Gravitational wave estimation with bayesian compressed sensing and machine learning

Garett Brown*

17 August 2016

1 Motivation and Context

The worldwide excitement that LIGO's (advanced Laser Interferometer Gravitational wave Observatory) February announcement made has refocused the attention of many scientists and experts [1], considering Einstein's theory of General Relativity (GR) was put forward 100 years ago [2]. Gravitational waves (GWs) are oscillations of space time produced by a changing mass distribution (like two stars orbiting around each other) or by a massive energy release (like supernova explosions). They are a consequence of GR and were first predicted by A. Einstein.

Until recently, GW were not possible to detect due to technological limitations. Experimental equipment is needed to be able to detect changes in length down to 10^{-20} meters. On top of this technical challenge, there was a mathematical challenge to fit the measured data with Einstein's GR model on the fly. Of the initial attempts to solve the Einstein equations only solutions were found for only certain cases due to the nonlinear nature of Einstein's GR model. Post Newtonian (PN) theory provides approximations to GW, and Numerical Relativity (NR) provide 'exact solutions', but both provide different kind of GW waveform models. Numerical solutions can still computationally expensive, even for supercomputers, this meant that there was need to further reduce the models without any significant loss of accuracy.

For detection and analysis we need to compute a bank of theoretical templates, i.e. waveform models, we need around 10^5 templates, but it depends on the number of parameters employed to compute the waveform and the region of parameter space to explore. Once the bank is computed we cross-correlate each signal with the data stream from the detector. The longer the and more complex the signal is the more expensive this process is and there is need for fast and accurate methods for GW analysis. The work presented here will be a preliminary study to improve previous works [3, 4] using applied mathematical techniques like bayesian compressed sensing and machine learning techniques. These techniques come from previous work done on sparse bayesian modelling as exemplified by the relevance vector machine [5].

The previous work begins by dividing the computation into an 'offline' portion, a 'start up' portion, and an 'online' portion. The 'offline' portion takes more time to set up the system for fitting the data, but in the end it makes the 'online' time significantly shorter. The method considers the fact that the parameter space is too large to consider so rather than evaluating a parameter space with 10^{35} possibilities, for every frequency in the frequency space some key assumptions are made. Firstly, only coalescing binary systems are considered because they produce the strongest (highest amplitude) GWs. The parameter space is then further reduced by considering

*Advisor: Priscilla Canizares

only combinations of binary systems which produce the strongest GWs. These assumptions and further reduction techniques create a basis with a reduced number of vectors needed to recreate the intended waveform. Further details of the model reduction can be found in the previous work [3].

After reducing the basis down to a more manageable size the online portion of computing the necessary weights can be done. Computing these weights can still take some time and the resulting weights show that many of them are effectively negligible. Thus the application of Tipping's work is used to further optimise the number of basis and obtain the correspondent weights. The algorithm works to reduce the required basis vectors down by randomly assigning some to be non-zero and then working out which ones are essential and which ones are not.

2 Methodology

In order to bring together the previous works, the results from the offline portion had to be processed and transferred into Tipping's Matlab code [6]. The text files generated by the 'offline' code were simply be read into Matlab and reformed to create a orthogonal basis matrix along with the waveform that I was trying to model with the basis. The data I used from the reduced parameter space had already been reduced to an $N \times M$ matrix with N bases evaluated at M frequencies ($N < M$). It also had a reduced waveform that simplified a waveform evaluated at all M frequencies to a smoother waveform evaluated at N frequencies chosen by the greedy algorithm (see Fig. 1) [3]. Thus, I needed to choose which N frequencies the basis vectors would be evaluated at to create a square $N \times N$ matrix. In order to further reduce the error, I strategically choose which frequencies were to be used in the algorithm. And finally, Tipping's algorithm simply takes this matrix and vector and runs through the machine learning and returns the minimum necessary weights needed to recreate the waveform with the basis.

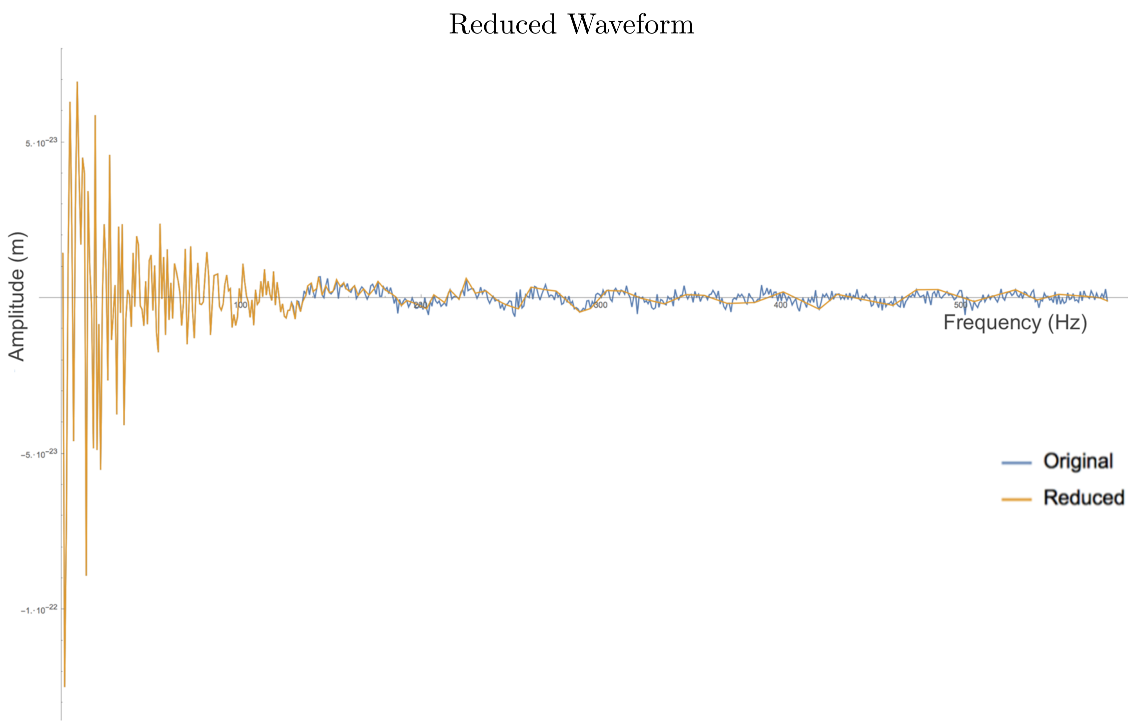

Figure 1: The blue line is the original waveform and the yellow line is the reduced waveform created by the greedy algorithm. As it is shown, the greedy algorithm smoothed out the tail end of it.

3 Results

With the data I used (see Appendix A) I had a square basis of about 200 vectors. The algorithm took around 6 seconds to retrieve the weights needed for both the real and imaginary parts. I decided to choose the same frequencies that were used to reduce the waveform to create the square matrix, but unfortunately the initial results were less than ideal as the error between the initial waveform and the basis and weights was around 0.3. However, by analysing the waveform and more strategically choosing which frequencies were used in the algorithm I was able to reduce the error further to approximately 0.04. It isn't much of an improvement, but the idea is to reduce the number of non- zero basis as much as possible so that during the 'start up' and 'online' portions of the algorithm time is saved by avoiding the costly calculation of weights required to fit the incoming data.

By using all 200 basis vectors, one could solve for the exact weights needed to fit the waveform by solving the $\vec{h} = \mathbf{B}\vec{w}$, where $\vec{h}$ is the vector describing the waveform, $\mathbf{B}$ is the square matrix of basis vectors, and $\vec{w}$ is the vector of weights. When the frequencies chosen for the square basis were the same as the ones chosen by the greedy algorithm for the waveform, then the relative error was $\vec{h} - \mathbf{B}\vec{w} = (7.14283 \times 10^{-12}) + (2.67064 \times 10^{-12})\, i$, for the real and imaginary part. Unfortunately, this same choice of frequencies resulted in an error of $0.33198{+}0.32871\, i$ when the Tipping algorithm was used — however, only 79 and 80 basis vectors (real and imaginary respectively) were non-zero (see Fig. 2 and Fig. 3). I managed to compromise slightly by working to strategically choose which

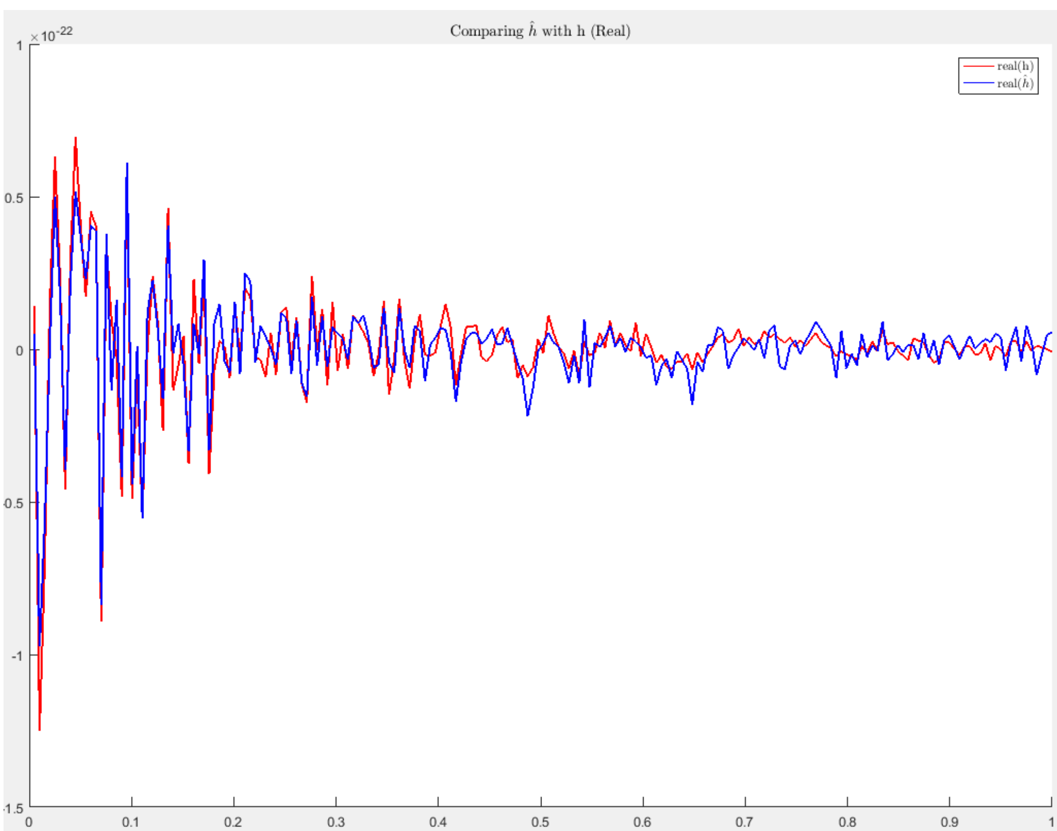

Figure 2: Plot of the real part of the original waveform (h) in red, and the reconstructed waveform ($\hat{h}$) in blue with frequencies that were chosen by the greedy algorithm. The vertical axis is amplitude (meters) and the horizontal axis is normalised frequencies (Hz).

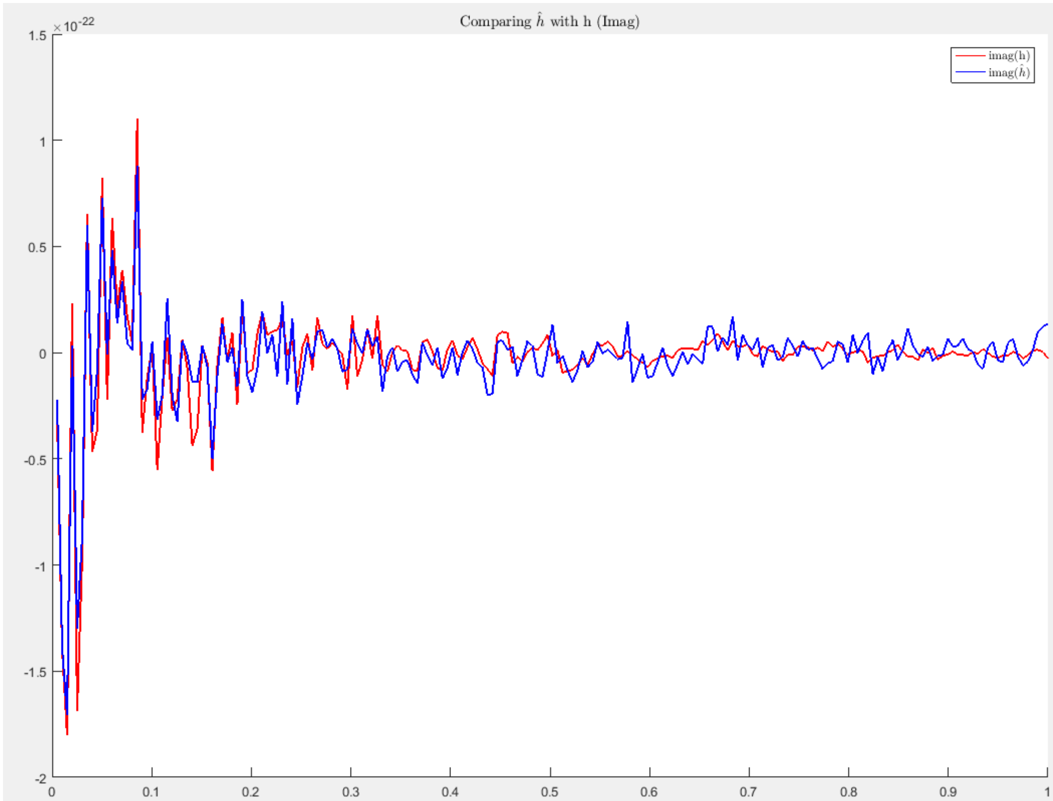

Figure 3: Plot of the imaginary part of the original waveform (h) in red, and the reconstructed waveform ($\hat{h}$) in blue with frequencies that were chosen by the greedy algorithm. The vertical axis is amplitude (meters) and the horizontal axis is normalised frequencies (Hz)

frequencies to evaluate the bases at. When I did the algorithm chose 112 and 120 basis vectors (real and imaginary) and the error was reduced to $0.0739726 + 0.0374448\,i$ (see Fig. 4 and Fig. 5). The increase in the number of basis vectors is not desirable, but a reduction in error is necessary. However, by using these strategically chosen basis vectors and solving for $\vec{h}$ with $\vec{h} = \mathbf{B}\vec{w}$, the error between these two was significantly higher $(0.0171 + 0.0039\,i)$. With a single order of magnitude drop in error after a 50% increase in basis vectors that don't 'fit' even if all of them are included calls out that there may be some sort of balance between the choice of frequencies through the greedy algorithm and my strategic way of choosing them.

The strategy on choosing which frequencies I would keep was a rough estimate based on the characteristics of the original waveform. Essentially, the idea is to create a type of a mask to select basis vectors that are more suitable for the waveform that I am fitting. Since the first part of the waveform has lower frequency modes and the second part of the waveform has higher frequency modes modulated by a lower frequency, the idea is to use fewer low frequency basis vectors and more higher frequency basis vectors. This is because I have already been able to accurately fit the first portion of the waveform, but I've been struggling to accurately fit the second portion (see Fig. 2 and Fig. 3). I decided to make a more rigorous selection of which basis vectors were chosen so I randomly took about 5% of the frequencies from the first 10% then I took about 10% from the next 10% and then I took about 85% from the remaining 80%. I also used a seed of 100. While this method did seem to produce improved results, it was a crude approach to say the least. Using a more distributive method (something like a decaying exponential) to achieve a similar result my be more efficient and more effective. However more work must be done in order to know for sure.

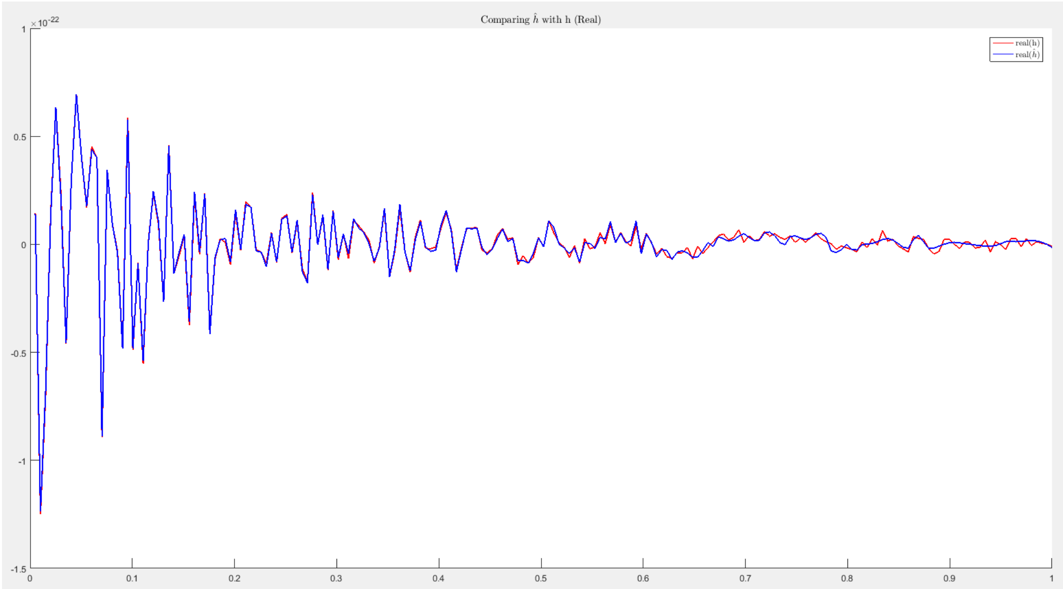

Figure 4: Plot of the real part of the original waveform (h) in red, and the reconstructed waveform ($\hat{h}$) in blue, with the frequencies chosen strategically. The vertical axis is amplitude (meters) and the horizontal axis is normalised frequencies (Hz).

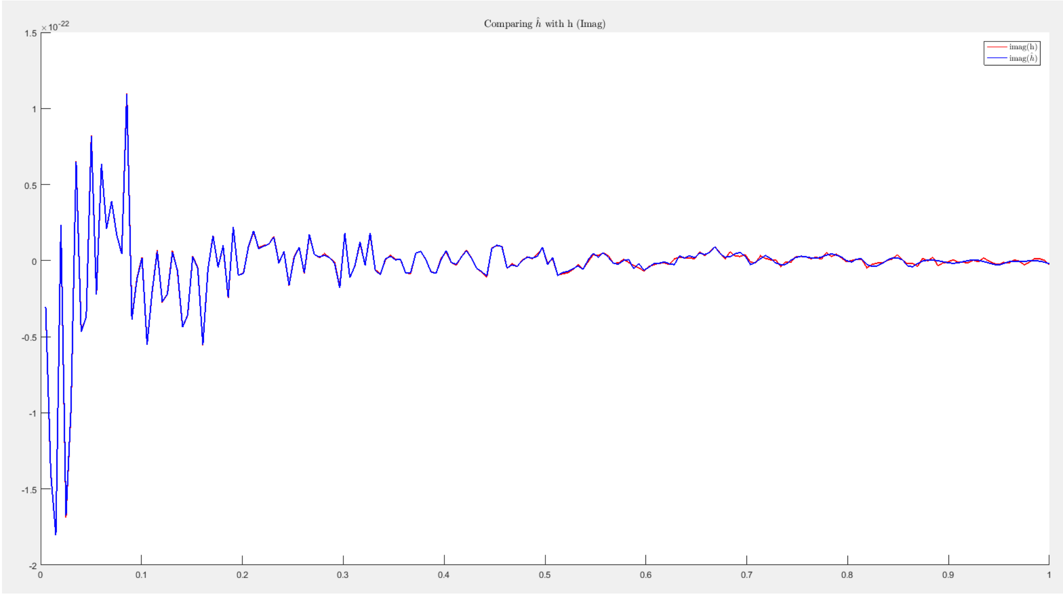

Figure 5: Plot of the imaginary part of the original waveform (h) in red, and the reconstructed waveform ($\hat{h}$) in blue with frequencies that were chosen strategically. The vertical axis is amplitude (meters) and the horizontal axis is normalised frequencies (Hz).

Finally, the data and work done with this bayesian statistics method was with the Post-Newtonian, TaylorF2 model [7] which retains some complexity, but it is certainly not the most complex model out of the PN expansions. Due to the how nature of TF2 and the length of the waveform the Tipping algorithm had a tendency to struggle to fit it in its entirety. If the waveform were longer, and the evolution of events were more distinct, then the problem could be divided up and worked separately. This could be effective, especially because of how quickly (and more or less accurately) the algorithm is at reconstructing a waveform with a reduced basis. Thus by reducing the complex waveform into smaller problems, more accuracy may be achievable. Also, the interesting physics and information is found in the data where the amplitude is low, the frequencies are high, and coalescing binaries are nearly merged and/or are merging. In the end, the results of this have been positive and promising, but more work needs to be done in order to use this technique to do gravitational wave parameter estimations.

A Appendix: Data

Here is the data (Table 1) that I used from the previous work to feed into the Tipping algorithm (Advanced LIGO noise from http://arxiv.org/abs/0901.4936v4. Basis have been whitened):

Table 1: Data used for this analysis.

CBC 3.5PN ROQ rule Parameter Range:	
Low Mass	3
High Mass	30
Data Taking:	
ObsTime	2
Sampling Rate	600
First Frequency	10.0
Greedy Settings:	
Tolerance	10^{-6}
Sampling in Parameter Dimension:	
Training Size	26565
Error Bounds for Noise Free Integrals:	
Estimated EIM error (single functions)	5.0499×10^{-05}
Estimated ROQ error (single functions)	9.1859×10^{-05}

References

[1] Benjamin P Abbott, Richard Abbott, TD Abbott, MR Abernathy, Fausto Acernese, Kendall Ackley, Carl Adams, Thomas Adams, Paolo Addesso, RX Adhikari, et al. Observation of gravitational waves from a binary black hole merger. *Physical review letters*, 116(6):061102, 2016.

[2] Albert Einstein. Zusammenfassung der mitteilung 'erklärung der perihelbewegung des merkur aus der allgemeinen relativitätstheorie.'. *Königlich Preußische Akademie der Wissenschaften (Berlin). Sitzungsberichte*, 47:803, 1915.

[3] Priscilla Canizares, Scott E Field, Jonathan R Gair, and Manuel Tiglio. Gravitational wave parameter estimation with compressed likelihood evaluations. *Physical Review D*, 87(12):124005, 2013.

[4] Priscilla Canizares, Scott E Field, Jonathan Gair, Vivien Raymond, Rory Smith, and Manuel Tiglio. Accelerated gravitational wave parameter estimation with reduced order modeling. *Physical review letters*, 114(7):071104, 2015.

[5] Michael E Tipping, Anita C Faul, et al. Fast marginal likelihood maximisation for sparse bayesian models. In *AISTATS*, 2003.

[6] Michael E Tipping. *SparseBayes Software (V2.0)*, 2016. Sparse Bayesian Models (and the RVM), Accessed 17 August 2016.

[7] Alessandra Buonanno, Bala R Iyer, Evan Ochsner, Yi Pan, and Bangalore Suryanarayana Sathyaprakash. Comparison of post-newtonian templates for compact binary inspiral signals in gravitational-wave detectors. *Physical Review D*, 80(8):084043, 2009.