

LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
– LIGO –
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type	LIGO-T010158-00-R	10/15/01
SIMULATION OF LIGO INTERFEROMETERS RESPONSE TO GRAVITATIONAL WAVES AND NOISE		
Mihail Amarie, Alan Weinstein		

Distribution of this draft:

This is an internal working note
Of the LIGO Project

California Institute of Technology
LIGO Project – MS 51-33
Pasadena, CA 91125
Phone (626)395-2129
Fax (626)304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – MS 20B-145
Cambridge, MA 01239
Phone (617)253-4824
Fax (617)253-7014
E-mail: info@ligo.mit.edu

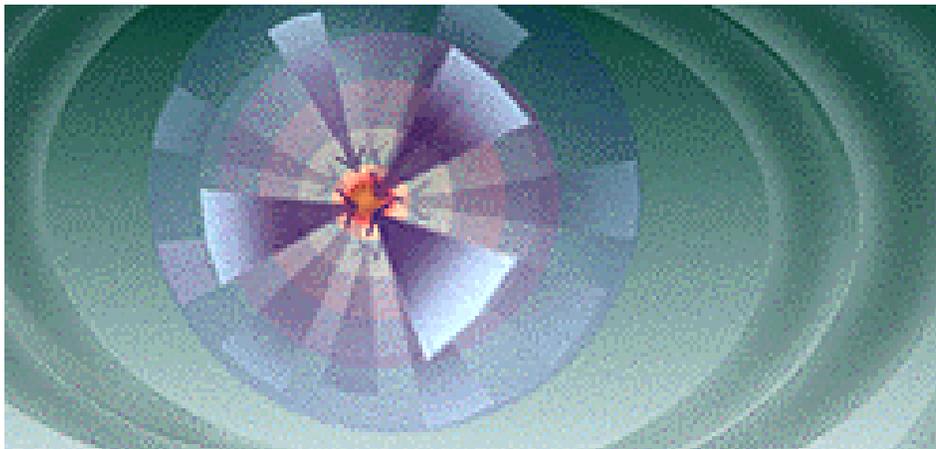
WWW: <http://www.ligo.caltech.edu>

**SIMULATION OF LIGO INTERFEROMETERS
RESPONSE TO GRAVITATIONAL WAVES
AND NOISE**

Mihail Amarie

Mentor: Alan J. Weinstein

30 September 2001



Artist's conception of a ripple created by an asymmetric Supernova Burst¹

Abstract

The Gravitational wave channel data stream from multiple interferometers in the LIGO (and other gravitational waves detectors) projects is expected to contain mostly noise, with occasional signals from astrophysical sources of GW (gravitational waves) bursts.

The LIGO Data Analysis System (LDAS) searches through individual detector data streams for such bursts and populates a database with these "event triggers".

In preparation for real data, I have developed simulations of the results of such searches, populating a test database. I then performed a study of the statistics of single interferometer bursts, as well as coincidence events from multiple interferometers due to signals from simulated GW bursts.

1. Introduction

a) About LIGO:

LIGO² (Laser Interferometer Gravitational-Wave Observatory) is a facility dedicated to the detection of cosmic gravitational waves. They were predicted a long time ago, but because the technique was not sufficiently advanced, could not have been detected because their effect on objects on Earth is very small. The method used in detecting such small oscillations is laser interferometry.

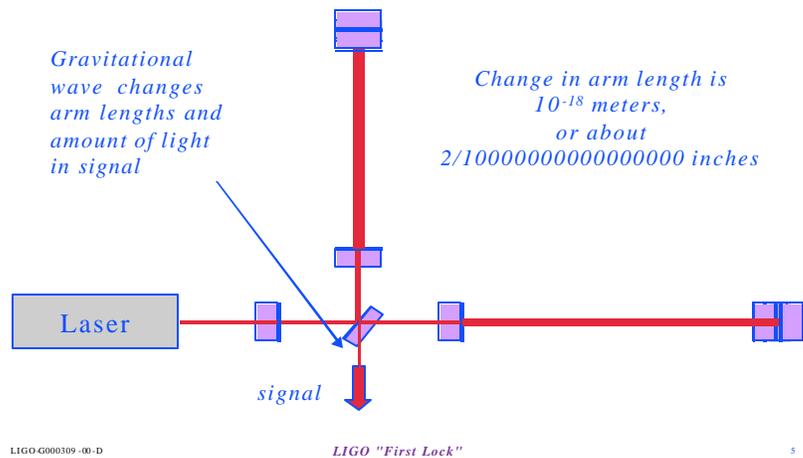


Fig 1. A schematic of the LIGO optical configuration; a power recycled Michelson interferometer with Fabry-Perot arms.

The device used for this purpose, shown in figure 1 is a power recycled Michelson interferometer with Fabry-Perot Arms that can detect displacements much less than the diameter of a nucleus. It consists of a laser that emits electromagnetic waves of very precise frequency in order not to affect the measurements. This electromagnetic wave goes through a beam splitter, which splits it into 2 components. Each one of them enters a Fabry-Perot resonant cavity in which the very small displacement caused by a possible gravitational wave is amplified many times (the number of bounces of the light in the Fabry-Perot resonant cavity). All this amplification given the displacement is still very small. In order to detect such small displacements of about 10^{-18} m the device must be isolated from the environmental noise. In order to exclude the eventuality of an accidental source of vibrations located on Earth

three similar devices were built 3000 km away one from another (two at Hanford WA and one at Livingston LA). The location of the LIGO detectors and the orientation of the arms of the interferometers are shown in the figure below:



Fig 2. The LDAS³ and the LIGO sites.

b) About my summer work:

As a SURF student I was a member of the "40m" group at Caltech. The "40m" is a prototype laboratory for the LIGO project, which is a Michelson interferometer with two arms of 40 meters each. Even though I was a member of the "40m lab" group, my work was not very much connected to its activity; as my project title states, I was working with the data analysis of LIGO. My summer activity was focused on two distinctive aspects of the data analysis. During the

first half I used to work mostly in signal processing, while in the second part I did single and multi-interferometer coincidence analysis.

As the main development environment I used [Root](#)⁴, which is mainly a C++ interpreter, which also comes with many libraries. These libraries are very useful for statistical calculations as well as for histogram accumulating and plotting.



Root Logo

My contribution to the LIGO project consisted of the development of a set of macros in preparation for a mock data challenge of the LIGO Data Analysis System (LDAS³). Some of the macros process signals coming from interferometers, especially from the gravitational wave channels, and other macros perform a coincidence statistical analysis for more than one interferometer.

2. Signal Processing:

For the first part of the period I was preparing the environment for the tools I was going to use and studying their documentation. As a part of this process I installed the [LAL](#)⁵ (LIGO/LSC Algorithm Library) and [LALWRAPPER](#)⁶. LAL is a C library of algorithms that LIGO uses to process its data. LALWRAPPER is its C++ wrapper. It allows LAL code to be used in the

LDAS environment, which uses C++ coding. LDAS is the tool that will perform the advanced GW signal searching for LIGO. It has the advantage of being able to use Beowulf computer clusters, which provide sufficient processing power for the system. A Beowulf cluster is a ‘network’ of computers connected to each other, forming an inexpensive multiple-processor supercomputer, able to divide a task and perform it in a much shorter time than only one computer could do it.

Using Root, I developed a series of macros with functions that generate simulated noise from the LIGO interferometers and provide the necessary calculation tools and some applications.

The functions I built are:

✦ *hstrain* ($t, m1, m2$), returns the value of the GW strain of a chirp⁷ where $m1$ and $m2$ are the masses of the compact bodies in terms of the solar mass and t is the time we want the strain returned (0 is the end time), so it should be negative.

An example of output of this function can be seen in Figure 3.

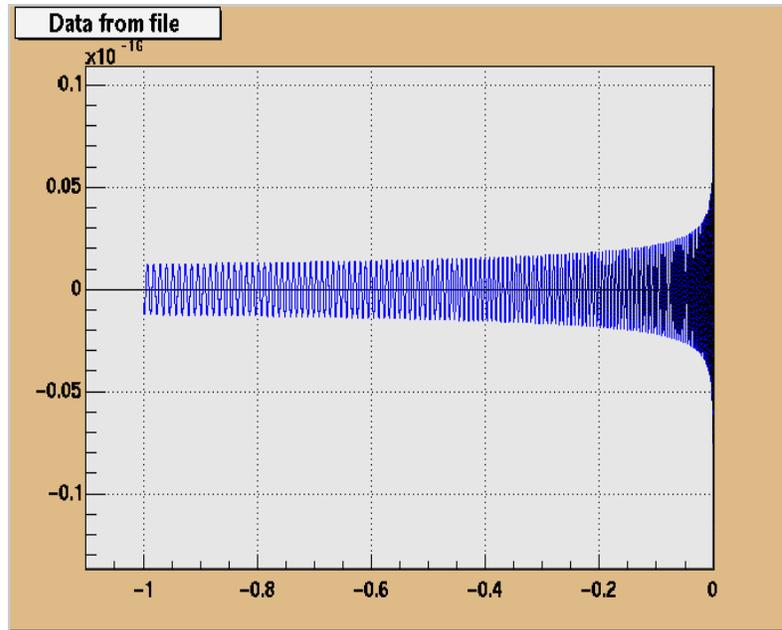


Fig 3. A chirp in the time domain.

The chirp is the waveform produced by two compact bodies that orbit one around the other emitting gravitational waves. As they emit gravitational waves they lose energy and get closer and move with higher angular velocity. This corresponds to the increase both in frequency and in amplitude seen in the figure above.

✦ *cross(x,y,z)* calculates the cross product of the 3D vectors x and y and "returns" the vector z

✦ *gnoise(x,m)* generates gaussian noise with sigma 1 in the vector x of length m

✦ *addgn(x,m,a)* adds gaussian noise of sigma float a to the vector x of size m

An example of output of gnoise function can be seen in Fig. 4.

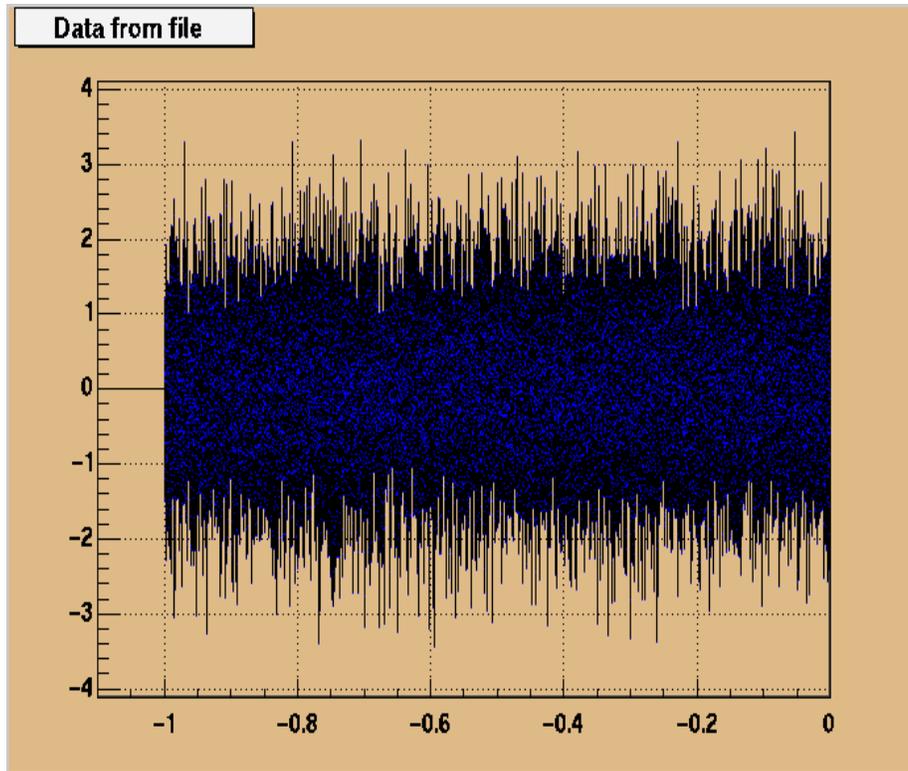


Fig 4. Gaussian white noise.

The white noise consists of random numbers in the time domain with a gaussian distribution with mean 0. This noise has a uniform distribution in the frequency domain too, which is why it is referred to as white noise.

✦ *addhp(x,m,a) adds chi-square noise of sigma float a to the vector x of size m*

In all the functions above when a parameter or a returned value refers to a vector, it represents the pointer to the array of values of that vector.

In addition to all these functions I also developed some simple functions that perform vector calculations like addition, scalar multiplication and dot

product.

Using the above functions and the [Frame Library](#)⁸ I developed two applications. A frame is a file format used for storing interferometer data from all the channels. It was chosen so that data from all the interferometers can be used together.

- The first one is a simple burst filter. It combines a burst signal with noise, takes its Fourier transform, and then slices the frequency domain and looks for signals exceeding a given threshold in each of the slices. In the testing application of the filter I randomly inserted a signal in half of the samples and I tested the efficiency and the fake rate of the filter by varying the signal to noise threshold. The efficiency is the ratio of number of detected real events to inserted 'real' events and the fake rate is the ratio of the number of detected fake events to the number of events where there was no signal. In fig. 5 I used white noise and the signal is a chirp (that's why the slope is so steep) with the maximum amplitude 1.5 times higher than the standard deviation of the white noise. The x-axis is the ratio $(\text{threshold} - \text{noise})/\text{noise}$, and it is lower than one might expect because of the low values of the signal I chose. (I chose that both graphs could be compared on the same axis).

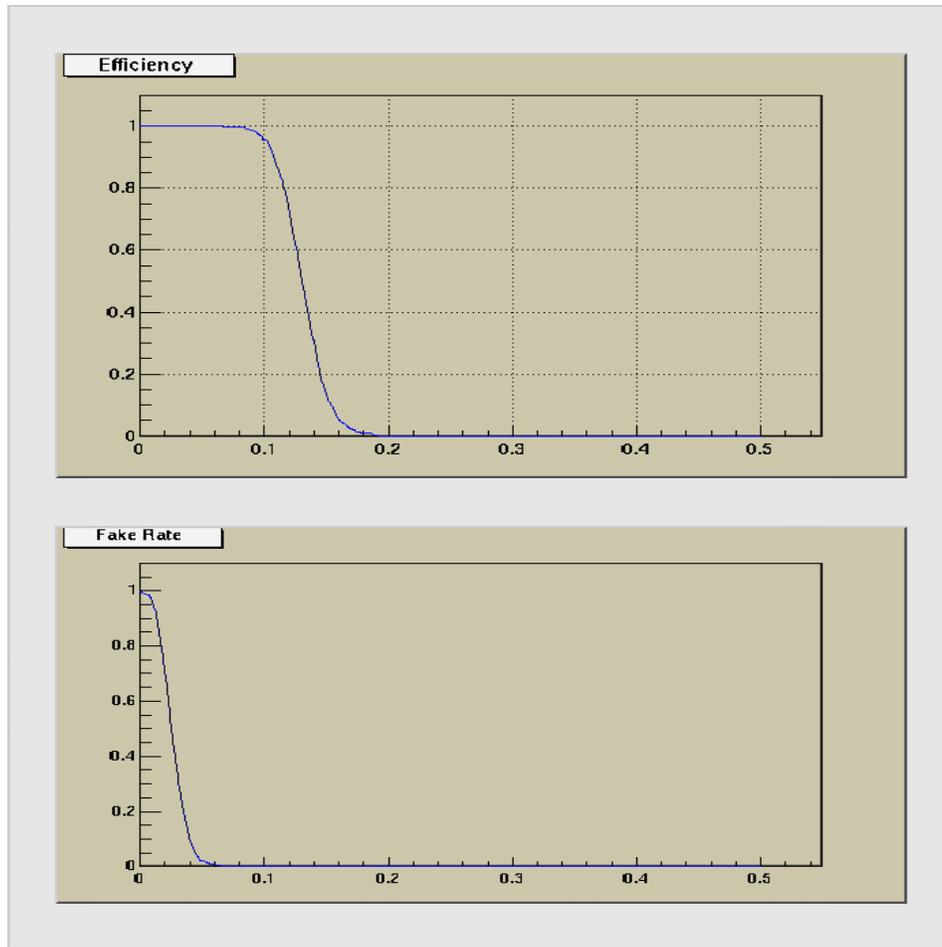


Fig. 5 The Fake Rate and the efficiency of a simple burst filter

■ The second application is a macro that creates the spectrogram of a given time series. It is a spectrogram function that creates and plots the histogram of the power spectrum of the signal versus time. The output can be displayed in various formats, like Lego, contours, or 3D. This function slices the time series in smaller intervals and by applies a Hanning⁹ window; and takes the power spectrum of each slice. As the Hanning window decreases the amplitude of the signal at the edge of each slice, in order not to lose any information, it is necessary to overlap the slices. The overlapping I used in this case was $\frac{1}{2}$ (half of a given slice

appears in the next slice too), but it can be changed in order to get the optimum resolution of the spectrogram. The resolution is determined, by both length and overlapping. The longer a slice in time is, the higher is the resolution of the frequency domain, and the lower is the resolution of the time domain. Increasing the overlapping can increase this, but it will be artificial, because no extra information is added to the spectrogram. In Figure 6, we show the spectrogram of a composite signal for which I used a $\frac{1}{2}$ overlapping.

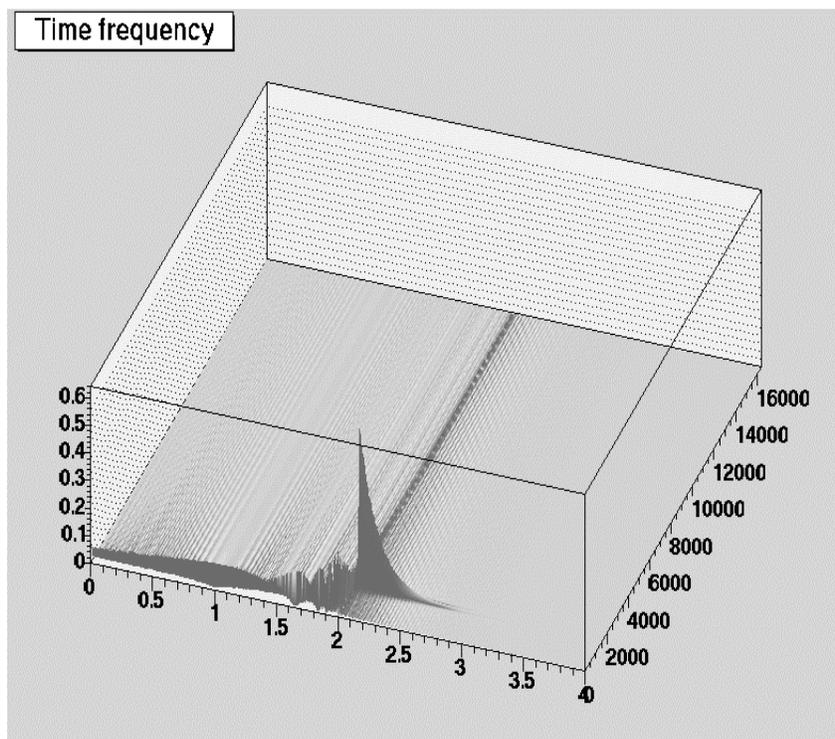


Fig. 6 A 3D time frequency spectrogram

As you can see in Fig. 6, this is a composite signal that consists of a chirp, a burst and a ringdown. It is supposed to be the kind of a signal produced by

the collapse of two compact objects. Lacking a good simulation of the inspiral collapsing burst, the one I used for this is a [ZM](#) (Zwerger-Muller)¹⁰ waveform, which is a simulation of an asymmetric supernova collapse. All the time series of the signal I used for the previous spectrogram can be seen in the following figure:

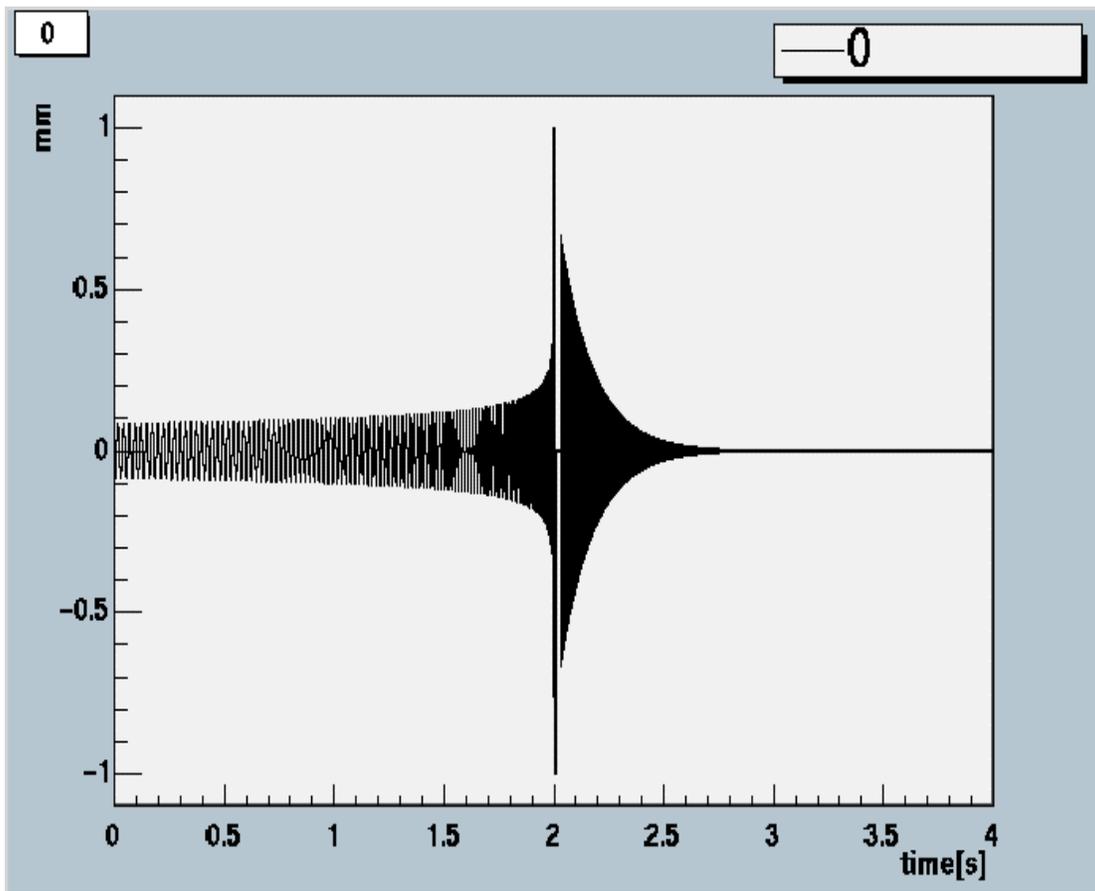


Fig. 7 The signal used for the previous spectrogram (chirp, a short burst and a ringdown).

3. Coincidence Analysis:

During the second part of the summer I analyzed the coincidence statistics of two or more interferometers (especially two, more work is required to understand triple and higher coincidences). It started with the study of the mathematics (statistics) of the system to find a clue of what to expect from it. Then I generated fake events and simulated the behavior of the interferometer. I built a class that stores all the important features of an interferometer that can be recorded in a database and then I built a series of functions that analyze events coming from multiple interferometers.

a) Rates for random coincidences:

Let's consider two different events that occur with the rates r_1 and r_2 .

The probability of event 1 to occur in the elementary time interval dt is $dp_1 = r_1 * dt$, and the probability of event 2 to occur in the same interval of time is $dp_2 = r_2 * dt$.

Let P_1 and P_2 be the probabilities that one of the events occur in the time interval τ , then disregarding which event to consider:

$dP(\tau) = (1-P(\tau)) * r * dt$ is the probability that the event didn't occur in the time interval τ times the probability that it occurred in a subsequent time dt .

Now integrating over time,

$$\int_0^P \frac{1}{1-P} dP = \int_0^t r dt, \text{ we get:}$$

$$1 - P = e^{-rt}$$

$$P = 1 - \tilde{a}^{-rt} \quad \text{and} \quad P_1 = 1 - \tilde{a}^{-r_1 t} \quad \text{and} \quad P_2 = 1 - \tilde{a}^{-r_2 t}$$

The probability of a coincidence in the time interval dt is:

$$dP = dP_1 * P_2 + dP_2 * P_1$$

$$dP = r_2 * H(1 - \tilde{a}^{-r_1 t}) dt + r_1 * H(1 - \tilde{a}^{-r_2 t}) dt, \quad \text{and if}$$

we consider the fact that we can count a coincidence both when one event occurred after the other or, before, we get:

$$dP = 2 * r_2 * H(1 - \tilde{a}^{-r_1 t}) dt + 2 * r_1 * H(1 - \tilde{a}^{-r_2 t}) dt$$

So the coincidence rate is:

$$r = 2 * r_1 * H(1 - \tilde{a}^{-r_2 t}) + 2 * r_2 * H(1 - \tilde{a}^{-r_1 t})$$

In the limit of $\tau \ll r_1^{-1}$, and $\tau \ll r_2^{-1}$, the formula reduces to $r \sim 4r_1 r_2 \tau$; in this limit it, agrees with the M-fold coincidence¹¹ formula:

$$r = M H 2 t L^{M-1} \prod_{i=1}^M r_i,$$

which for $M=2$ yields $r = 4r_1 r_2 \tau$.

b) Some numeric results:

For all the numeric results below, the time window used is 10 ms (the Hanford-Livingston window, given by the light travel time between the two sites).

■ *If we were willing to tolerate a fake coincidence rate less than 1/15 year, then the fake rate would be 0.83/h*

■ *If we were willing to tolerate a fake coincidence rate less than 1/year, the fake rate will be 3.24/h*

■ *If we were willing to tolerate a fake coincidence rate less than 1/hour, the fake rate will be 300 h⁻¹*

The graphic below shows the dependence of the coincidence rate versus fake rate, considering the two fake rates to be equal ($r_1 = r_2 = r$)

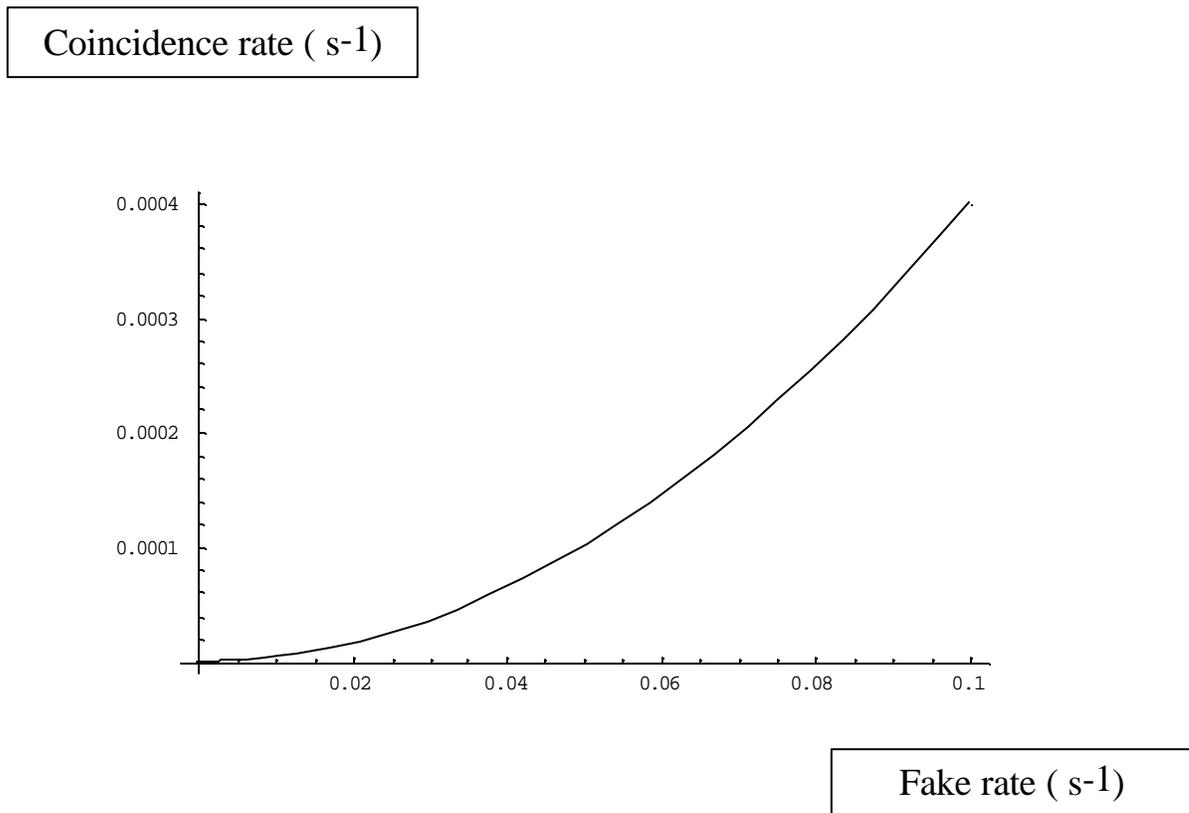


Fig 8. The coincidence rate vs. Fake rate dependence for a 10 ms window

c) **The interferometer class:**

It is a C++ class, named `ifo` that stores information about an interferometer. Its members are: name, position, start time, length, lock time periods, events (bursts), signal to noise ratio and confidence. These are columns in the single data burst `.ilwd` file, used for storing interferometer data in the LIGO [database](#)¹²:

- *Name is a field that stores the name of the interferometer, for example “Hanford 2k”*
- *The position consists of 2 members, one of them named latitude that stores the latitude of the interferometer and the other one that stores the longitude. Both are real (double) and expressed in degrees.*
- *The start time is the GPS starting time in seconds. It is named `st`.*
- *The length is the duration in seconds of the observation.*
- *The lock time period is an array of 0 and 1 (short) that specifies if the interferometer was in or out of lock for that second.*
- *Event is an array of times measured from the starting time at which an event occurs (real or fake).*
- *Signal to noise ratio (`snr`) is an array that characterizes each event and contains its signal to noise ratio.*
- *Confidence is also an array that characterizes each event, but in my case contains the information about its real nature (0 if it is fake and 1 if it is a burst).*

In Figure 9 there is a graphic representation of in lock segments and fake

and real burst events from 4 interferometers:

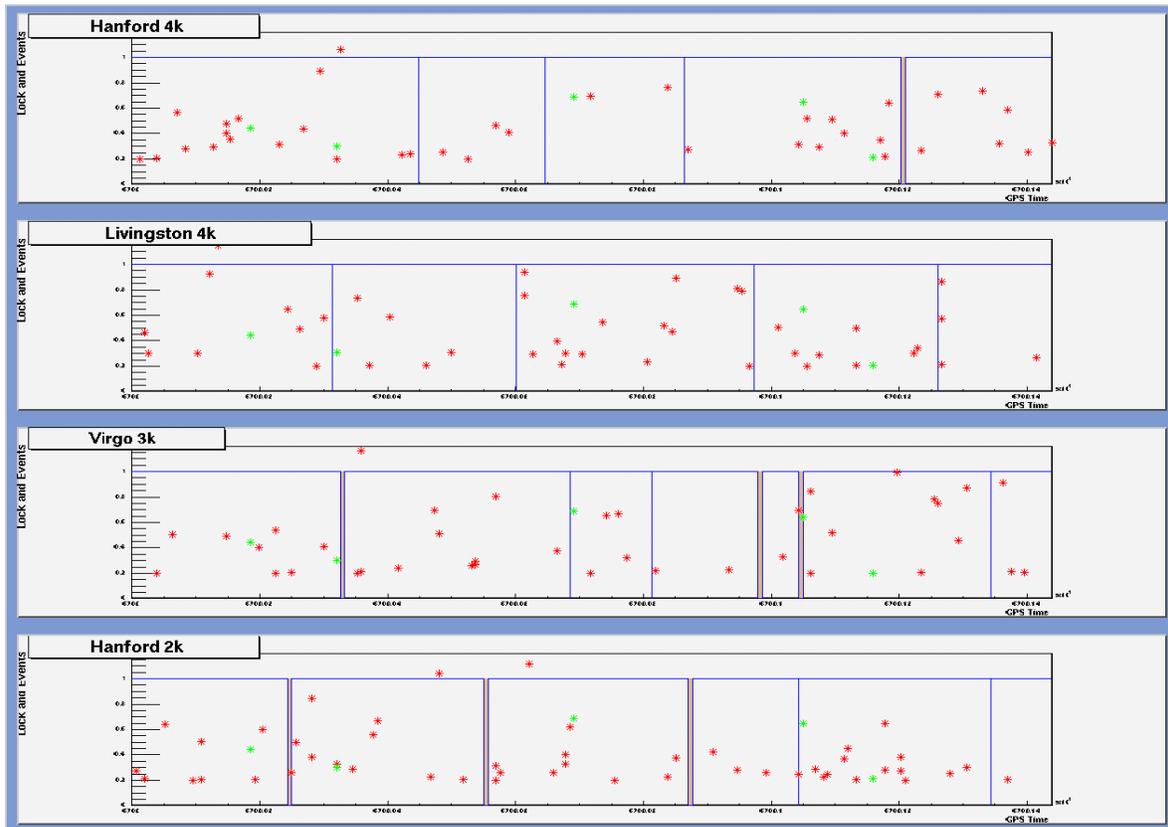


Fig. 9 The behavior of interferometers along 4 hours

In Figure 9, the blue line indicates if the interferometer was in lock or not, the red stars are fake events, the green stars are real bursts and the height at which they are situated represents their signal to noise ratio. While generating the real events I chose a random direction on the sky at a given time when the burst reaches the center of the earth, and then I calculate the moment of time at which it reaches the interferometer knowing its location on earth ignoring the polarization of the gravitational wave and the orientation of the interferometer, as shown in Figure 10.

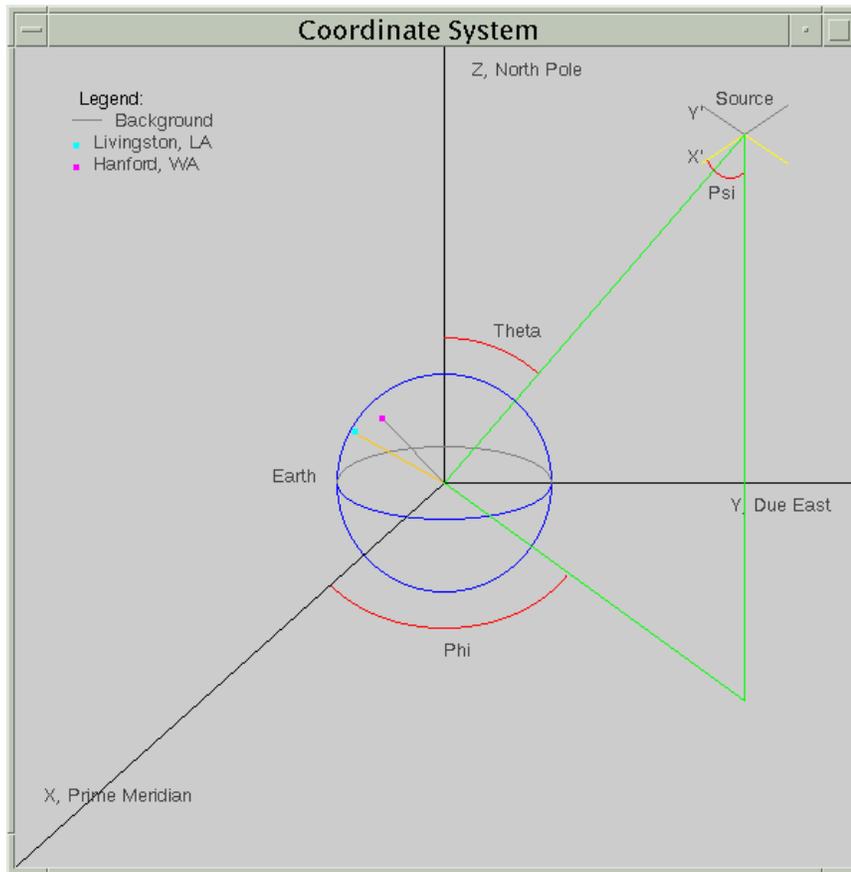


Fig.10: Location on earth's surface of the interferometers and an incident gravitational wave.¹³

d) Coincidence statistics:

For this I choose two interferometers (2 structures of type ifo), calculate the distance and the time window between them, and delay one interferometer with respect to the other with a given time. Then I count the coincidences in the time window. I do this for various delay times and then plot the number of coincidences versus the delay time. If there is a peak centered on 0 then it means that there were events in coincidence between the two

interferometers. In the following graphics, the time delay stays all the time close to zero, so that we can see on the graphic how the width of the peak increases when the distance between interferometers decreases. This may cause the same coincidence to be counted twice, which does not affect the height of the peaks. Anyway, the time delay graphics are usually made, using a wide time delay and the central peak appears just as a line. Also the real event rates are much higher, than LIGO expects. The rates used are just an artifact to show better the way this method works.

- ***Time delay statistics Hanford-Livingston:***
Time = 2h; dist = 3005.7 km; surface dist=3034 km; time window=10.02ms; fake rate = 50 h⁻¹; event rate = 20 h⁻¹

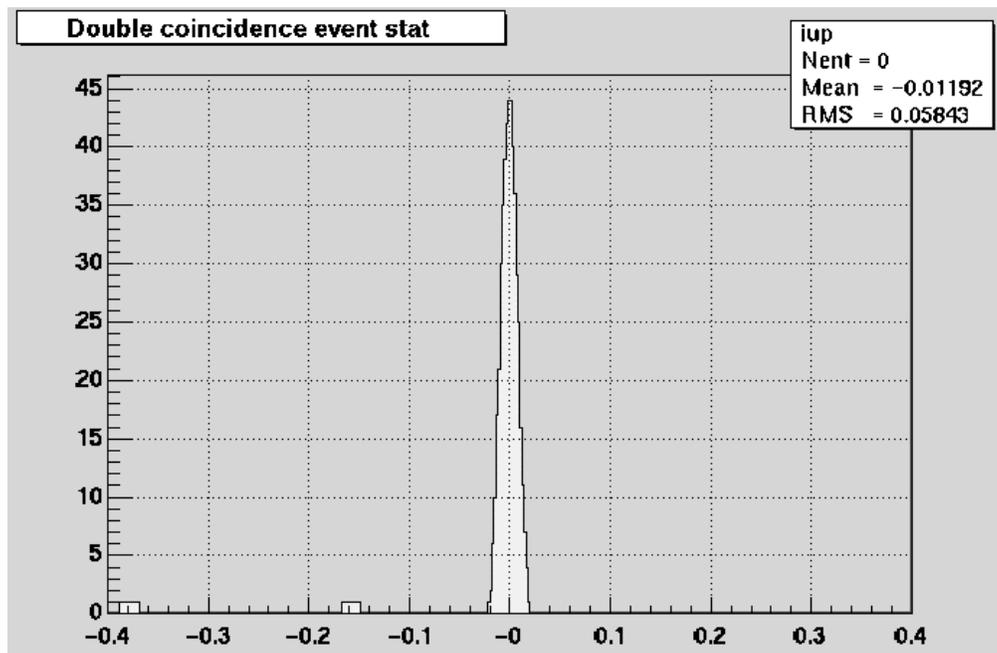


Fig.11 Time delay for Hanford-Livingston

- Time delay statistics Hanford-Virgo:**
 Time = 2h; dist = 8169.88 km; surface dist=8867 km; time window=27.23ms; fake rate = 50 h⁻¹; event rate = 20 h⁻¹

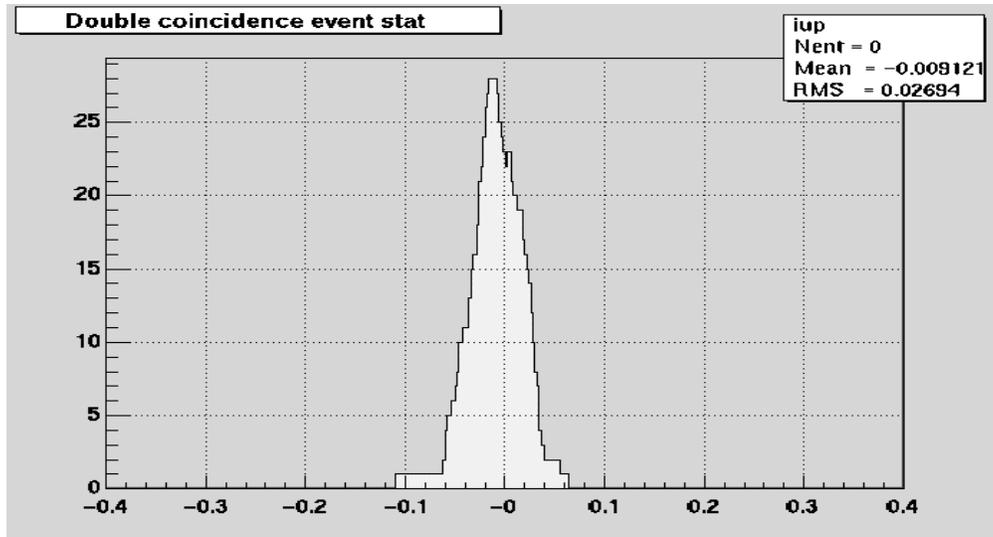


Fig.12 Time delay graph Hanford-Virgo

- Time delay statistics Hanford-Livingston:**
 Time = 2h; dist = 3005.7 km; surface dist=3034 km; time window=10.02ms; fake rate = 1000 h⁻¹; event rate = 20 h⁻¹

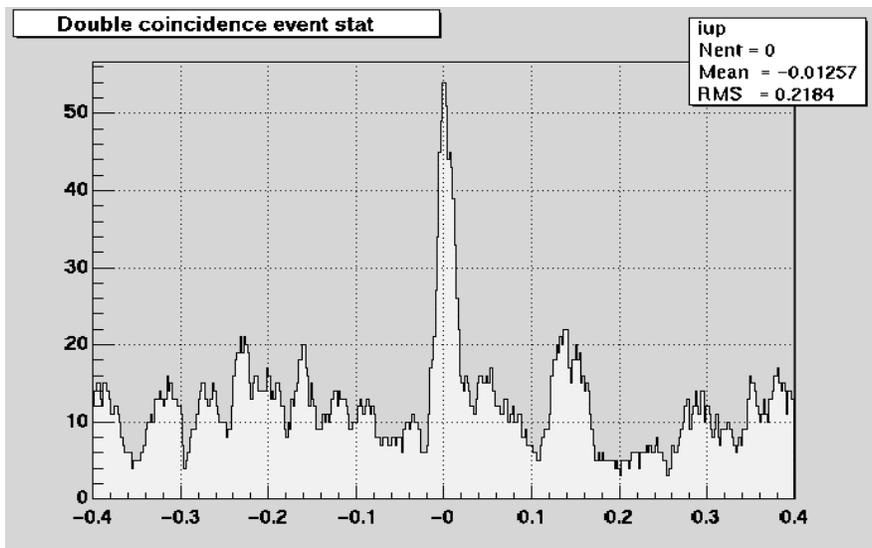


Fig.13 Time delay graph Hanford-Livingston

- *Time delay statistics Hanford-Livingston:*
Time = 2h; dist = 8169.88km; surface dist=8867 km; time window=27.23ms; fake rate = 1000 h⁻¹; event rate = 20 h⁻¹

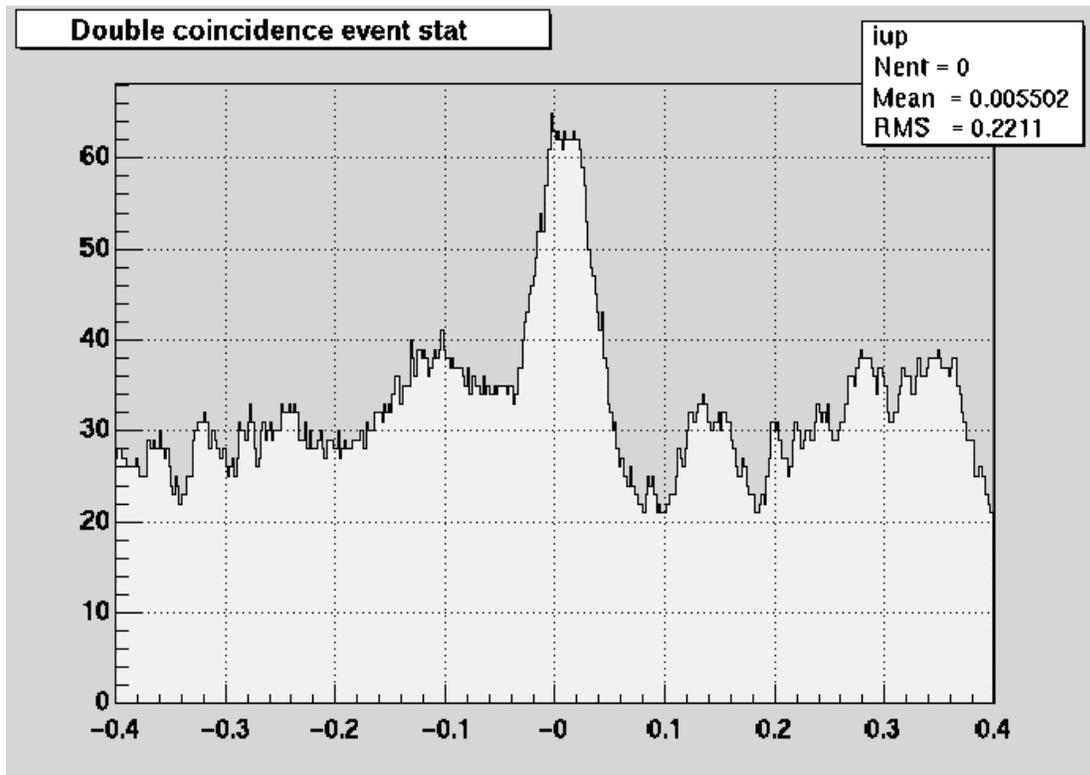


Fig.14 Time delay graph Hanford-Virgo

e) File writing and reading:

I wrote all the information in the ifo class in an [.ilwd](#)¹⁴ single interferometer burst file. This is one of the formats a file should have in order to be inserted in a database. All these files have to be inserted in a database.

For the writing and reading purposes I used two functions wilwd (for writing) and rilwd (for reading), which are members of the ifo class. The reading part was much harder than the reading one. In order to get convinced

about that here is an example of an .ilwd file I wrote:

```
<?ilwd?>
<ilwd name='ligo:ldas:file'>
  <ilwd name='sngl_burstgroup:sngl_burst:table' size='11'>
    <ilwd name='sngl_burstgroup:sngl_burst:process_id' size='10'>
      <char_u      comment='process_id:20010426221849333706000000'      dims='13'
name='sngl_burstgroup:sngl_burst:process_id'>\040\001\004\046\042\030\111\063\0
.....
.....
    </ilwd>
      <int_4s dims='10' name='sngl_burstgroup:sngl_burst:start_time'>672359334 672359335 672359336 672359337
672359338 672359339 672359340 672359341 672359342 672359343</int_4s>
      <int_4s  dims='10'  name='sngl_burstgroup:sngl_burst:start_time_ns'>316620000 316620001 316620002
316620003 316620004 316620005 316620006 316620007 316620008 316620009</int_4s>
      <real_4 dims='10' name='sngl_burstgroup:sngl_burst:duration'>9.8832410e+08 9.8832410e+08 9.8832410e+08
9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08\
9.8832410e+08</real_4>
      <real_4  dims='10'  name='sngl_burstgroup:sngl_burst:central_freq'>9.8832410e+08 9.8832410e+08
9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410\
e+08 9.8832410e+08</real_4>
      <real_4 dims='10' name='sngl_burstgroup:sngl_burst:bandwidth'>9.8832410e+08 9.8832410e+08 9.8832410e+08
9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+0\
8 9.8832410e+08</real_4>
      <real_4 dims='10' name='sngl_burstgroup:sngl_burst:amplitude'>9.8832410e+08 9.8832410e+08 9.8832410e+08
9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+0\
8 9.8832410e+08</real_4>
      <real_4  dims='10'  name='sngl_burstgroup:sngl_burst:snr'>9.8832410e+08 9.8832410e+08 9.8832410e+08
9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.88\
32410e+08</real_4>
      <real_4 dims='10' name='sngl_burstgroup:sngl_burst:confidence'>9.8832410e+08 9.8832410e+08 9.8832410e+08
9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+08 9.8832410e+\
08 9.8832410e+08</real_4>
    </ilwd>
  </ilwd>
```

An example of an .ilwd file

4. Things to be done in the future and conclusion

- Writing documentation for every macro I made.
- Working on improving the macros (the ones in the first part have room for a lot of improvement).
- Functional database input and output.
- Multiple coincidences.

- More realistic rates.

In conclusion during the summer I accomplished tasks related to signal processing and coincidence analysis. I built a simple burst filter and a spectrogram function, I simulated the output of the Gravitational wave channel of a real interferometer, made time delay plots for coincidences and wrote files in the .ilwd format. Learning how to use some tools, like Root and manipulate data was an important accomplishment.

For many of these I have to thank:

- ✚ Alan Weinstein, my mentor, for guiding me throughout the project
- ✚ Richard George, for C++ help
- ✚ Kathy Cooksey, for fig. 10
- ✚ Guillaume Michel, for presenting me his results in the field
- ✚ Benoit Mours, for introducing me to the Frame Library

¹ <http://www.ligo.caltech.edu/~ajw/bursts/bursts.html>

² www.caltech.edu

³ www.ldas-sw.ligo.caltech.edu

⁴ <http://root.cern.ch>

⁵ <http://fermat.utb.edu/cgi-bin/cvsweb.cgi/lal-devel/>

⁶ <http://fermat.utb.edu/cgi-bin/cvsweb.cgi/lalwrapper-devel/>

⁷ 300 Years of Gravitation; K.S. Thorne; Cambridge University Press 1987

⁸ <http://wwwlapp.in2p3.fr/virgo/FrameL/>

⁹ Numerical Recipes in C, The Art of Scientific Computing

by [William H. Press](#), [Saul A. Teukolsky](#), [William T. Vetterling](#), [Brian P. Flannery](#)

¹⁰ Zenger Muller supernova bursts Astronomy & Astrophysics (A&A, 1997, 320, 209-227)

¹¹ G. A. Prodi et al., International Journal of Modern Physics, Vol. 9, No. 3, June 2000, page. 243

¹² http://www.ldas-test.ligo.caltech.edu/ldas_outgoing/logs/LDASdatabase.html

¹³ With the help of Kathy Cooksey.

¹⁴ <http://ldas-sw.ligo.caltech.edu/ldas/ldas-0.0/doc/db2/doc/html/ilwdformat.html>