

**Improving Searches for Gravitational Waves using Signal Isolation Tests  
SURF Progress Report 2: August 1, 2005**

Sebastian Cassel

Supervisor: Peter Shawhan

**I. INTRODUCTION**

The LIGO search for gravitational waves (GW) from binary inspiral sources relies on a matched filtering technique. This can be shown to be the optimal method for detection when the noise present is Gaussian and stationary. However, in practice the detector noise is sometimes variable and ill-behaved. The result is many noise-induced signals being recorded as GW events. This project intends to introduce a test that will screen out most of the noise-induced signals.

At the time of the last report<sup>1</sup>, the characteristics of noise-induced signals and hardware injected signals had been investigated. The noise-induced signals were seen to possess elevated signal-to-noise ratio (SNR) and  $\chi^2$  time series in the vicinities of their triggers. The variation of their signal magnitudes was also greater than those of the injected signals. These properties were used to construct a test that will be discussed in section III.

**II. FURTHER STUDY OF INJECTED SIGNALS**

The effect of template mismatch on signals was investigated. This was important in order to understand how the waveforms might differ for various true signals. It would be necessary to develop a test that did not discriminate against the characteristics observed.

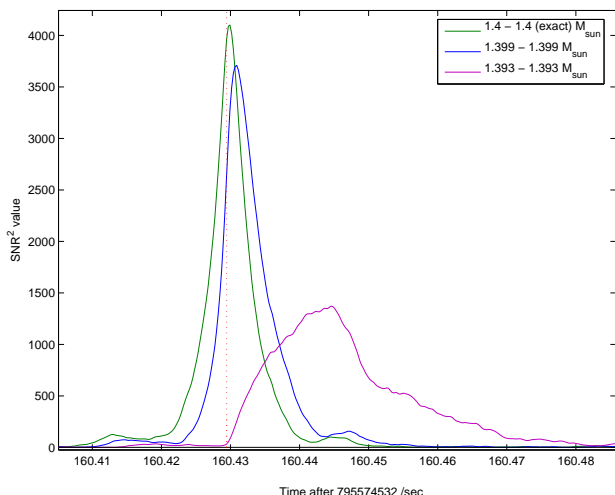


FIG. 1: Plot of  $\text{SNR}^2$  for Strong Injected Signal

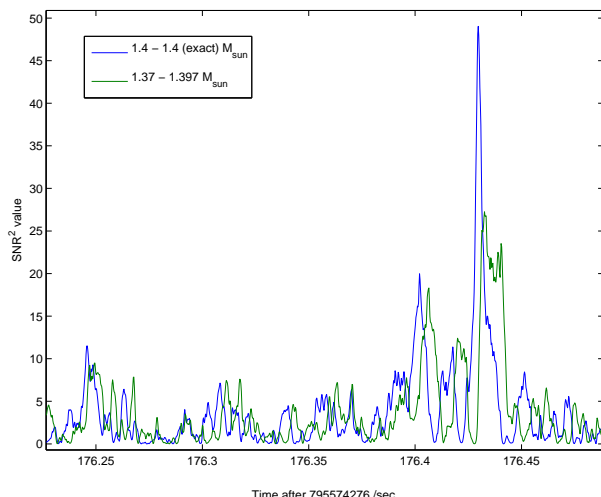


FIG. 2: Plot of  $\text{SNR}^2$  for Weak Injected Signal

Fig 1 shows the matched filter output from a strong injected signal for three different templates. The masses for the injected signal were 1.4-1.4 solar masses. When the template masses exactly matched the source masses, a symmetrical waveform was observed. As the template masses were lowered, the SNR peak shifted forward in time and reduced in magnitude. For templates that were equally spaced around the true masses (above and below), there was a mirror symmetry of the SNR signal about the true coalescence time. Fig 2 shows a similar effect for the weak injected signal.

The  $\chi^2$  behaviour was also investigated. On the next page, Figs 3-5 show how the strong SNR signal varied with different templates. Again, a similar effect was seen with the weak injected signal. The waveform from the exact template has a symmetrical nature. As the masses were lowered, the  $\chi^2$  minimum moved forward in time, and rose. The side-peaks also assumed some antisymmetric nature. Figs 4 and 5 demonstrate the mirror symmetry between templates above and below the true values.

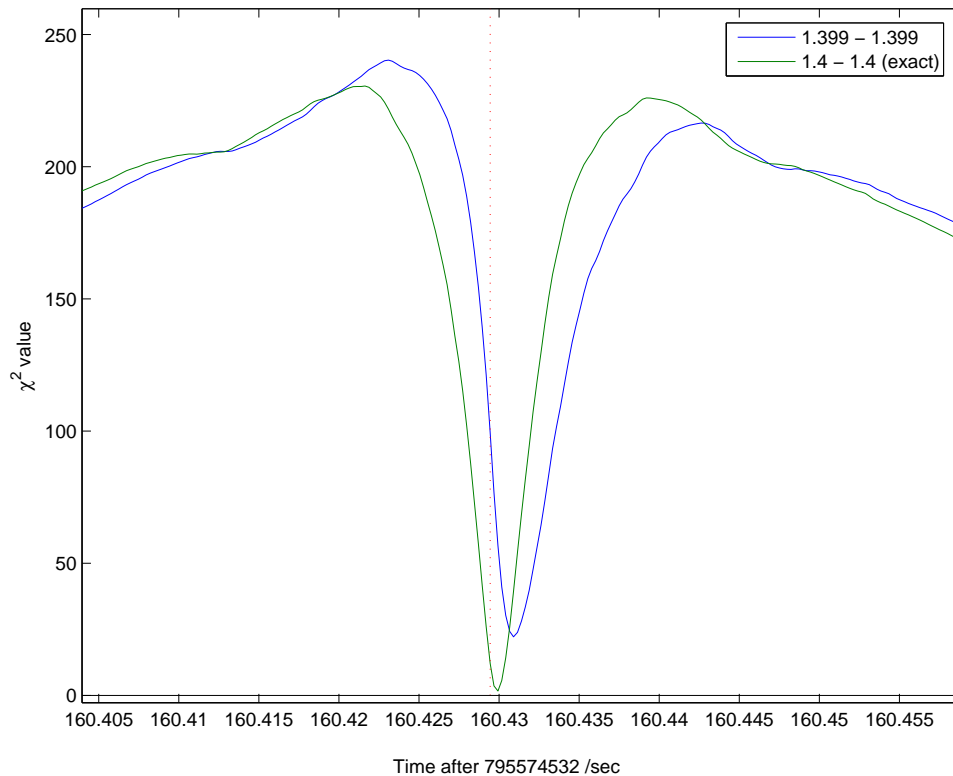


FIG. 3: Plot of  $\chi^2$  for strong injected signal

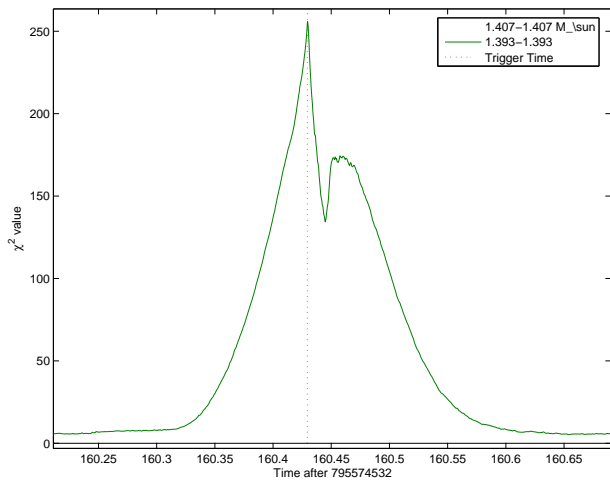


FIG. 4: Plot of  $\chi^2$  for strong injected signal

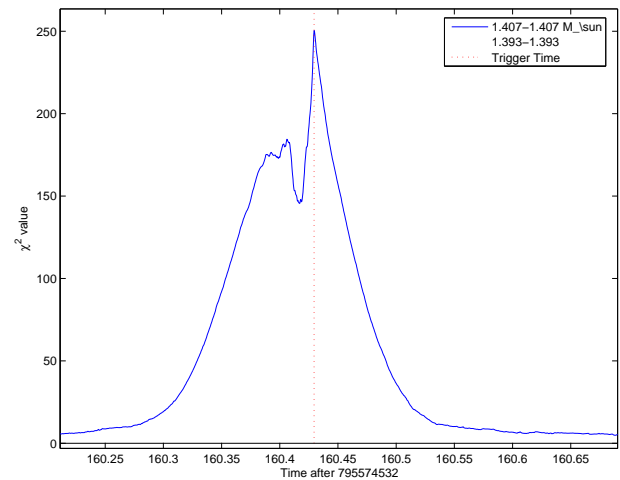


FIG. 5: Plot of  $\chi^2$  for strong injected signal

It should be noted that a large set of the noise-induced signals also possess these side-peaks. I expect that their magnitude is directly related to the SNR of the trigger. Further investigation could determine whether there is any difference in their relationship for true or noise-induced signals.

Other studies found that some noise-induced signals only had elevated signal magnitudes *after* the trigger. Also, some signals only had elevated signal magnitudes in close vicinity to the trigger. This must be considered during the development of the test. An offset time might miss the excess signal. When the time window tested is too long, the

presence of the excess signal might again be neglected by averaging. Similarly, for a time-above-threshold test, the fraction of time spent above the threshold could be reduced to a negligible value.

### III. SIGNAL ISOLATION TEST

#### A. Development of LAL

The test that is being developed is being incorporated into the function `LALFindChirpFilterOutputVeto`. It is necessary to list what variables the function will use. This required alteration of the following files:

<code>FindChirpFilterOutputVeto.c</code>	(in directory <code>\$LAL_LOCATION/src/lal/packages/findchirp/src</code> )
<code>FindChirpFilterOutputVeto.h</code>	(in directory <code>\$LAL_LOCATION/src/lal/packages/findchirp/include</code> )
<code>FindChirpFilter.c</code>	(in directory <code>\$LAL_LOCATION/src/lal/packages/findchirp/src</code> )

The source code was modified so that some arguments of the `LALFindChirpFilterSegment` function were passed along to the `LALFindChirpFilterOutputVeto` function. For any arguments not listed by `LALFindChirpFilterSegment`, they needed to be added into its structure before inclusion into `LALFindChirpFilterOutputVeto`. This was done by changing the files listed below:

<code>FindChirpFilter.c</code>	(in directory <code>\$LAL_LOCATION/src/lal/packages/findchirp/src</code> )
<code>FindChirp.h</code>	(in directory <code>\$LAL_LOCATION/src/lal/packages/findchirp/include</code> )
<code>inspiral.c</code>	(in directory <code>\$LAL_LOCATION/src/lalapps/src/inspiral</code> )

Using the segments of time previously analysed, the developed code was tested to be working properly. To ensure some of the calculations were correct, messages were generated in the verbose output of the `lalapps_inspiral` program. In order to include the results of the test into the `SnglInspiralTable` structure, further changes<sup>2</sup> were also necessary (see Appendix).

#### B. $\chi^2(t)$ discrimination

The code responsible for the test was added to `FindChirpFilterOutputVeto.c`. A test was initially introduced that counted the number of samples over some  $\chi^2$  threshold in the vicinity of each trigger. Fig 7 shows a cumulative plot over time for triggers within a 2048s period containing the injected signals discussed in the previous report. Fig 6 shows the results from a noisy 2048s period.

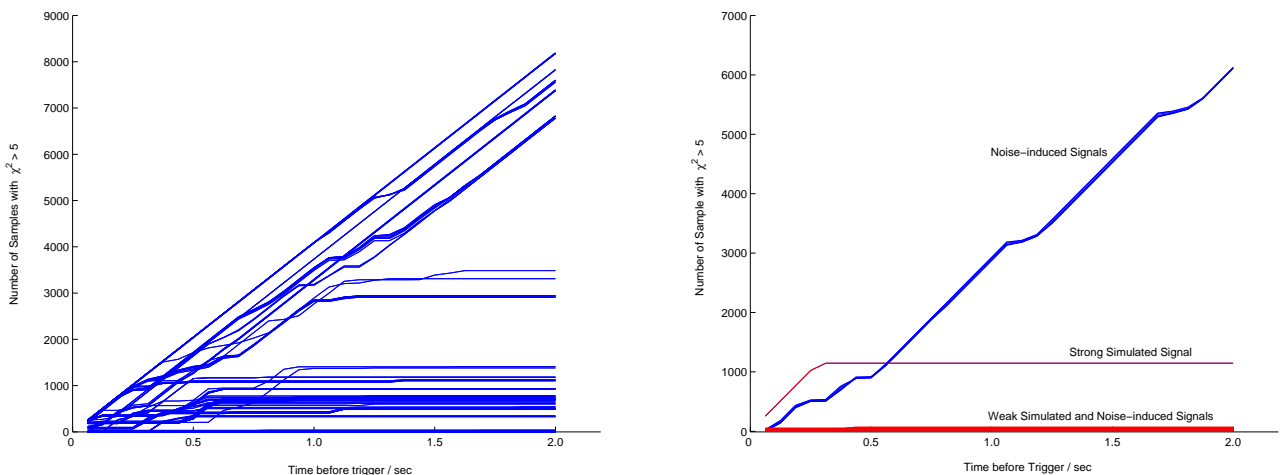


FIG. 6: Period only containing noise-induced signals of SNR 6-8 FIG. 7: Period containing injected signals of SNR 7 and 64

Cumulative Plots with Number of Samples with  $\chi^2 > 5$

The injected signal with a strong SNR had a consistently high  $\chi^2$  for the 0.3s preceding the trigger, as noted in the previous report. This information should be used to make the test account for these expectations. Many noise-induced signals frequently exceeded the threshold throughout the two second period. If the pure waveform of the  $\chi^2(t)$  were known in the vicinity of a trigger, it could be subtracted following a similar approach to Guidi<sup>3</sup>. Any excess signal close to the trigger could then be discriminated against in the test. Alternatively, an SNR dependent time offset could be used in order to avoid the side-peaks of the  $\chi^2$  signal.

Despite a large number of signals exceeding the threshold, many did not during the two second window observed. This introduces a problem with distinguishing between the weak injected signal, and some noise-induced signals. One option would be to have an SNR dependent threshold. However, the same problem would still be expected. Another option would be to consider the average value over some time window, and possibly the variance of the signal.

### C. Comparison with SNR( $t$ ) discrimination

One difference between the SNR and  $\chi^2$  signals is that the SNR is more variable. The  $\chi^2$  signal is fairly smooth, but the SNR signal goes from its peaks to roughly zero in very short timescales. This suggests that a time-above-threshold and crossings test<sup>4</sup> would produce similar results when applied to the SNR signal. However, differences between the two tests will occur when the SNR is elevated for long periods of time. A test discriminating against the average value in some time window would be unsuitable due to frequent values of very low magnitude.

A time offset would again be suitable due to width of the peak of a pure signal. Some threshold test seems to be the most appropriate choice at this stage. In order to use information about how high a signal is over a threshold, instead of assigning a positive result with one unit in the test, the excess SNR could be used.

### D. Test Parameters

Before the test can be implemented on a full set of data, an appropriate offset time for the SNR and  $\chi^2$  must be determined. The time window must be decided and any SNR dependence. Suitable thresholds must also be considered. The effect of different variables will be tested, such as the template masses and low frequency cutoff.

I have begun to look at large sets of software injected triggers, using the CIT computing cluster. In order to avoid poor quality data, the following data quality flags were used to veto time periods:

ADC_OVERFLOW	H1UP_H2UNLOCKED	JET
CALIB_LINE_DROPOUT	INJECTION_BURST	OUT_OF_LOCK
CALIB_LINE_V03_1_SEC	INJECTION_PULSTART	OUTSIDE_S4
CALIB_LINE_V03_60_SEC	INJECTION_STOCHASTIC	WIND

Only the software injected signals are being considered. In order to determine the offset based on the pure signal, triggers within noisy periods will not be included.

## IV. FUTURE WORK

The test will be developed and tested to produce useful results. Further refinements to the test will be considered. Once confident in the form of the test, it will be applied to a full data run, and the effectiveness determined. The characteristics of noise-induced signals with large SNRs should be investigated, as well as the injected signals.

## APPENDIX A: ALTERATIONS TO LAL CODE

Index: src/insprial/insprial.c

```

=====
RCS file: /usr/local/cvs/lscsoft/lalapps/src/insprial/insprial.c,v
retrieving revision 1.179
diff -r1.179 insprial.c
1536d1535
< #if 0
1539d1537

```

```

< #endif
1942c1940
<             &eventList, fcFilterInput, fcFilterParams ), &status );
---
>             &eventList, fcFilterInput, fcFilterParams, tmpltCurrent ), &status );

```

Index: packages/findchirp/include/FindChirp.h

```

=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/findchirp/include/FindChirp.h,v
retrieving revision 1.78
diff -r1.78 FindChirp.h
788c788,789
<     FindChirpFilterParams      *params
---
>     FindChirpFilterParams      *params,
>     InspiralTemplateNode      *tmpltCurrent

```

Index: packages/findchirp/include/FindChirpFilterOutputVeto.h

```

=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/findchirp/include/FindChirpFilterOutputVeto.h,v
retrieving revision 1.4
diff -r1.4 FindChirpFilterOutputVeto.h
116c116,120
<     FindChirpFilterOutputVetoParams  *params
---
>     FindChirpFilterOutputVetoParams  *params,
>     COMPLEX8Vector                   *qVec,
>     REAL4                             qNorm,
>     InspiralTemplateNode              *tmpltCurrent

```

Index: packages/findchirp/src/FindChirpFilter.c

```

=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/findchirp/src/FindChirpFilter.c,v
retrieving revision 1.111
diff -r1.111 FindChirpFilter.c
90c90,91
<     FindChirpFilterParams      *params
---
>     FindChirpFilterParams      *params,
>     InspiralTemplateNode      *tmpltCurrent
475c476
<     if ( params->filterOutputVetoParams )
---
>     if ( 1 || params->filterOutputVetoParams )
477,478c478
<     LALFindChirpFilterOutputVeto( status->statusPtr, eventList, input->segment, params->chisqVec,
deltaT, params->filterOutputVetoParams );
---
>     LALFindChirpFilterOutputVeto( status->statusPtr, eventList, input->segment, params->chisqVec,
deltaT, params->filterOutputVetoParams, params->qVec, norm, tmpltCurrent );

```

Index: packages/findchirp/src/FindChirpFilterOutputVeto.c

```

=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/findchirp/src/FindChirpFilterOutputVeto.c,v
retrieving revision 1.8
diff -r1.8 FindChirpFilterOutputVeto.c
64c67,71

```



---

```
> \process:process_id:0\,\%s\,\%s\,\%s\,%d,%d,%22.16e,%d,%d,%22.16e,%22.16e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%e,%d,%22.16e,%e,%1f,%11d
```

Index: packages/support/src/LIGOLwXML.c

```
=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/support/src/LIGOLwXML.c,v
retrieving revision 1.33
diff -r1.33 LIGOLwXML.c
457a458,460
> tablePtr.snglInspirarTable->test0,
```

Index: packages/support/src/LIGOLwXMLRead.c

```
=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/support/src/LIGOLwXMLRead.c,v
retrieving revision 1.49
diff -r1.49 LIGOLwXMLRead.c
1141c1141,1144
< {event_id, -1, 40},
---
> {test0, -1, 40},
> {event_id, -1, 41},
1400c1403,1415
< else if ( tableDir[j].idx == 40 )
---
> else if ( tableDir[j].idx == 40 )
> {
> thisEvent->test0 = r8colData;
> }
> else if ( tableDir[j].idx == 41 )
```

Index: packages/tools/include/LIGOMetadataTables.h

```
=====
RCS file: /usr/local/cvs/lscsoft/lal/packages/tools/include/LIGOMetadataTables.h,v
retrieving revision 1.42
diff -r1.42 LIGOMetadataTables.h
372a373,375
> REAL8 test0;
```

---

<sup>1</sup> S. Cassel, SURF Progress Report 1: July 5, 2005

<sup>2</sup> A. Rodriguez, [<http://www.lsc-group.phys.uwm.edu/iulgroup/investigations/s3index.html>]

<sup>3</sup> G. Guidi, Class. Quantum. Grav. **21**, S1767 (2004)

<sup>4</sup> P. Shawhan, E. Ochsner, Class. Quantum. Grav. **21**, S1757 (2004) [gr-qc/0404064]