

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

Technical Note **LIGO-T060112-00-Z** May 30, 2006

WaveBurst
S5 version

S. Klimenko (UF), I. Yakushin (LLO), G. Mitselmakher (UF)

Distribution of this draft:

LSC

This is an internal working note
of the LIGO Project.

California Institute of Technology	Massachusetts Institute of Technology
LIGO Project - MS 18-34	LIGO Project - MS 20B-145
Pasadena, CA 91125	Cambridge, MA 01239
Phone (626) 395-2129	Phone (617) 253-4824
Fax (626) 304-9834	Fax (617) 253-7014
E-mail: info@ligo.caltech.edu	E-mail: info@ligo.mit.edu

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

Contents

1	Introduction	2
2	Meyer wavelet	3
3	Linear Predictor Filter	11
3.1	Symmetric filters	11
3.2	LPE filter in wavelet domain	11
3.3	Effect of LPE on the GW waveforms	11
4	Improvements to whitening procedure	16
5	Bug in the rank calculation	16
6	Modifications to the production script	16
7	Comparison of S4 and S5 waveburst version results on the first few months of S5 run	18
8	References	24

1. Introduction

The first version of waveburst algorithm, described in [1] and [2], was implemented in LAL and LDAS and used for the S2 burst analysis [3], [4].

For the S3 [5] and S4 [6] analysis the same algorithm was implemented in a C++ library called Wavelet Analysis Tool (WAT) which is part of the DMT package. The pipeline itself is implemented as a ROOT [7] script and the ROOT jobs ran under Condor.

For the S4 run [6],[8], we upgraded the algorithm to perform analysis at several time-frequency resolutions [9], which made it more sensitive to a wider class of burst signals.

For the S5 run [10] further upgrades are made: the Symlet_60 wavelet is replaced by Meyer_1024 wavelet and the linear predictor error (LPE) filter is used for data conditioning. The Meyer_1024 wavelet has much longer filter (1024 coefficients) which significantly reduce spectral leakage between wavelet layers. The LPE filter removes significant fraction of energy from power lines and violin modes. It improves the detection efficiency for burst signals in the vicinity of these lines. Both modifications also improve the detection efficiency at low frequencies (64-100Hz).

Since the S4 waveburst version of waveburst is described in details in [9] , in this document we only describe the difference between S5 and S4 versions rather than presenting the whole algorithm from the beginning. However, to make clear what part of the pipeline is modified, we list below its main stages:

- Apply wavelet transformation to GW channel from each detector to obtain time-frequency scalograms of data with a certain time-frequency resolution;
- Apply data conditioning: whitening, variability corrections, etc;
- Select 10% of loudest (black) pixels in each detector;
- Compute rank of sdelected pixels;
- Compare scalograms for each pair of detectors and select only those pixels that are in time-frequency coincidence (within some window) in all detectors;
- Do clustering of the remaining pixels separately in each detector;
- Compute properties of clusters (time: central, start, stop, duration; frequency: central, minimum, maximum, bandwidth; hrss, SNR, etc) in each detector;
- Use non-parametric significance (gs) of a multidetector trigger to select candidate GW events.
- Apply amplitude cut which requires that hrss of a trigger detected in H1 is no more than twice different from hrss of the coincident trigger reconstructed in the H2 detector.

We typically set the geometric significance threshold so that the background rate at the end of the waveburst pipeline is about $20\mu Hz$. The background is estimated by running the pipeline on time shifted data (we use 100 time lags shifting the LLO and

LHO detector with respect to each other). After that the r-statistics cut [11] and vetoes [12] are applied to reduce the rate down to about $0.1\mu Hz$.

2. Meyer wavelet

The bandwidth of wavelet layers depends on the wavelet decomposition depth. A single wavelet decomposition step divides an input time series $X(t)$ into two wavelet series $X_a(t)$ (low-pass filtering) and $X_d(t)$ (high-pass filtering). The sampling rate of $X(t)$ is twice larger than one for the wavelet series. The $X_a(t)$ represents the low frequency sub-band (approximation layer) and the $X_d(t)$ represents the high frequency sub-band (detailes layer) of the original time series. It can be seen by zeroing either $X_a(t)$ or $X_d(t)$ data and then applying the inverse wavelet transformation which produces time series $Y_a(t)$ ($X_d = 0$) or $Y_d(t)$ ($X_a = 0$). In case when $X(t)$ is white gaussian noise and the Haar wavelet is applied, the corresponding power spectra of $Y_a(t)$ and $Y_d(t)$ are shown in Figure 1.

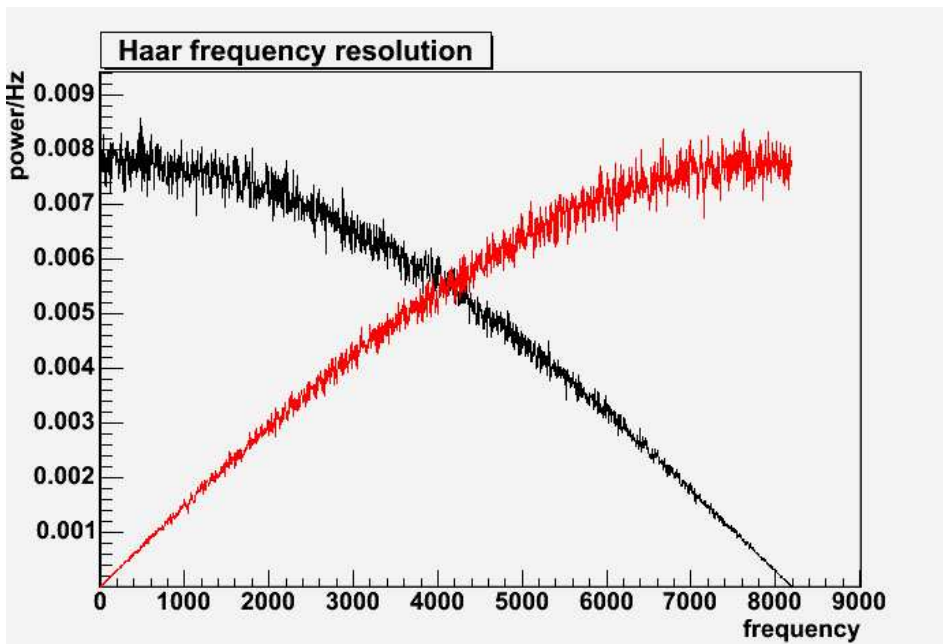


Figure 1. Power spectra $Y_a(t)$ (black) and $Y_d(t)$ (red) for Haar wavelet.

One can see that there is a significant spectral leakage between the frequency bands, which makes Haar a poor wavelet for time-frequency analysis.

Applying the wavelet decomposition steps, formally we can divide data into sub-bands with a desirable frequency resolution. This is the *transformation frequency resolution* (TR). It should be distinguished from the *wavelet frequency resolution* (WR), which is defined by spectral leakage between wavelet sub-bands (see Figure 1). In turn the spectral leakage depends on the length of the wavelet filter.

Figure 2 shows the spectral leakage from the approximation wavelet layer for Haar, Symlet_16 and Symlet_60 (where 16 and 60 denote the number of the wavelet filter coefficients). As one can see, the longer is the filter impulse response, the lower is the spectral leakage between the wavelet sub-bands. For example, also see the PSD of approximation signal after 2 wavelet decomposition steps (Figure 3).

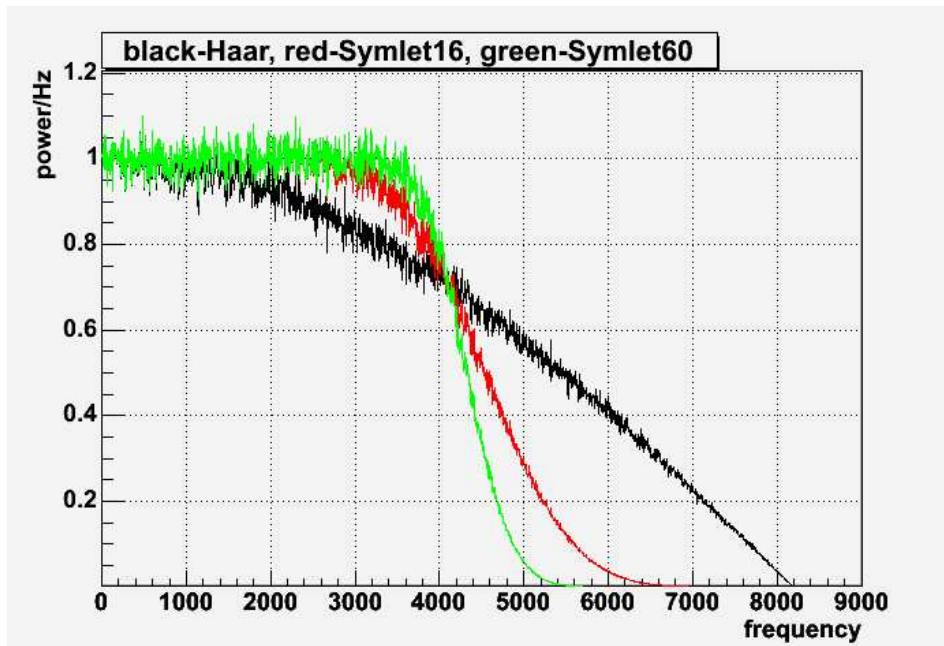


Figure 2. Comparison of spectral leakage for Haar and Symlet wavelets after one decomposition step

In general, given a time series with sampling rate f and a wavelet with filter length N , the actual wavelet frequency resolution is $WR \sim f/N$. This is the reason why in the S4 analysis we used Symlet wavelet with the filter length 60. But even in case when Symlet_60 wavelet is applied to time series with $f = 4096Hz$, the actual wavelet frequency resolution is $WR \sim 64Hz$ which is not adequate for detection of long and narrow-band GW signals like high Q sine-gaussian signals. WaveBurst performs analysis at different TR resolutions with the best one of $TR = 8Hz$. Obviously the wavelet filter length of 60 is not adequate for this resolution and longer filters should be used. Unfortunately there is no known Symlets with filter length greater than ~ 64 . Long Symlet filters can be calculated in principle with the algorithm suggested by Daubechie, but we did not see any references to long Daubechies/Symlet filters so far. The calculation of long filters involves the ratios of large numbers (like factorials) and it is quite nontrivial computational task.

Since long Symlet filters are not readily available, we suggest to use Meyer filters instead, which can be easily calculated for filter length < 1000 . Another small advantage of the Meyer filter is that it is perfectly symmetric (linear phase highpass and lowpass filters). For reference on Meyer wavelets see [13]

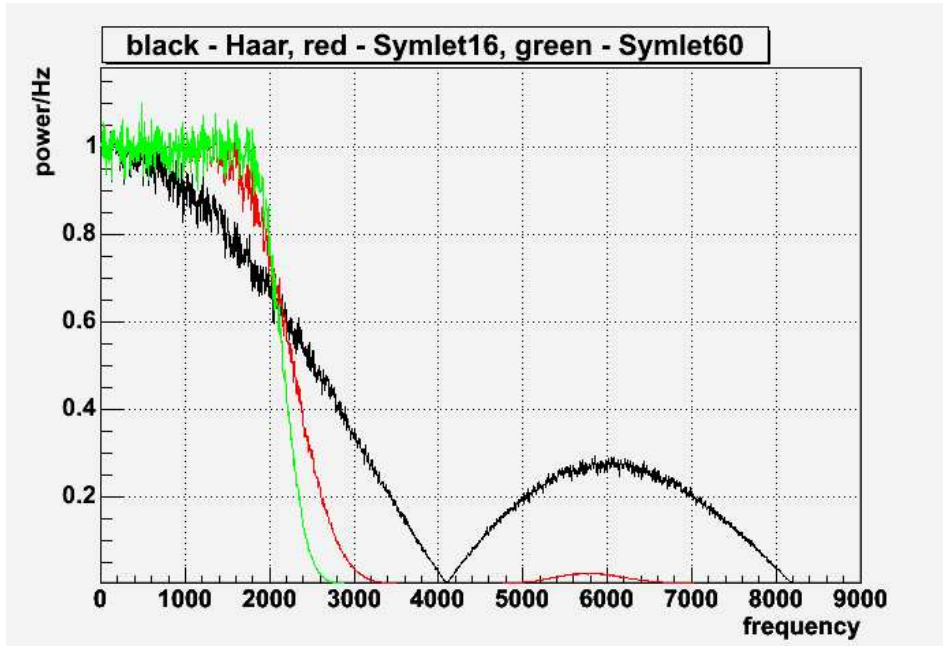


Figure 3. Comparison of spectral leakage for Haar and Symlet wavelets after two decomposition steps

The following figures show the frequency resolution and spectral leakage for one step (Figure 4) and three step (Figure 5) Meyer_1024 wavelet transformation in comparison with the Haar and Symlet_60 wavelets. Obviously the Meyer_1024 wavelet is much better bandpass filter than the Haar and Symlet_60 wavelets.

Note, the wavelet functions (Figure 6) for Meyer and Symlet_60 wavelet are very similar. It is not surprising because any symmetric wavelet becomes a Shannon wavelet in the limit of long filters ($N \rightarrow \infty$). A disadvantage of the Meyer filters is that for local support they have to be truncated. The result of this is that the Meyer wavelets are "approximately" orthonormal. It means that the Parseval identity is approximate and the forward-inverse wavelet transformation does not reproduce the original time series "exactly" as for the Symlet wavelets. From the other side the Meyer filters can be constructed in such a way that the Parseval identity holds with better than 0.01% accuracy, which is more than adequate for the analysis. The histograms of the residual signal (original $X(t)$ - forward-inverse $X(t)$) are shown for Meyer_1024 (Figure 7) and for Symlet_60 (Figure 8) wavelets.

Figure 9 (see also zoomed plot on Figure 10) demonstrates how wavelet works as a bandpass filter. For this example a white Gaussian noise (sampling rate 4096Hz) was transformed with 7-step wavelet transformation, which corresponds to the $TR = 16\text{Hz}$. Then all wavelet layers except for the wavelet layer 33 representing frequency band $528 - 544\text{Hz}$ were zeroed and the inverse transformation was applied. Performing FFT of the obtained signal we expect to see a bandlimited spectra with power in $528 - 544\text{Hz}$ band. The Haar wavelet exhibits a significant long-range spectral leakage. The Haar filter

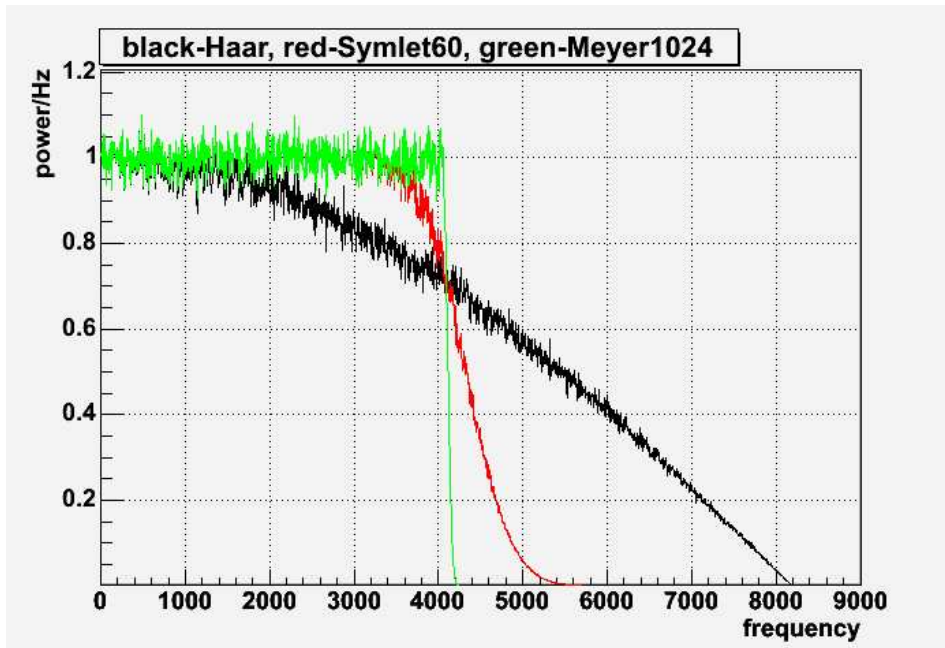


Figure 4. Comparison of spectral leakage for Haar, Symlet_60 and Meyer_1024 wavelets after one decomposition step

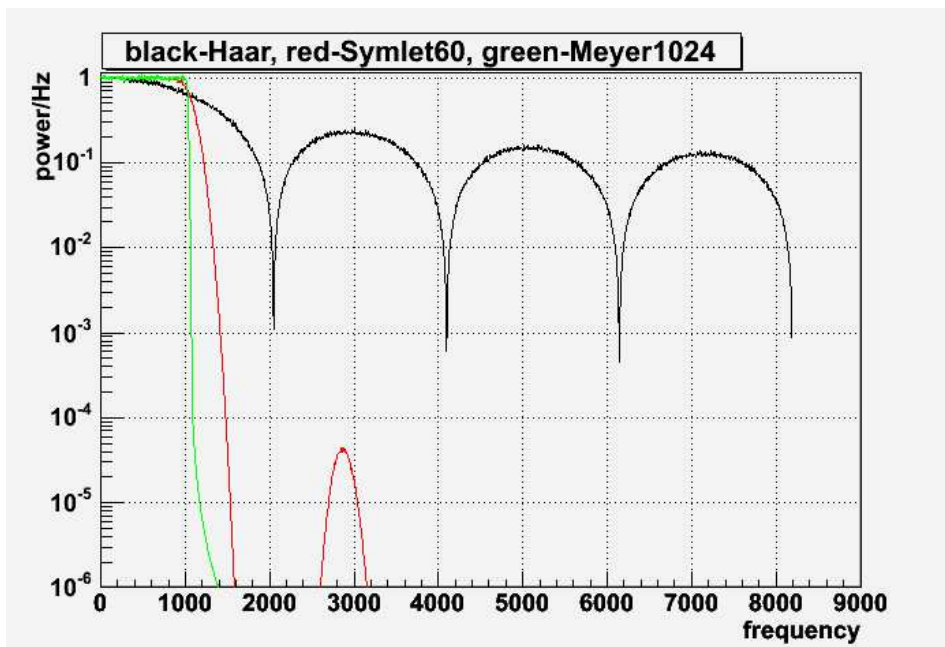


Figure 5. Comparison of spectral leakage for Haar, Symlet_60 and Meyer_1024 wavelets after three decomposition steps

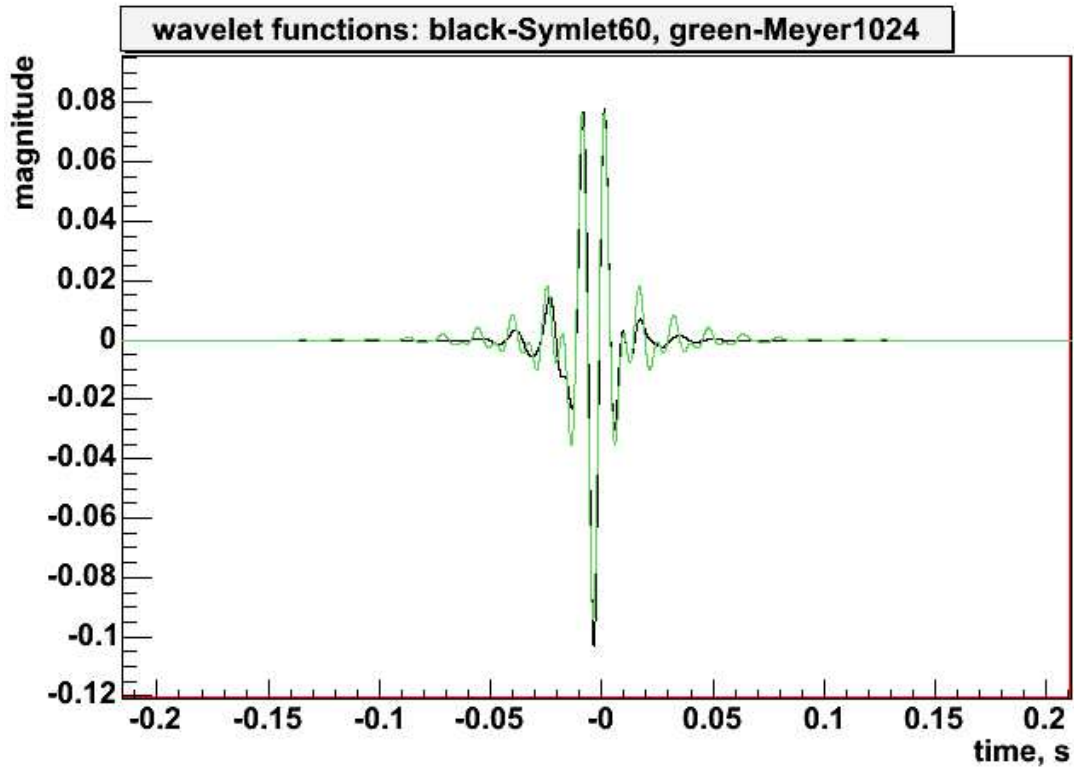


Figure 6. Wavelet functions for Symlet_60 and Meyer_1024 wavelets

is too short and it is not a good tool for this task. The `Symlet_60` does not have any long-range leakage, but it has a significant leakage into neighbor bins (within a window defined by its $WR \sim 64Hz$ resolution). The `Meyer_1024` wavelet shows no significant leakage.

Use of the `Meyer_1024` wavelet improves the WB performance. The WB whitening is performed at $WR=4Hz$. A corresponding scalogram for L1 detector and `Symlet_60` transformation (Figure 11) shows a line at $\sim 70Hz$ which is in fact due to the spectral leakage from nearby $60Hz$ power line. The `Meyer_1024` wavelet localizes the line energy much better (Figure 12), which significantly improves the WB sensitivity at $70Hz$.

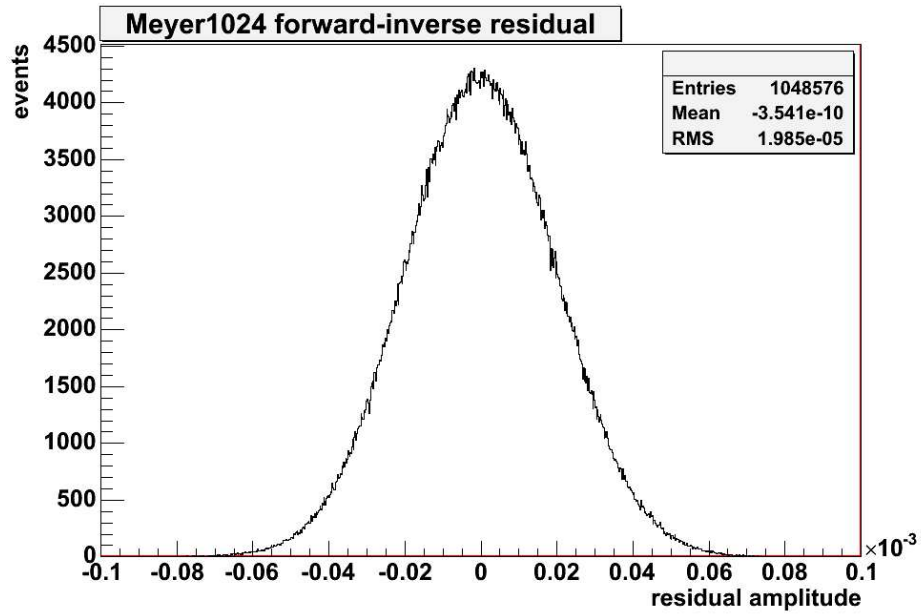


Figure 7. Meyer_1024 forward-inverse residual

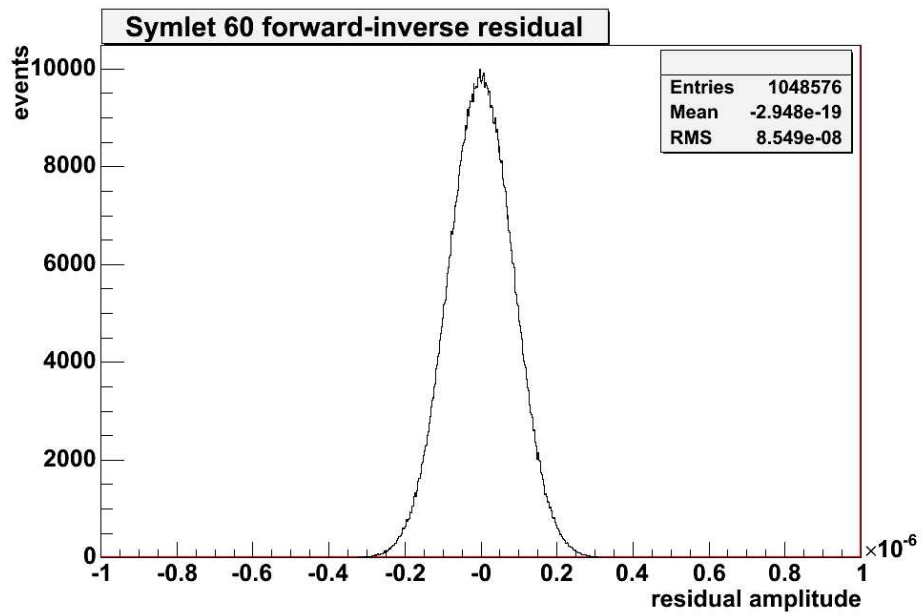


Figure 8. Symlet_60 forward-inverse residual

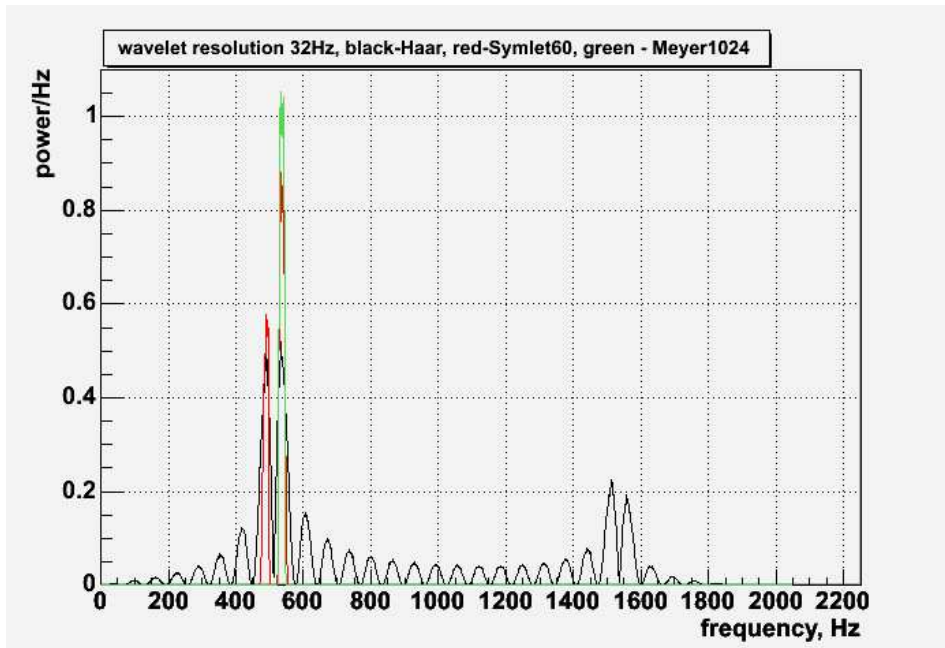


Figure 9. Wavelet resolution 32 Hz for Haar, Symlet_60 and Meyer_1024 wavelets

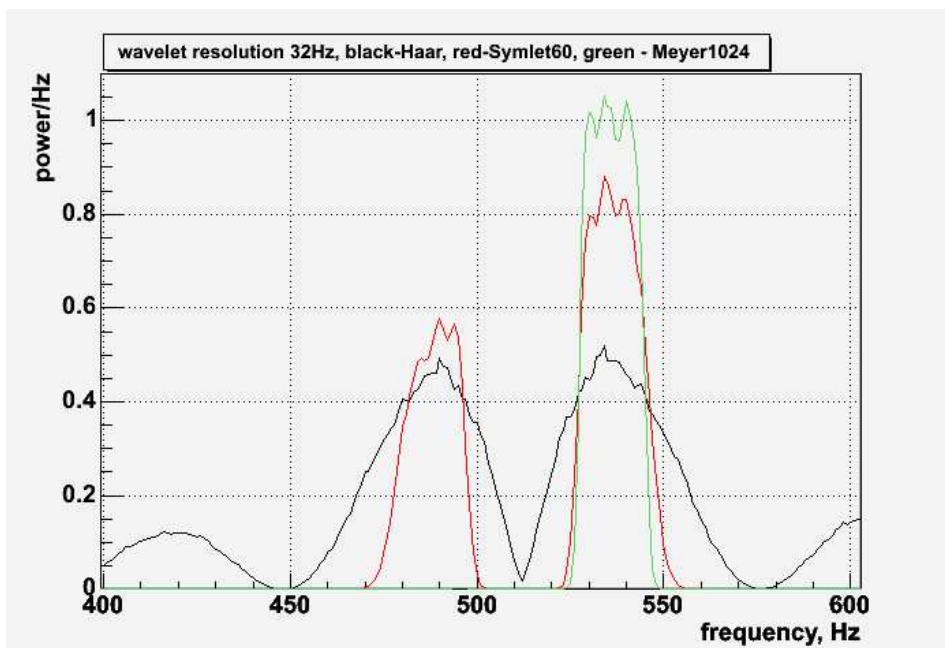


Figure 10. Wavelet resolution 32 Hz for Haar, Symlet_60 and Meyer_1024 wavelets, zoomed

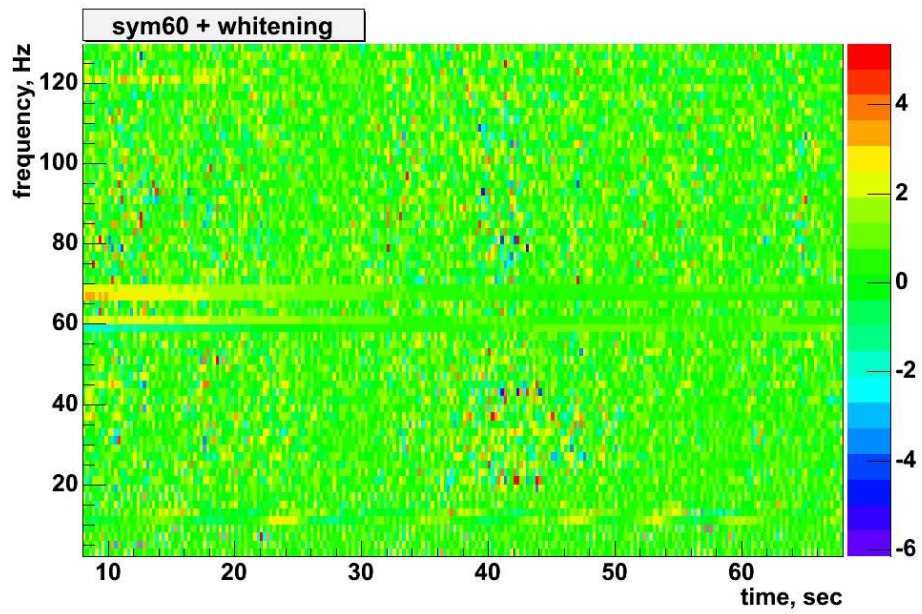


Figure 11. Scalogram with a $70Hz$ line using Symlet₆₀ wavelet

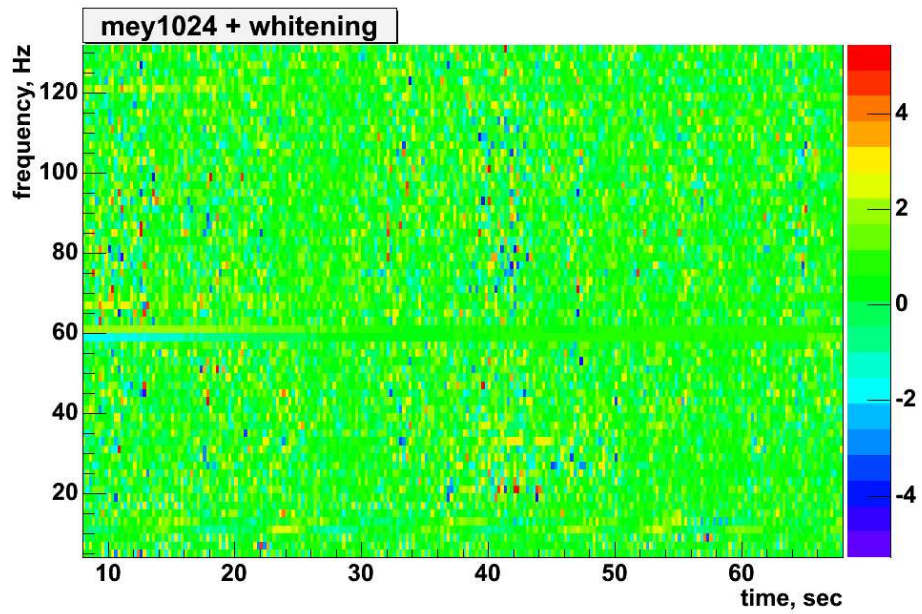


Figure 12. Scalogram with a $70Hz$ line using Meyer₁₀₂₄ wavelet

3. Linear Predictor Filter

During the S4 analysis we started observing that some data artifacts (like power lines) present a certain problem for the WB analysis. They were not a problem for S2 and S3 runs, but during the S4 run the detector sensitivity was considerably increased and the power lines started to affect the analysis. Therefore for the S5 analysis we have looked at different options for the line/artifacts removal. One of the options is the linear prediction error (LPE) filter.

The linear prediction error filters are used to remove "predictable" components of a given signal $x(t)$. They have been used before in LIGO data analysis [14] in time domain. In this case the output of the LPE filter is a whitened time series $x'(t)$. Unfortunately this approach can not be directly used in the WB analysis because:

- construction of symmetric LPE filters needs to be clarified,
- it interferes with the WB parameter reconstruction, such as estimation of the noise $rms(f)$ and signal hrss,
- there is also a question how LPE filter affects the GW signal waveforms.

3.1. Symmetric filters

For construction of symmetric LPE filters we follow the approach described in [15] The symmetric LPE filters can be constructed from the backward and forward LPE filters by using classical Levinson algorithm or the split lattice algorithm.

3.2. LPE filter in wavelet domain

Since each wavelet layer is a time series, rather than applying LPE filter to a time series $x(t)$, one can perform a wavelet decomposition $x(t) \rightarrow w(f, t)$ first and then calculate and apply LPE filter $F(f)$ individually to each wavelet layer. A set of filters $F(f)$ remove predictable components (like lines) in wavelet layers producing data $w'(t)$. A filtered time series $x'(t)$ can be reconstructed from $w'(t)$. The example PSD of filtered S4 data is shown in Figure 13. As one can see, when applied in wavelet domain the LPE filter removes data artifacts but preserves the power spectral density of the noise floor.

3.3. Effect of LPE on the GW waveforms

Lets consider a time series $x(t)$ with GW signal $h(t)$ in it: $x(t) = n(t) + h(t)$, where $n(t)$ is the noise. The LPE filter is constructed from an estimate of the autocorrelation function of the noise $a(t) = \sum[x(t) * x(t-T)]$. If the signal is quite strong (signal energy is comparable with the energy of the noise in the signal bandwidth) it will affect (or even dominate) the estimation of $a(t)$ and therefore when the corresponding LPE filter is applied it will partially remove the GW signal. It may affect the GW detection and the estimation of the GW amplitude. This is definitely a problem for strong injections. The signal cancelation also depends on the data length used for filter training. For

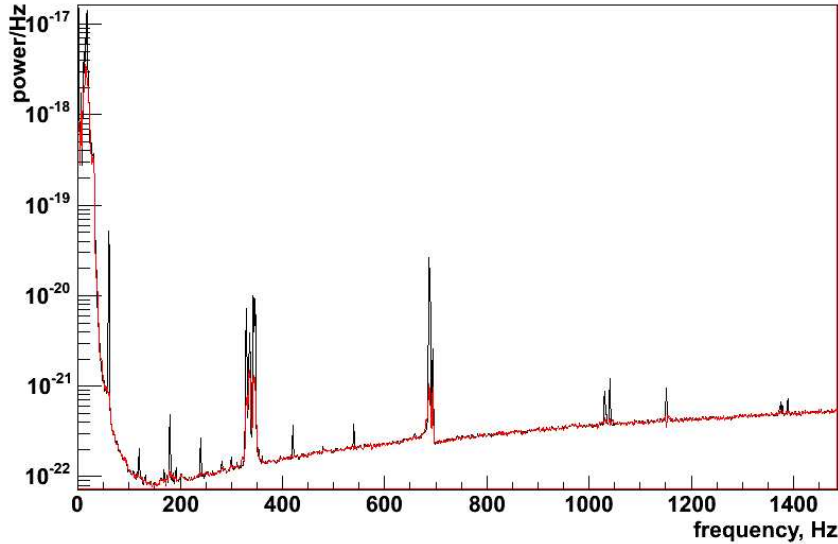


Figure 13. Power spectra of original (black) and LPE filtered (red) noise

example, Figure 14 shows how the LPE filter (the training length of 120 sec) affects the 270Hz sine-gaussian injection with $hrss = 10^{-21}$ (S4 data, L1 detector). The difference of $LPE(x(t) + h(t))$ and $LPE(x(t))$ is responsible for residual signal away from the injections (Figure 15).

To reduce the effect of the injected signal on the $LPE(x(t) + h(t))$ filter construction, 5% of loudest samples are removed from the data before training the filter. See corresponding injected signal (Figure 16) and residual signal (Figure 17).

Normalized wavelet scalograms without LPE (Figure 18) and with LPE (Figure 19) demonstrate how the LPE filter is used for data conditioning. It removes most of the line structure present on the TF plot due to power lines and violin modes.

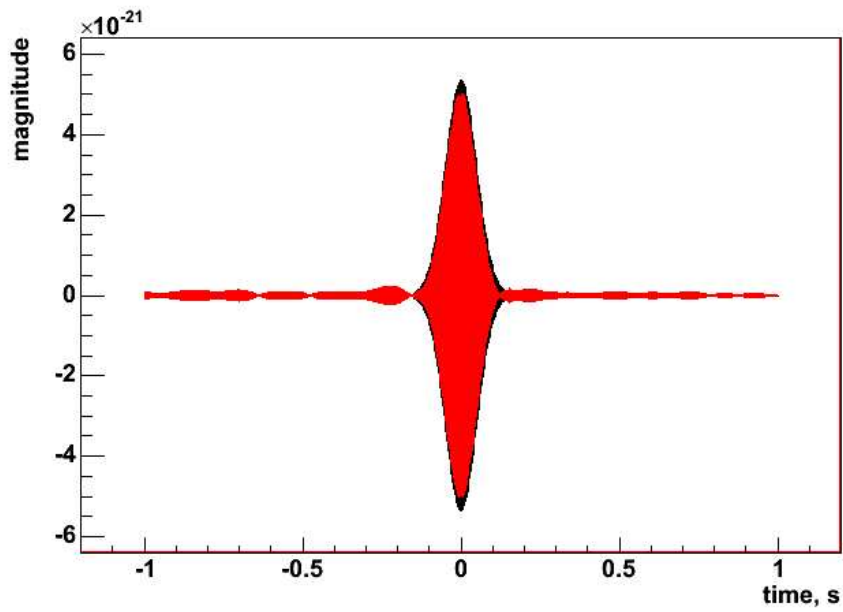


Figure 14. Effect of LPF on 270 Hz sine-gaussian. Black curve is an injection. The red curve is obtained as a difference between LPE filtered data with injection $x(t)+h(t)$ and LPE filtered data without injection $x(t)$. One can see that the red curve has lower amplitude, which means that the LPE filter trained on data $x(t) + h(t)$ cancels out some of the injected signal. The difference between $LPE(x(t) + h(t))$ and $LPE(x(t))$ is responsible for residual signal away from the injections.

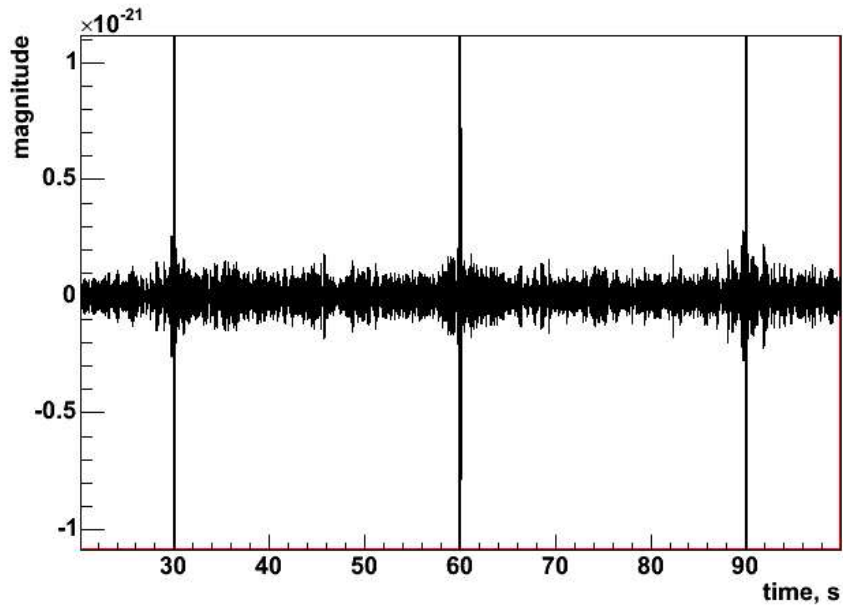


Figure 15. Residual signal

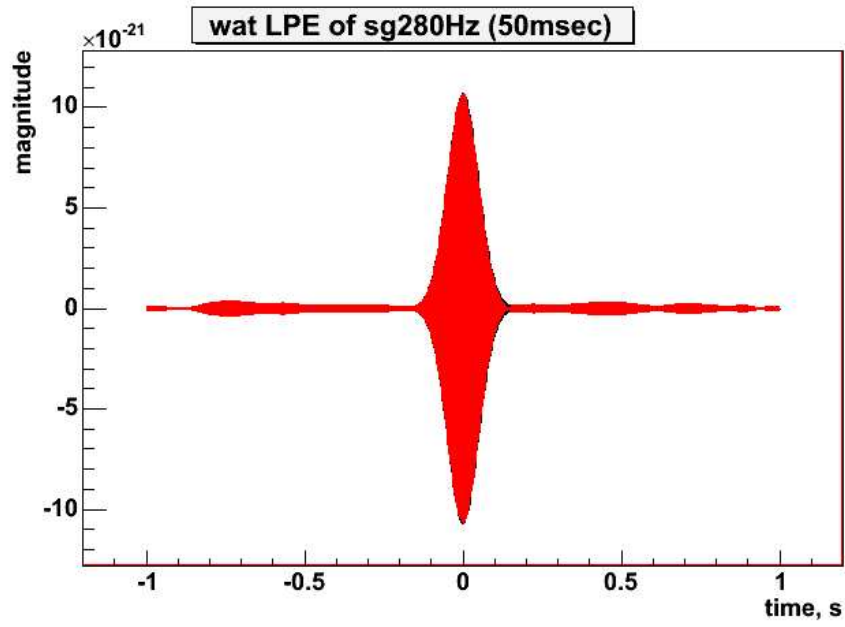


Figure 16. Injected signal with 5% of strongest samples removed for the LPE filter construction.

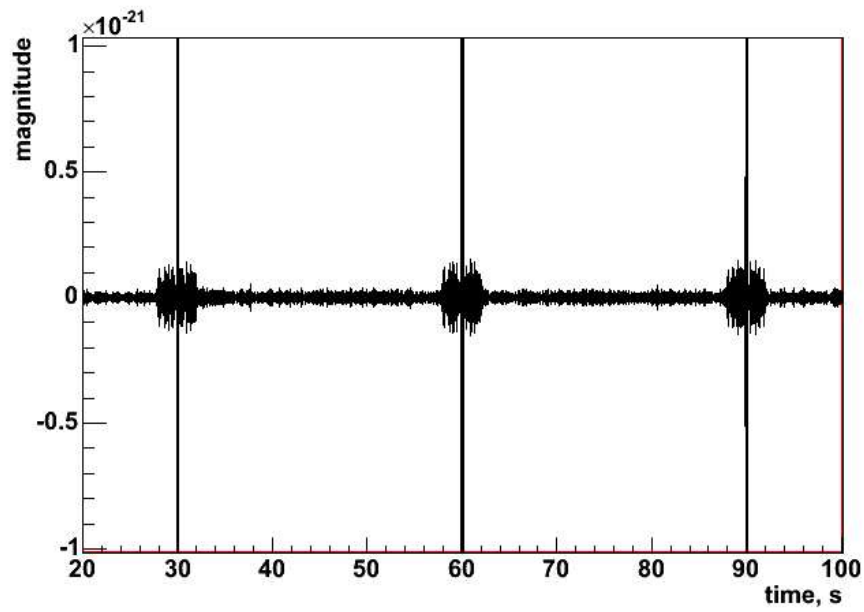


Figure 17. Residual signal with 5% of strongest samples removed for the LPE filter construction

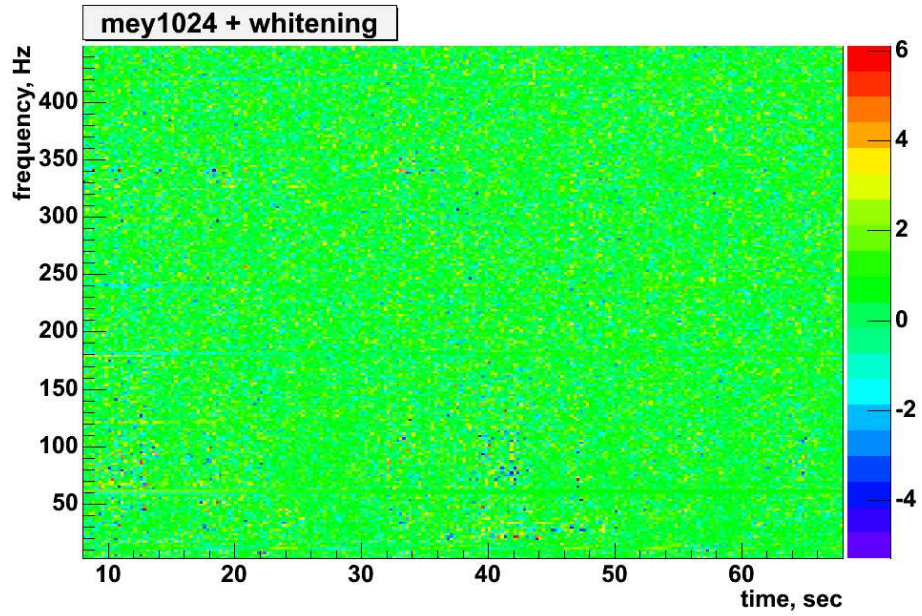


Figure 18. Normalized wavelet scalogram before the LPE filter is applied.

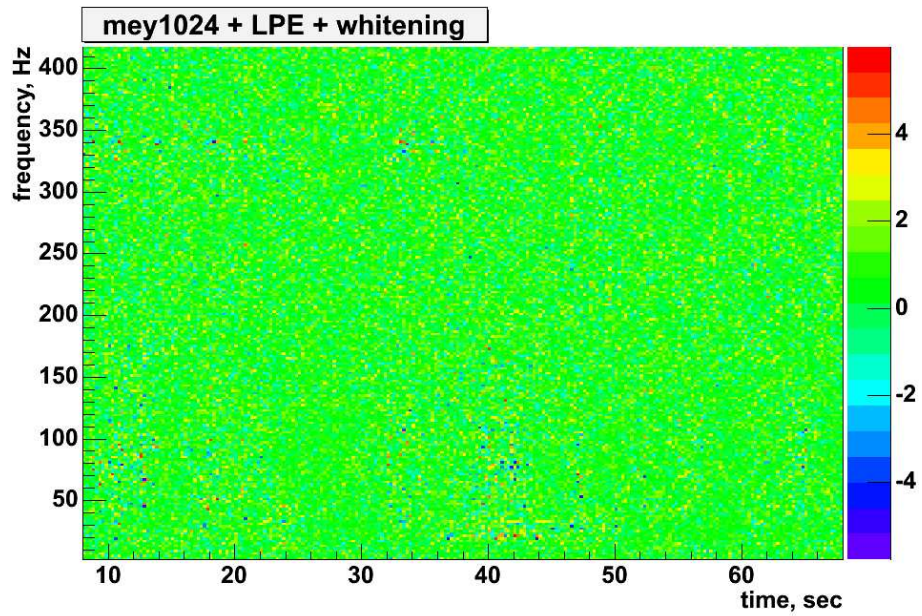


Figure 19. Normalized wavelet scalogram after the LPE filter is applied.

4. Improvements to whitening procedure

Whitening procedure was modified for more robust handling of non-standard time intervals. In the S4 waveburst version we observed some boundary artifacts (excess of triggers at the beginning or end of a job) in the presence of power line instabilities on a few segments, which duration was not divisible by 60 seconds. The corrections introduced in S5 version removed those artifacts. Also we introduced sliding window for the estimation of the detector noise. It considerably reduced the boundary artifacts (for comparison, see Figures 30 and 31).

5. Bug in the rank calculation

The rank statistic is calculated in the `rSignificance()` function. The algorithm is described in [9]. Given a data segment $x(t)$, the pixel rank is calculated with a running window T (5sec):

```
      |Ts|
****|*****xxxx*****|*****
<-  window T  ->
```

Rank is assigned to all pixels in the window, but only rank of pixels in the middle of the window (marked xxx) is saved. The length of the xxx section is equal to the window sliding step T_s . Sliding the window by T_s one can rank all pixels in the data set x . In this way each pixel is always ranked against neighbour pixels.

The S4 implementation of the algorithm is a little bit different from what is described above and what was intended - the ranked pixels are located not in the middle of the window, but in the beginning:

```
      |Ts|
****|xxxx*****|*****
<-  window T  ->
```

This bug is responsible for inconsistent wavelet scalograms for strong injections. Namely, the highest rank can be assigned to relatively weak pixels on the edges of injected signal. However, the correct picture would be when the highest rank is assigned to a pixel in the center of the injected signal. Figures 20 and 21 show wavelet rank scalograms before and after bug fix.

6. Modifications to the production script

The WB script was modified to avoid multiple calculation of the rank statistic for each time lag. Before the whitened TF series were shifted and then the rank statistic was calculated. Now the rank statistic is produced first and then the rank TF series are shifted in time. This modification reduced the WB run time by factor of 3.

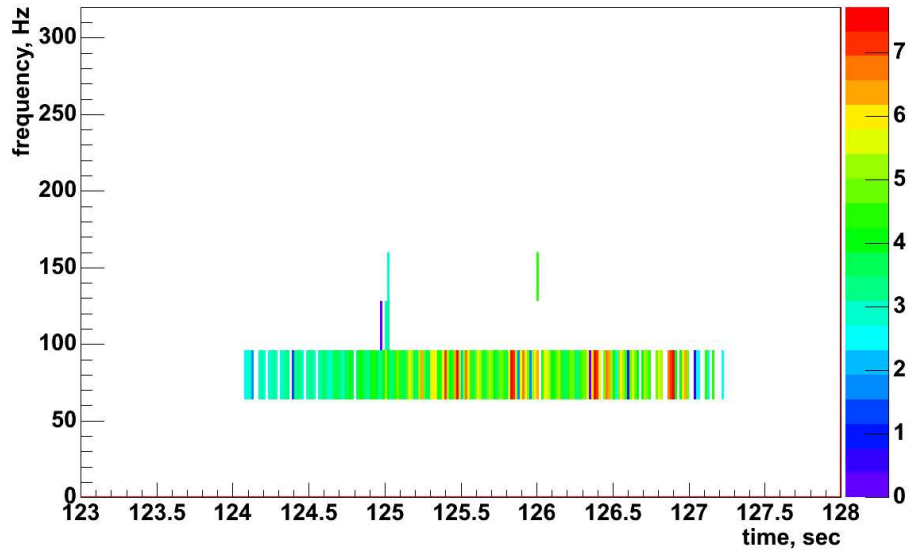


Figure 20. Incorrect rank assignment for long sine-gaussian injections.

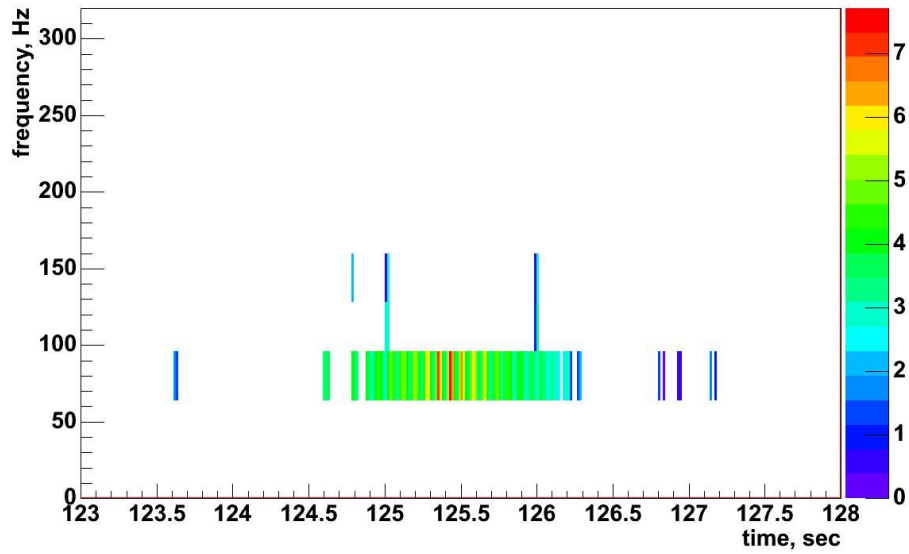


Figure 21. Rank assignment for long sine-gaussian injections after the bug fix.

7. Comparison of S4 and S5 waveburst version results on the first few months of S5 run

Figures 22 and 23 show comparable sensitivities for most of the frequencies for both searches at about the same background rate. Notice significant improvement in sensitivity for 70 Hz sine-gaussian due to better data conditioning (LPE) and longer wavelet filter.

Frequency reconstruction is somewhat better in the S5 version (Figure 24) than in the S4 version (Figure 25). Time reconstruction (Figures 26 and 27) is improved as well.

The hrss reconstruction is better for the S5 version (Figure 28) than for the S4 version (Figure 29) because less energy leaks outside of the detection band for signals with frequencies close to the detection band boundaries (70 and 2000 Hz sine-gaussians).

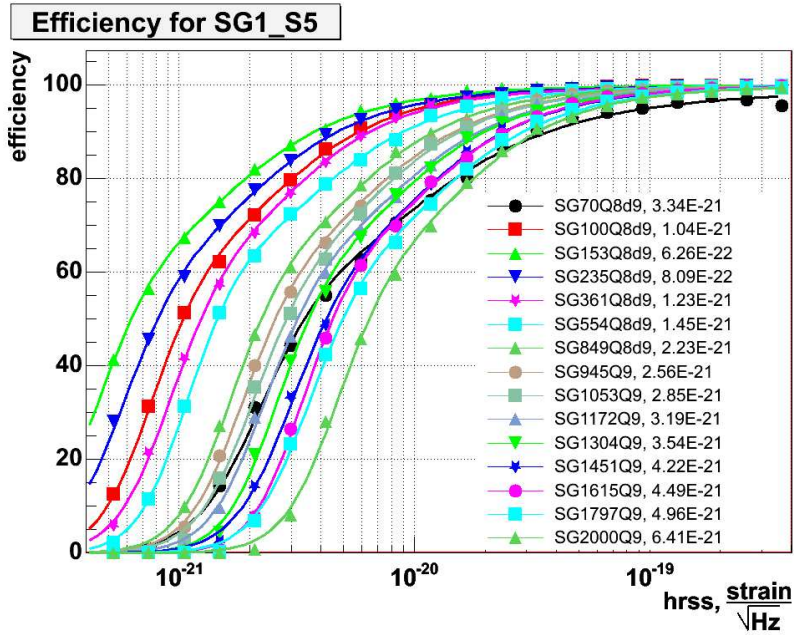


Figure 22. Efficiency for S5 version of waveburst, SG1_S5, $gc = 2.1$, background rate $18\mu\text{Hz}$.

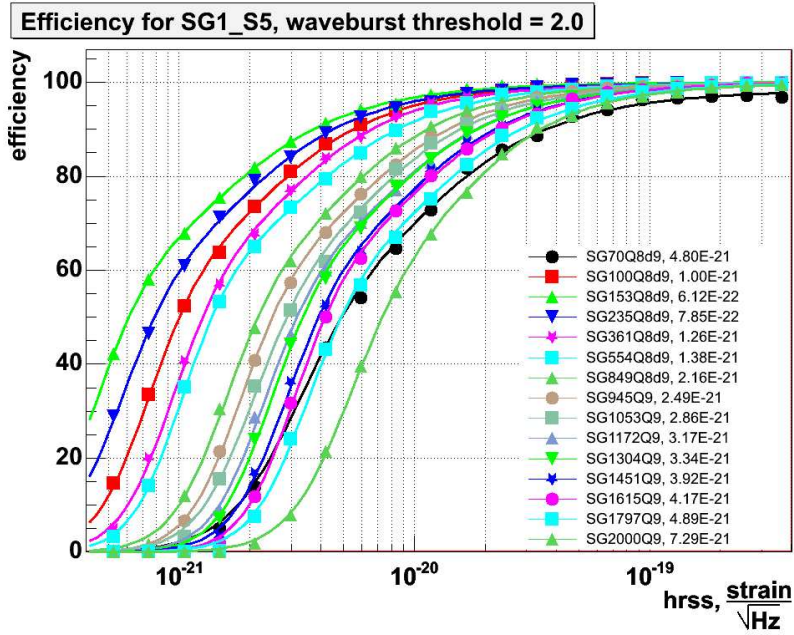


Figure 23. Efficiency for S4 version of waveburst, SG1_S5, $gc = 2.0$, background rate $19\mu\text{Hz}$.

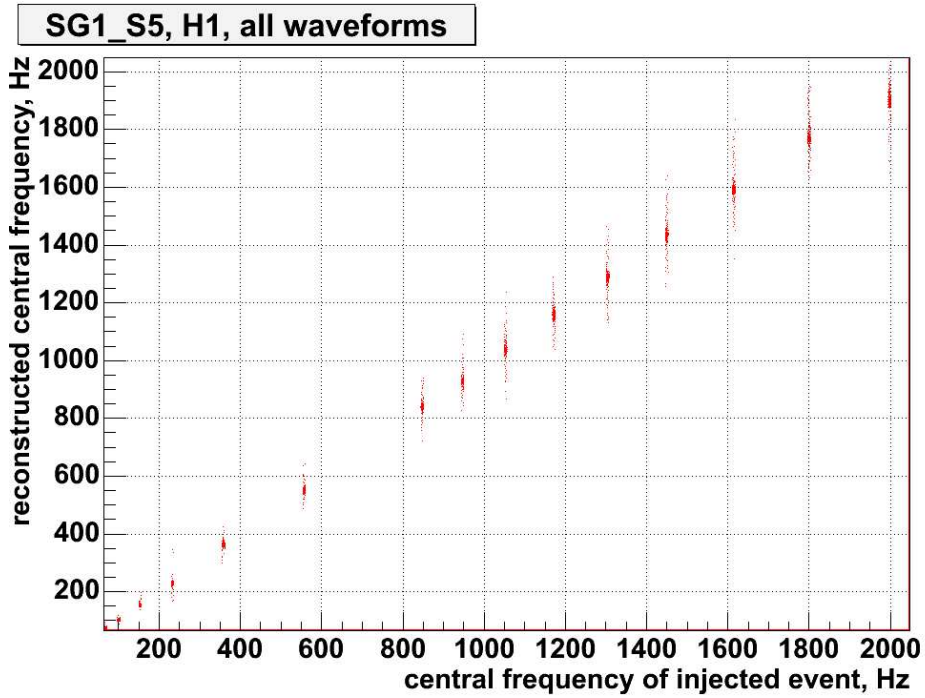


Figure 24. Frequency reconstruction, S5 waveburst version, SG1_S5.

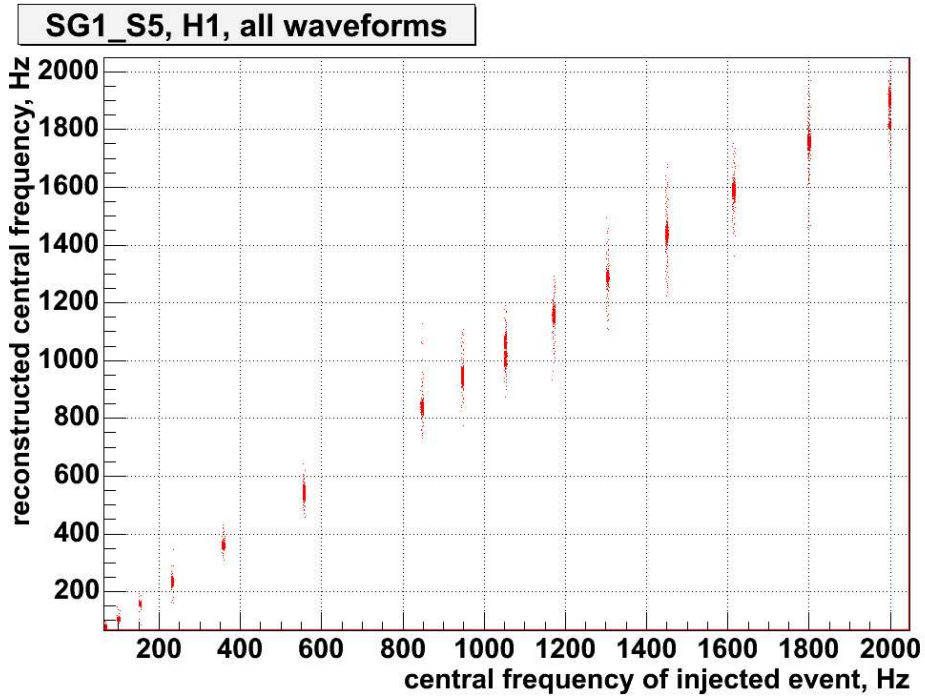


Figure 25. Frequency reconstruction, S4 waveburst version, SG1_S5.

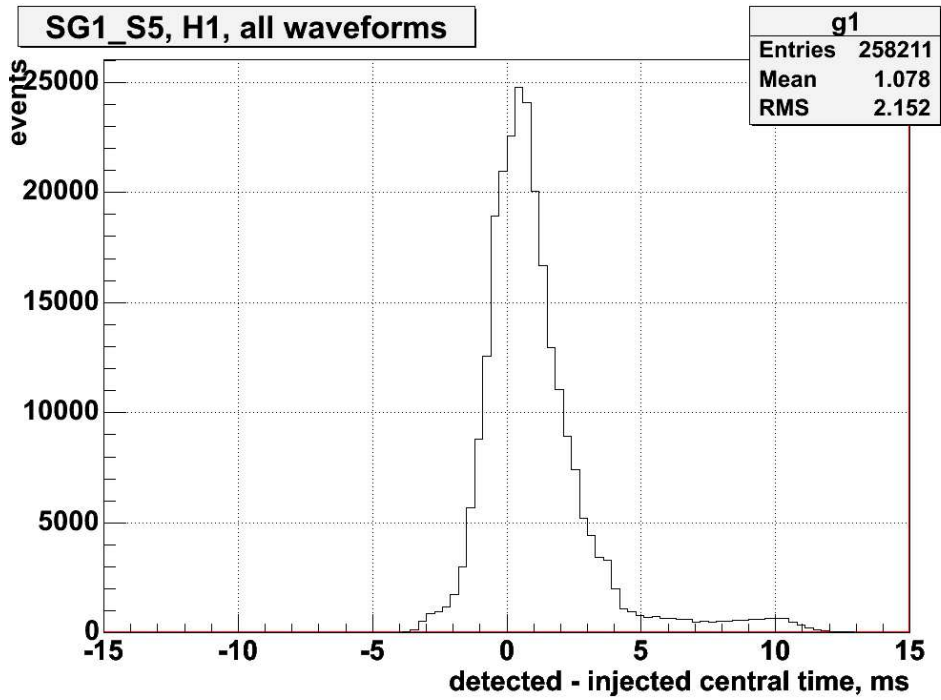


Figure 26. Time reconstruction, S5 waveburst version, SG1_S5.

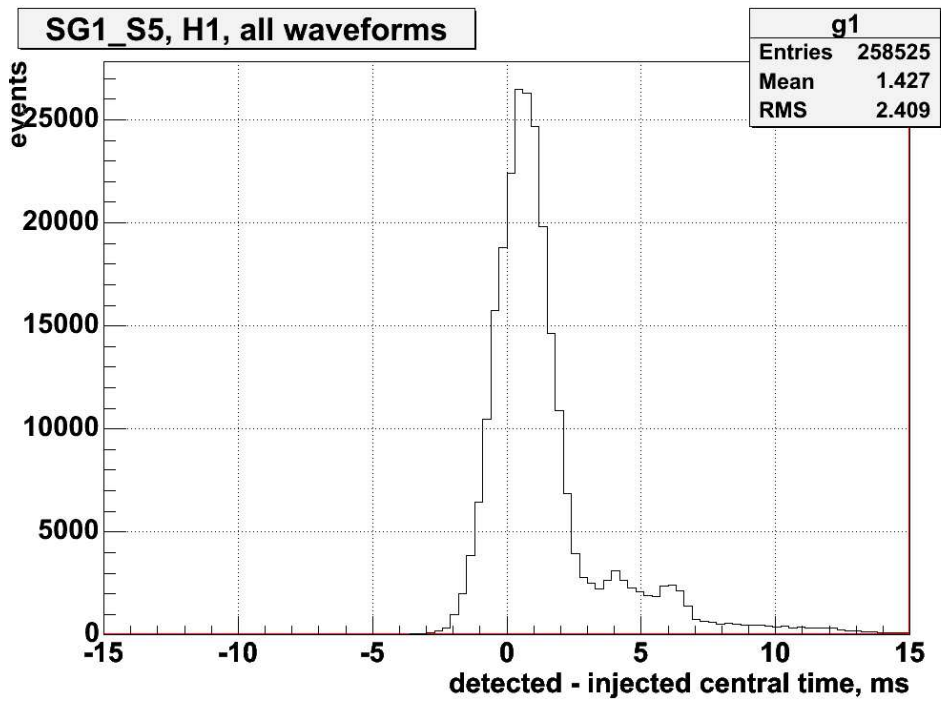


Figure 27. Time reconstruction, S4 waveburst version, SG1_S5.

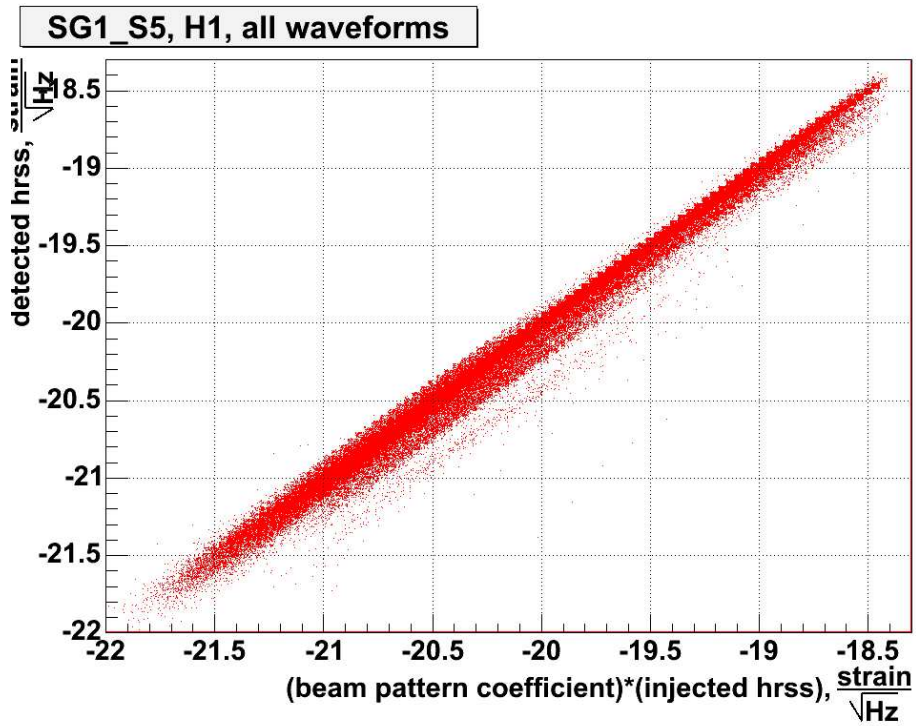


Figure 28. hrss reconstruction, S5 waveburst version, SG1_S5.

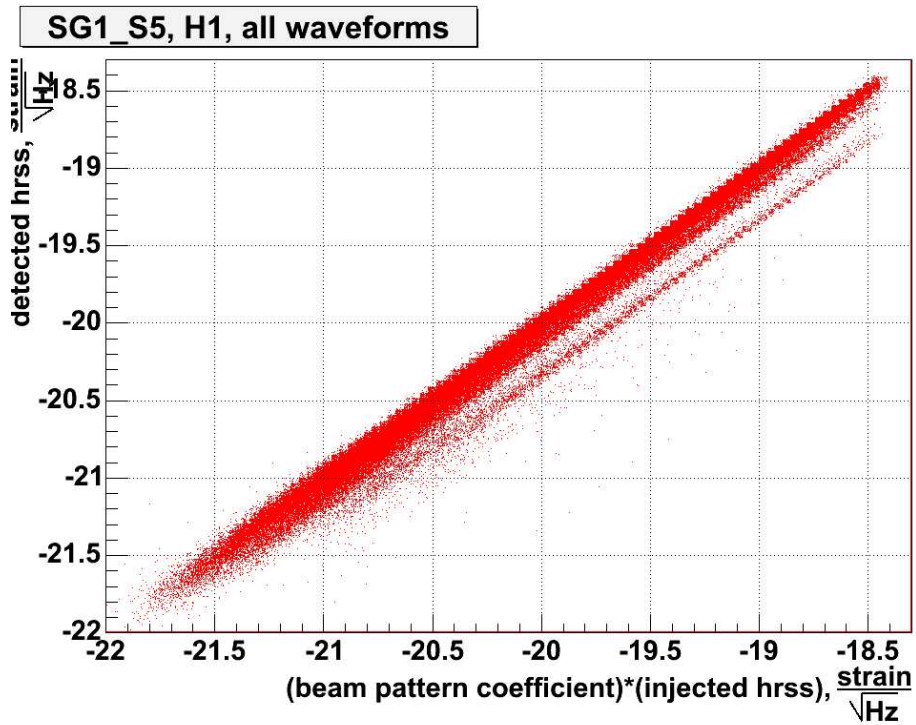


Figure 29. hrss reconstruction, S4 waveburst version, SG1_S5.

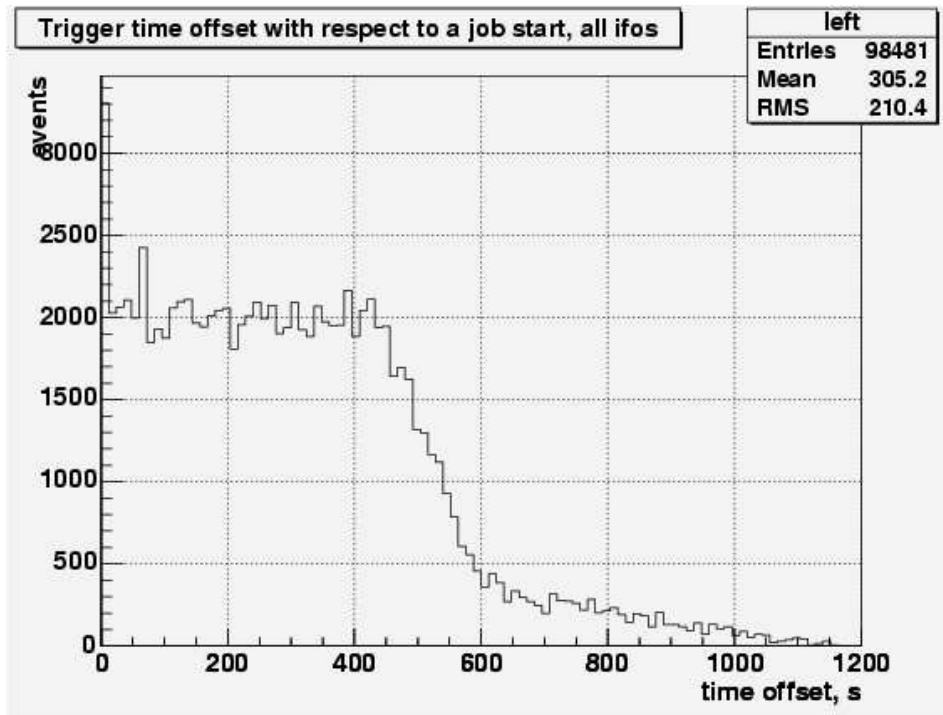


Figure 30. Boundary artifact, S4 waveburst version.

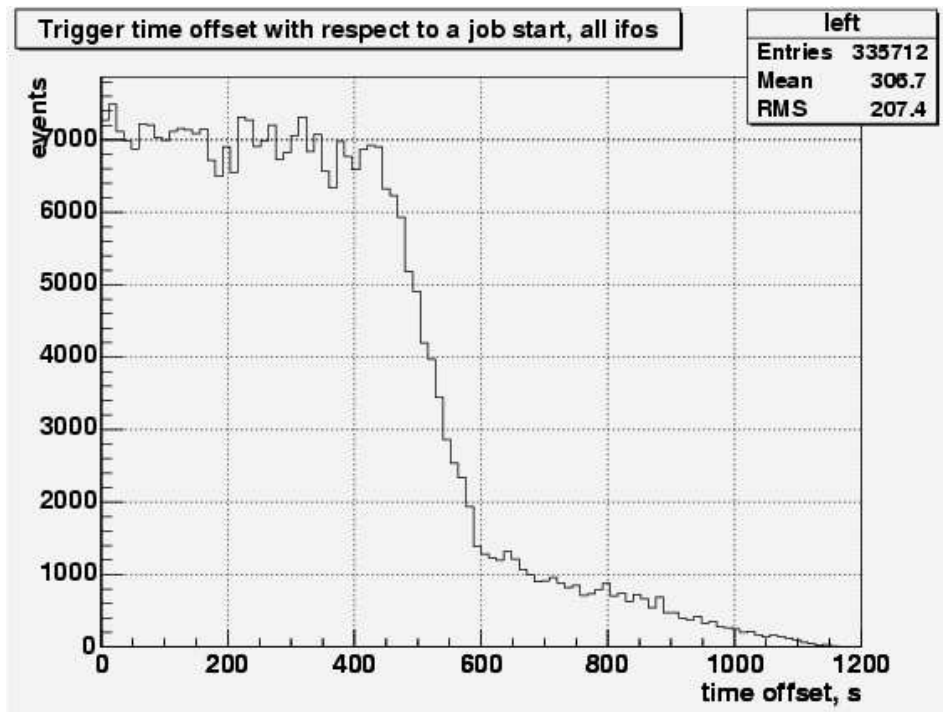


Figure 31. No more boundary artifact in S5 waveburst version.

8. References

- [1] S. Klimenko and G. Mitselmakher. A wavelet method for detection of gravitational wave bursts. *Class.Quant.Grav.*, 2004. Proceedings of GWDAW-8, 2003 conference.
- [2] S. Klimenko, I. Yakushin, and G. Mitselmakher. *WaveBurst*. LIGO note T040040-00-Z.
- [3] S. Klimenko, I. Yakushin, M. Rakhmanov, and G. Mitselmakher. Performance of the WaveBurst algorithm on LIGO data. *Class.Quant.Grav.*, pages 1685–1694, 2004. Proceedings of GWDAW-8, 2003 conference.
- [4] LIGO Scientific Collaboration. Upper limits on gravitational wave bursts in LIGO’s second science run. *Phys.Rev. D.*, 72, 2005.
- [5] LIGO Scientific Collaboration. Search for gravitational wave bursts in LIGO’s third science run. *Class.Quant.Grav.*, 23:29–39, 2006. Amaldi-6 conference proceedings.
- [6] Peter Shawhan for LIGO Scientific Collaboration. Search for Gravitational Wave Bursts in Data from the LIGO S4 Run. Presentation at APS meeting, April 2006. LIGO-G060041-01-Z.
- [7] ROOT web page. <http://root.cern.ch>.
- [8] I. S. Heng, L. Cadonati, S. Klimenko, P. Sutton, and I. Yakushin. LIGO-GEO S4 Analysis Update. LSC Meeting, November 19-20, 2005, Cambridge MA. LIGO-G050627-00-Z.
- [9] S. Klimenko, I. Yakushin, and G. Mitselmakher. *WaveBurst version 5*. LIGO-T050222-00-Z.
- [10] Igor Yakushin for LIGO Scientific Collaboration. Search for Gravitational Wave Bursts in LIGO’s S5 run. Presentation at APS meeting, April 2006. LIGO-G060170-00-Z.
- [11] L. Cadonati. Coherent waveform consistency test for LIGO burst candidates. *Class.Quant.Grav.*, 21(1695-1704), 2004.
- [12] Alessandra Di Credico. Gravitational wave burst vetoes in the LIGO S2 and S3 data analyses. *Class.Quant.Grav.*, 22:1051–1058, 2005.
- [13] B. Vidakovic. *Statistical modeling by wavelets*. 1999.
- [14] S. Chatterji. Data conditioning for gravitational wave burst analysis. LSC Meeting, August 18th - 21st, 2003, Hannover, Germany. LIGO-G030439-00-Z.
- [15] P. Delsarte and Y. Genin. On the splitting of classical algorithms in linear prediction theory. *IEEE*, ASSP-35(5), 1987.