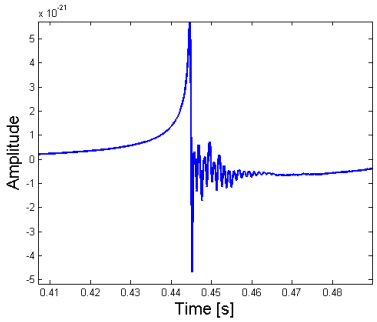
In this report we determine the success of a nested sampling Bayes factor algorithm, which determines the probability of one waveform model over another once a simulated waveform has been injected into GEO 600 noise. We inject waveforms from Abdikamalov *et. al* 2010 [2] and also waveforms from Dimmelmeier *et. al* 2008 [1] that both model the magnetorotational

mechanism from different progenitors. We also have tested various bandpass filters on both catalogs according to ET sensitivity in an attempt to simulate how ET would measure a signal.

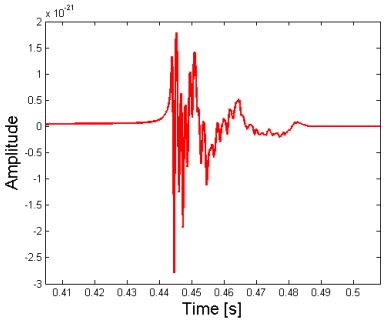
1.1 Supernova models

The main goals of Bayesian inference when applied to these waveform catalogs are to identify whether a signal is present and to determine which waveform catalog a signal corresponds. There are two models for the same mechanism that we primarily study in this report, and we determine our ability to differentiate between the two catalogs with the algorithm.

A white dwarf that accretes matter from a companion star usually results in a Type 1a supernova, but in the case of an oxygen-neon-magnesium white dwarf it is possible that accretion may trigger a gravitational collapse. In this case, the collapsing core will rebound in a core bounce, lead to the formation of a proto-neutron star, and then culminate in a collapse-driven supernova. This scenario results in a relatively small explosion energy and it would be difficult to observe with electromagnetic means. The use of gravitational wave searches for AIC waveforms can therefore aid in the detection of this type of event, which has not yet been detected either with gravitational or electromagnetic observations[2]. The magnetorotational mechanism is a process in which the magnetic field is amplified during the core collapse and post-bounce phases leading to a jet-like explosion that occurs along the axis of rotation. The Dimmelmeier catalog utilizes the same mechanism but for supernovae that occur in the life cycles of high mass stars[1]. The similarity in morphology between the two catalogs also gives the opportunity to test the algorithm's ability to distinguish similar waveforms from one another.



(a) Abdikamalov



(b) Dimmelmeier

1.2 Singular Value Decomposition

SVD allows us to decompose a catalog of signals into a relatively small number of basis vectors that effectively describe the catalog. In order to decompose the waveforms, we arrange a catalog into a matrix $\mathbf{H}$ in which the columns H_i represent the time series data of each waveform in the catalog. Let the matrix $\mathbf{H}$ have dimensions $M \times N$ where M is the number of waveforms and N is the number of time series data points. We can calculate the covariance matrix $\mathbf{C}$ of $\mathbf{H}$ with [4]

$$\mathbf{C} = \frac{1}{M}\mathbf{H}\mathbf{H}^T \tag{1}$$

The eigenvectors of the matrix $\mathbf{C}$ form a set of basis vectors that completely describe the parameter space of the catalog. The corresponding eigenvalues of these basis vectors describe how well the vectors span the parameter space. In practice, this allows us to use a relatively small number of basis vectors to reconstruct a waveform with reasonable accuracy. However, the calculation of determining the eigenvectors of a matrix is computationally expensive because the dimension N is quite large (in our case 32678, two seconds of data at the GEO and LIGO sampling rate of 16384 Hz). We avoid this by finding the eigenvectors $\mathbf{v}$ of $\mathbf{H}^T\mathbf{H}$ where

$$\mathbf{H}^T\mathbf{H}\mathbf{v}_i = \lambda_i\mathbf{v}_i \tag{2}$$

where $\mathbf{v}_i$ and λ_i are corresponding eigenvectors and eigenvalues, respectively. We then multiply both sides from the left by $\mathbf{H}$ so that

$$\mathbf{H}\mathbf{H}^T\mathbf{H}\mathbf{v}_i = \lambda_i\mathbf{H}\mathbf{v}_i \tag{3}$$

Let us rewrite eq. (1) in the form $\mathbf{C} = \mathbf{H}\mathbf{H}^T$ so that from eq. (2)

$$\mathbf{C}(\mathbf{H}\mathbf{v}_i) = \lambda_i(\mathbf{H}\mathbf{v}_i) \tag{4}$$

and we see that the eigenvectors of the covariance matrix are now $\mathbf{H}\mathbf{v}_i$. We can now calculate the eigenvectors of $\mathbf{C}$ using the smaller $M \times M$ matrix $\mathbf{H}\mathbf{H}^T$ because $M << N$. This method reduces the computation costs and we now have a set of basis vectors describing our catalog H [4].

We utilize for our purposes a linear model from Rover et. al given by [6]

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{5}$$

where $\boldsymbol{\epsilon}$ is the noise vector, $\boldsymbol{\beta}$ is the vector of principal component coefficients that we must determine with our algorithm, $\mathbf{X}$ is the matrix with

columns formed by our chosen number of principal components, and finally $\mathbf{y}$ is the vector that represents the waveform with noise added. We already know the matrix $\mathbf{X}$ and the data measurement $\mathbf{y}$ (in this case a simulated waveform injected into noise). In this model we assume the prior information to be a uniform distribution $P(\boldsymbol{\beta})$, which leads to our efforts in using Bayesian inference.

1.3 Bayes factor

Bayesian methods often provide practical solutions to problems that would otherwise be very difficult with normal statistical methods. One such method that we use to describe the success of a supernova waveform model is called the "Bayes factor," which allows us to weigh two hypotheses against one another- in this case, two competing models of waveforms or weighing a waveform against noise (a null hypothesis).

Let the signal that we receive from a detector be represented by the data D, and the two competing models (hypotheses) as H_1 and H_2. In general, Bayes theorem is given by [7]

$$p(\theta|D, H_n) = \frac{p(D|\theta, H_n) \times p(\theta|H_n)}{p(D|H_n)} \tag{6}$$

where $p(D|\theta, H_n)$ is known as the *likelihood* which represents the modification of the probability by the data collected, $p(\theta|H_n)$ is the *prior* which represents our state of knowledge about the model beforehand, $p(D|H_n)$ is the *evidence*, a quantity that takes a crucial role in model selection, and $p(\theta|D, H_n)$ is the *posterior* which represents our final state of knowledge about the truth of the model given the data measured. The posterior in this case is the probability distribution over the parameter space of θ, which is one of the principal component coefficients

The Odds Ratio O for our two models H_1 and H_2 is defined to be

$$O_{1,2} = \frac{p(D|H_1)}{p(D|H_2)} \times \frac{p(\theta|H_1)}{p(\theta|H_2)} \tag{7}$$

Thus the odds ratio is the product of the ratio of the two priors and the ratio of the two evidences. The favored model can be found depending on whether $O_{1,2}$ is greater than or less than unity. Often the two priors for each model are equivalent, so to determine the success of one model over another we use the ratio of the evidences, which we call the *Bayes factor*. In this application we will use the logarithm of the Bayes factor for clarity,

and one model will be favored over another based on the sign of the result. Thus the logarithm of the Bayes factor is given by

$$\log B = \log p(D|H_1) - \log p(D|H_2) \tag{8}$$

However, because the two models we have chosen are dependent on θ, we must integrate over the free parameter space to determine the evidences $p(D|H_1)$ and $p(D|H_2)$. In order to remove the dependence of $p(D|H_n)$ on θ, we utilize the technique of marginalization to integrate over all possible values of θ. This is given by [7]

$$p(D|H_n) = \int p(D|\theta, H_n)p(\theta|H_n)\,\mathrm{d}\theta \tag{9}$$

We cannot solve this analytically for our purposes, but we can circumvent this problem by using a nested sampling algorithm.

1.4 Nested sampling

The nested sampling algorithm allows us to find the evidence and the posterior given the prior and likelihood probability density functions. The advantage to this method is that it simultaneously searches for peaks in the distribution and computes the evidence integral in each dimension of the free parameter space of the PC basis vectors, and it is robust in the analysis of problems arising from a high dimensionality parameter space [9].

Let $Z = p(D|H_n)$, and let us also rewrite eq. (6) to clarify the calculation of the evidence integral of eq. (9) [9]

$$p(\theta|H_n) \times p(D|\theta, H_n) = Z \times p(\theta|D, H_n) \tag{10}$$

The requirement that both the prior and posterior be normalized by definition means that the magnitude of the evidence Z is determined by the likelihood $p(D|\theta, H_n)$. We must sum the product of the prior and the likelihood for every point in our free parameter space θ so that we can determine the evidence Z. We have stated that the integral that determines Z is not analytic, so we must use a subset of points in θ to approximate the result using a summation.

We will consider a stochastic sampling of the prior distribution to produce a set of N live points θ_k where $k = 1, 2, ...N$. We can now approximate the integral of eq. (5) with the summation

$$\sum_{k=1}^{N} L_k w_k \tag{11}$$

Where L_k is the likelihood $p(D|\theta_k, H_n)$ which represents a fraction of the prior over the kth sample and w_k denotes the *weight*

$$w_k = p(\theta_k|H_n)d\theta \tag{12}$$

We must now find the weights associated with each point θ_k. We can think of each point as lying on a contour surface of equal likelihood in the parameter space of θ, and the fraction of the prior enclosed by each contour surface is the prior mass, denoted X_k. We can now rewrite our summation in one dimesion as

$$Z = \int L(X)\,\mathrm{d}X = \sum_k L(X_k)\Delta X \tag{13}$$

We know that the prior distribution is normalized, so X_1 has a probability distribution which we equate to the distribution of a new variable $t \in [0, 1]$, which is the maximum of our N random numbers drawn from a uniform distribution. We continue to take points from the prior limited to a volume enclosed by X_k with a higher likelihood than L_1. We do this iteratively so that the volume of the prior enclosed shrinks geometrically. Thus the integral converges, and we can approximate the weight of each sample as $w_k = X_k - X_{k-1}$. We now need only to specify a condition where the integral terminates, and Veitch and Vecchio [9] have found by experience that it should continue while $L_{max}w_i > Z_i e^5$.

Our nested sampling algorithm calculates the logarithm of the evidence for a waveform reconstructed with a certain set of basis vectors that we choose beforehand. Once we have a value for that waveform with the basis vectors, it is a simple matter to subtract the log evidence of the waveform with other basis vectors or the evidence of the noise only to determine the Bayes factor B. We are able to determine in this way the success of the algorithm in correctly identifying a waveform. For example, if we have the log evidence of a waveform from the Abdikamalov catalog reconstructed with Abdikamalov basis vectors, we can compare it with the log evidence for the same waveform reconstructed with the Dimmelmeier basis vectors. We are interested in this but also that the algorithm does not incorrectly identify glitches and other noise as waveforms over the injected waveforms.

2 Project log

I will now outline the methods and progress of the research chronologically to give a context for the following section of results.

2.1 Using the nested sampling algorithm

The nested sampling and PCA algorithms are the main tools of this research, and so my first task was to learn how to use them properly and also to gain more experience in programming with Matlab. This was an important task because I had to be confident in modifying the architecture of a pre-written program. I was able to use the algorithm and I began simply by finding the Bayes factors for randomly selected Abdikamalov and Dimmelmeier waveforms against one another. This followed the methodology outlined in Logue 2011 [5]. I was able to successfully use the algorithm and its companion functions and expand it to use 6 PC basis vectors for extended accuracy.

2.2 Filtering waveforms

After assuring that I could use the algorithm with confidence, I began to test various bandpass filters on three different waveform catalogs. Filters allow us to simulate the effect of instrument sensitivity on the waveform simulations. My objective was to determine that the unfiltered basis vectors could faithfully reconstruct a filtered waveform. I utilized the method of the *match parameter* from Heng 2009 [4] to determine how well the basis vectors reconstruct a waveform. This parameter is given by

$$\mu_i = \left|\left| \sum_{j=1}^{Z} (H_i, e_j) e_j \right|\right| \tag{14}$$

where (H_i, e_j) is the inner product of the waveform with the orthonormal basis vector e_j. If we use all of the basis vectors, μ_i will go to unity, but Heng 2009 [4] found the number of PC basis vectors needed for a minimal match parameter $\mu_i \approx 0.80$ is approximately 8.

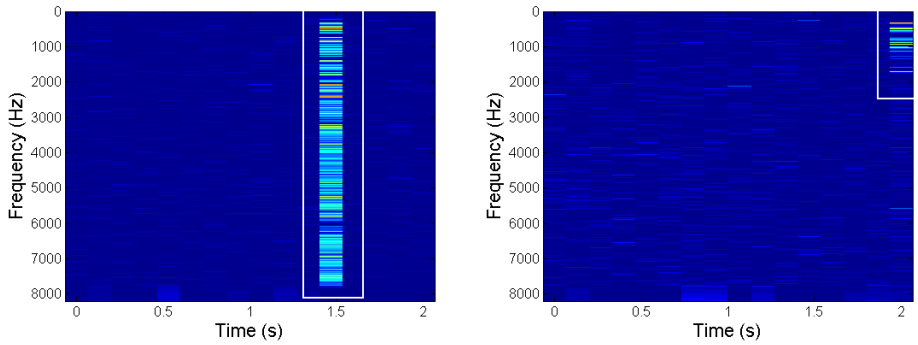
I examined the worst match parameter and the match parameter average for both an unfiltered and filtered version of the same catalog, and I found that the minimum match parameter for the filtered waveforms decreases dramatically compared to the unfiltered waveforms. The average values were comparable for both, and the average match parameter for a filtered catalog with a bandpass of 10 - 4000 Hz was usually only 0.01 greater than the unfiltered catalog. We determined that currently filtering is not a good method to use on Dimmelmeier or Abdikamalov waveforms because it removes most of the important features of these morphologically similar waveforms. It also makes the waveforms even more difficult to reconstruct

from the unfiltered basis vectors. However, filtering is successful in simulating real detection of Murphy waveforms because it removes low frequency noise and real features that we will not be able realistically to detect. A better simulation of detection conditions will be to decrease the SNR before utilizing the algorithm.

GEO data characteristically has a large amount of low frequency noise, and the unfiltered noise can have amplitudes of up to ten orders of magnitude greater than the amplitude of a simulated signal. We used a bandpass of 100-4000 Hz to filter the GEO noise after splitting it into segments. The filtered segments were buffered by 1 second on both sides to eliminate phase offset artifacts from the filtering process.

2.3 Finding Glitches

Glitches are an interesting case of noise that we can use to test the algorithm. We identified the glitches by visually inspecting spectrograms of each two second segment of data. The myspecgram function resolves the two seconds into several pieces and finds a power spectrum of each of these segments, which is then modified with a running median across the time domain in each frequency bin to highlight areas with dramatic features. Glitch 1 is quite dramatic with amplitude differences in most frequencies, while Glitch 2 occurs primarily at low frequencies.



(a) Glitch 1 (b) Glitch 2

Figure 1: Finding glitches in GEO data spectrograms

We simulated the worst-case scenario of signal detection by aligning the glitch beginnings with the peaks of the simulated waveforms. It is unlikely that we would be able to detect such a signal, but it is interesting to see

how the algorithm performs in this case. Figure 2 shows the superposition of Glitch 1 over a real waveform scaled up three orders of magnitude.

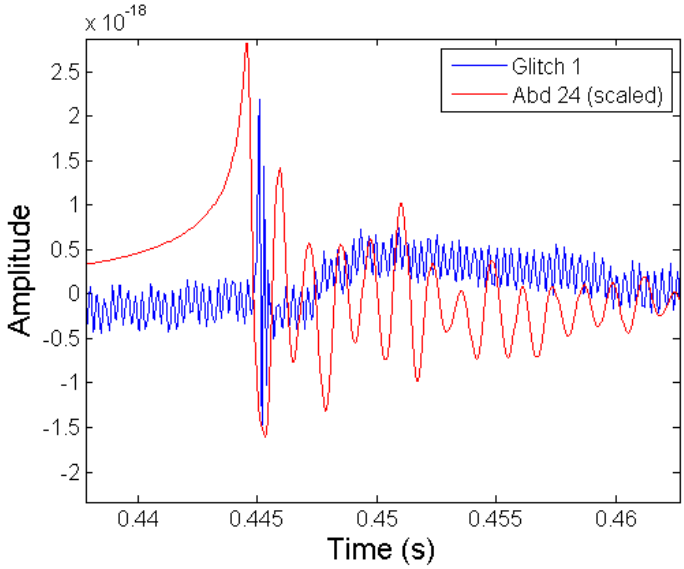


Figure 2: Glitch 1 superimposed over a scaled Abdikamalov waveform

3 Results

The first test of success for the SMEE algorithm was determining whether it can correctly identify waveforms over noise. Gaussian white noise at LIGO S5 levels does not pose a problem for the algorithm, and in Logue 2011 [5] the algorithm can clearly identify most waveforms over noise and also over one another. The next type of noise we utilized in this report is Gaussian white noise at ET levels, and then finally we tested ET colored Gaussian noise. Table 1 shows the performance of the algorithm and that it can distinguish waveforms over Gaussian noise. The next step in our analysis was to test the injected waveforms in real GEO noise. Finally, we compared the Bayes factors for injected waveforms reconstructed with both the Abdikamalov and Dimmelmeier basis vectors to see the algorithm's ability to distinguish between the two similar catalogs.

S5 level white	SNR	ET level white	SNR	ET level colored	SNR
208.765	2.749	**7825.555**	16.283	**7760.660**	19.296
373.371	3.625	**14690.416**	22.558	**14566.667**	25.692
1205.021	6.064	**42408.846**	35.286	**42707.800**	42.726
2388.058	8.613	**84961.247**	47.774	**85089.910**	57.511
3410.238	9.831	**123469.900**	57.483	**123148.690**	67.984
3923.465	10.742	**136490.835**	59.534	**136511.876**	71.053
4735.902	11.678	**167058.826**	69.522	**167318.275**	77.154
4945.594	11.773	**180764.519**	70.606	**180502.872**	80.123
4353.667	10.623	**149043.924**	63.736	**149341.357**	73.862
3710.118	10.237	**130038.372**	62.191	**129242.494**	68.187

Table 1: Bayes Factors for 10 Abdikamalov waveform evidences against Gaussian noise evidence and the Signal to Noise Ratio for each waveform injection

GEO Noise 1	SNR
19859.630	19.399
19860.168	19.866
19860.762	20.476
19860.078	19.852
19860.086	21.361
19860.116	18.803
19860.285	20.608
19860.878	18.227
19858.613	26.193
19858.870	20.307

Table 2: Bayes Factors for 10 Abdikamalov waveform evidences against GEO noise evidence and the Signal to Noise Ratio for each waveform injection (still using Gaussian model for SNR and noise evidence and uniform priors)

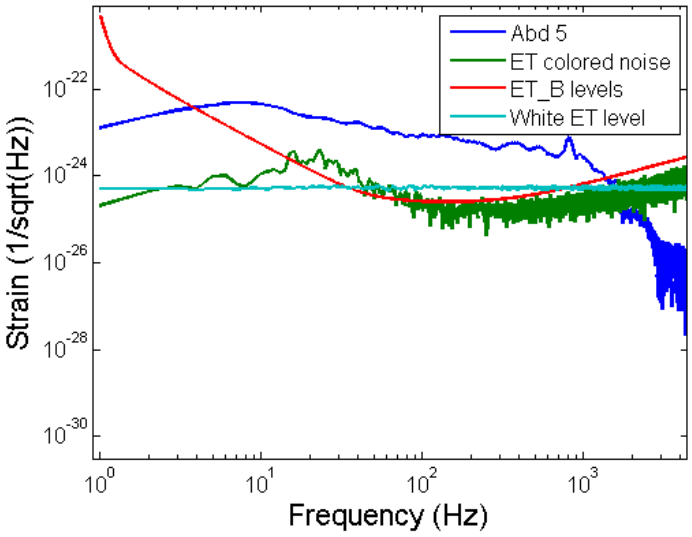


Figure 3: Power spectral density for Abdikamalov simulation 5 compared to ET white noise, ET colored Gaussian noise, and the expected ET_B strain

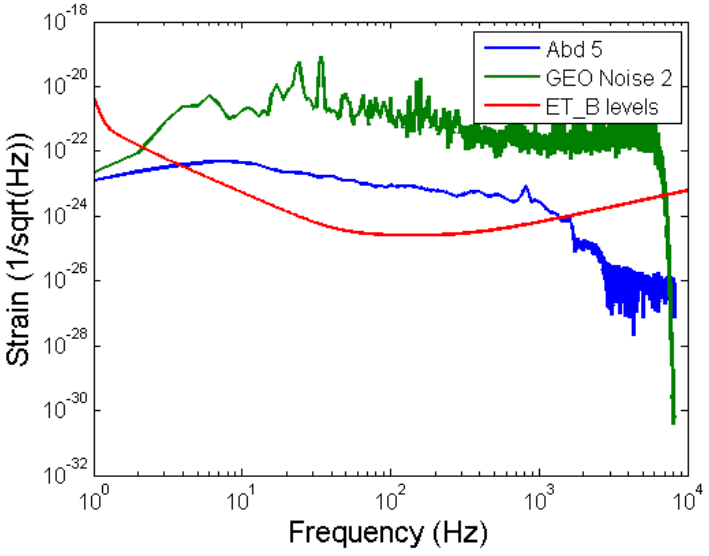
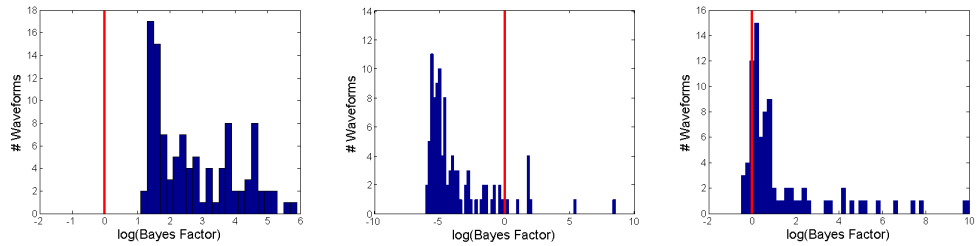


Figure 4: Power spectral density for Abdikamalov simulation 5 compared to GEO noise 2 and the expected ET_B strain

We can see in Figure 4 that the GEO power spectrum is actually higher at all frequencies than the example waveform Abd 5 and also higher than the predicted ET strain. In Figures 5-8 we examine the algorithm's performance on injected waveforms into the various sections of noise when it is calculating the Bayes factor for evidence using 6 Abdikamalov PC basis vectors versus the evidence using 6 Dimmelmeier basis vectors for the reconstruction of the injected waveform and that perticular piece of noise (or aligned glitch).



(a) Noise Segment 1 (b) Noise Segment 2 (c) Noise Segment 3

Figure 5: Bayes factors for all Abdikamalov waveform evidences against GEO noise segment evidence for three different segments of noise. Positive Bayes factors favor Abdikamalov while negative ones favor the Dimmelmeier model.

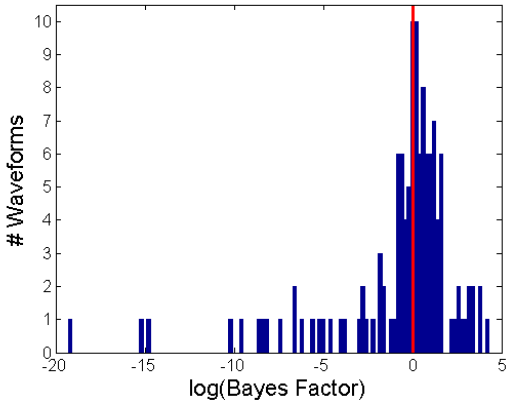
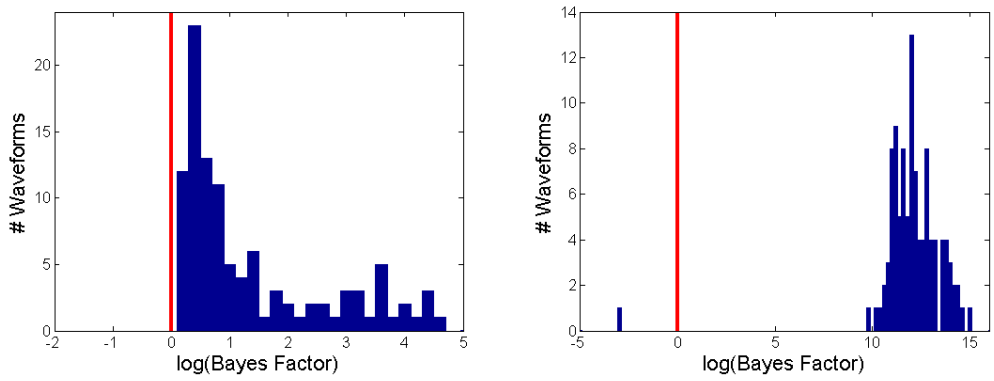


Figure 6: Bayes factors for all Dimmelmeier waveform evidences against GEO noise segment 3. Positive Bayes factors favor Abdikamalov while negative ones favor the Dimmelmeier model.



(a) Glitch 1 (b) Glitch 2

Figure 7: Bayes factors for all Abdikamalov waveform evidences against GEO noise segment evidence for three different segments of noise. Positive Bayes factors favor Abdikamalov while negative ones favor the Dimmelmeier model.

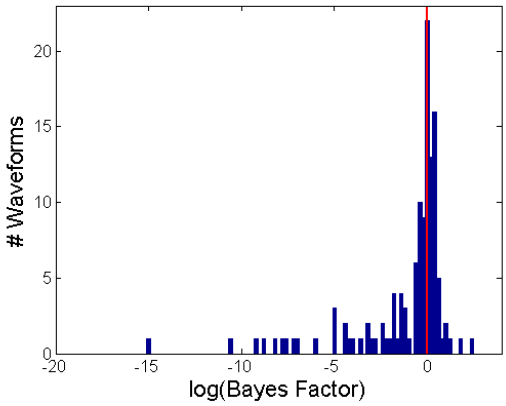


Figure 8: Bayes factors for all Dimmelmeier waveform evidences against GEO Glitch 1. Positive Bayes factors favor Abdikamalov while negative ones favor the Dimmelmeier model.

4 Conclusions

Clearly we see that the evidence for injected waveforms is much greater than the evidence for Gaussian noise in Table 1. If our sources of noise are indeed Gaussian, we can be reasonably confident that SMEE will correctly identify the waveform over the noise. However, our main focus in this report is on waveform identification in GEO noise, and we see that the Gaussian noise model is not adequate or accurate. We show the Bayes factors for the same ten waveforms against GEO noise evidence in Table 2, and it is evident that there is some problem in using the Gaussian model to find noise evidence. The SNR is also calculated using a Gaussian model. The large factors in Table 2 probably result from a scaling issue because the waveform amplitudes are up to 3 orders of magnitude smaller than the noise amplitudes. The almost uniform Bayes factors and SNRs also suggest that the algorithm cannot readily identify the waveform from the noise, possibly also due to the scaling issue. It is also evident in the power spectrum of Figure 4 that detecting a signal will be nearly impossible even with filtered GEO noise in most cases with the noise from GEO overwhelming the injected signal. We see that the signal is detectable in ET noise and with white Gaussian noise at ET levels in Figure 3.

In Figures 5-8, we see that when waveforms are injected it makes some difference in the algorithm's response suggesting that the waveforms are not totally overwhelmed. Giving the algorithm two choices of basis vectors to reconstruct a segment of noise with an injected waveform shows that the algorithm cannot properly distinguish between Abdikamalov and Dimmelmeier waveforms. There is a clear bias toward the Abdikamalov basis vector evidence in some noise segments, but in Figure 8 we see that the algorithm cannot readily distinguish the injected Dimmelmeier waveforms. The Bayes factors cluster around certain numbers in all of Figures 5-8, but this is likely due to the noise. In Figure 5b the algorithm seems to prefer the Dimmelmeier evidences, but the noise must have some feature that more closely remembles the Dimmelmeier basis vectors. In Figures 5c, 6, and 8, the Bayes factors cluster around zero. Only in certain injections does the algorithm prefer the correct set of basis vectors, but it is not reliable enough to identify waveforms over noise or over one another.

If we want to assure that this algorithm can work with real sources of noise, the first step in further analysis will be to improve the prior probability distribution functions to reflect our knowledge about the noise sources. We have still been able to examine SMEE's preferences of waveforms, but it is necessary that we know whether we can reliably identify a waveform against

noise. After we have accomplished this, we can then move to the task of adjusting the algorithm to identify signals against one another. We have successfully shown that the algorithm only performs its function in Gaussian noise scenarios, but that more adjustment is needed for success in real data.

References

[1] H. Dimmelmeier, C. D. Ott, A. Marek, and H.-T. Janka. Gravitational wave burst signal from core collapse of rotating stars. *Phys. Rev. D.*, 78(6), 2008.

[2] Abdikamalov *et. al.* Axisymmetric general relativistic simulations of the accretion-induced collapse of white dwarfs. *Phys. Rev. D.*, 88, 2010.

[3] H. Grote and the LIGO Scientific Collaboration. The GEO 600 status. *Class. Quantum Grav.*, 27, 2010.

[4] Ik Siong Heng. Rotating stellar core-collapse waveform decomposition: a principal component analysis approach. *Classical and Quantum Gravity*, 26(10):105005, 2009.

[5] J. Logue. 1st year research report: Inferring the core-collapse supernova mechanism with gravitational waves.

[6] Christian Roever, Marie-Anne Bizouard, Nelson Christensen, Harald Dimmelmeier, Ik Siong Heng, and Renate Meyer. Bayesian reconstruction of gravitational wave burst signals from simulations of rotating stellar core collapse and bounce. *Phys. Rev. D.*, 80, 2009.

[7] D. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial*. Oxford Science Publications, 2 edition, 2006.

[8] ET Science Team. Einstein gravitational wave telescope conceptual design study. *ET-0106C-10*, 2011.

[9] J. Veitch and A. Vecchio. Bayesian coherent analysis of in-spiral gravitational wave signals with a detector network. *Phys. Rev. D.*, 81(6), 2010.

5 Appendix – Data files, functions, and scripts

- Folder **GEOdata**

- **glitchandData3.mat** 5 segments of GEO data (Col 1-3 random noise, Col 4,5 glitches)
- **hour08** and **hour13** raw segments of GEO data.
- **timedata26206.mat, timedata27434.mat, timedata42606.mat** filtered (100-4000Hz) two minute segments from hour08 and hour13
- **splitdata.m** splits the raw segments into a chosen length, filters, and buffers the segments. From 60 minutes of data you get 29 two minute segments because the first and last second are buffered.

- Folder **injectData100**
  For each file the columns are as follows: Col 1 – SNR, Col 2 – noise evidence using Gaussian noise model, Col 3 – Evidence for waveform reconstructed with Abdikamalov basis vectors, Col 4 – Evidence for waveform reconstructed with Dimmelmeier basis vectors.
  - **AbdInjectGlitch1_100.mat, AbdInjectGlitch2_100.mat** 106 Abdikamalov waveforms injected into two GEO glitches.
  - **AbdInjectNoise1_100.mat, AbdInjectNoise2_100.mat, AbdInjectNoise3_100.mat** 106 Abdikamalov waveforms injected into three random sections of GEO noise from glitchandData3.mat
  - **DimInjectGlitch1.mat, DimInjectGlitch2.mat** 128 Dimmelmeier waveforms injected into two minute sections of GEO glitches.
  - **DimInjectNoise1.mat, DimInjectNoise2.mat, DimInjectNoise3.mat** 128 Dimmelmeier waveforms injected into three random sections of GEO noise from glitchandData3.mat
  - **gaussnoiseBayes.mat** Five pieces of white Gaussian noise at GEO levels injected blindly into algorithm.

- Folder **injectData500**
  This folder contains similar data to injectData100 but the data has been bandpass (500-4000Hz) filtered incorrectly.

- Folder **Simulation**
  - **data_*** sets of waveform simulations from each catalog labeled accordingly.
  - **vectors_*** sets of PC basis vectors for simulations from each catalog labeled accordingly.
  - **ET_noise.mat** two minutes simulated Gaussian noise colored from ET sensitivity curve.

  - **gaussNoisegeolevel.mat** Five pieces two minute long segments of white Gaussian noise at GEO levels.

- Folder **Functions**
  Scripts and functions for various purposes

  - **myspecgram.m** function makes a spectrogram of a 2 second piece of data with a chosen time resolution in fractions of a second (e.g. If you want it in 0.25 s increments, res = 0.25)
  - **nested_2.m** script that finds the Bayes factors with scaling adjusted for GEO noise and prior ranges for $10^{-}19$.
  - **splitdata.m** function splits long time series data into segments of your choosing, filters, and buffers