Improving Searches for Gravitational Waves using Signal Isolation Tests

Sebastian Cassel
Supervisor: Peter Shawhan
(Dated: September 23, 2005)

The procedure for detecting gravitational wave (GW) signals from binary inspiral sources is reviewed. The characteristics of true and noise-induced signals are discussed and a discriminating test proposed. The approach involves considering the time history of the matched filter output in the vicinity of a recorded event. The test only uses the GW channel data and is complementary to the existing χ^2 veto. The effectiveness of this test when applied to the S4 data of the Hanford 4km detector is presented. A comparison with previously developed tests offered. An SNR time above threshold test has shown to be 82% effective at vetoing noise-induced triggers whilst neglecting less than 1% of the true signals.

I. INTRODUCTION

The Laser Interferometer Gravitational Wave Observatory (LIGO) aims to detect gravitational waves (GWs) from various sources throughout the universe. These waves are consequences of the finite speed at which information can travel through space. The discovery of GWs would allow stringent tests for the theory of general relativity. It could also allow the possibility of gathering evidence of a grand unified theory.

The LIGO detectors have a similar design to a Michelson interferometer. Due to the spin of the graviton, GWs must have a quadrupole nature to lowest multipole order. This means that if a GW propagates through the detector, one arm will contract while the perpendicular arm expands in length. The interference fringes would shift, allowing a calculation of the strain, s, in the detectors. The strain determined can be separated into that due to noise, n, and that due to GWs, h.

$$s(t) = n(t) + h(t) \tag{1}$$

The detection of GWs from binary inspiral sources[1] was only considered in this project. That is where two astrophysical objects are in orbit around each other. If an accelerating quadrupole (or higher multipole) moment exists, GWs are emitted. These carry away energy and angular momentum, causing the orbits of the bodies to spiral into each other. Eventually the bodies begin to merge, a stage which will be referred to by the coalescence time.

This project investigated the GW signal data from LIGO's Hanford 4km detector. The purpose was to develop a test which could discriminate between true and noise-induced GW signals. The time history characteristics of signals were investigated. The aim was to be able to eliminate 90% of the false GW events, whilst neglecting less than 1% of the true events.

A. Detection Algorithm

The GW waveform during the inspiral stage is believed to be well known[2]. The optimal method of detection of signals with known waveforms can be shown to be a matched filter. However, it must be assumed that the noise present is stationary and Gaussian. Defining the Fourier transform as in (2), the matched filter output, z, is calculated as in (3), where $Q(t)$ is the theoretical waveform or *template*, and $S_n(f)$ is the noise power spectrum. The template is complex with the real and imaginary parts corresponding to waveforms with orthogonal inspiral phase.

$$\tilde{s}(f) \equiv \int e^{-2\pi i f t}\, s(t)\, dt \tag{2}$$

$$z(t) \equiv \int \tilde{s}(f) \,.\, \frac{\tilde{Q}^*(f)}{S_n(f)}\, df \tag{3}$$

The complex filter assumes a stationary inspiral phase for each component waveform. The magnitude of the output is also normalized to produce the signal-to-noise ratio, SNR or ρ, where $<\rho^2> = 2$. By taking the magnitude, the maximum SNR over the full range of inspiral phases is recorded at any time. The SNR is the scaled correlation of the measured signal with the template where the coalescence time is aligned with the current time index.

A discriminating test[3] is also currently used that determines whether the signal shows the correct time-frequency evolution. It is evaluated by dividing frequency space into p bins, and calculating the filter outputs, z_i, within each

limited frequency range. The frequency ranges are chosen such that the output from each bin has the same magnitude, if the signal perfectly matches the GW template. Thus, the output of (4) should be small for true GWs.

$$\chi^2 \equiv p \sum_{i=1}^{p} \left| z_i - \frac{z}{p} \right|^2 \tag{4}$$

During this report, the parameter $r^2 \equiv \chi^2/p$ will be used to avoid the sensitivity to the number of frequency bins. The analysis presented uses 15 frequency bins, where the expectation, $< r^2 > = 2(p-1)/p \approx 2$, and the standard deviation, $\Delta(r^2) = 2\sqrt{2(p-1)}/p \approx 0.7$. These values vary slightly when template mismatch is accounted for.

A GW candidate is selected from the analyzed data when the following conditions are met: the SNR exceeds a threshold of 6.0, and $r^2 < 5.0$. These are the thresholds used in the analysis for this report as well as those recommended by the LIGO documentation. However, they can be varied in order to control the number of candidates selected, or *triggers*.

Due to the ill-behaved nature of the noise observed in LIGO detectors, many noise-induced triggers are created. During the latest data collection run (S4), it was expected that, at the most, a couple of binary inspiral events might be detected. However, over 10^7 triggers were recorded. For the analysis presented here, all triggers within LIGO data were taken to be noise-induced. The triggers deemed as true GWs were those that were injected into the LIGO data, by simulation of the expected response.

One limitation to the current method of selecting GW candidates is that only one instant in time is considered. During noisy periods, it might be expected that there is an excess of signal for sustained periods of time. There may also be other characteristics within the time series that differ between true and noise-induced signals. The time histories of the SNR and r^2 signals were investigated in order to find distinguishing features.

It should be noted that there are also coincidence tests between multiple detectors, and environmental sensors[4,5] that attempt to eliminate noise-induced triggers. The latter is used to produce data quality (DQ) flags which can be used to veto certain times. This project was not involved in the development of those applications.

II. INVESTIGATION OF TRIGGER TIME HISTORY CHARACTERISTICS

A. Hardware Injected Signals

During the data collection, a few simulated inspiral signals were injected into the detector by driving the end mirrors. The injections listed in Table I were inspected. In order to inspect signals with negligible noise, periods of time in which the only DQ flag was the inspiral injection were used.

Binary Masses / Solar Mass	Effective Distance / Mpc	GPS Start Time	Figure
1.4 - 1.4 (exact)	20	795574427	3
1.4 - 1.4 (exact)	2	795574667	1

TABLE I: Sequence of Hardware Injected Signals

A template exactly matching the mass parameters of the injections was used. Triggers were observed 25s after the injected signals were initiated. This corresponds to the inferred coalescence time. The injection in Fig 1 also produced a secondary trigger, with reduced SNR, 30s after its GPS start time. This was neglected for analysis.

Fig 1 shows a sharp peak in the SNR at the inferred coalescence time, but with some intrinsic width. There are also subsidiary peaks at approximately ± 0.015s of the trigger time. The maximum sensitivity of the LIGO detectors is of the order of 100 Hz. This would suggest that the subsidiary peaks are due to the template being $\sim 2\pi$ out of phase with the injected signal. I expect that the full form of this peak can be reconstructed by the auto-correlation function of the template.

The r^2 time series possesses a minimum at the trigger time. However, there are large side-peaks at about ± 0.008s of the trigger time, as well as bumps occuring at ± 0.015s. The maxima of the r^2 side-peaks occur when the template is $\sim \pi$ out of phase with the injected signal. The cause for these side-peaks is illustrated by Fig 2. At the inferred coalescence time, each sub-filter has an equal magnitude. However, since each sub-filter corresponds to a different frequency range, the signals oscillate with different periods. The result is large values for the r^2 statistic. It should be possible to recreate the r^2 time series characteristics due to the pure GW in a similar way as that for the SNR signal.

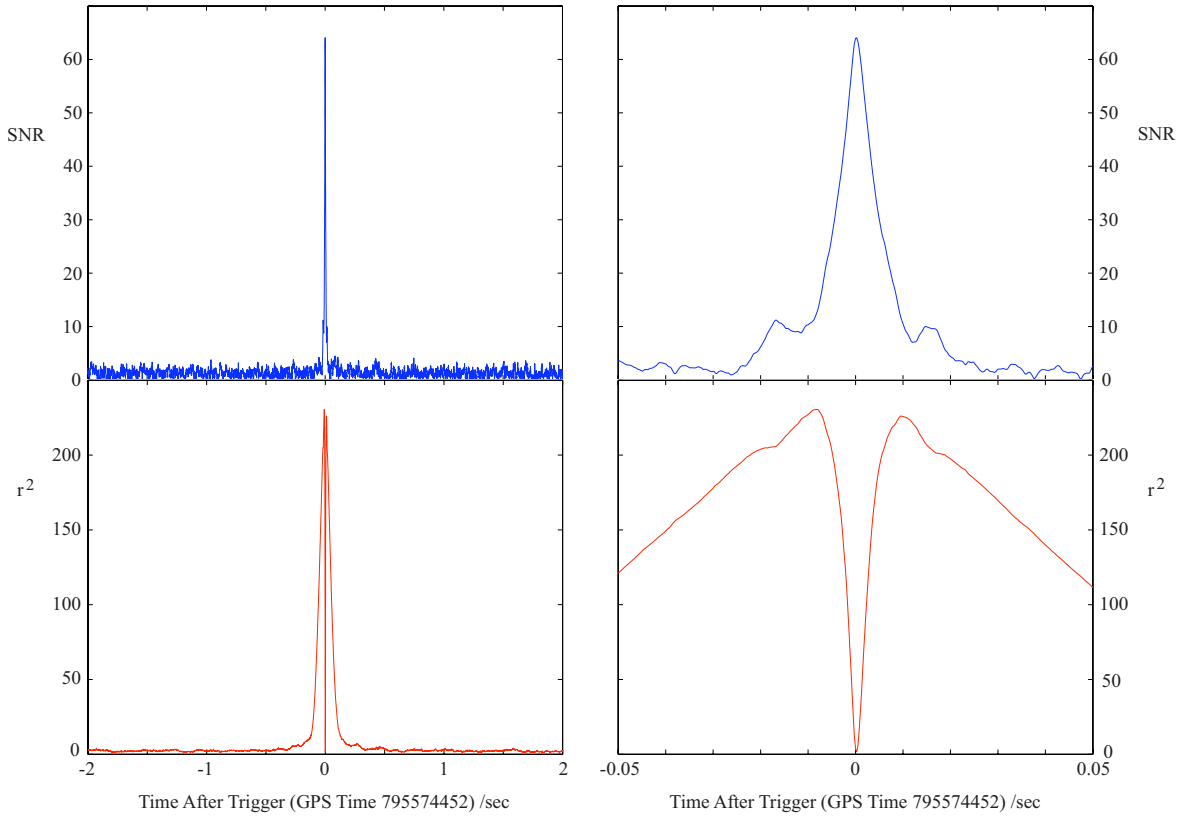

FIG. 1: Plot of the SNR and r^2 time series for a strong hardware injected signal (with a 1.4-1.4 solar mass template)

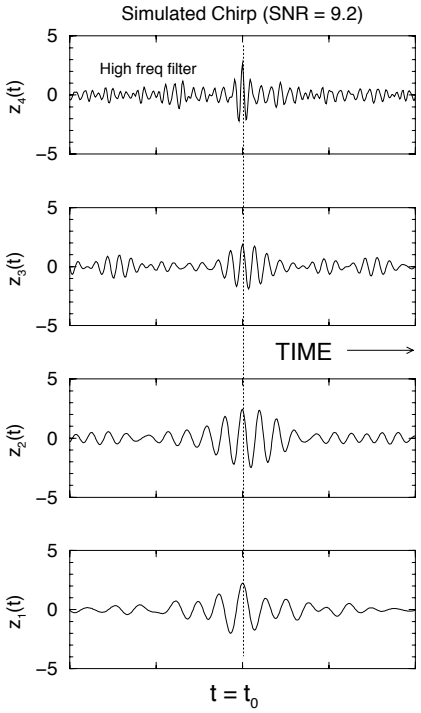

FIG. 2: The output of $p = 4$ single-phase filters for a simulated chirp signal added into a stream of detector noise. (Taken from Phys. Rev. D **71**, 062001 (2005) by B. Allen)[3]

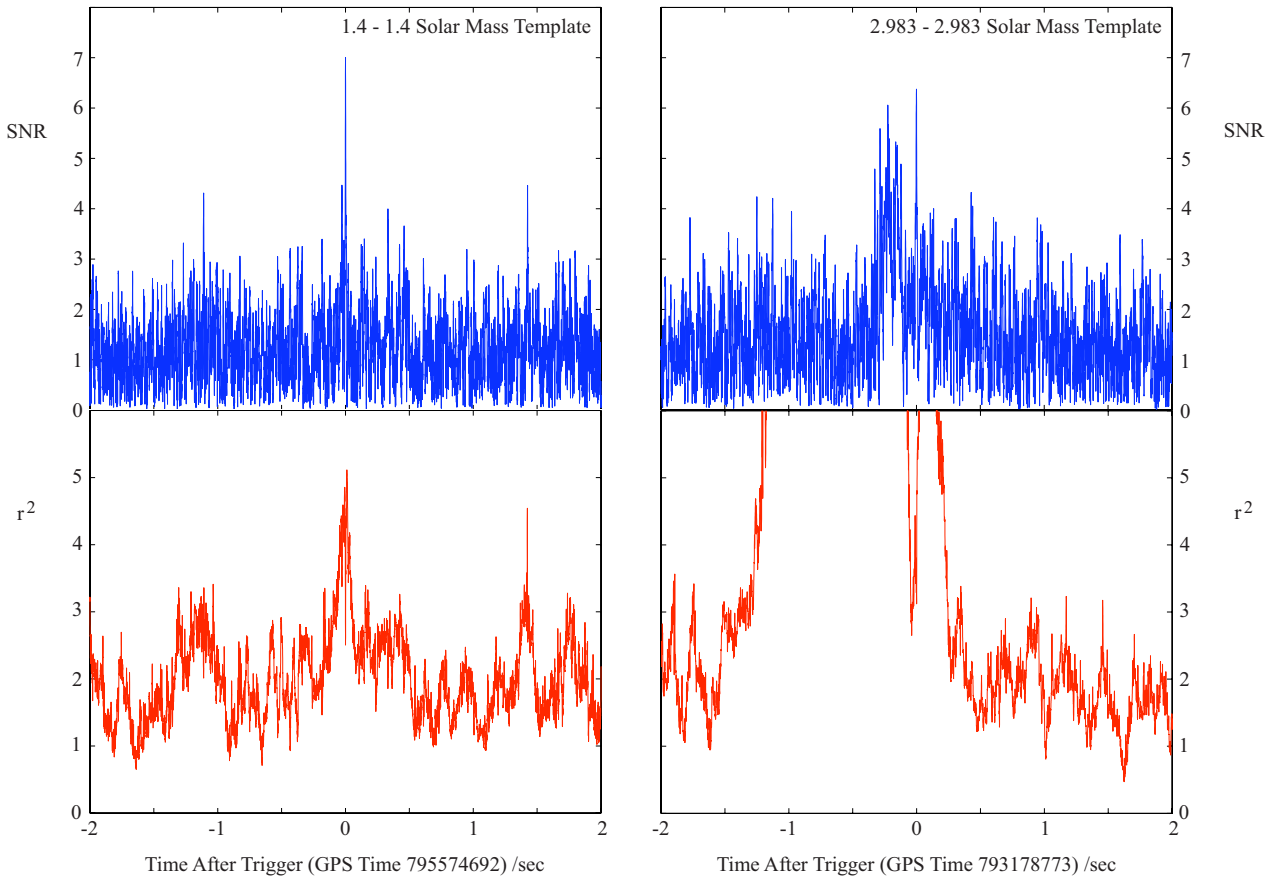

FIG. 3: Plot of the SNR and r^2 time series for a weak hardware injection (left) and a noise-induced trigger (right).

B. Noise-Induced Triggers

Triggers during periods with no DQ flags present were then investigated. This choice was taken in order to avoid investigating triggers that could be eliminated by environmental sensors. Figure 3 compares the signal histories of a hardware injected signal and a noise-induced trigger with similar SNRs. The effect of noise on the hardware injected signal is now more apparent. The subsidiary SNR peaks are unnoticeable and the r^2 side-peaks are more variable.

Away from the injected signal's trigger, the noise appears to be well-behaved with an SNR of $\sim \sqrt{2} \pm 1$ and r^2 value of $\sim 2 \pm 0.7$ as expected for stationary Gaussian noise. However, for the noise-induced trigger, the signal time histories are much more variable. At the recorded trigger time, the SNR exceeds 6.0 and the r^2 value is less than 5.0, so it is selected as a GW candidate. However, the noise is clearly excessive in the vicinity of the trigger.

It appears that the r^2 time series is much more sensitive to the presence of this noise-induced trigger. However tests on both signal histories were developed in order to find the most effective when applied to all triggers. Excessive signal magnitudes were observed in the following regions surrounding noise-induced triggers:

(i) Within the first second preceeding the trigger.

(ii) Within the first second following the trigger (although to a lesser extent than in (i)).

(iii) Only during the periods that the injected signals were also large.
(Sometimes much greater than that observed for the injected signals though)

(iv) Signals of an order of magnitude greater, around five seconds before the recorded trigger.
(The earlier signals were usually not selected as candidates due to high r^2 values)

Only period (i) was investigated in this project for developing a test due to time constraints. However, a fraction of the triggers did not possess excess signals during (i) but instead during some of the other periods. There were still a few noise-induced triggers though, that seemed indistinguishable from the injected signals under inspection of the SNR and r^2 time series.

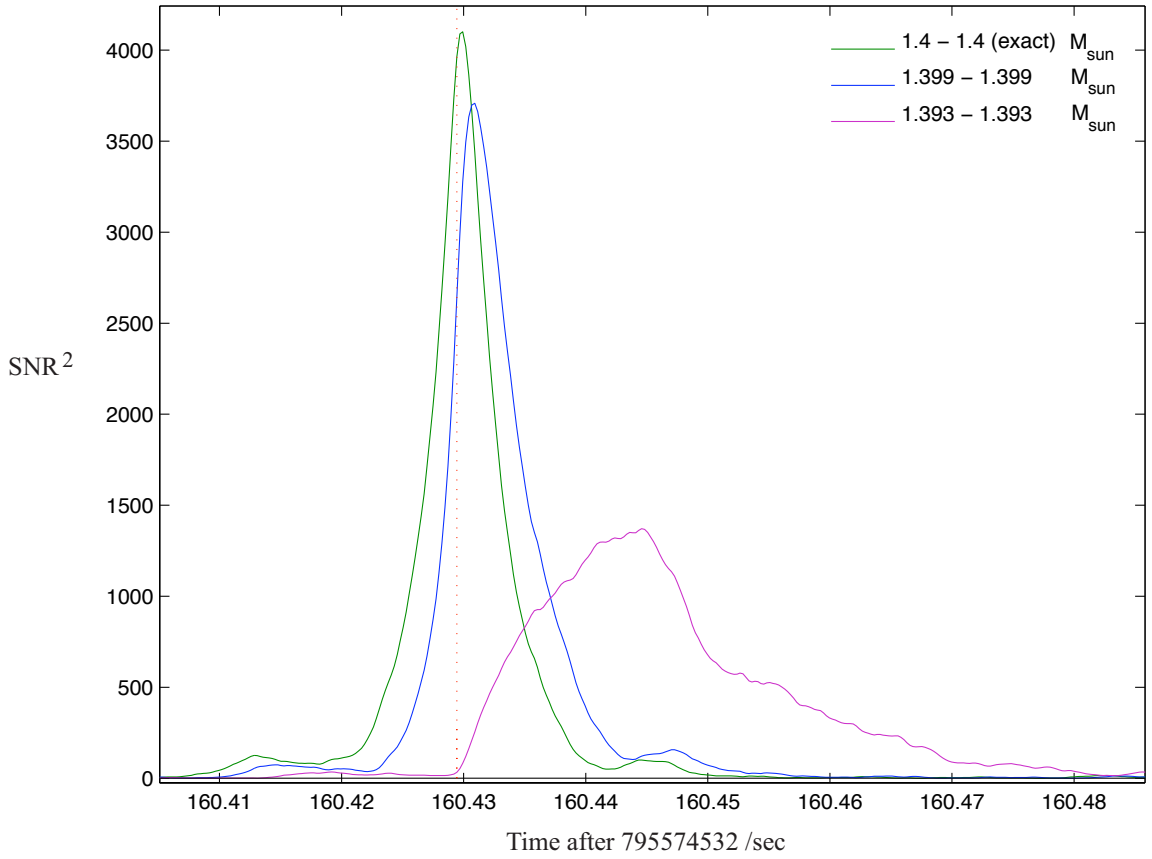

FIG. 4: Plot of SNR^2 for Strong Injected Signal

C. Effect of Template Mismatch

Before a test was developed, it was important to understand the effect that template mismatch had on the form of a pure GW signal. The hardware injection signal shown previously in Fig 1 is now shown again in Figs 4-7, except with the filters using templates that have slightly mismatched mass parameters. It can be seen that the lighter mass templates infer a later coalescence time. An antisymmetry of the signal also develops about the new inferred coalescence times. Figs 6 and 7 demonstrate that a mirror symmetry about the true coalescence time exists between templates above and below the true mass parameters by the same amount.

It should be noted that template mismatch causes the SNR peak to reduce in magnitude, and the r^2 minimum to increase in value. This is accounted for in the analysis software by introducing some SNR dependence to the r^2 threshold. The maximum mismatch allowed for is chosen by the operator before analysis begins.

III. TYPES OF DISCRIMINATING TESTS

A. Previously Developed Tests

A *Crossings Test* was developed by Shawhan and Ochsner[6] that inspected the time history of the SNR signal within the first half-second preceeding a trigger. The number of times the signal crossed some SNR dependent threshold is counted. If the number of crossings is too high, the trigger is vetoed.

A *Time Above Threshold Test* has also been developed by Rodriguez[7]. The time that the r^2 signal spends above a certain threshold during the two seconds preceeding the trigger is counted. If the time spent is too high for that expected, given its SNR, it is vetoed. Using a threshold of $r^2 = 10.0$, it has been shown to be 54% effective at eliminating noise-induced triggers, whilst only disregarding 1% of the simulated true GWs. These results were determined by application of the test onto the S3 data from the Hanford 4km detector.

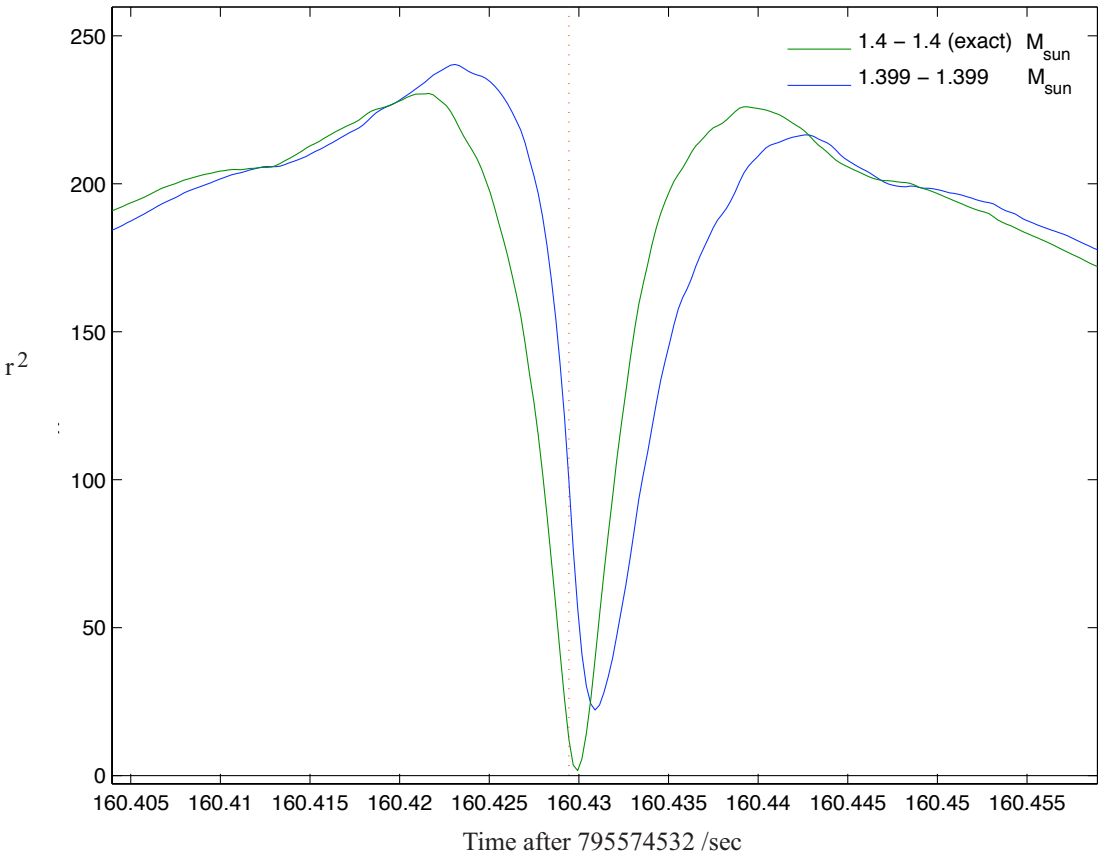

FIG. 5: Plot of r^2 for Strong Injected Signal

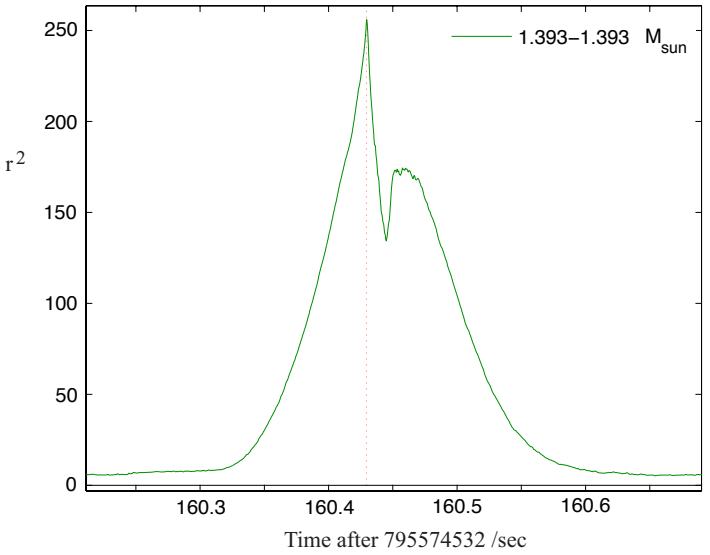

FIG. 6: Plot of r^2 for Strong Injected Signal

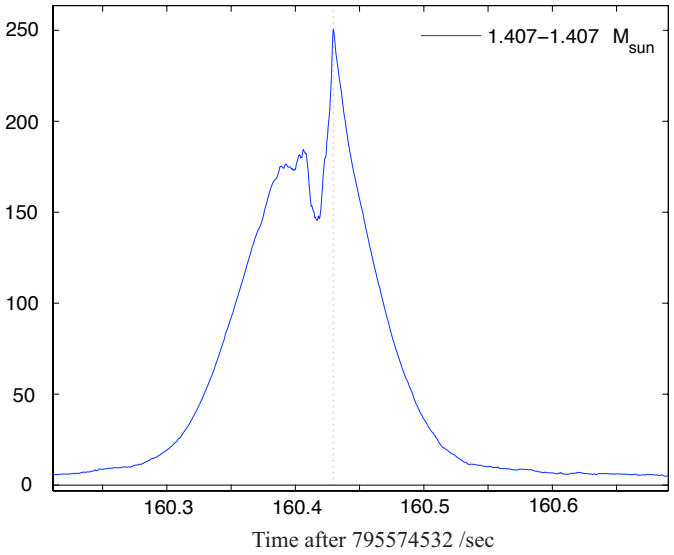

FIG. 7: Plot of r^2 for Strong Injected Signal

Guidi has also developed a test[8] that determines the maximum value of a modified SNR signal during the first second that preceeds a trigger. The expected contribution of a GW that would produce the trigger is subtracted from the filter output. The maximum of the modified SNR signal should now be due to noise, and approximately follow a Fréchet distribution. If the recorded maximum corresponds to a value that falls below some confidence threshold, it is vetoed. Unlike the other two tests, this veto only depends on one point in time, where the modified signal happens to be greatest.

B. Test Developed for Analysis

On inspection of the time history characteristics, a test that discriminated against excess signal magnitudes seems appropriate. Since it is known that a pure signal has some intrinsic width, a time offset was developed in order to exclude these periods of time. The SNR signal is seen to vary from its peaks to near zero very quickly. This suggests that a test that determines the average SNR would not be too effective in determining if periods of excess signal were present. This also explains the reason for choosing a crossings test.

A crossings test, however, would not distinguish whether the signal were just oscillating about the threshold, or frequently reaching very high values. Also, if the SNR exceeded the threshold for long periods of time, as sometimes observed, the crossings test would be insensitive to that behaviour.

For these reasons, a time above threshold test was developed for the SNR signal. A parallel test was also developed, in order to be sensitive to how greatly the signal exceeded the threshold. Instead of assigning an equal value for every sample that exceeded the threshold, the excess SNR^2 would be used. This is effectively integrating the SNR^2 (minus the threshold) over time, whereas only counting periods of time with positive values. This test shall be referred to as the *Excess of Threshold Test*. It should be noted that this is a purely empirical approach to the problem. The reason for using the SNR^2 was for greater simplicity in the computation algorithm. It would also discriminate more against large signals.

The r^2 time series display different characteristics to the SNR signals. It is much more smooth, so a test that determines the average would be more sensitive to periods of excess signal. The effectiveness of discriminating against large averages would be compared to the previously developed time above threshold test. However, it must be noted that the tests developed in this project were chosen to only inspect a time interval of one second. Thus, the results presented here do not attempt to compare different approaches directly.

C. Implementation

The LIGO Algorithm Library (LAL) was used for the analysis. The code for the lalapps_inspiral program was modified in order to carry out the tests described in the previous section. By analyzing a large number of injected signals, the offset times for the SNR and r^2 tests were determined. These times were chosen by inspecting the spread of times that it took for the SNR to first drop below 3.0 and the r^2 value below 2.0.

In determining when the r^2 condition was met, the signal was averaged over 0.02s intervals in order to neglect fast fluctuations. The algorithm for finding the offset time also ignored the first 0.01s, where the r^2 minimum was known to be. The value of 2.0 was chosen so that the influence of the GW would have been known to have become negligible.

IV. APPLICATION OF TESTS

The analysis was carried out by sampling the data 4096 times a second. A template bank spanning between 1 and 20 solar masses for each partner was used. This would have allowed detection of GWs from the majority of neutron star and black hole binaries. The template bank was also organised so that the minimal mismatch between a recorded signal and the closest template was 3%. However, in order to increase the chance of detection, the r^2 threshold was applied with an SNR dependence[3], effectively allowing a maximum mismatch of 10% between signal and template.

A. Choice of Offset Time

For determination of the width that a true signal possesses, simulated signals were injected every 600s into the data stream. The program lalapps_sire was used to select triggers that were within 1ms of the expected coalescence times. 2000 triggers were selected. The list of extragalactic sources in the inspscs.dat file was extended to include sources ranging from the edge of the Milky Way to the Large Magellanic Cloud. This was to ensure a wide range of SNRs

recorded. Mass configurations were also constructed that would match the range of the template bank. This was so that the effect of template masses on the offset time could be determined.

Figs 8-11 show the times that it took for each trigger to satisfy the conditions, specified in section III(C). The results for noise-induced data are also included for comparison. It was expected (and observed) that some triggers would take much longer than the rest, due to the effect of noise. These outliers were ignored when deciding what offset time should account for a pure GW.

The offset time was observed to depend slightly on the low frequency cutoff parameter, as well as the template mass parameters. The effects were deemed negligible though. Also, during the investigation it was noted that the magnitude of the r^2 side-peaks scaled with SNR^2.

B. Effectiveness of Tests

The results of the different tests are listed in Table II from when applied to the Hanford 4km S4 data. The injections were selected from the unchanged inspscs.dat file. This corresponds to the physical distribution of sources in our detectable neighbourhood. The mass parameters were taken from the BNSMasses.dat file which used Monte Carlo methods to model the distribution. A low frequency cutoff of 70Hz was used for the analysis.

The result of each test is characterized by X, and the last two columns correspond to the fraction of vetoed noise-induced and injected triggers respectively. Also, the SNR variable is substituted by ρ here. There are 2327 recovered injections, and 2×10^7 noise-induced triggers. The veto conditions were chosen after inspection of the spread of data points as shown in Figs 12-15. The aim was to maximise efficiency against the noise-induced triggers, whilst neglecting less than 1% of the injections. Where the fraction of false dismissals was much less than 1%, only a small change in the veto condition caused the 1% limit to be exceeded.

Type of Test	Threshold	Offset Time (ms)	Time Window (seconds)	Veto Condition(s)	Efficiency Noise (%)	Efficiency True (%)
r^2 Time Above Threshold	$r^2 = 12$	None	2.0	$X > 0$ ms $\quad \rho \leq 12$ $X \geq 0.3\,\rho^2 \quad 12 < \rho < 70$	16.7	0.5
r^2 Time Above Threshold	$r^2 = 10$	None	2.0	$X > 0$ ms $\quad \rho \leq 10$ $X \geq 0.4\,\rho^2 \quad 10 < \rho < 70$	30.6	0
ρ^2 Time Above Threshold	$\rho^2 = 9$	$2.5\,\rho^{2/3}$	1.0	$X \geq 45\,\rho \quad \rho < 20$	78.0	0.7
ρ^2 Time Above Threshold	$\rho^2 = 12$	$2.5\,\rho^{2/3}$	1.0	$X \geq 8\,\rho \quad \rho < 30$	53.3	2.4
ρ^2 Time Above Threshold	$\rho^2 = 9$	$35 + 5\,\rho^{2/3}$	1.0	$X \geq 45\,\rho \quad \rho < 70$	65.6	1.0
ρ^2 Time Above Threshold	$\rho^2 = 6$	$35 + 5\,\rho^{2/3}$	1.0	$X \geq 100\,\rho \quad \rho < 70$	68.4	1.0
ρ^2 Excess of Threshold	$\rho^2 = 9$	$35 + 5\,\rho^{2/3}$	1.0	$X \geq 1000 \quad \rho < 70$	69.9	1.0
ρ^2 Excess of Threshold	$\rho^2 = 9$	$35 + 5\,\rho^{2/3}$	1.0	$X \geq 20\,\rho^2 \quad \rho < 70$	69.6	1.0
ρ^2 Excess of Threshold	$\rho^2 = 6$	$35 + 5\,\rho^{2/3}$	1.0	$X \geq 2500 \quad \rho < 70$	62.2	1.0
ρ^2 Excess of Threshold	$\rho^2 = 6$	$35 + 5\,\rho^{2/3}$	1.0	$X \geq 50\,\rho^2 \quad \rho < 70$	69.6	1.0
Average r^2 Test		$40\,\rho$	1.0	$X \geq 1.2\,\rho^{1/2} \quad \rho < 70$	61.2	1.0
r^2 Time Offset Test		5	3.0	$X \geq 40\,\rho$ ms $\quad \rho < 70$	54.5	0.8
Maximum r^2 Value		None	0.25	$X > 1.1\,\rho \quad \rho \leq 10$ $X \geq 0.11\,\rho^2 \quad 10 < \rho < 70$	53.6	0.1

TABLE II: Results from Application of Tests

The 'r^2 time above threshold' test above was that developed by Rodriguez[7]. The 'average r^2 test' calculates the average of the signal over some time interval. The 'r^2 time offset test' was developed by choosing veto conditions that isolated a section of noise-induced triggers as shown in Fig 10. Finally, the 'maximum r^2 value' is just the largest value of the signal in some time window.

An effort was made to set the veto conditions so that the very large SNR, injected triggers were not selected. The maximum noise-induced SNR was measured as 65. Thus, for safety and computational ease, an upper limit of $\rho = 70$ was usually chosen for the tests. In other tests, the large noise-induced SNRs occupied the same area of the graph as the injections. For those cases, a smaller upper limit on the SNR was chosen to avoid only vetoing injected signals.

The most efficient test was found to be a combination of the ρ^2 time above threshold test and the maximum r^2 value test. It vetoed 82% of the noise-induced triggers, whilst neglecting 0.9% of the injected triggers. Further simultaneous applications of other tests affected the results only a negligible amount.

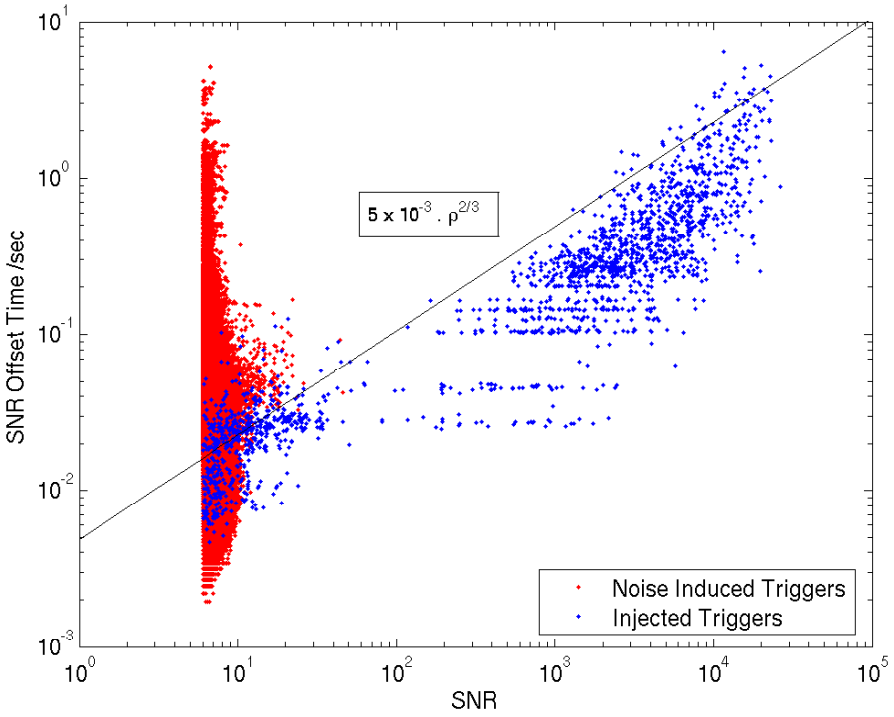

FIG. 8: Plot of SNR Time Offset

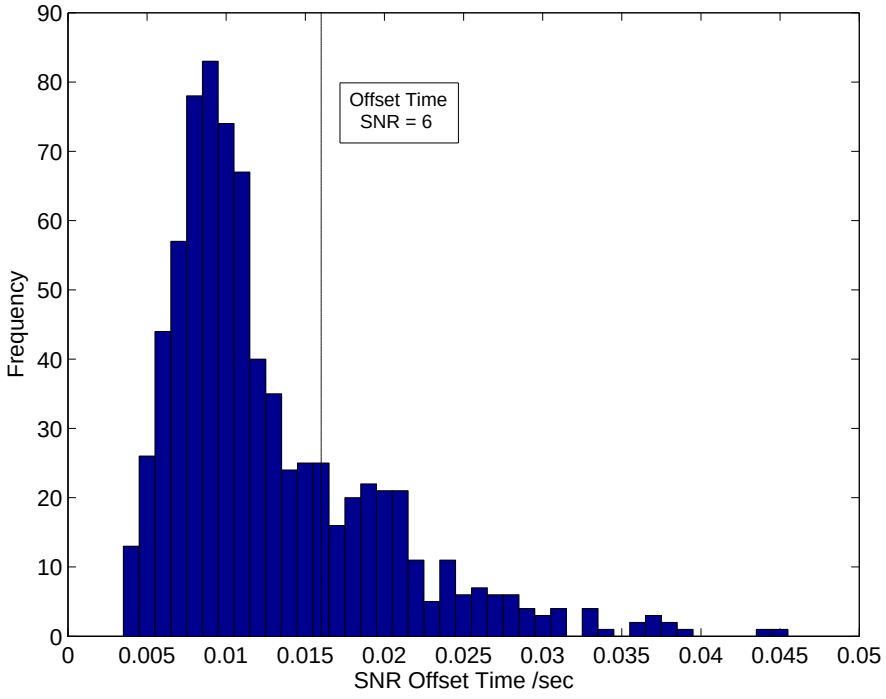

FIG. 9: SNR Time Offset Histogram for Injections where $6 < \rho < 7$

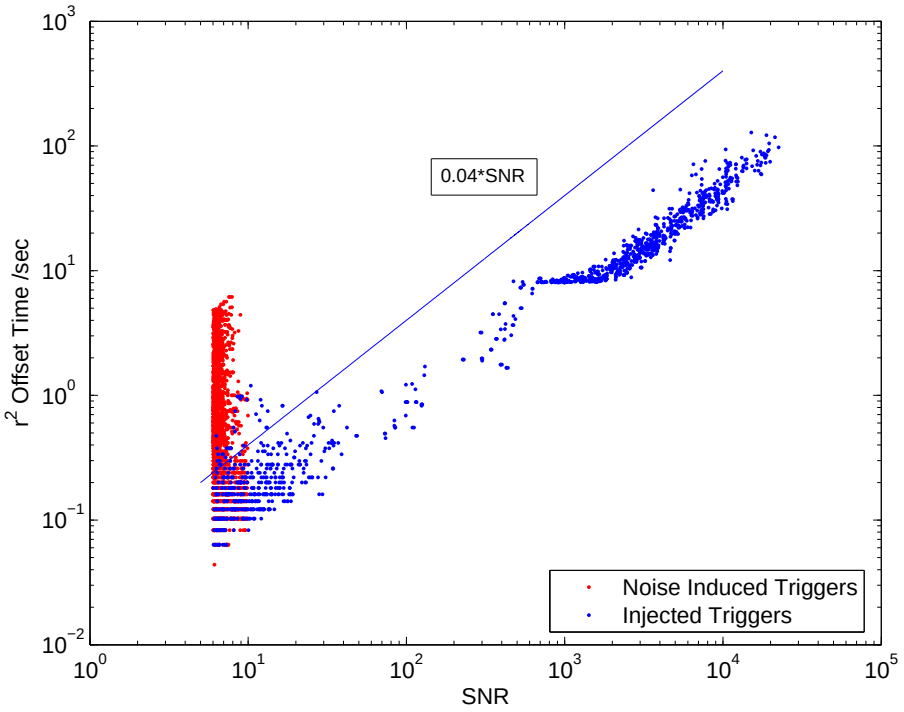

FIG. 10: Plot of r^2 Time Offset

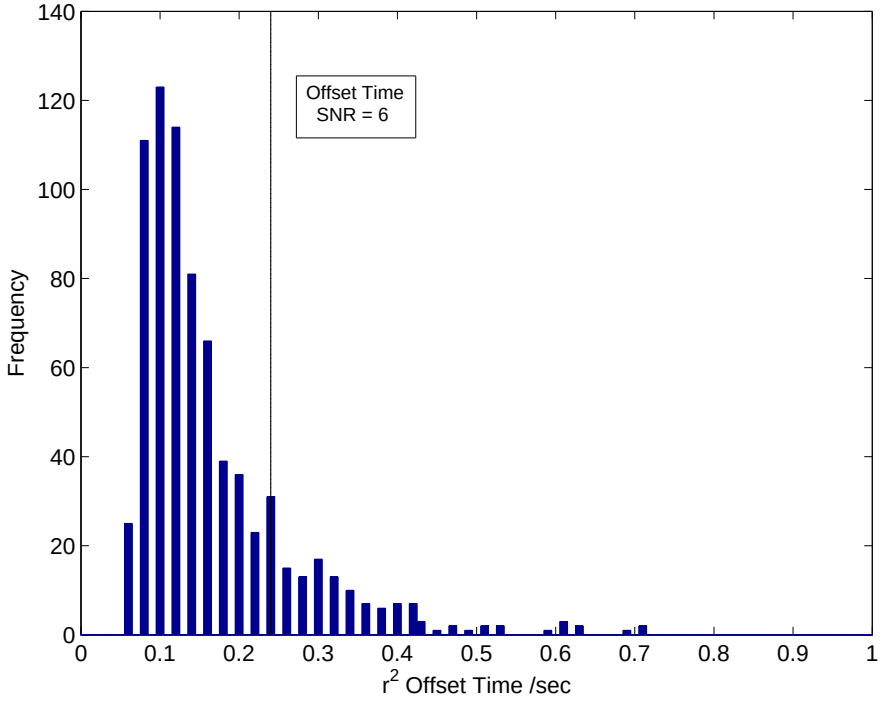

FIG. 11: r^2 Time Offset Histogram for Injections where $6 < \rho < 7$

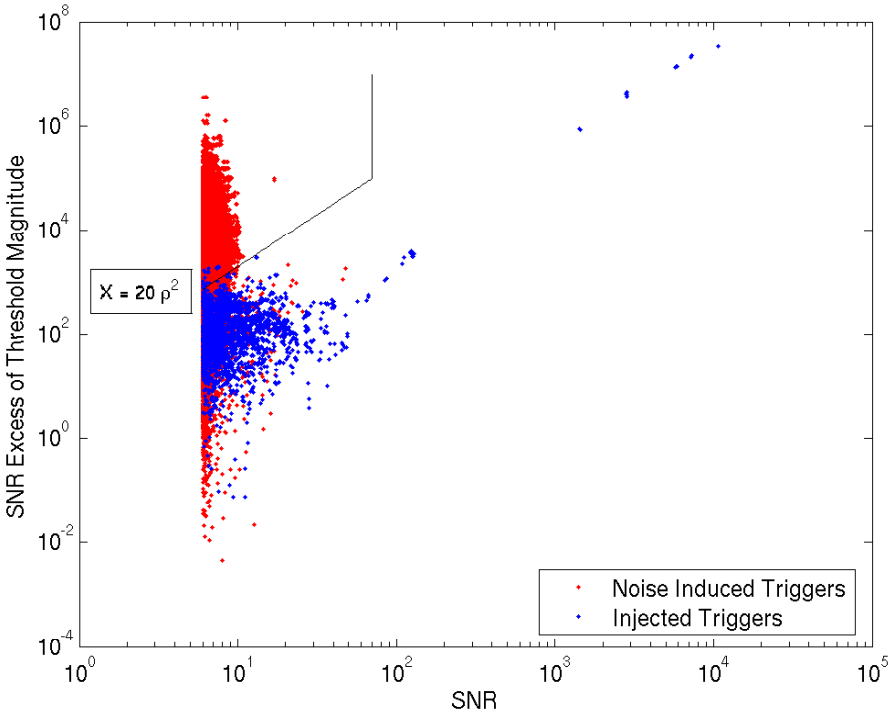

FIG. 12: Plot of ρ^2 Excess of Threshold ($\rho^2 = 9$, Offset Time $= 35 + 5\,\rho^{2/3}$ ms)

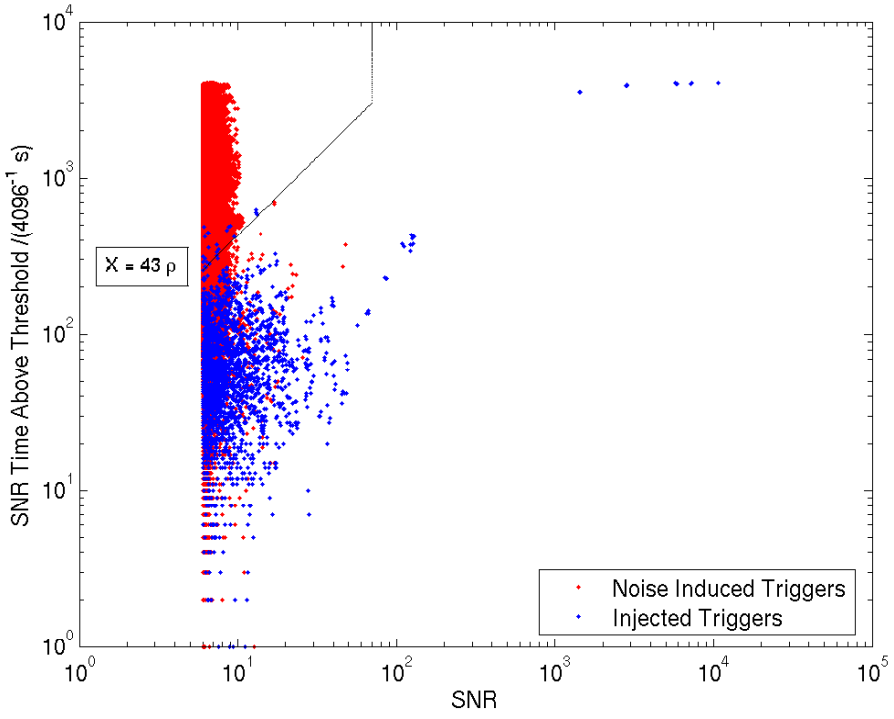

FIG. 13: Plot of ρ^2 Time Above Threshold ($\rho^2 = 9$, Offset Time $= 35 + 5\,\rho^{2/3}$ ms)

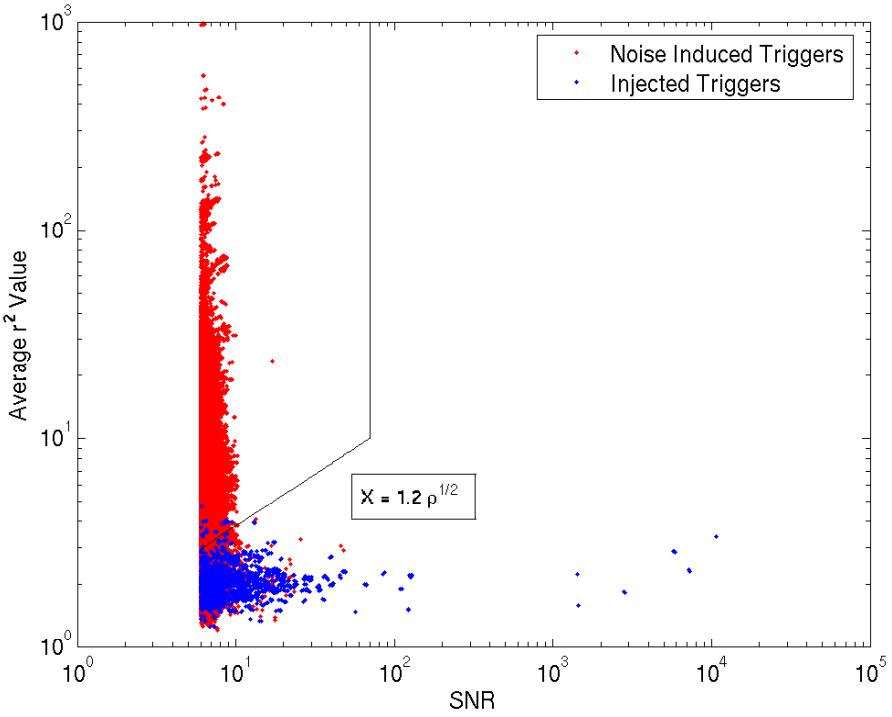

FIG. 14: Plot of Average r^2 Test During One Second Interval

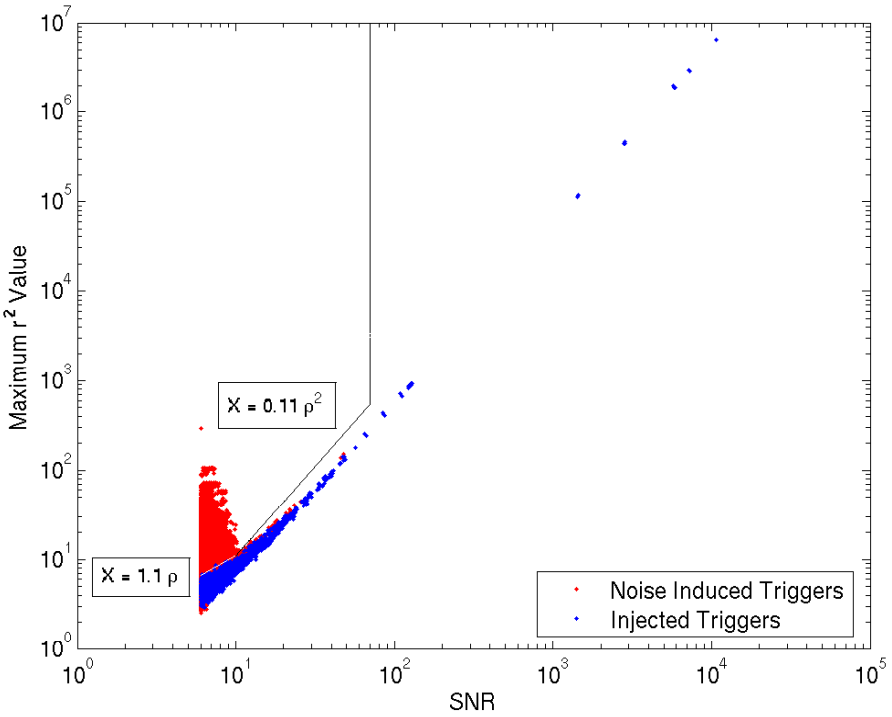

FIG. 15: Plot of Maximum r^2 Value During 0.25s Preceding Trigger

V. DISCUSSION

Due to time constraints, I have not been able to determine the most effective thresholds for each test. However, it is clear that the particular veto conditions developed by Rodriguez for the S3 data are not as effective on the S4 data. This could suggest that the noise has become more well-behaved. The limiting factor for the test's efficiency is the large number of triggers that never exceed the threshold. I anticipate that a lower threshold would prove to be much more effective.

The SNR offset time was developed to include a constant offset of 35 ms. This was added on top of the value determined by analysis of the graphs. It was used as it seemed that pure signals always had an influence during this period of time. However, it is obvious that it had a detrimental effect on the efficiency. Inspecting the histograms in Figs 9 & 11, it appears that the offset times chosen were both too long anyway. For the next run, I would choose to halve the offset time.

These results also indicate that an excess of threshold test is not any more powerful than a time above threshold test. Overall, these tests are most effective at vetoing small SNR triggers. This may be because only small SNR noise-induced triggers were inspected. However, the large SNR triggers only correspond to a small fraction of those recorded.

In further investigations, it would be helpful to obtain more injected signals for analysis. This would make the distribution of the values more apparent. It is also important to identify the optimum thresholds for each test. This project has identified a set of tests that can eliminate 82% of the noise-induced triggers while only neglecting 0.9% of the injected signals. The goal of 90% was not reached but the tests developed are still very effective.

In section II(C), on the effect of template mismatch, it was apparent that one signal would frequently produce triggers in a number of different templates. In order to reduce the number of triggers, only one trigger should need to be recorded. This would be the one with the largest SNR and lowest r^2.

In order to improve the power of the discriminating tests, I believe that following the method used by Guidi would be beneficial. The contribution from the pure GW to the matched filter output would be subtracted. This would leave a signal that should only be due to noise. An offset time would not be required, and a time above threshold test could be used to detect excessive signals.

[1] LIGO Scientific Collaboration, Phys. Rev. D **69**, 102001 (2004)
[2] C. M. Will, A. G. Wiseman, Phys. Rev. D **54**, 4813 (1996)
[3] B. Allen, Phys. Rev. D **71**, 062001 (2005)
[4] N. Christensen, Class. Quantum. Grav. **22**, S1059 (2005)
[5] N. Christensen, P. Shawhan, G. González, Class. Quantum. Grav. **21**, S1747 (2004)
[6] P. Shawhan, E. Ochsner, Class. Quantum. Grav. **21**, S1757 (2004)
[7] A. Rodriguez, [http://www.lsc-group.phys.uwm.edu/iulgroup/investigations/s3index.html]
[8] G. Guidi, Class. Quantum. Grav. **21**, S1767 (2004)

APPENDIX A: ALTERATIONS TO LAL CODE

Type of Test	Variable
. r^2 Time Above Threshold	. rsqveto_duration
ρ^2 Excess of Threshold	snrtest1, other1
ρ^2 Time Above Threshold	snrtest2, other2
Average r^2 Test	avrtest
r^2 Variance	testvar
r^2 Time Offset Test	test0
Time of r^2 Side-Peak	other3

```
Index: $LAL_LOCATION/src/lal/packages/findchirp/src/FindChirpFilterOutputVeto.c, revision 1.8
===================================================================
line 60:
void
LALFindChirpFilterOutputVeto(
    LALStatus                           *status,
    SnglInspiralTable                  **eventList,
    FindChirpSegment                    *segment,
    REAL4Vector                         *chisqVec,
    REAL8                                deltaT,
    FindChirpFilterOutputVetoParams     *params,
    REAL4TimeSeries                     *rhosqVec,
    COMPLEX8Vector                      *qVec,
    REAL4                                qNorm,
    InspiralTemplateNode                *tmpltCurrent,
    INT4                                 sampleRate
    )

{
  SnglInspiralTable    *event;
  double                SegmentTime,Chi2,Chi2Bins,SNR2,avrtest,testvar;
  double                test0,test1,test2,samp_time,samp_int,snhctest,jend;
  double                snrtest1,snrtest2,other1,other2,other3,other4;
  int                   i,j,sample,istart;
===================================================================
line 128:
  event = *eventList;
  while ( event )
  {
    SegmentTime = (double)(event->end_time.gpsSeconds)+1.0e-9*(double)(event->end_time.gpsNanoSeconds)
     -(double)(segment->data->epoch.gpsSeconds);

    if( modf( sampleRate*SegmentTime, &samp_int) < 0.5 ) sample = (int)(samp_int);
    else                                                 sample = (int)(samp_int)+1;

    Chi2Bins = (double)(segment->chisqBinVec->length) - 1;

    test0 = 0.0; test1 = 0.0; test2 = 0.0; avrtest = 0.0; testvar = 0.0; samp_time = 80.0;

    snrtest1 = 0.0; snrtest2 = 0.0; other1 = 0.0; other2 = 0.0; other3 = 0.0; other4 = 0.0;

    SNR2 = double(qNorm)*(pow(double(qVec->data[sample].re),2) + pow(double(qVec->data[sample].im),2));

    jend = sampleRate*2*pow(log10(SNR2),2)/samp_time;
    if( jend*samp_time + 20.0 > sample ) jend = floor(sample/samp_time);

    for ( j = 1; j < jend; ++j){
      snhctest = 0.0;
      istart = sample - 20 - (int)samp_time*j;
      if ( istart < 1 ) istart = 1;
      for ( i = istart; i < istart + samp_time; ++i ){
Chi2 = (double)(chisqVec->data[i]);
snhctest = snhctest + Chi2;
```

```
      }
      if( test0 < 0.1 && snhctest/samp_time < 2.0 ){
test0 = j*samp_time + 20.0;
break;
      }
    }

    for( i = sample; i > sample - 1001; --i ){
      Chi2 = (double)(chisqVec->data[i]);
      if( Chi2 > test1){
test1  = Chi2;
other3 = (double)(sample - i);
      }
    }

    istart = (int)(sample - 0.04*sampleRate*sqrt(SNR2));
    if( istart - 1*sampleRate < 1 ) istart = 1*sampleRate + 3;
    for ( i = istart; i > istart - 1*sampleRate; --i ){
      Chi2 = (double)(chisqVec->data[i]);
      avrtest = avrtest + Chi2;
      testvar = testvar + pow(Chi2,2);
    }
    avrtest = avrtest/(1.0*sampleRate);
    testvar = sqrt( testvar/(1.0*sampleRate) - pow(avrtest,2) );

    for ( i = sample; i > sample - 4.0*test0; --i ){
      if( i < 1 ) break;
      SNR2  = (double)(qNorm)*(pow((double)(qVec->data[i].re),2) + pow((double)(qVec->data[i].im),2));
      if( SNR2 < 1.0 ){
test2 = (double)(sample - i);
break;
      }
    }

    istart = (int)(sample - 20*pow(SNR2,1.0/3.0) - 0.035*sampleRate); /* 10*pow() - 0  */
    if( istart - 1*sampleRate < 1 ) istart = 1*sampleRate + 1;
    for ( i = istart; i > istart - 1*sampleRate; --i ){
      SNR2  = (double)(qNorm)*(pow((double)(qVec->data[i].re),2) + pow((double)(qVec->data[i].im),2));
      if( SNR2 > 9.0 ){
snrtest1 = snrtest1 + SNR2 - 9.0;
snrtest2 = snrtest2 + 1.0;
      }
      if( SNR2 > 6.0 ){
other1 = other1 + SNR2 - 6.0;
other2 = other2 + 1.0;
      }
    }

    event->test0      = test0/sampleRate;
    event->test1      = test1;
    event->avrtest    = avrtest;
    event->testvar    = testvar;
    event->gwf_number = other3;

    /* event->gwf_number = 15.0*(double)(tmpltCurrent->tmpltPtr->number)+(double)(segment->number); */
    event->snroff     = test2;
    event->snrtest1   = snrtest1;
    event->snrtest2   = snrtest2;
    event->other1     = other1;
    event->other2     = other2;


    event = event->next;
  }
}
```