LIGO-T020137-00-E

Burst Pipeline in the E7 Playground

S. Ballmer, L. Cadonati, E. Katsavounidis
LIGO-MIT
DRAFT 2.3

July 24, 2002

Abstract

We analyzed a subset of LIGO data from Engineering Run #7 (E7). The subset, referred to as "playground", has been used to define a strategy to be applied on the entire E7 set. This report describes the steps we took and the considerations that guided our choices. Particular emphasis was given to the veto strategy, threshold tuning and sanity checks.

1 Burst data analysis pipeline

The scientific goal of the bursts working group is to set an upper limit on unmodeled sources of gravitational radiation. This is implemented in the data analysis pipeline shown in Figure 1.

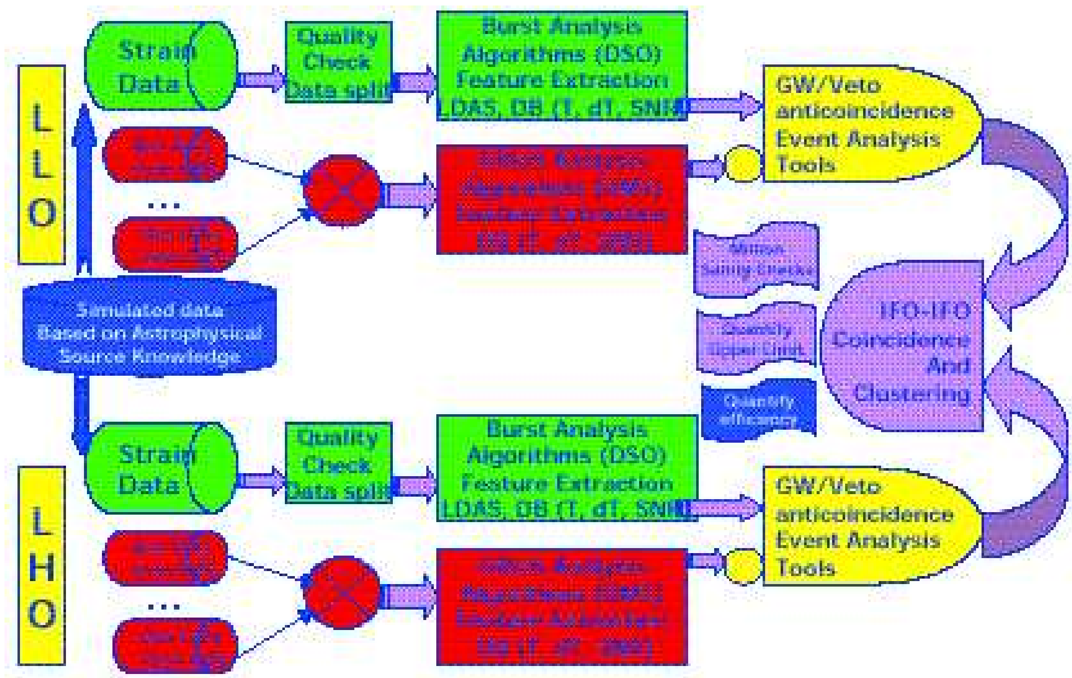


Figure 1: *The burst data analysis pipeline*

The first step consists of running search codes on the gravitational channel (AS_Q) time series. These codes, usually referred to as DSOs (Dynamic Shared Objects), look for Gravitational Wave (GW) signatures in a number of different approaches:

1. **slope** (authors: Arnaud et al, Ed Daw) looks for changes in the signal slope in the time domain only,

2. **tfcluster** (author: Julien Sylvestre) looks for excess of power in clusters of pixels in the time-frequency domain, and

3. **power statistics** (authors: Patrick Brady et al.) searches for excess of power above noise in a time-frequency band.

The DSOs are run within LIGO's Data Analysis System (LDAS). They extract a number of features for the candidate GW events and store them onto LIGO's event and analysis databases (DB). The database information for the candidate events is then retrieved and analyzed in a post processing phase [1].

The LIGO detectors not only record gravitational wave signals, but also continuous time series from a number of interferometric and environmental channels. These channels, in the hypothesis that they have null or very weak coupling to a GW, can be used to flag data that were affected by noise (environmental and/or internal to the instrument). These instances thus yield vetoes for any GW candidate the AS_Q channel might have

[1] In the pipeline work here described, LIGO's DBs were used in a very limited way. This work involved extensive exploratory directions for which it was deemed unnecessary to invoke the DB function before a close to final pipeline version was reached.

simultaneously recorded. Another set of search codes (generally referred to as DMTs, or Data Monitoring Tools) are executed on the auxiliary channels, looking for glitches in the time series.

Both DSO and DMT codes generate triggers, logic states signaling that the time series being analyzed has fulfilled some well defined selection criteria. Depending on the nature of the search algorithm, features of the triggers are extracted and stored in LIGO's databases. As a general rule of nomenclature, triggers coming from the DSOs processing of the Gravitational Wave channel will also be referred to as "candidate" events, while the DMT-based analysis of auxiliary channels yields possible veto triggers or simply "vetoes".

Our activity has mostly been in the post-process analysis of triggers (both candidate events and vetoes). The studies presented in this report have been performed using the **EventTool** (D. Siggs and M. Ito), a set of C++ libraries running in the framework of **ROOT**, the data analysis package developed at CERN for particle physics. The code is designed to work with large sets of events. It allows to load database entries and perform selection cuts, coincidence analysis and statistics on them.

In the framework of a single-interferometer characterization, we explored the possible use of vetoes to sensibly reduce the event rate at each site. The next section will describe this process in extensive detail. We will also walk through the mechanism of coincidence between two interferometers (the 4Km at Livingston, LLO-L1, and the 2Km at Hanford, LHO-H2) and the determination of upper limits in a Feldman Cousin approach. Finally, we will probe the pipeline efficiency with the aid of injected simulations.

2 Data stretches and recipes for trigger selection

LIGO's E7 Run took place from December 28, 2001 to January 14, 2002, yielding 112 hours of triple coincidence among the Livingston (LLO-4Km or L1) and Hanford (LHO-2Km or H2, LHO-4Km or H1) interferometers. Of these data, roughly 10% was hand-picked to function as a "playground" set over which analysis pipelines were going to be defined and optimized. The investigation here presented is limited to the subset of the E7 playground data labeled L1H1H2 (i.e., triple interferometer coincidences) [2].

The start time and nominal duration of the lock stretches are listed in table 1. The last column in the table reports the time actually used in the analysis: due to a *de facto* implementation of the burst search DSOs, only integer multiples of 360 seconds have been analyzed.

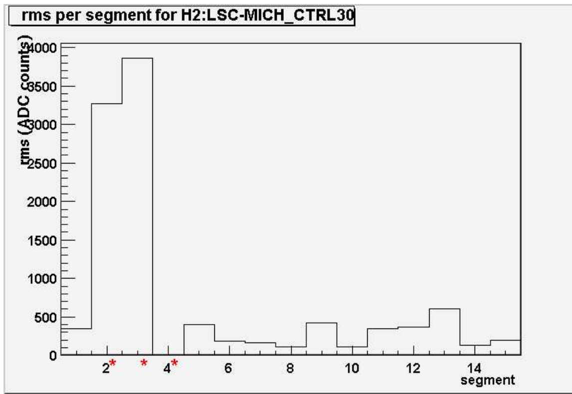
The set contains 15 stretches, but only 12 of those have been included in our study. Segment 4 (278 sec) was too short to satisfy the 360 sec requirement. Segments 2 and 3 have been excluded from the analysis because of their H2 behavior: figure 2.a shows how the rms value on the H2:LSC-MICH_CTRL channel [3] was abnormally large in those two stretches. The slope DSO crashed because of the high event rate. A large resonance at about 350 Hz was also observed in the power spectrum (see figure 3). [4]

The three burst DSOs were run over the total exposure of 3.2 hours. This was mostly the work of Alan Weinstein who, following prescriptions by the DSO authors, has run the search codes on the MIT-LDAS cluster.

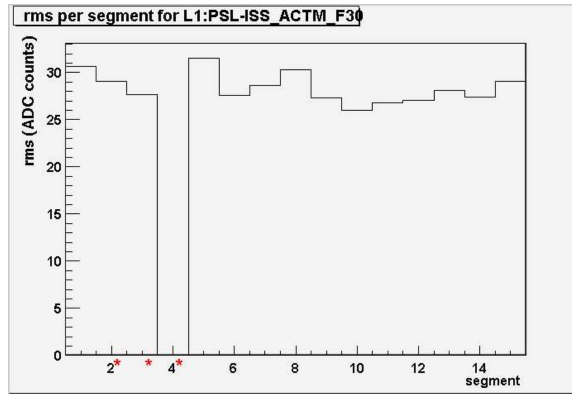
[2] The E7 bursts analysis will make no use of the LHO-4Km data due to the instrument's much poorer sensitivity respect to the LHO-2Km and LLO-4Km during E7.

[3] H2:LSC-MICH_CTRL is the most effective veto channel we found for H2:LSC-AS_Q; we will describe in the next section how we came to this conclusion.

[4] We should consider using the rms monitor as a veto, as the inspiral group does; if this behavior is repeated throughout E7 we might have to deal with these abnormal stretches in a more systematic way!



(a)



(b)

Figure 2: *RMS value per segment, calculated on the veto auxiliary channels described in section 5, at the two sites. Note that at H2 the second and third segments are particularly loud, while elsewhere the rms is pretty much stable throughout the playground. Those two segments have not been used in the analysis. The gap at the 4th segment is due to its being too short to be analyzed by the DSOs.*

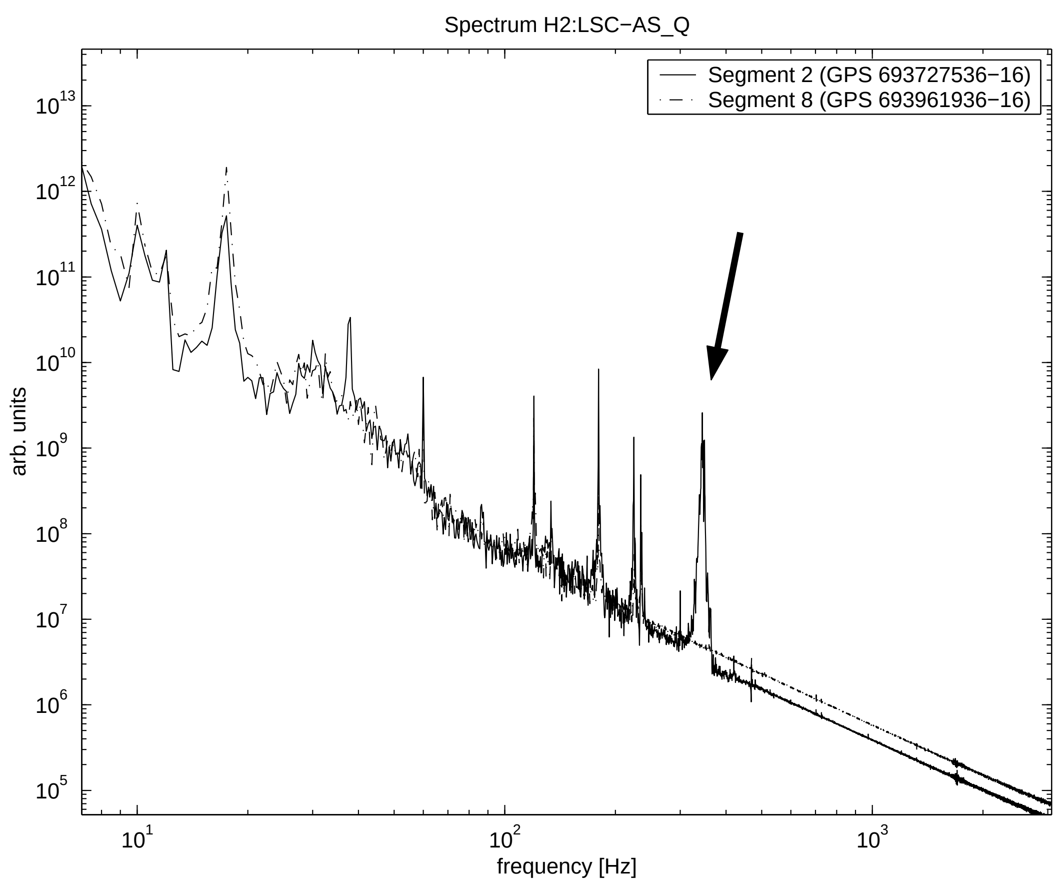


Figure 3: *Comparison between the power spectra in stretch 2 (large RMS) and stretch 8 (smaller RMS) at the H2 interferometer: note the broad resonance at about 350 Hz.*

Table 1: *Lock segments in the L1H1H2 portion of the E7 playground data set.*

Segment	GPS start time	Duration (sec)	Used Duration (sec)
1	693641112	934	720
2	693727208	1010	0
3	693768213	1139	0
4	693793929	278	0
5	693794236	873	720
6	693836549	1150	1080
7	693879478	1067	720
8	693961597	945	720
9	694067698	1143	1080
10	694086870	1222	1080
11	694250412	1226	1080
12	694254869	1225	1080
13	694420642	1123	1080
14	694613993	1197	1080
15	694843214	1081	1080
TOTAL			11520 sec

In output, we have events, that is triggers (time intervals) with given sets of variables associated to them, according to a standard SINGL_BURST database table. Since each DSO gives different meanings to the standard entries of the database, it is worth pointing out what we use in the analysis, as power of the event:

1. **slope**: power = AMPLITUDE × AMPLITUDE
2. **tfcluster**: power = SNR × AMPLITUDE − which means, for a given cluster in the time-frequency plane, the product of the average power per pixel times the number of pixels constituting the cluster.
3. **power statistics** power = SNR.

Note that while for slope and tfcluster we are looking at a total power, inclusive of the underlying background, the power statistics effectively provides a signal to noise ratio. It will be important to keep this in mind when interpreting the results of the analysis.

We introduce, at this point, two cuts on the DSO triggers:

1. **time from lock start $>$ 70 sec**: we noticed that in a few instances, particularly at H2, the first minute after the start of locked acquisition is very noisy (see, for instance figure 4). This likely corresponds to a ringdown of the violin modes and harmonics. Thus, we decided to always exclude the first 70 seconds of each stretch.
2. **frequency cut**. For the tfcluster DSO, we focus on the 300-3000 Hz frequency range: all the time-frequency clusters that do not overlap this interval are left out of the analysis. The slope DSO only works in the time domain, so the frequency cut has no real meaning and/or effect. The power statistics allows a search to be performed over a user-defined range of frequencies. The power DSO has been run on the 300-1024Hz band, without any special treatment of the resonant frequencies (violin modes etc) that are present in the interferometer's spectrum. Some level of tuning at the level of analysis output is to be expected.

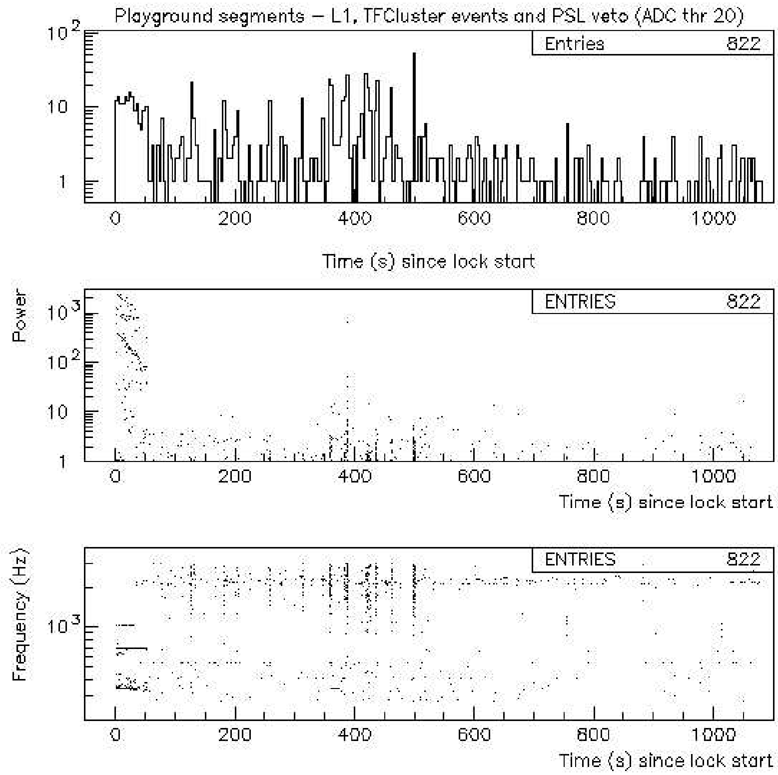


Figure 4: *distribution of time from the stretch start for L1, after application of a PSL veto (see next section). Note the large number of events in the first minute.*

3 The absGlitch glitch finder

A number of auxiliary interferometer channels were also analyzed by more than one DMT glitch-finding algorithms. There were three fundamental categories of glitch search codes that were run on the E7 data:

- time domain, fixed threshold: a glitch is recorded in the time domain (after possibly some filtering) when the signal crosses a fixed threshold. This threshold can be tuned but does not depend on the noise level.
- time domain, relative threshold: the signal (in the time domain) needs to be a certain number of RMS's above the Gaussian noise level. The noise average and variance is calculated on a stretch by stretch basis and the glitch-finding threshold is accordingly adjusted.
- time-frequency and template matching algorithms: glitches of specific frequency content and/or of known overall waveform signature may be searched by invoking time-frequency analyses, like the ones performed by the Bursts DSOs, or the full matched filter techniques used by the Inspiral DSOs to identify binary inspiral signatures.

All the work presented in this report has been performed with a time domain, fixed threshold glitch search code, referred to by its authors (Rauha Rahkola and Masahiro Ito) as **absGlitch**. We have performed exploratory work on the other two ways of finding glitches, without going over the entire series of investigations for their functioning. We will defer for the future their implementation in the bursts pipeline.

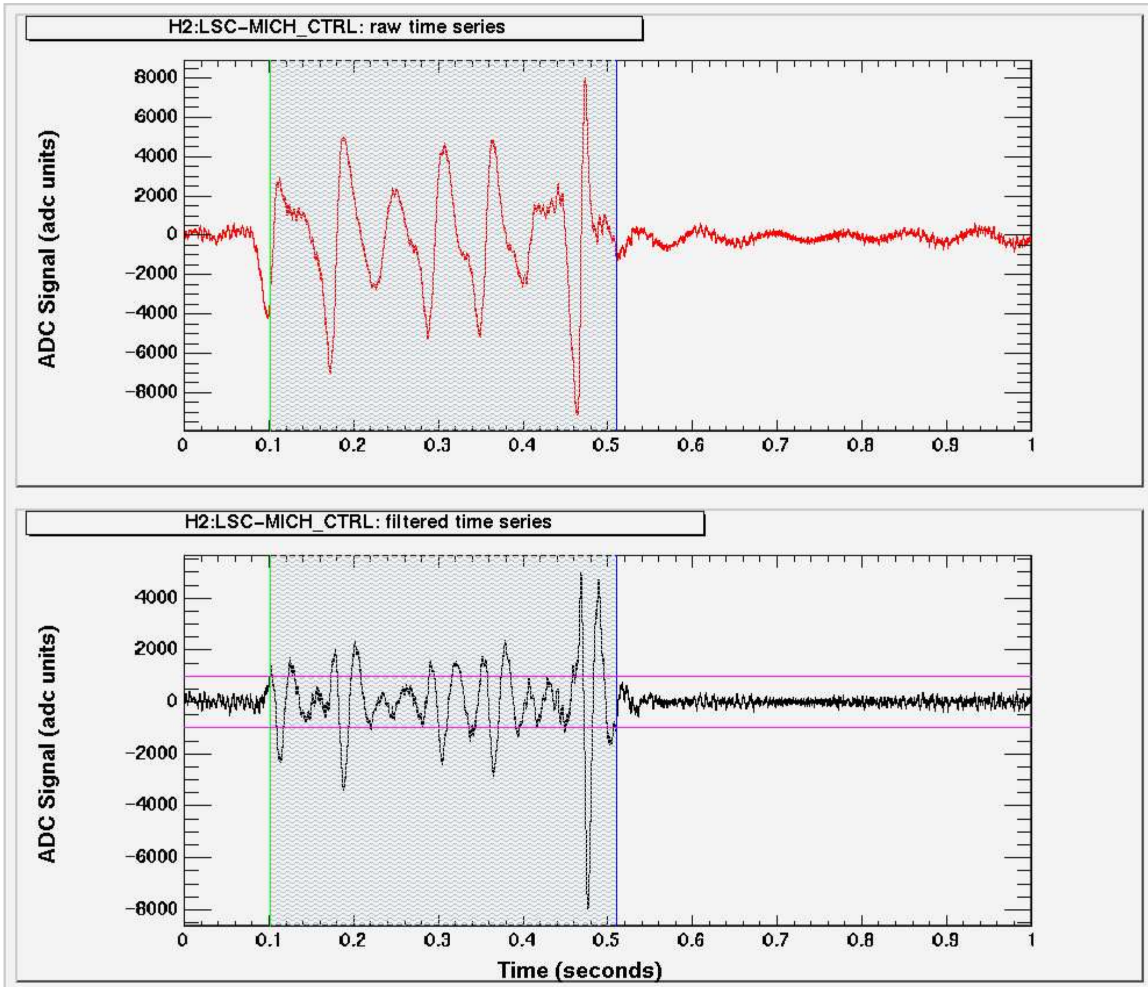


Figure 5: *The figure shows the raw H2:LSC-MICH_CTRL time series in the upper half and the 30Hz high-pass filtered one in the lower half. The used threshold is shown in magenta; the trigger is shaded with its start in green and its end in blue.*

The absGlitch code scans a given channel in the time-domain and filters the raw data with a customizable IIR filter. As the filtered time series crosses a preselected, absolute threshold, the start time of the trigger is recorded. When the time series falls below the threshold again, absGlitch starts a timer. If the time series recrosses the threshold within a predefined time, the trigger is extended, otherwise the last threshold crossing is the end time of the trigger. This behavior is illustrated in figure 5.

4 The veto strategy

Our goal is to verify whether any of the auxiliary channels can be used as a sensible veto for AS_Q. We will start from a definition of the nomenclature and a general description of the fundamental strategy. Comments on the efficiency of vetoes at the two sites will follow in section 5.

We introduce here the variables that will be used in the veto characterization:

- N = number of event triggers detected by the DSO;
- N_V = number of event triggers that are vetoed;
- $\varepsilon_V = \frac{N_V}{N}$ veto efficiency;
- T = total measurement time (real time);
- $T_D = \sum t_j$ total dead time, sum of the duration of all the veto triggers;
- $T_L = T - T_D$ live time;
- $\tau_D = \frac{T_D}{T}$ dead time fraction;
- $\tau_L = \frac{T_L}{T}$ live time fraction;
- $\zeta = \frac{(1-\varepsilon_V)N}{T_L}$ residual rate: this will represent the background for GW detection, once the veto is applied.
- η = ratio of the veto efficiency and the accidental background in the veto process, which can also be described as peak-to-total ratio in a lag (or time shift) plot (see for instance figure 8-a).
- the veto condition is satisfied if there is any overlap between the candidate trigger and the veto trigger:

$$[Start, Start + Duration)_{gw} \bigcap [Start, Start + Duration)_{veto} \neq 0$$

A general statement can be made about a good veto as the one offering a large efficiency ε_V and a small dead-time fraction τ_D. For this reason, we will use $\varepsilon_V - \tau_D$ plots from different possible veto channels, in order to choose the one that best performs. Once the veto channel is chosen, we need to tune the threshold on veto trigger identification, through the optimization of the following two figures of merit:

1. the residual rate ζ: we want to minimize the background for gravitational wave candidates;
2. η: the ratio of efficiency to accidentals should be maximized.

We should add, though, that the optimization of the final product of the pipeline, the upper limit for gravitational events, will be the ultimate tool on the selection of a veto strategy.

These being our guiding principles, we will have to come to terms with the actual quality of the data and the fact that we are dealing, at the end of the pipeline, with the statistics of small numbers. Minimization and maximization will have to be applied very carefully.

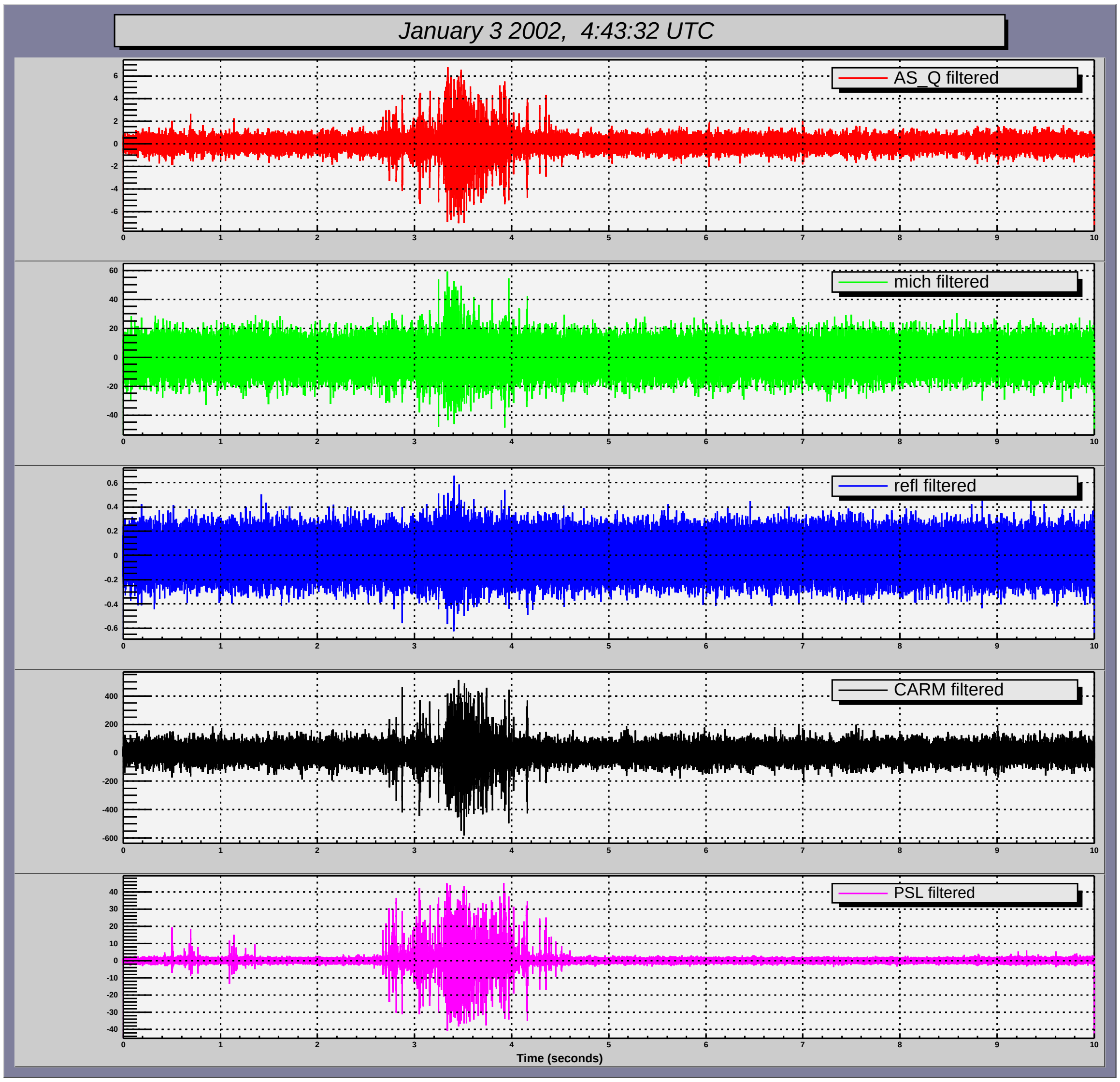


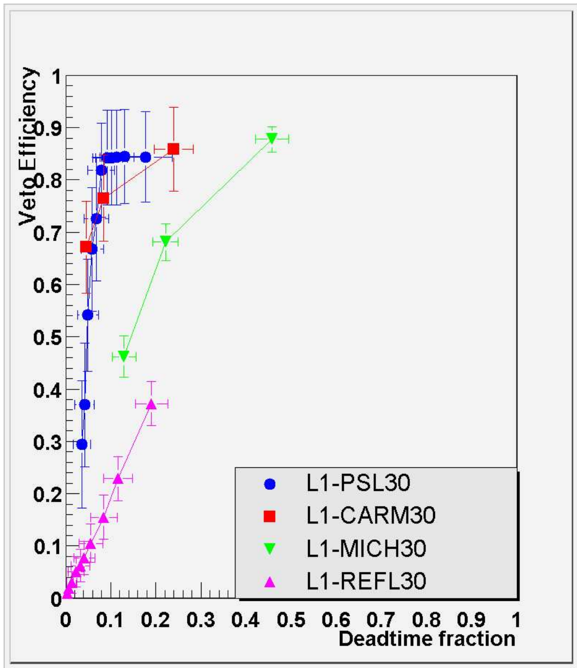
Figure 6: *A typical E7 glitch at L1. At the top is the AS_Q time series, filtered with a 300 Hz high-pass filter. Below are MICH_CTRL, REFL_Q, CARM_CTRL and the PSL channel, all filtered at 30Hz.*

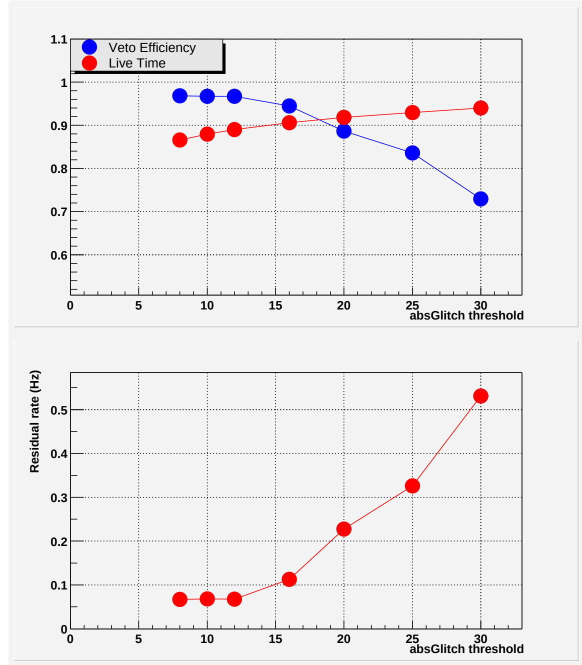
5 Single IFO characterization

5.1 L1

During E7 we observed frequent glitches in the GW channel in coincidence with glitches on the PSL channel. An example of this coincidence is shown in figure 6. Given the nature of the auxiliary channel, we can make the statement that these glitches are all instrumental. The PSL-related glitches on AS_Q also tend to show on other channels that are more or less coupled to the differential signal: MICH_CTRL, CARM_CTRL, REFL_Q. We explored these channel as possible vetoes, as well.

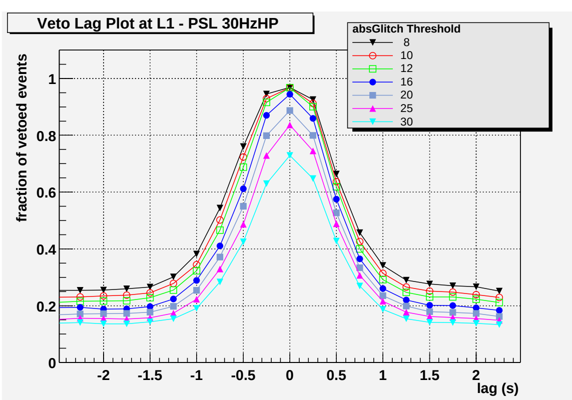
An efficient veto will throw away most such glitches at a small cost for the lifetime. For this reason, we used $\varepsilon_V - T_D$ plots such as the one in figure 7-a to compare the performance of different veto channels as vetoes. We ran absGlitch on each channel with various thresholds and obtained the curves shown in the plot (each line is parametrized

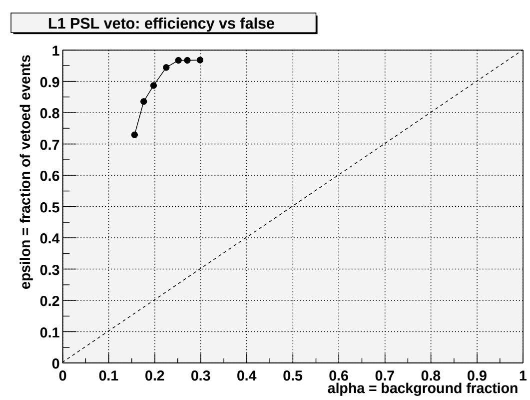




(a) **(b)**

Figure 7: *Search for a veto at L1: (a) efficiency vs dead-time for different thresholds and channels; (b) efficiency, live time and residual rate as functions of the PSL threshold at L1.*





(a)
Veto lag plot

(b)
Peak vs tail

Figure 8: *(a) Veto lag plot for the PSL veto at L1, obtained with different thresholds and a time shift (lag) between PSL and AS_Q time series (±2.25 sec). (b) 2D plot of the veto efficiency (peak at lag=0 in (a)) vs the background (tail of plot in (a)), calculated as average of events picked up with lag=±2 sec and ±2.25 sec.*

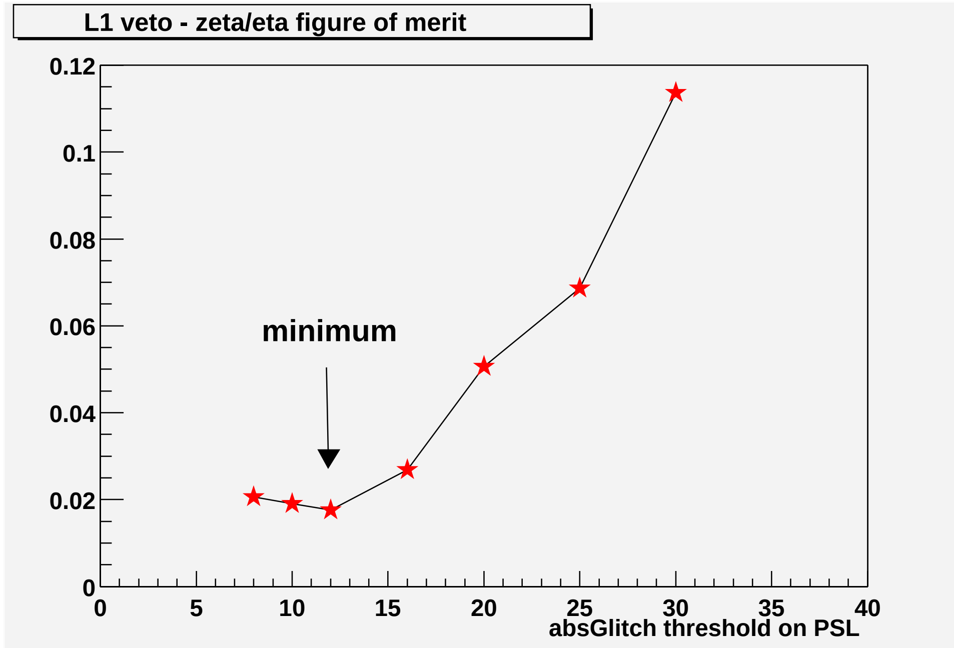


Figure 9: $\rho = \zeta/\eta$ *ratio for the PSL veto at L1, function of the threshold on the PSL channel. the ratio has a minimum at threshold=12.*

with the absGlitch threshold). The ideal veto is somewhere in the upper-left corner of this kind of plot. The PSL is the channel that clearly offers the best performance, among the channels we tested.

Next is a decision on the threshold to be used on the PSL. The plots in figure 7-b offer a tool to make this decision. In particular, from the plot of the residual rate ζ we can come to the conclusion that the best threshold is between 8 and 12, the threshold values for which we get maximum veto efficiency and a stable live-time rate.

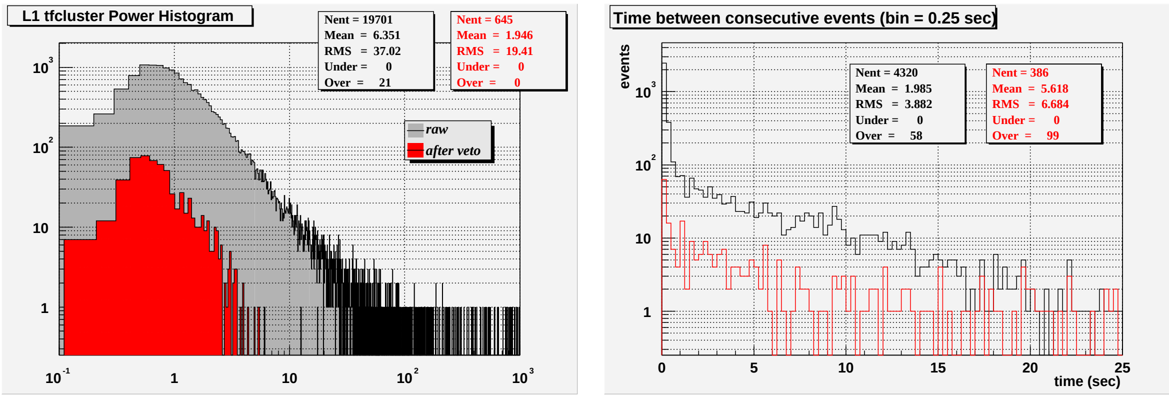
For the second figure of merit we want to maximize the ratio of peak to tail in a lag plot such as the one in figure 8-a: we shifted the time series on the auxiliary channel and recorded the number of candidate events in coincidence with the veto triggers, for different absGlitch thresholds. We estimated the background, or accidental coincidence rate, α by averaging the fraction of vetoed events at lag ± 2 sec and ± 2.25 sec. The veto efficiency ε_V is obtained at lag=0. The relation between the two, as a function of the absGlitch threshold, is shown in figure 8-b. The figure shows the veto is qualitatively good, but the ratio $\eta = \varepsilon_V/\alpha$ is monotonically increasing with the threshold. This, per se, does not help the veto optimization.

However, in a naive approach, since ζ wants to be minimized and η maximized, one can think about minimizing, as an ultimate figure of merit, the ratio:

$$\rho = \frac{\zeta}{\eta} = \frac{1-\varepsilon_V}{\varepsilon_V} \cdot \frac{N\alpha}{T_L} = \frac{\text{residual events}}{\text{vetoed events}} \cdot (\text{rate of accidental vetoes}) \quad (1)$$

The behavior of this quantity as a function of the veto threshold, for PSL at L1, is shown in figure 9. This quantity does indeed present a minimum, in correspondence with the threshold value we would have chosen anyway based on the previous arguments:

$$\text{PSL veto threshold} = 12.$$



(a) (b)

Figure 10: *effect of the veto on the L1 power histogram (a) and the distribution of the delay between events (expected to be exponential for Poisson processes)(b). Data from the Tfcluster DSO.*

5.2 H2

While the choice of a veto for L1 was practically a no-brainer, the situation is much less straightforward at H2. It is true that the H2 event rate (tfcluster output) was much lower than the L1 rate to start with and, in fact, the residual number of events at L1, after the veto, is comparable to that of H2 without any veto. In principle, we could say L1 sets the rules of engagement for the expected rates and the veto at H2 is not be necessary to optimize the final pipeline result.

Nevertheless, from the point of view of a single IFO characterization, we can go through the same tests made for L1. Through signal hand-scanning, we identified four veto candidates: MICH_CTRL, POB_Q, CARM_CTRL and REFL_Q. Note that all of these channels are somehow coupled to the gravitational wave channel, so their use as a veto will have to be performed with a grain of salt (which we will try to quantify in section 5.3). A sample coincidence between AS_Q and the candidate veto channels is shown in figure 11.

A look at the $\varepsilon_V - T_D$ plot in figure 12-a immediately shows that none of these veto options is close to the quality of the L1 veto; the best option is the MICH_CTRL (H2:LSC-MICH_CTRL). We can then ask ourselves which is the threshold that optimized the two figures of merit ζ and η, described in section 9. Figures 12-a and 13 suggest that the only real contribution in vetoing bursts comes from setting high thresholds on the veto (2000 was the largest we tested). As we lower the threshold, there is a linear increase of accidental coincidence rate between AS_Q and MICH_CTRL, as suggested by figure 13-b.

The ratio $\rho = \zeta/\eta$, introduced in eq. 1, also does not provide a firm indication: as the threshold increase, ρ monotonically decreases, up to some asymptotic condition that we have not quantified ($\rho \to 0/0$ as ε_V and α simultaneously drop zero).

For the purpose of a single IFO characterization, we will then use the largest threshold we tested (2000). Figure 15-a shows the effect of such veto on the H2 power spectrum (tfcluster data): the event number is not altered by much, but the high-power tail is completely eliminated.

One can then raise the very legitimate doubt that MICH_CTRL is so closely connected to AS_Q that we are at risk of vetoing all GW events, just because they also affect the

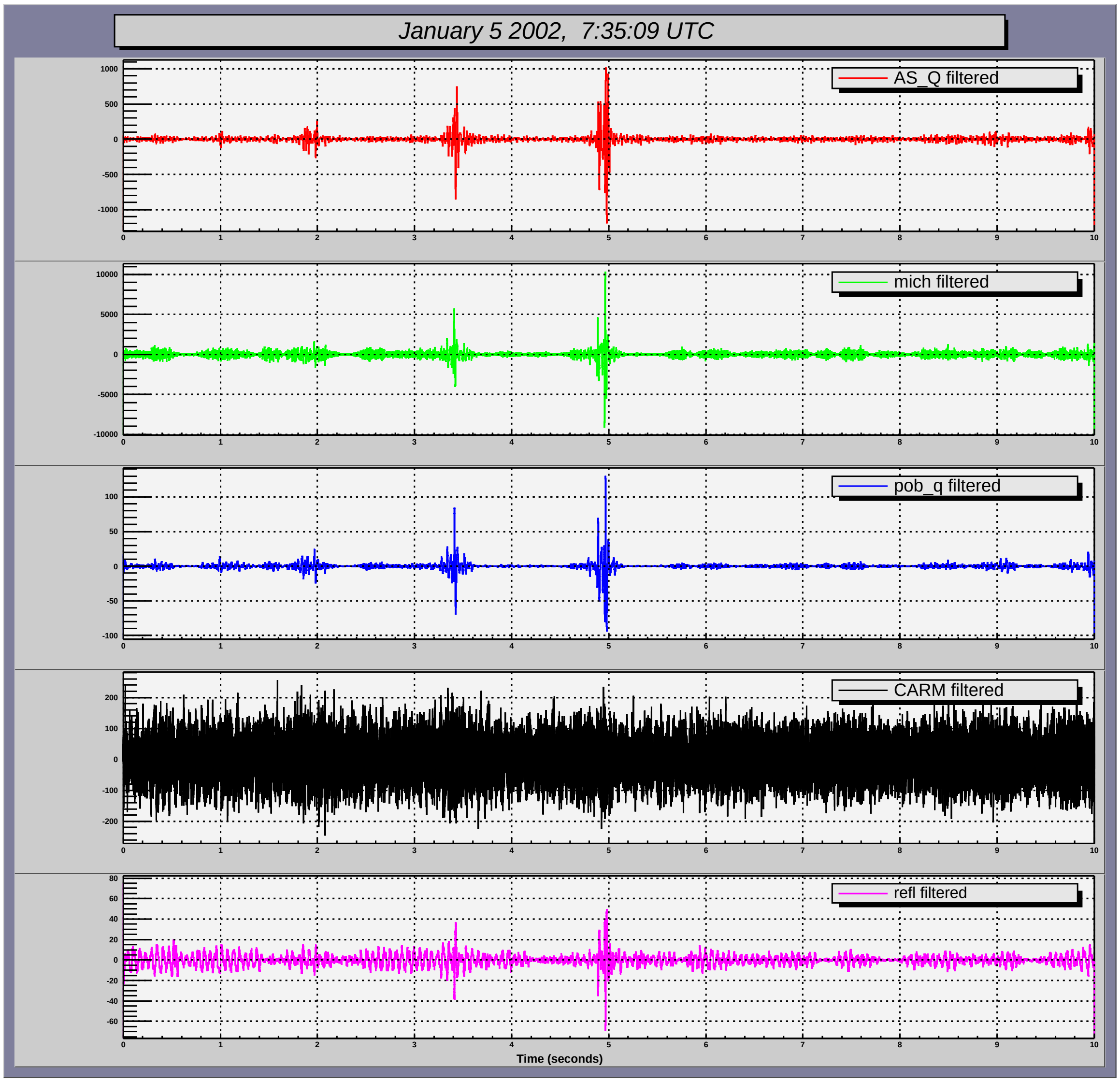
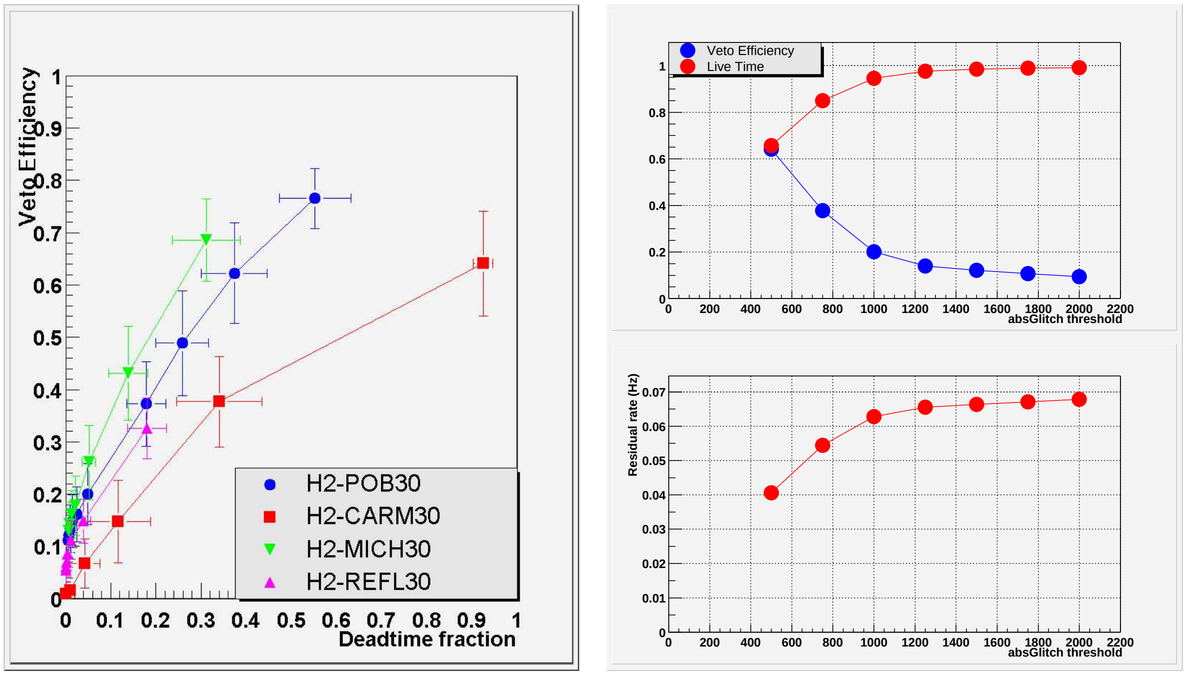
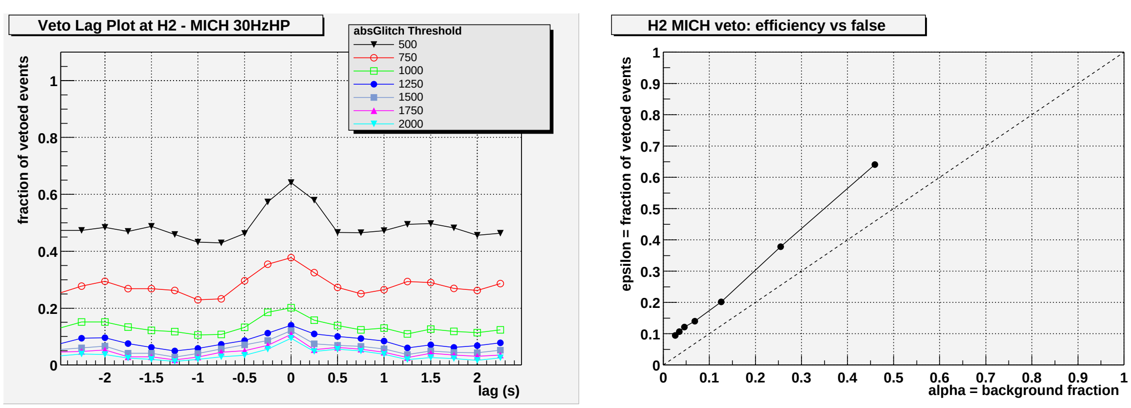


Figure 11: *A typical E7 glitch at H2. At the top is the AS_Q time series, filtered with a 300 Hz high-pass filter. Below are MICH_CTRL, POB_Q, CARM_CTRL and REFL_Q, all filtered at 30Hz.*



(a) **(b)**

Figure 12: *Search for a veto at H2: (a) efficiency vs dead-time for different thresholds and channels; (b) efficiency, lifetime and residual rate as functions of the MICH threshold at H2.*



(a) *Veto lag plot* **(b)** *small Peak vs Tail*

Figure 13: *Lag plot and peak vs tail plots for H2 and the MICH_CTRL veto. The veto is, in this instance, much less efficient than the L1-PSL veto, but we should remember the initial event rate is much smaller.*

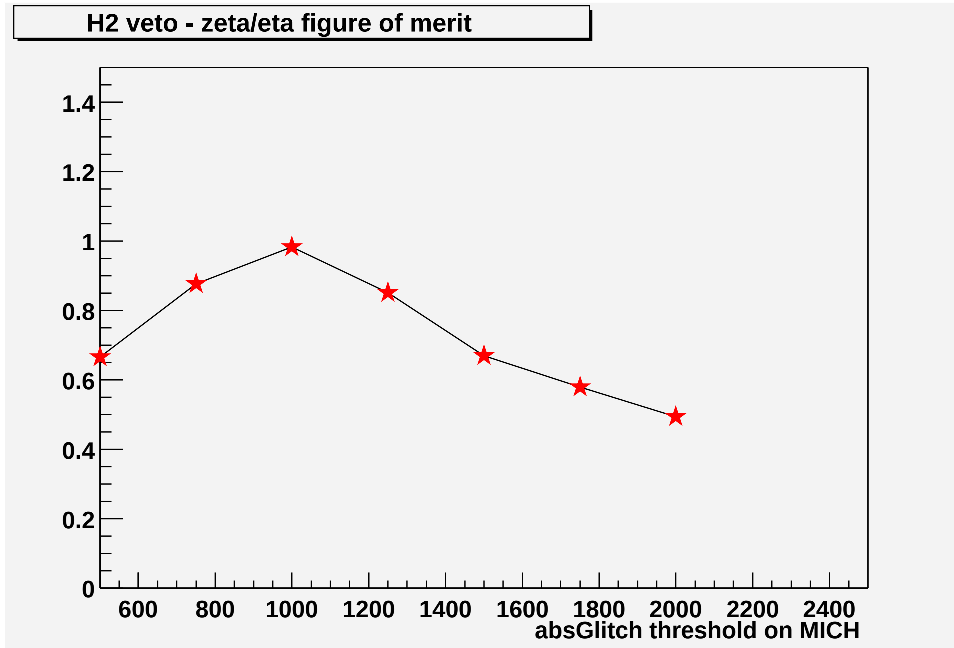
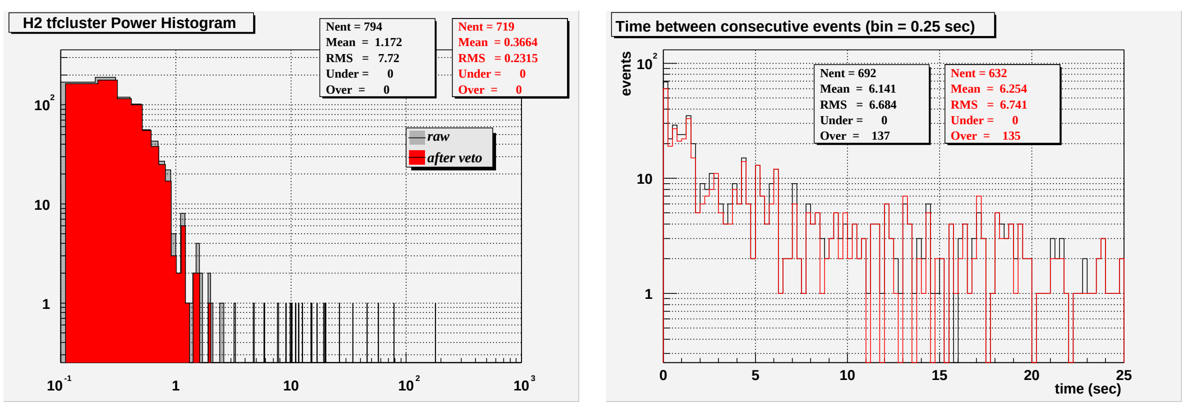
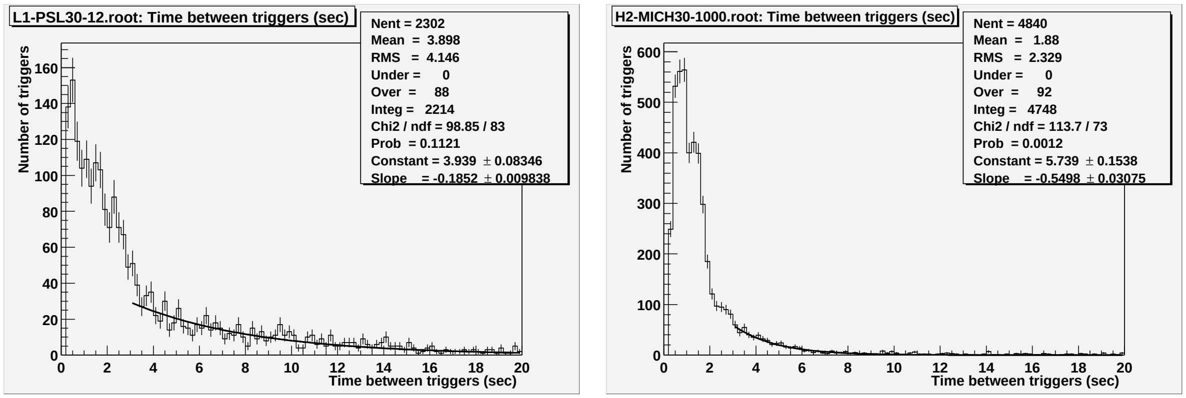


Figure 14: $\rho = \zeta/\eta$ *ratio for the MICH_CTRL veto at H2, function of the veto threshold. This quantity does not have a real minimum - it monotonically decreases toward some asymptotic condition.*



(a) (b)

Figure 15: *effect of the veto on the H2 power histogram (a) and the distribution of the delay between events (expected to be exponential for Poisson processes)(b). Data from the Tfcluster DSO.*



(a) *L1:PSL-ISS_ACTM_F*

(b) *H2:LSC-MICH_CTRL*

Figure 16: *Time between triggers*
(a) Channel: L1:PSL-ISS_ACTM_F, Filter: 30Hz highpass, 4th order butterworth, Threshold: 12 counts
(b) Channel: H2:LSC-MICH_CTRL, Filter: 30Hz highpass, 4th order butterworth, Threshold: 1000 counts

MICH_CTRL. This issue is addressed in section 5.3.

In the case of the tfcluster DSO, whether we do use a veto at H2 or not is not really affecting the upper limits we can set, an argument in favor of not using a veto at H2 until all questions are answered.

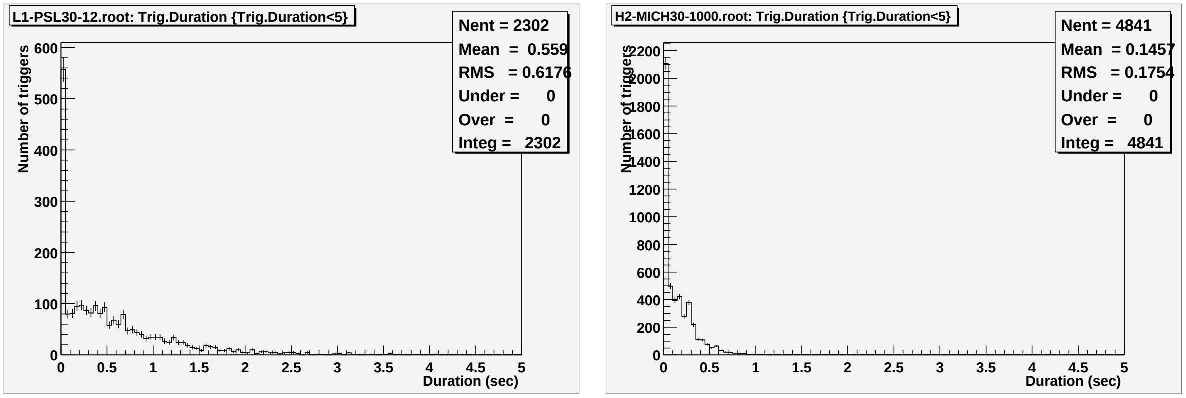
5.3 H2:LSC-AS_Q - H2:LSC-MICH_CTRL cross coupling

During E7 the H2 data had a large number of very dominant glitches, clearly visible in most of the LSC channels (see figure 11). Because these glitches were clearly of instrumental origin (in fact it was found later that a servo instability caused them), we tried to come up with a veto for them. As described above H2:LSC-MICH_CTRL seemed to be the most sensitive channel to detect them.

It is known that in principle H2:LSC-MICH_CTRL is also coupled to the differential arm degree of freedom, but a lot less than H2:LSC-AS_Q. In order to be sure that we are not throwing away valid candidate events we plotted (in figure 19, blue) the square root of the total power reported by TFCLUSTER during a glitch in H2:LSC-MICH_CTRL (reported by absGlitch) vs the maximal amplitude of H2:LSC-MICH_CTRL during that glitch. Also we added corresponding points for some of the Jan. 11 injected signals (red): “Sombrero” (∼250Hz,4x), “Wiggle” (∼200Hz,1x) and “Blip2” (∼300Hz,2x) waveforms. The signals were quite broadband, especially “Sombrero” and “Wiggle (see figure 20 or the Jan. 12 Livingston ilog entry by Szabi Marka).

These waveforms were injected to H2:LSC-ETMX_EXC only but, since both H2:LSC-AS_Q and H2:LSC-MICH_CTRL are only sensitive to the differential arm degree of freedom, this does not matter. Also note that the maximal amplitude for these events, after 30Hz high-pass filtering was read out by hand, since 6 out of 7 were too small to be found by absGlitch. The single event with the highest maximal amplitude turned out to be the (only) “wiggle” waveform, i.e. the lowest frequency one. At least for these 7 events the amplitude ratio $\frac{H2:LSC-MICH\_CTRL}{H2:LSC-AS\_Q}$ indeed seems to be frequency dependent (figure 21).

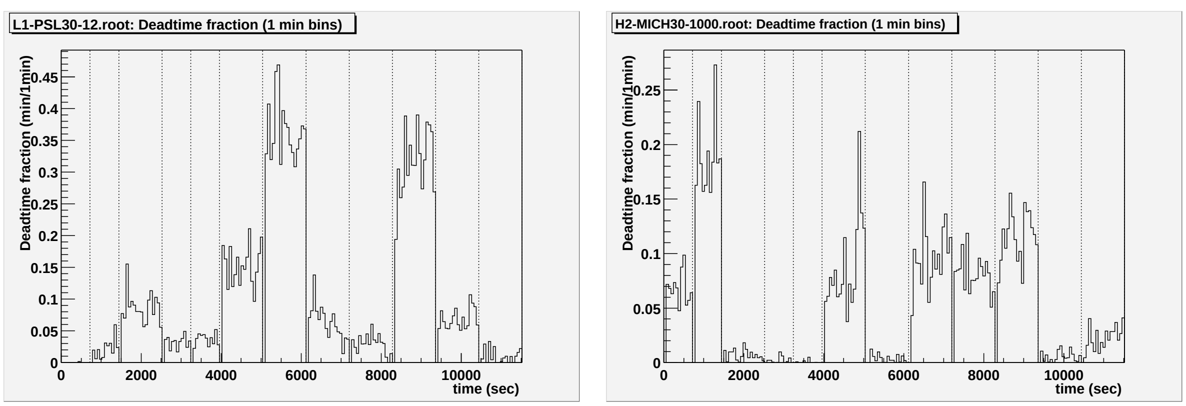
The vetoed events (blue) in figure 19 clearly fall into two classes. Most of the events



(a)
L1:PSL-ISS_ACTM_F

(b)
H2:LSC-MICH_CTRL

Figure 17: *Duration distribution*
(a) Channel: L1:PSL-ISS_ACTM_F, Filter: 30Hz highpass, 4th order butterworth, Threshold: 12 counts
(b) Channel: H2:LSC-MICH_CTRL, Filter: 30Hz highpass, 4th order butterworth, Threshold: 1000 counts



(a)
L1:PSL-ISS_ACTM_F

(b)
H2:LSC-MICH_CTRL

Figure 18: *Deadtime fraction vs time (Rate weighted with Duration vs time)*
(a) Channel: L1:PSL-ISS_ACTM_F, Filter: 30Hz highpass, 4th order butterworth, Threshold: 12 counts
(b) Channel: H2:LSC-MICH_CTRL, Filter: 30Hz highpass, 4th order butterworth, Threshold: 1000 counts

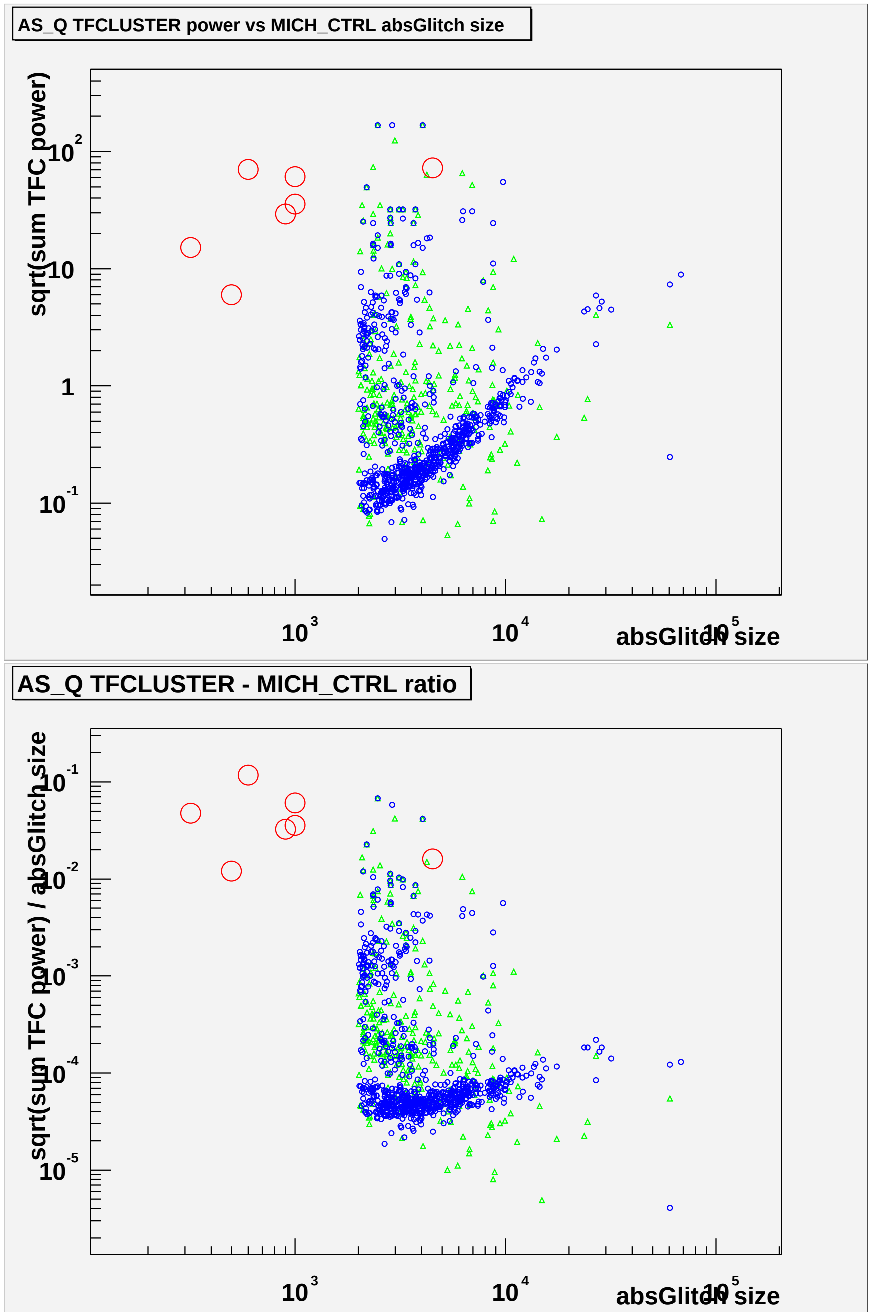
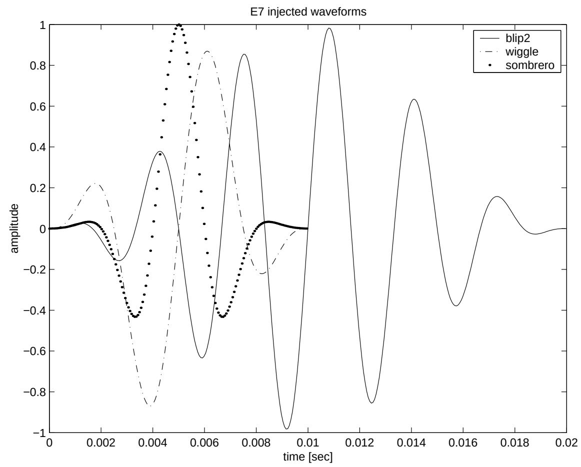
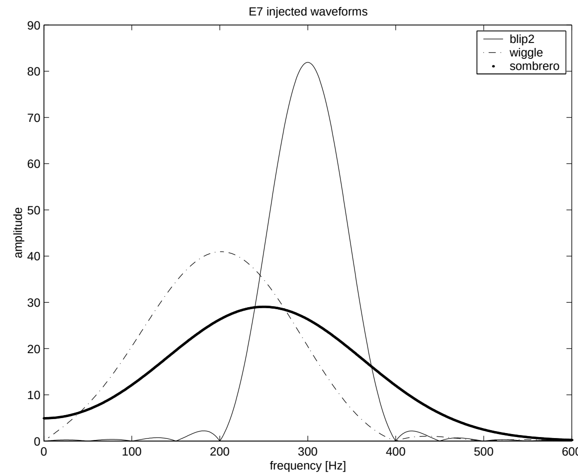


Figure 19: $\sqrt{\sum TFCpower(AS\_Q)}$ *(above) and* $\frac{\sqrt{\sum TFCpower(AS\_Q)}}{absGlich\ size\ (MICH\_CTRL)}$ *(below) vs absGlich size (MICH_CTRL): The vetoed H2 E7 playground events are plotted as small blue circles. They clearly fall into 2 classes. The red circles correspond to events injected into H2:LSC-ETMX_EXC. The green triangles correspond vetoed events for a 2sec shifted veto stream, i.e. they give an impression of the random background. (Cuts: absGlitch threshold=2000, veto-trigger duration < 0.15sec)*



(a)
time domain



(b)
frequency domain

Figure 20: *The waveforms of the injected events in figure 19 in time (a) and frequency domain (b)*

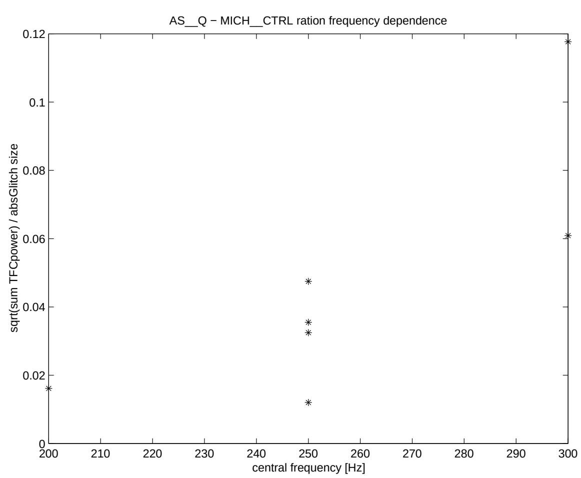


Figure 21: *The ratio* $\frac{\sqrt{\sum TFCpower(AS\_Q)}}{absGlich\ size\ (MICH\_CTRL)}$ *vs the central frequency for all 7 injected events, 'Sombrero' (250Hz,4x), 'Wiggle' (200Hz,1x) and 'Blip2' (300Hz,2x)*

cluster in the lower branch. These events show up about 100x stronger in H2:LSC-MICH_CTRL than what we would expect for an astrophysical event with the same power in H2:LSC-AS_Q and are therefore of instrumental origin.

Moreover, from figure 22 we can see that these 2 event classes have a different frequency behavior. The bigger lower branch contains almost exclusively events with a central frequency below 100Hz, whereas most of the other events are above 100Hz. The aforementioned servo instability is probably responsible for most of these below 100Hz events.

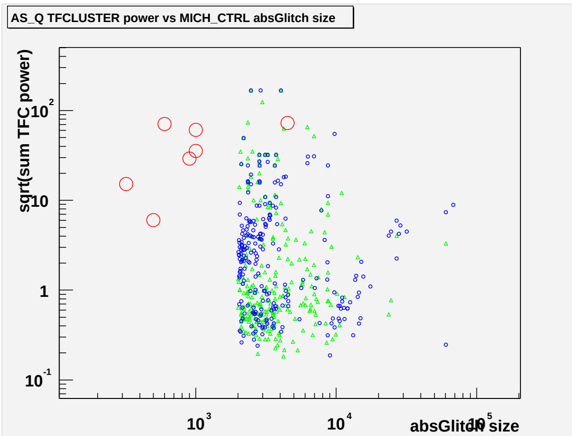
We certainly want to get rid of these instrumental events. The cleanest cut would be to throw away all events that

- are in coincidence with a H2:LSC-MICH_CTRL veto trigger, and
- have $\frac{\sqrt{\sum TFCLUSTER\, power}}{absGlitch\ size} < 10^{-3}$, and
- are below 100Hz.

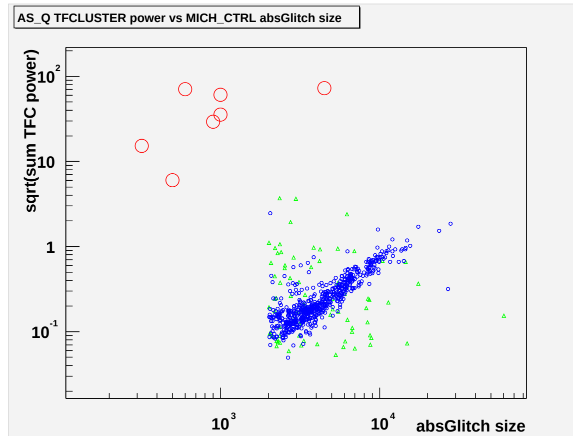
However, since the sensitivity of the instrument was not good below 100Hz anyway, it seems reasonable to just throw away everything below 100Hz. Because this already takes care of most of the instrumental glitches of the Hanford 2km interferometer, applying a additional veto (like the H2:LSC-MICH_CTRL one) then becomes obsolete.

Unfortunately this is not possible for slope since we don't have the frequency information there. Therefore a H2:MICH_CTRL veto is more important for slope. A straightforward application without cross-checking the event size in AS_Q is at risk of cutting valid very high amplitude candidates. Quantifying this statement is hard though because we do not have a full understanding of the frequency dependence of the amplitude reported by slope.

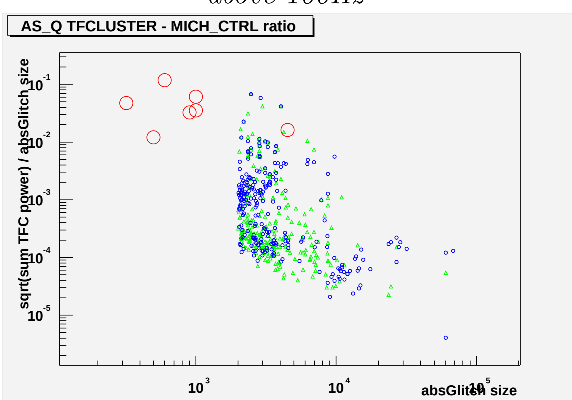
If we want to learn something about the $\frac{H2{:}LSC-AS\_Q}{H2{:}LSC-MICH\_CTRL}$ amplitude ratio independent of the used search codes we can use the calibration swept sine measurements. From the transfer functions DARM_ERR_EXC → AS_Q and DARM_ERR_EXC → MICH_CTRL (figure 24a,top; 24b,black) we can get the frequency dependence of the amplitude ratio $\frac{H2{:}LSC-AS\_Q}{H2{:}LSC-MICH\_CTRL}$ for injected signals. Figure 24a,bottom also shows the transfer functions MICH_ERR_EXC → AS_Q and MICH_ERR_EXC → MICH_CTRL. They give us the same ratio for pure michelson glitches (24b,red) - note that it is at least 100 times lower than for DARM excitations.



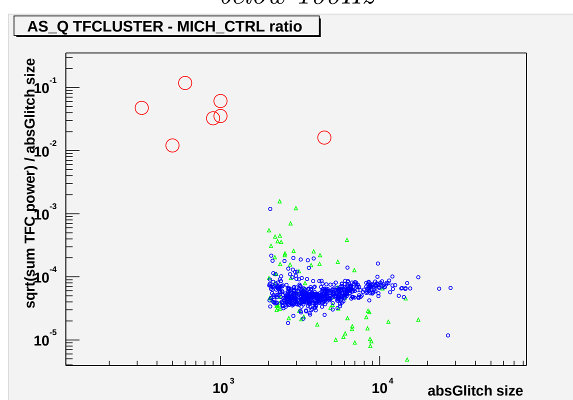
(a)
above 100Hz



(b)
below 100Hz

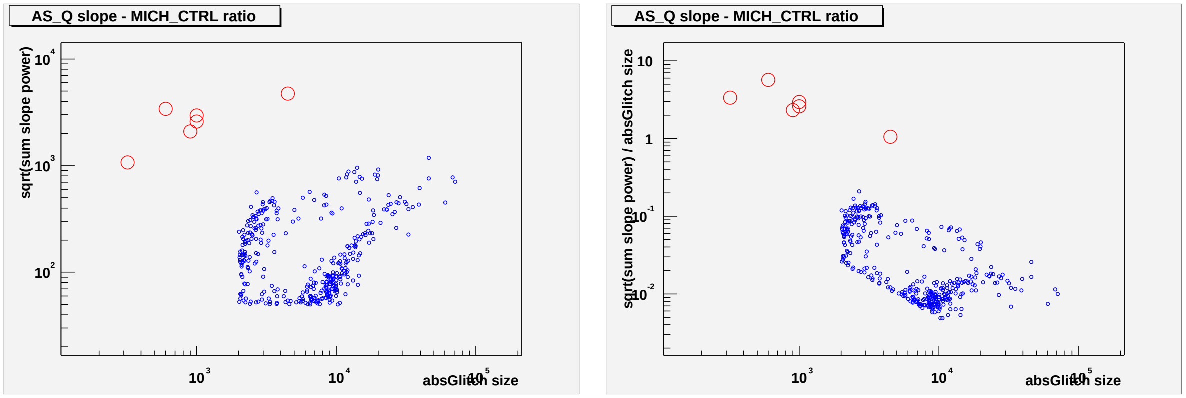


(c)
ratio, above 100Hz



(d)
ratio, below 100Hz

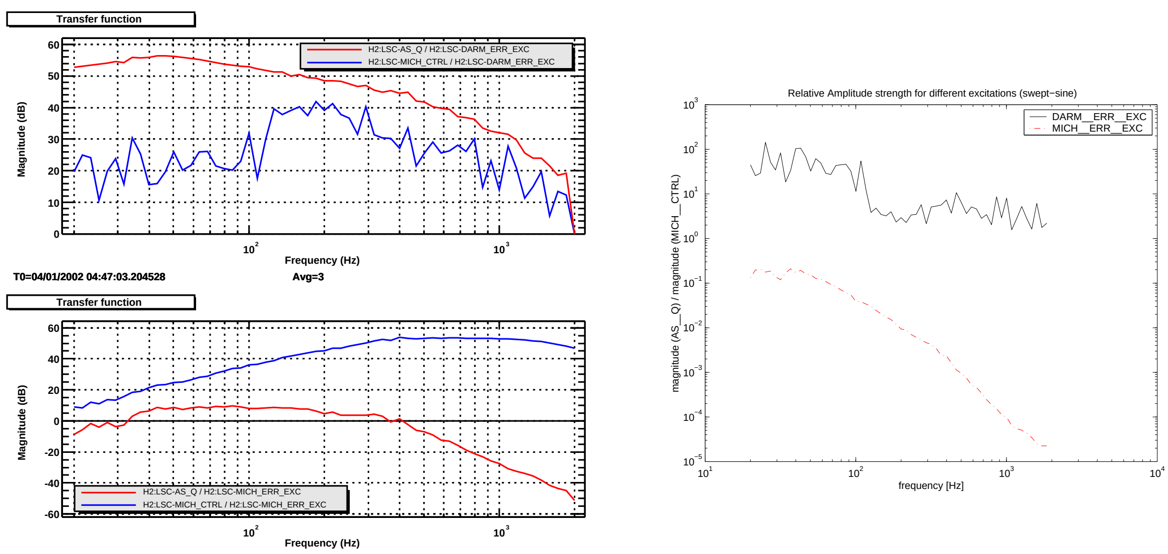
Figure 22: *Same plot as figure 19, but only the events with central frequency above 100Hz (a,c), below 100Hz (b,d) are included. (The injected events are left in all figures as a reference.)*



(a) *amplitude scatter plot*

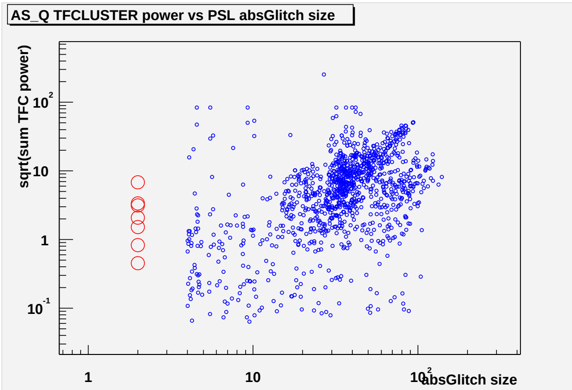
(b) *ratio vs gltch amplitude*

Figure 23: $\sqrt{\sum slope\ power(AS\_Q)}$ *(a) and* $\frac{\sqrt{\sum slope\ power(AS\_Q)}}{absGlich\ size\ (MICH\_CTRL)}$ *(b) vs absGlich size (MICH_CTRL): The vetoed H2 E7 playground events are plotted as small blue circles. As for TFCLUSTERS they clearly fall into 2 classes. The red circles correspond to events injected into H2:LSC-ETMX_EXC. (Cuts: absGlitch threshold=2000, veto-trigger duration* $<$ *0.7sec)*

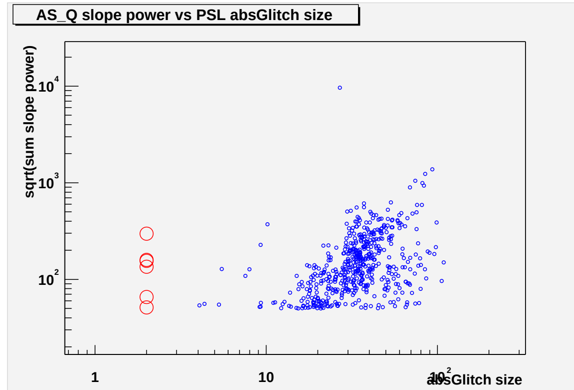


(a) **(b)**

Figure 24: *H2 excitation transfer functions based on calibration swept sine*
(a,top) Transfer functions DARM_ERR_EXC $\rightarrow$ *AS_Q and MICH_CTRL*
(a,bottom) Transfer functions DARM_ERR_EXC $\rightarrow$ *AS_Q and MICH_CTRL*
(b) $\frac{AS\_Q}{MICH\_CTRL}$ *for DARM_ERR_EXC (black) and DARM_ERR_EXC (red) excitations. For glitches in the Michelson cavity the amplitude ratio* $\frac{AS\_Q}{MICH\_CTRL}$ *is at least a factor 100 smaller than for DARM glitches.*



(a)
TFCLUSTER



(b)
slope

Figure 25: *L1:LSC-AS_Q vs L1:PSL-ISS_ACTM_F scatter plots for TFCLUSTERS (a) and slope (b). The maximal amplitude in L1:PSL-ISS_ACTM_F (=absGlitch size) of the injected signals (red circles) is an upper limit, i.e. the signals were* **not** *seen in L1:PSL-ISS_ACTM_F.*

5.4 L1:LSC-AS_Q vs L1:PSL-ISS_ACTM_F scatter plot

For completeness we also produced the same type of scatter plot for the Livingston Veto channel (L1:PSL-ISS_ACTM_F), including the signal hardware injections (figure 25). As expected, none of the injected signals was acutually seen in L1:PSL-ISS_ACTM_F, consequently all absGlich sizes of the injected signals are upper limits.

Based on that we are certain that we are not vetoing astrophysical events at Livingston.

Table 2: *Single-IFO characterization with the tfcluster DSO/ NOTE: in this context, clustered means events with the SAME time stamp. This is not a very rigorous definition of clustering, but it is useful to get an idea. We decided to deal with events as singles all the way through and think about clusters only at the end of the pipeline.*

TFCLUSTER 300-3000 Hz	L1		H2	
BURSTS	All	Clustered	All	Clustered
IN	19701	4320	794	676
Vetoed	19056	3934	75	62
OUT	645	386	719	614
VETO TRIGGERS	PSL HP30Hz Thresh = 12		MICH HP30Hz Thresh=2000	
IN	1572		469	
True	1057		48	
False	515		421	
Total duration	1174 sec		84 sec	
LiveTime	9506 sec (89.0%)		10596 sec (99.2%)	
Residual rate	0.07 Hz	0.04 Hz	0.07 Hz	0.06 Hz

Table 3: *Single-IFO characterization with the slope DSO/ NOTE: these numbers have been obtained with my present favorite choice of thresholds on the DSO - they are bound to change with a different slope threshold. (LC)*

SLOPE	L1 (amplitude > 55)	H2 (amplitude > 75)
BURSTS IN	9000	1827
BURSTS Vetoed	8873	341
BURSTS OUT	127	1486
GDS-TRIGGERS	PSL HP30Hz Thresh = 12	MICH HP30Hz Thresh=2000
IN	1571	469
True	1102	140
False	469	329
Total duration	1174 sec	84 sec
LiveTime	9506 sec (89.0%)	10596 sec (99.2%)
Residual rate	0.013 Hz	0.14 Hz

5.5 Conclusions on single IFO characterization

Our recommended choice of vetoes is to run absGlitch over the following channels:

- L1: PSL, filtered with 30Hz High Pass and threshold 12;
- H2: TFCLUSTER: No veto, or, if the below 100Hz region is of interest, MICH_CTRL, filtered with 30Hz High Pass and threshold 2000.
- H2: slope: MICH_CTRL, filtered with 30Hz High Pass and threshold 2000.

The consequence on rates, with the tfcluster and slope DSO, are summarized in tables 2 and 3.

6 IFO-IFO coincidence

Once the vetoes have been applied at each site, we have two lists of events that we want to combine, in order to perform a coincidence analysis. Here is the recipe we followed:

1. Looking at simulation injected data, we noticed that the simulated signal has a jitter of up to 0.25 sec (at least in tfcluster, where the time resolution is 0.125 sec).

2. We chose to start from a relatively large coincident window and select events according to the condition:

$$|H2_{Start} - L1_{Start}| < 0.5\, sec$$

   This can easily be replaced by an overlap condition, similar to the one used for vetoes, but we expect the difference in the final result to be small. This is a more general and conservative condition.

3. In tfcluster (and power), we can impose a frequency match requirement. The frequency resolution in tfcluster is 8 Hz; we allowed for a 10 sigma tolerance and required that the bandwidth of coincident events overlaps within $\pm 80\,\mathrm{Hz}$[5]

4. Finally, we perform a clustering in power: all the events taking place within 1 second are treated as one single event and their power is added.

This procedure identifies a handful of events that we want to examine with more care[6].

Note that both the time and the frequency coincidence requirement are still quite loose: there is room for improvement and we are trying to use the simulations (so far, only software injections in the time series) to do that. Nevertheless, these requirements, as they are, are already pretty good to identify only a handful of events, which can be looked on an event by event basis.

As far as the clustering goes, after the L1 veto, the residual rate is low enough that clustering becomes "easy matter". We show below that with tfcluster we are left with about 4 pairs of coincident events, scattered through 3.2 hours, and only one event needs "clustering".

Clustering is an issue before the veto application, because the glitches are usually broadband and do present several entries in the database (this is especially true for tfcluster, but slope presents the problem as well). We chose to treat each event as a single, in order to avoid some tricky technical aspects of clustering, and to postpone clustering at the end of the pipeline. Injected simulations are also typically broadband, but in that case, the requirement of 1 (or even 2) seconds between start times does enough of a good job for this analysis.

In order to see how the procedure works, we focus on tfcluster, where the DSO thresholds have already been optimized by Julien Sylvestre and the event rate is reasonable (more on the other DSOs later). The result we obtain is:

- 12 time coincidence: 12 coincidences;
- clustering (start time within 1 sec): 11 events;
- frequency coincidence: 3 events

[5] Julien here requires overlap of the bandwidth, while I am leaving a 80 Hz tolerance - it is not completely clear to me how different the frequency response can be at the two sites.

[6] with cross correlation? still to be determined

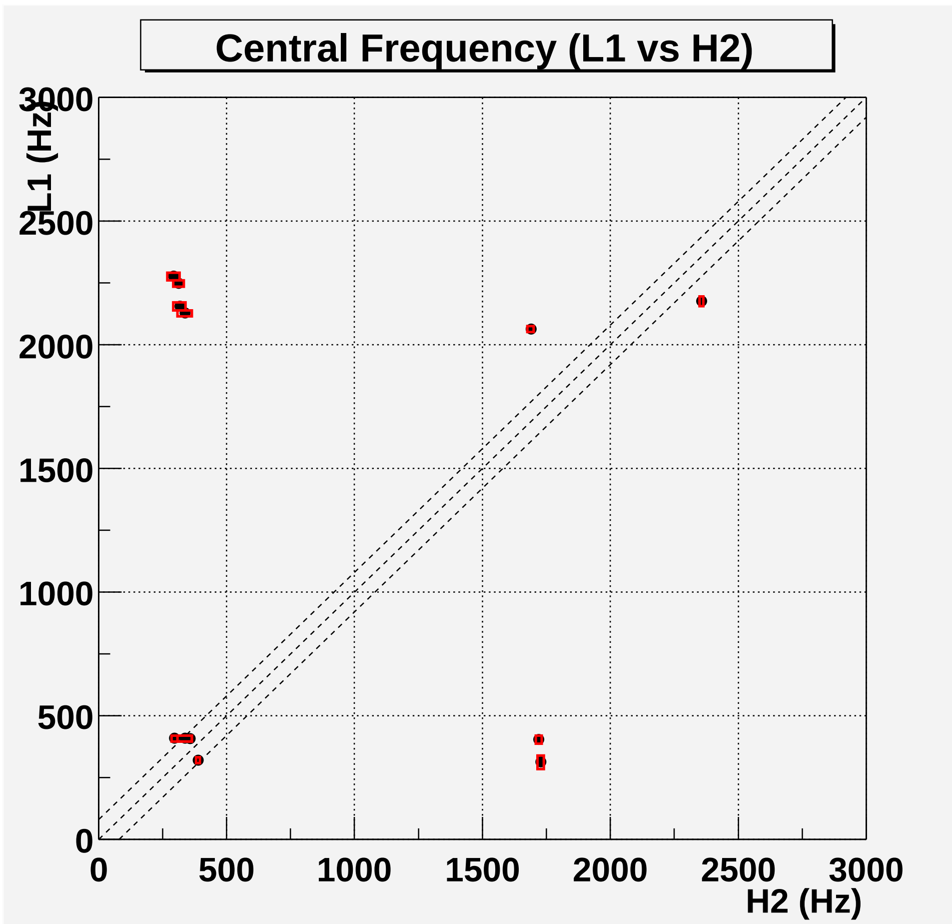


Figure 26: *tfcluster: frequency at L1 vs frequency at H2. The black dots represent the central frequency at one site versus the other. The red boxes have the bandwidth folded in. If the red box overlaps the band defined by the diagonal dashed line, the event passes the frequency cut, because the bandwidth at the two sites overlaps within 80Hz. Notice that in the injected simulations seen so far (see section* **??***) bursts are broadband, so in principle we could set a cut on bandwidth and all these events would be rejected. The injection process is still under study, though.*

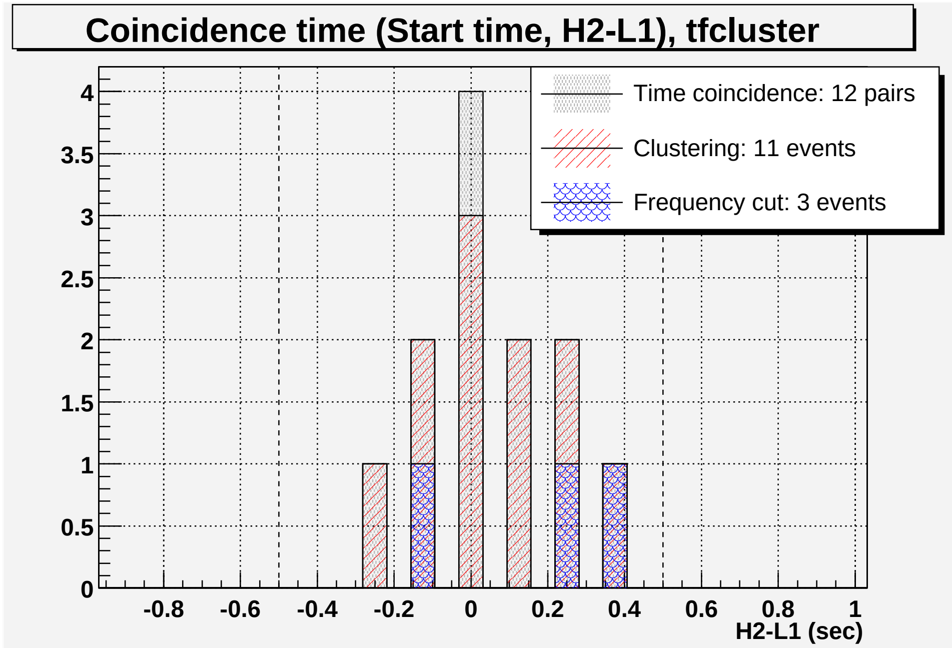


Figure 27: *tfcluster: histogram of the* **start time** *difference (H2-L1) for the coincident events. The effect of clustering and frequency cut are also shown.*

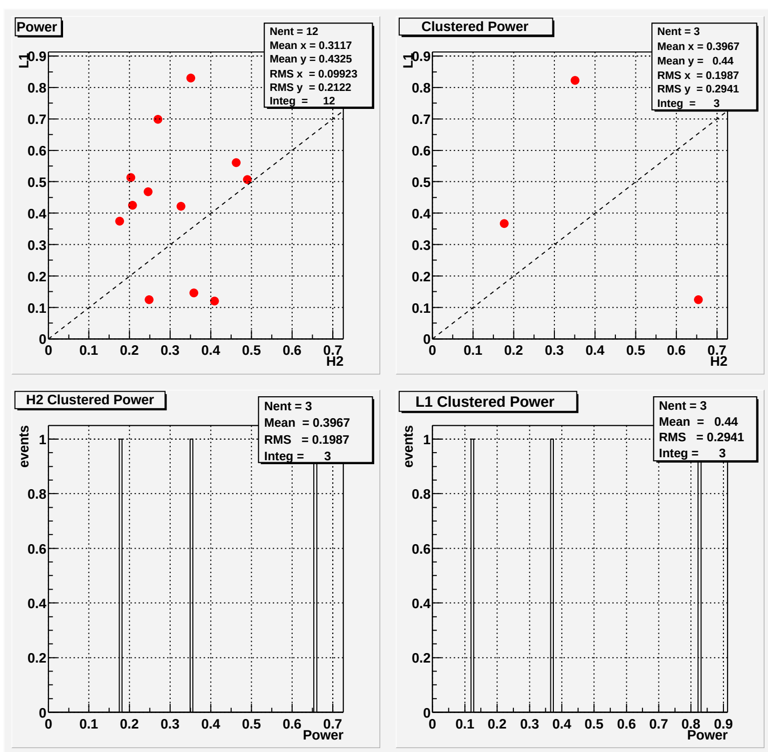
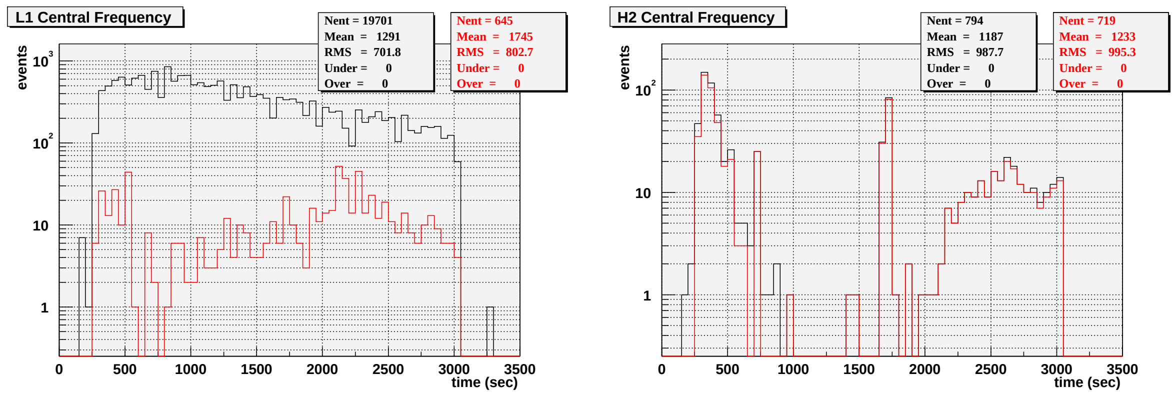


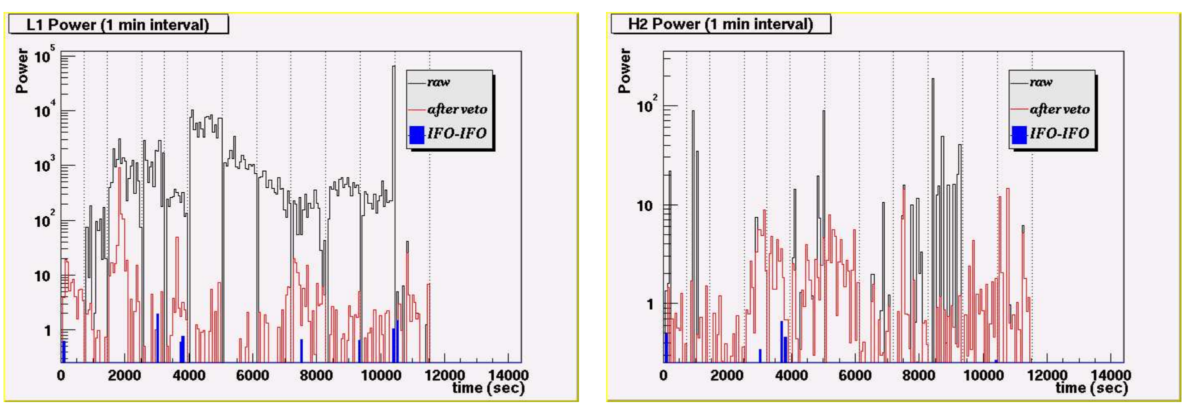
Figure 28: *tfcluster: the upper left plot shows the power at L1 versus the power at H2 for the 12 pairs of events that pass the time coincidence requirements. The other three plots only show the 3 (clustered) events that pass the frequency cut: L1 versus H2 scatter plot and the histogram for the two sites.*



(a)
L1

(b)
H2

Figure 29: *Histograms of tfcluster central frequency at L1 and H2, before and after the veto application (PSL-12 at L1, MICH-2000 at H2). Note the peaks at ∼350 Hz and ∼1700 Hz in the H2 histograms: they are reflected in the frequency content of the residual coincident events (see figure 26)*



(a)
L1

(b)
H2

Figure 30: *Power vs time (tfcluster) at each site: raw, after the veto application and after the IFO-IFO requirement.*

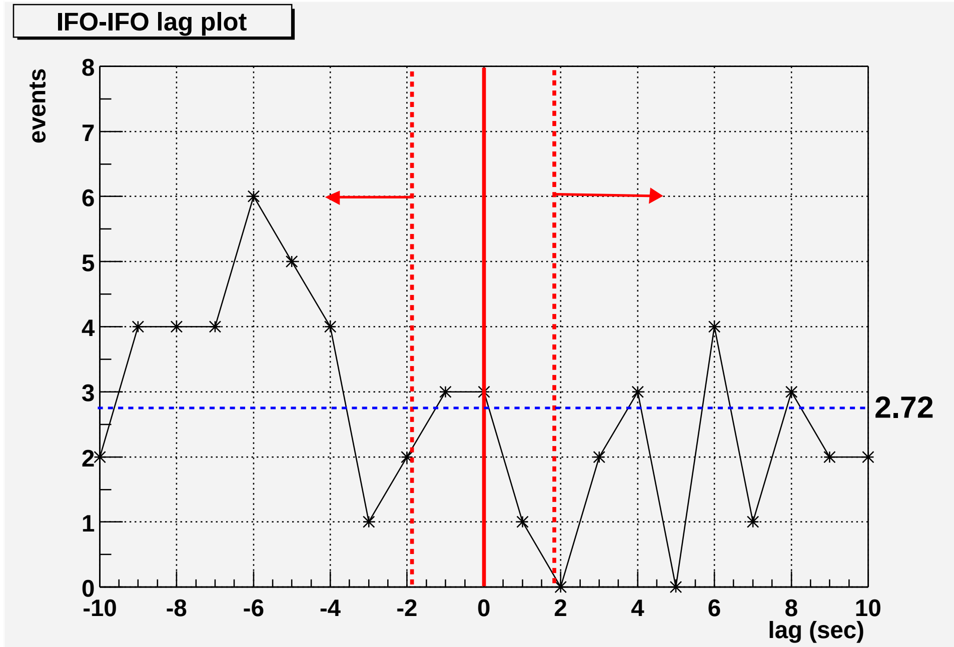


Figure 31: *tfcluster coincident events: lag plot*

7 Setting upper limits - the veto issue revisited

The number of final candidates can be used to set an upper limit. We followed the Feldman Cousin approach. We have $n_0 = 3$ final candidates for the L1H1H2 E7 playground and we estimate the background by shifting the time series at one of the two sites by a given time lag. The resulting lag plot is shown in figure 31, where the lag is changed from -10 sec to + 10 sec. An average of the coincident rate between ± 2 and ± 10 sec yields the background ($b = 2.72$ events for this veto choice). The values of b and n_0 can be used, in the Feldman Cousin approach, to set an upper limit with some confidence level, on the number of "real" events in the 3.2 hours we analyzed. The upper limit, divided by the total remaining live time, can then be used as an upper limit on rate.

We performed the upper limit estimation, for the tfcluster DSO, for several combinations of the veto thresholds. The idea was to revisit the question of what threshold to use in the veto if the figure of merit we want to optimize is the upper limit obtained at the end of the pipeline.

It turns out that the actual upper limit might not be the best quantity to minimize - we might want to minimize, instead, the background b. Since $n0$ is a discrete quantity and b is continuous, the fact that we are analyzing only 3.2 hours of data make the upper limit jump in discrete steps (1/10680=0.1 mHz, of the order of the variations of the upper limit itself!). The effect of this can be noticed when comparing figures 32 and 33.

The UL minimization suggests a threshold at 16 for L1, but that is a fluke due to the fact that the background has monotonically increased from 2.7 to 4.2 (as expected from the single IFO characterization), while the coincidences at 0 lag remain 3. There is also an issue with the error on both n_0 and b: we refer to Julien Sylvestre's thesis for a complete treatment (which we have omitted).

The minimum background is anywhere at threshold ≤ 12 on the L1 PSL veto, in agreement with what seen in section5. The other thing one can notice from these plots is that the threshold for the H2 veto has no relevance at all - we can easily neglect the veto at H2 and still get the same background/upper limit. This is a reassuring statement

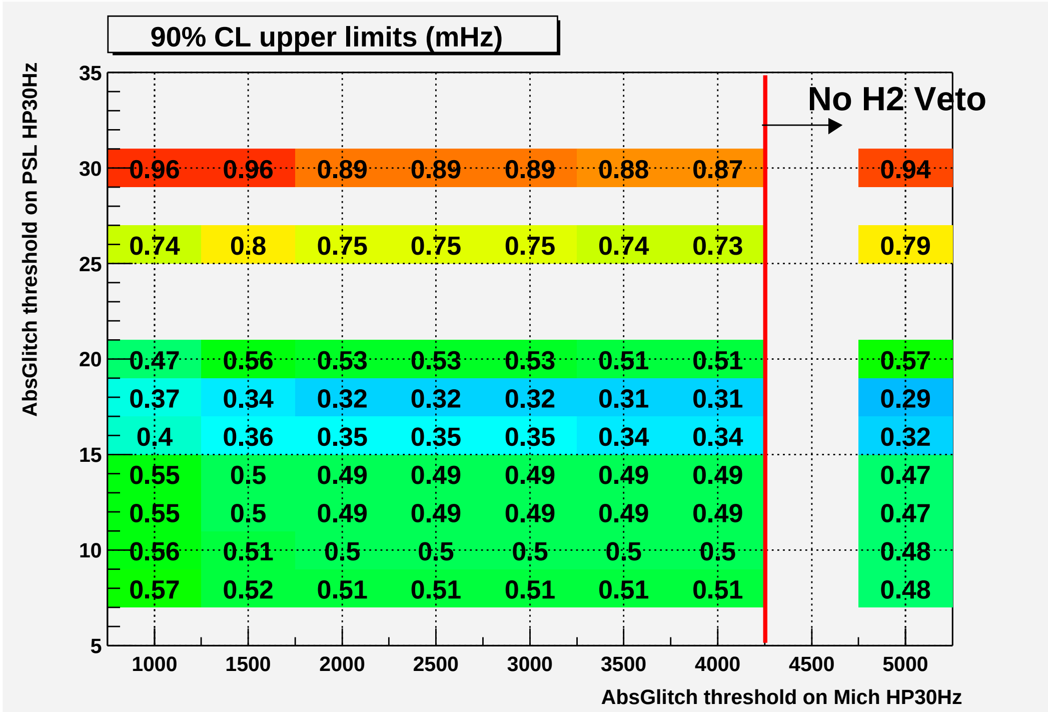


Figure 32: *Feldman Cousin upper limits (90% CL), functions of the veto thresholds at L1 and H2. Note that there is actually an error on these numbers (10-20%) due to the error on* n_0 *and* b*. For a complete treatment of this topic, we refer to Julien Sylvestre's PhD thesis.*

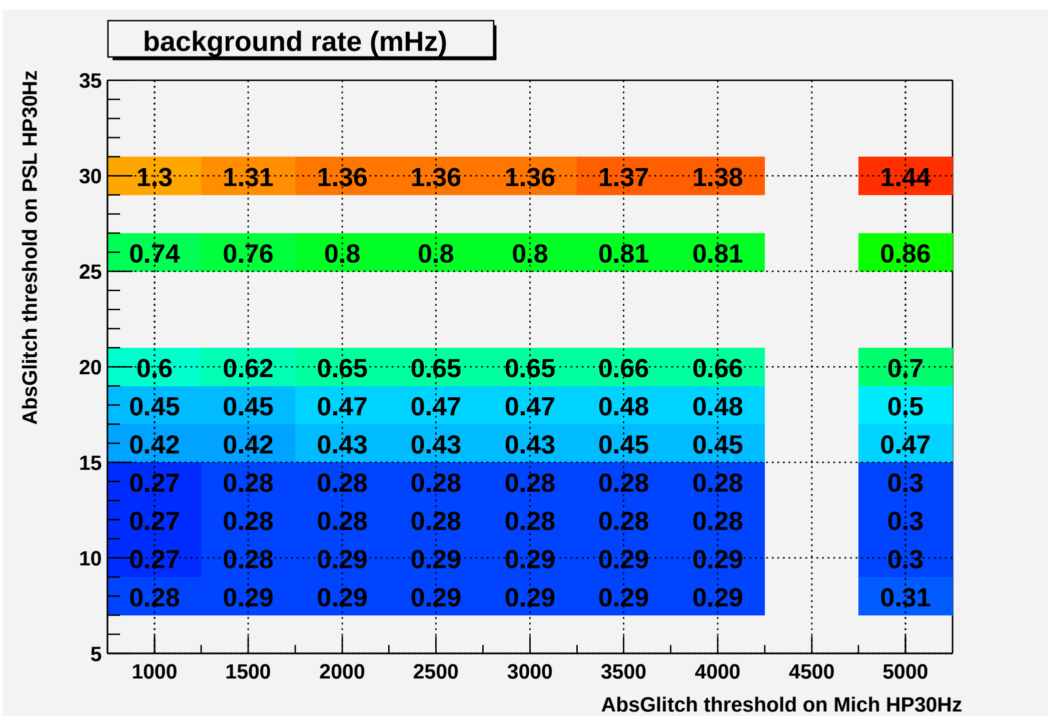


Figure 33: *Background estimated with the lag method (b), for different values of veto threshold at L1 and H2. Threshold=5000 at H2 actually corresponds to no veto applied at H2.*

- as we wait for an agreement on the legitimacy of the MICH_CTRL veto, we can safely operate without a veto (this statement might need to be revisited for the slope and power DSOs).

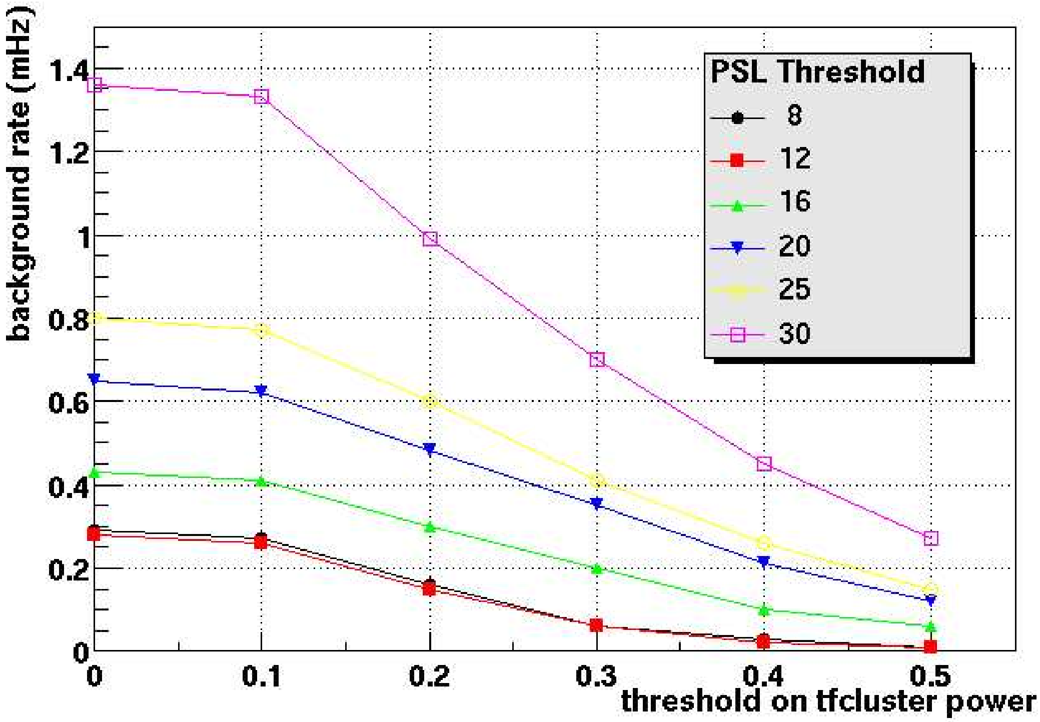


Figure 34: *Background rate, at the end of the pipeline, obtained with the tfcluster DSO, as a function of the DSO threshold and for various threshold for the PSL veto at L1.*

8 DSO threshold tuning

8.1 tfcluster

The threshold for tfcluster in E7 has been optimized by its author, Julien Sylvestre. A post process tune up of the threshold is still an interesting investigation, though, especially if we want to attempt tuning the veto threshold and the DSO threshold simultaneously.

For several different veto threshold at L1, we changed the threshold on tfcluster and monitored the variation of the final background (and, consequently, the upper limit). For ease of treatment, we can safely assume the veto at H2 is not relevant to this study, and disregard it.

Results are shown in figures 34 and 35. The background varies monotonically with the threshold, while the upper limit is "jumpy". This is due to the short measurement time and the relevance of the discrete steps in n_0 discussed at the end of the previous section.

Based on figure 35, we could in principle set a larger threshold, between 0.2 and 0.4 - but this is really fine tuning that is probably difficult to do within the playground. Note that, anyway, a threshold set at these levels does not impair by much the sensitivity to astrophysical signals (refer to the simulations).

8.2 slope

The slope DSO offers an alternative approach to the time-frequency analysis of tfcluster and the power statistics; its performances with software injections are well comparable to those of tfcluster. The only problem that is still open is to set a realistic threshold at the DSO level, something that we will try to target in this section.

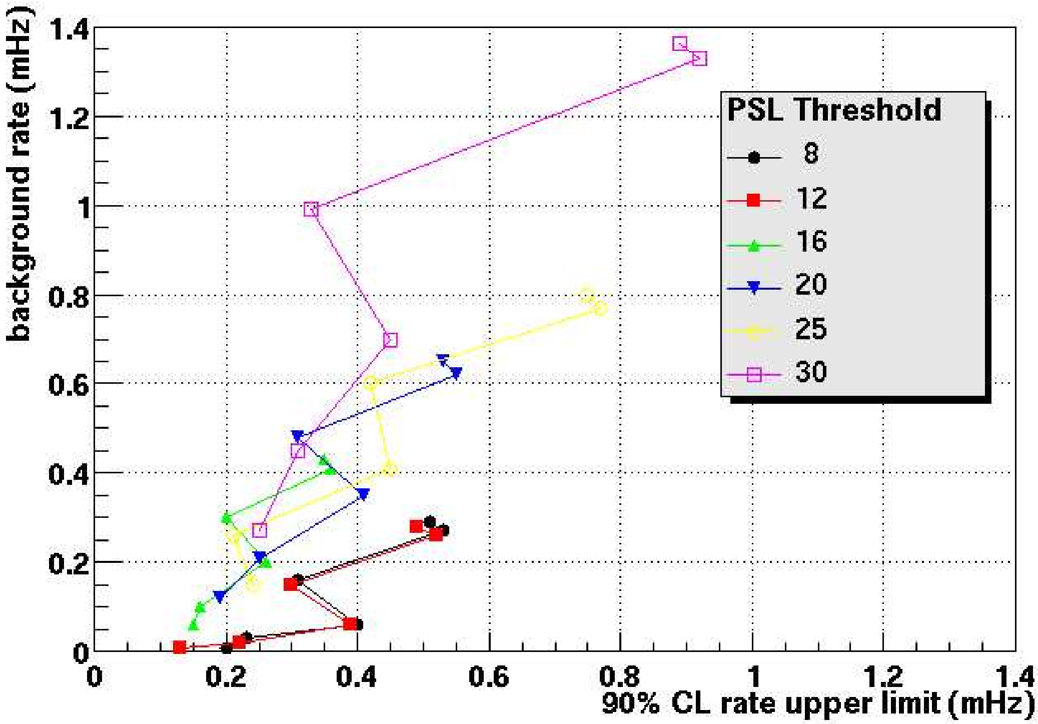


Figure 35: *background versus 90% upper limit in the E7 playground, for different values of the threshold on the tfcluster DSO and on the PSL veto at L1.*

The E7 playground has been analyzed by slope with an empirically chosen threshold: AMPLITUDE > 50, which means "slope power" > 2500. The number of events thus written to the database is pretty large and not necessarily significant, since most of the events happen in the neighborhood of the threshold.

Figure 36 shows the power histogram for a 360 sec stretch at H2, with the slope threshold set at, respectively, 2500, 3000, 3500 and 4000 in power. The only consequence is the elimination of low power events, identical to the result we would have obtained with a software cut on the event power in post-processing. This test ensures us that we can start from the existing low-threshold runs, tune the threshold for E7 playground within the EventTool and, once the consensus on the threshold is reached, set the new threshold at the DSO level for E7 final. This will solve some of the data handling issues we have had with slope so far.

The next thing to look at is how the slope threshold is affecting the background, at the end of the pipeline. As for tfcluster in the previous section, we define background as the rate of residual events, in coincidence at the two sites, after we apply a shift to the time series at one of the two sites. We averaged the result obtained with a lag ranging from ± 2 sec to ± 10 sec.

Figure 37 shows how the background changes with the slope threshold (which we have been varying within the EventTool). The background falls with a $1/x^6$ power law[7]: the continuous line connecting the points in the plot is an actual fit. The red vertical line marks the power threshold as it was set in the DSO. From a first look at this plot, it is clear that the threshold as it is is definitely too low. The data handling with slope, at this point is much more cumbersome than it needs to be.

[7]is this what we would expect?

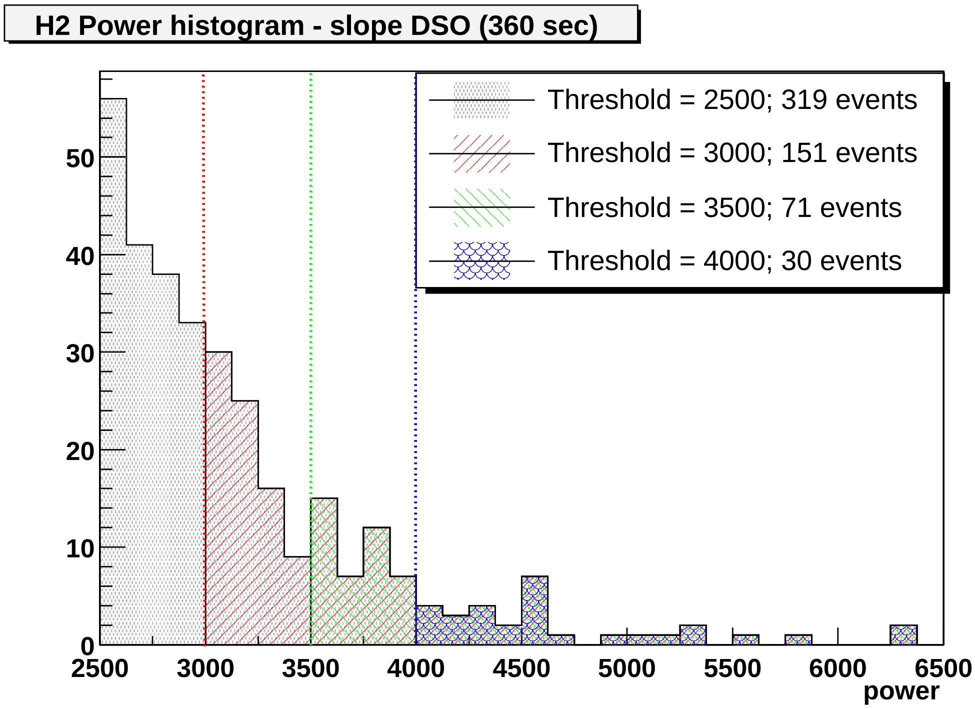


Figure 36: *Power histogram from 360 sec of H2 data, obtained with the slope DSO. We tested 4 thresholds, set at the DSO level. The only effect on the power histogram was the elimination of low power events. These results are identical to those obtained in a post-processing cut on the event power.*

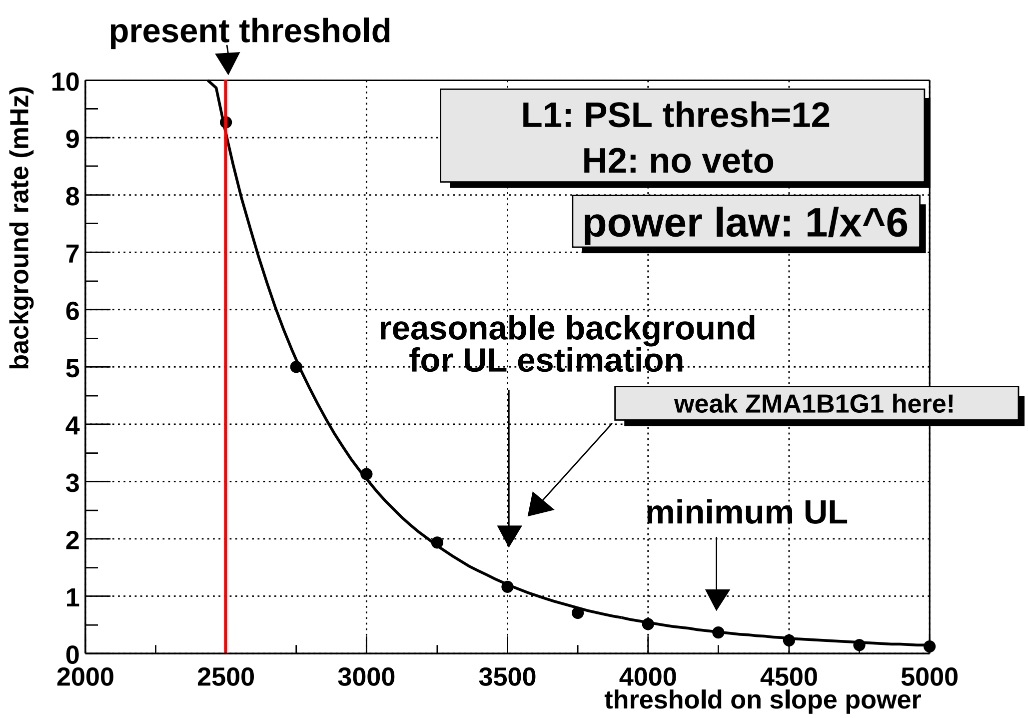


Figure 37: *Background rate, at the end of the pipeline, obtained with the slope DSO, as a function of the DSO threshold. The pipeline includes the optimal veto chosen at L1 with tfcluster and no veto at H2. Note that the background scales with the inverse sixth power of the DSO threshold.*

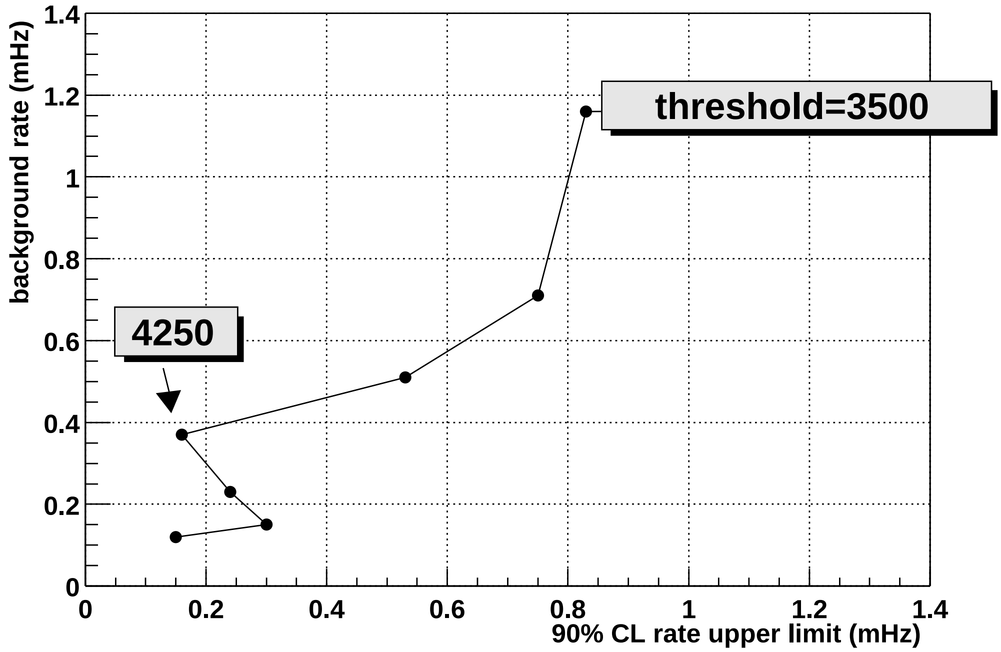
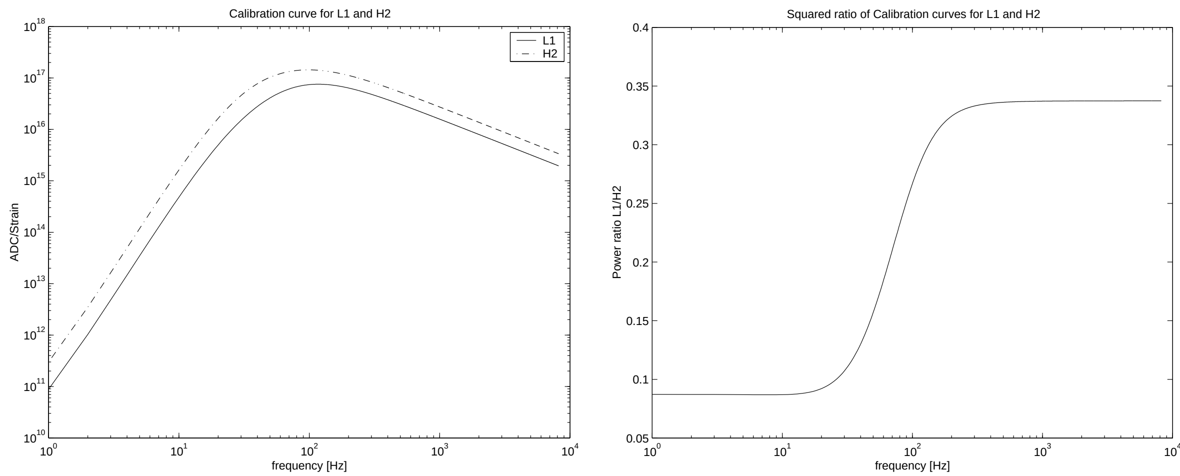


Figure 38: *background versus 90% upper limit in the E7 playground, for different values of the threshold on the slope DSO.*

The background rate, combined with the number of coincidence at zero lag, can be used to calculate an upper limit with some confidence level. Figure 38 shows how background and 90%CL upper limit change as the slope threshold is moved along the x-axis of figure 37. Due to practical reasons, we started from a threshold value of 3500, up to 5000. The minimum upper limit (comparable to what found by tfcluster) is reached at 4250: if minimizing the upper limit is our criterion, this should be the threshold value we set upon.

There is another factor to take into account, that is, how the threshold is affecting the sensitivity to astrophysical signatures. The faintest injected signals we found with slope were in the 3500-4000 neighborhood, but maybe we do need to come to compromises between the sensitivity and the upper limit. We will be able to say more after analyzing the new set of simulation Alan Weinstein produced.

In the case we will have some uncertainty at the time when the DSO needs to be run over E7, our recommendation is to increase the threshold to 3000 (which will reduce the event number by 50%) and then leave some room for improvement in the post processing.



(a)
E7 response functions

(b)
L1/H2 power ratio

Figure 39: *(a) The interferometer response functions (strain sensitivity) for L1 and H2 during E7.*
(Source: http://blue.ligo-wa.caltech.edu/engrun/Calib_Home/html/cal_summ.html)
(b) Their squared ratio gives the expected power ratio L1/H2.

9 Amplitude calibration

The goal of this section is to check our understanding of the power reported by the DSO's for the simulated injections. We are mostly interested in how the power depends on frequency and the interferometer, i.e neglect constant factors.

9.1 The interferometer response functions

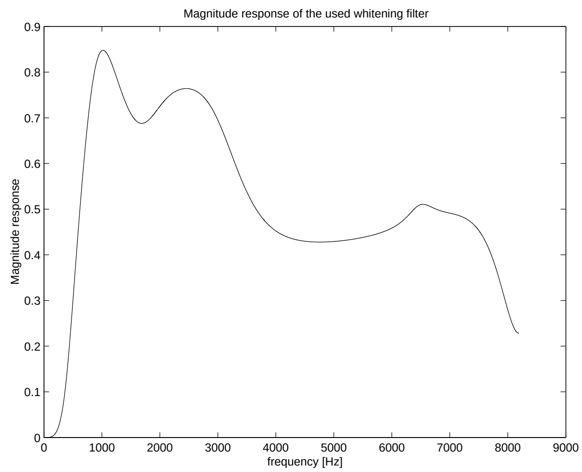
The software injected data needs to be filtered with the interferometer response function before we add it to the actual data stream. To do this we used the E7 pole zero fits for H2 and L1 published on the LIGO Amplitude Calibration Homepage (http://blue.ligo-wa.caltech.edu/engrun/Calib_Home; note that those pole zero fits are for displacement sensitivity curves, i.e they do not include the arm length.) Figure 39a shows the strain sensitivity curve for L1 and H2 during E7. Their squared ratio gives the expected Power ratio between the 2 sites for an ideal excitation at both sites, i.e. neglecting the antenna pattern. Above 300Hz this ratio is close to

$$\frac{Power\ L1}{Power\ H2} \approx \frac{1}{3} \qquad (2)$$

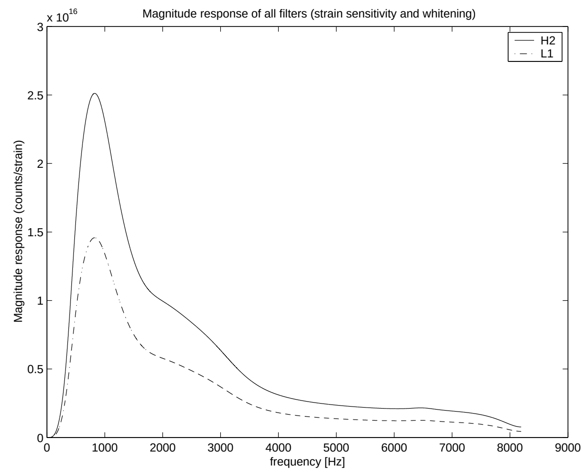
This agrees well with the result of the signal injections.

9.2 DSO prefiltering

Currently all DSO's filter the raw data with the same digital whitening filter. It's magnitude response is plotted in figure 40a. Combined with the interferometer response function we get the complete filter function that any strain signal sees before it is fed into the algorithms. (Figure 40b)



(a)
DSO whitening filter



(b)
Combined magnitude response

Figure 40: *(a) The magnitude response of the digital whitening filter used by all 3 DSO's. (b) Combined magnitude response (interferometer response function and whitening filter). Note the peak at 826Hz.*

For TFCLUSTER it is then easy to calculate the expected power for a sine-gaussian injection. TFCLUSTER defines power as

$$Power \propto \sum_i h^2_{filt\ i} = \sum_i \tilde{h}^2_{filt\ i} \tag{3}$$

where h_{filt} is the signal fed to the algorithm and $\tilde{h}_{filt}$ is its discrete Fourier transform (symmetric definition). For a narrow band sine-gaussian signal of the (used) form

$$h = A \cdot sin(2\pi f t) e^{-400t^2} \tag{4}$$

that is large compared to the noise level the power reported by TFCLUSTER is

$$Power \propto \sum_i \tilde{h}^2_{filt\ i} = |F(f)|^2 \cdot \sum_i \tilde{h}^2_i = |F(f)|^2 \cdot A^2 \cdot \frac{f_{sample}}{40}\sqrt{\frac{\pi}{2}} \tag{5}$$

$|F(f)|^2$ is the magnitude squared of the total filter function (figure 40b). This is in excellent agreement with the injected sine-gaussians if I assume a proportionality factor of $\frac{1}{2}$, see 41. (This factor $\frac{1}{2}$ could be explained if TFCLUSTERS neglects the complex conjugate part, i.e. the negative frequencies, of the Fourier transformed signal.)

At the moment we (at least I) do not fully understand the behavior of the slope power. The slope power of the injected events does not follow the same curve as for TFCLUSTERS, so clearly the slope algorithm has an intrinsic frequency dependence (see figure 42). Note that the current choice of parameters the slope algorithm seems to be most sensitive to events around 200Hz (figure 42a). When we include the overall transfer function this sensitivity region moves to 550Hz (figure 42b).

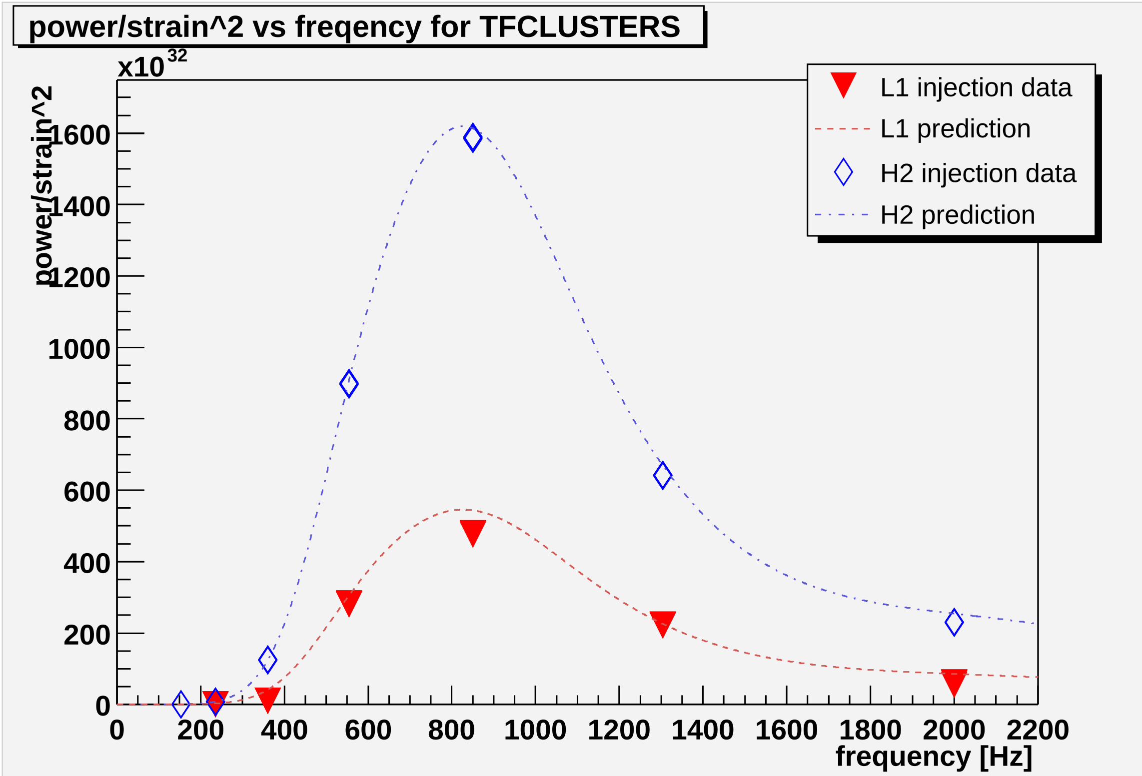
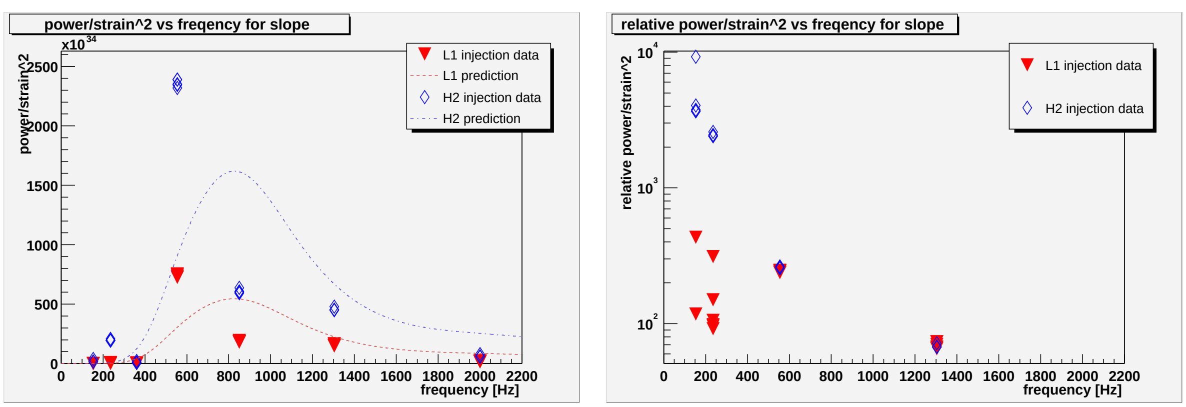


Figure 41: *Comparison between the predicted* $\frac{power}{strain^2}$ *and the* $\frac{power}{strain^2}$ *of the sine-gaussian injected events for* **TFCLUSTERS**. *Each visible marker actually corresponds to 4 injections with different strain - they are just too close to be distinguishable.*



(a)
absolute

(b)
relative to TFCLUSTER

Figure 42: $\frac{power}{strain^2}$ *of the sine-gaussian injected events for* **slope**:
(a) absolute; the predicting curve is for TFCLUSTERS, scaled by 100 to be visible.
(b) normalized by the power predicted for TFCLUSTERS, i.e. the points reflect the true frequency dependence of the slope algorithm.