Median based noise floor tracker (MNFT) : robust estimation of noise floor drifts in LIGO S1 data.

LIGO-T030019-00-Z

Soma Mukherjee
Max Planck Institut fuer Gravitationsphysik,
Am Muehlenberg 1, D-14476 Golm, Germany.
March 2003

Introduction

We are interested in tracking *temporal variation* in the statistical properties of the *noise floor* in a time series. The term noise floor is very loosely defined as that component of the power spectral density of a time series which would be left after removal of narrowband *technical* noise features (a.k.a., *line noise*). It is the component shown, for instance, in plots of interferometer design sensitivity. Further, we are interested in slow variations (*drifts*) of the noise floor. The drifts could be in both the shape as well as the level of the noise floor. We call the method that we have developed for this purpose **MNFT** (*Med*ian based *N*oise *F*loor *T*racker). The definition of the noise floor is, of course, fuzzy when there are technical line features that are not confined to narrow bands. However, we leave the matter of more rigorous definitions in the background for the moment.

There are two major problems in tracking the temporal behaviour of the noise floor:

1. Current interferometer data is dominated by high power line features in the time domain. So any statistical descriptor (mean, variance, etc.) estimated in the time domain actually measures the properties of the dominant components of the data and not the noise floor.
2. Besides line features, interferometer data also contains transients which can skew estimates made in the time domain. For example, transients coming at a high rate can temporarily masquerade as slow non-stationarity.

So to get at the noise floor time series, we need to eliminate or suppress the line features. After line removal, a measure of statistical property must be used that is sensitive to slow variations but not significantly affected by the presence of transients.

MNFT consists of the following steps:

1. Bandpass and resample the given time series *x(k)*, where *k* is the sample index. This is done in order to reduce the sampling rate and to restrict the analysis to the frequency band of interest. The resulting time series is called the *resampled time series r(k)*. (In this report, *x(k)* is also used to denote the k^{th} sample with the context clarifying the sense in which *x(k)* is being used.)
2. Construct an FIR filter that *whitens* the noise floor. See Section 0 for details. Whitening the data makes it convenient to compare with simulations. The output of the whitening filter is called the *whitened time series w(k)*.

3. Remove lines using a *notch filter.* A notch filter implemented as an FIR filter is a computationally inexpensive and a reasonably effective method to eliminate lines present in the data. The resulting data is called the *cleaned time series c(k)*.
4. Track variation in the second moment of *c(k)* using *a running median13*. The running median output is a time dependent estimate of the second moment and we denote this latter time series by *S(k).*
5. Obtain significance levels for the *sampling distribution* of the second moment via Monte Carlo simulations and set thresholds for detection of non-stationarity.

Details of each of the steps above are provided in the subsequent sections. The method is illustrated using the LIGO S1 data.

Data : lowpass and resample

We used LIGO S1 data from the Hanford 2k (H2) and the Livingston 4k (L1) interferometers to illustrate our method. The particular segments chosen here were also relevant to the externally triggered [5] search and indeed much of the motivation for this work derives from the particular needs of that analysis. The channel looked at for both the interferometers was L1 or H2:LSC-AS_Q (uncalibrated).
The sampling frequency for this channel was 16384 Hz. Figure (1) shows the timeseries envelope and the power spectrum of the data recorded between GPS times 715082714 and 715083088 s.

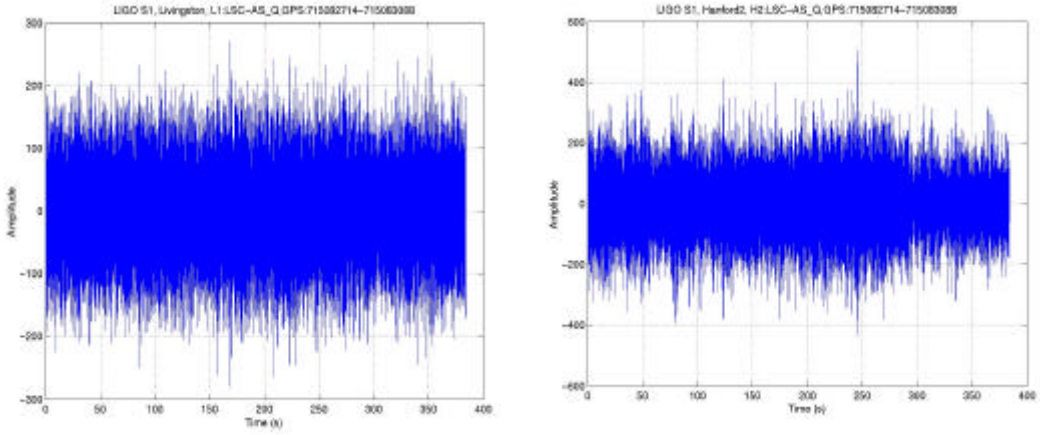

Figure 1. The timeseries of the raw data from the uncalibrated LSC-AS_Q channel of the L1 and H2 detectors.

The data was then lowpassed with a cutoff frequency of 4096 Hz and resampled down.

Whitening the data

A quiet stretch (without large transients) is identified visually and the PSD is computed. A *running median* is used on the PSD to obtain an estimate of the noise floor (in a sense this can be taken to be a *definition* of the noise floor). Since line features are *outliers* in the frequency domain series, a running median with a sufficiently large block size is a *robust estimate* of the noise floor PSD [1,8,10]. Figure (2) illustrates this process. This method of noise floor estimation has found many applications, viz. automated line detection [9].

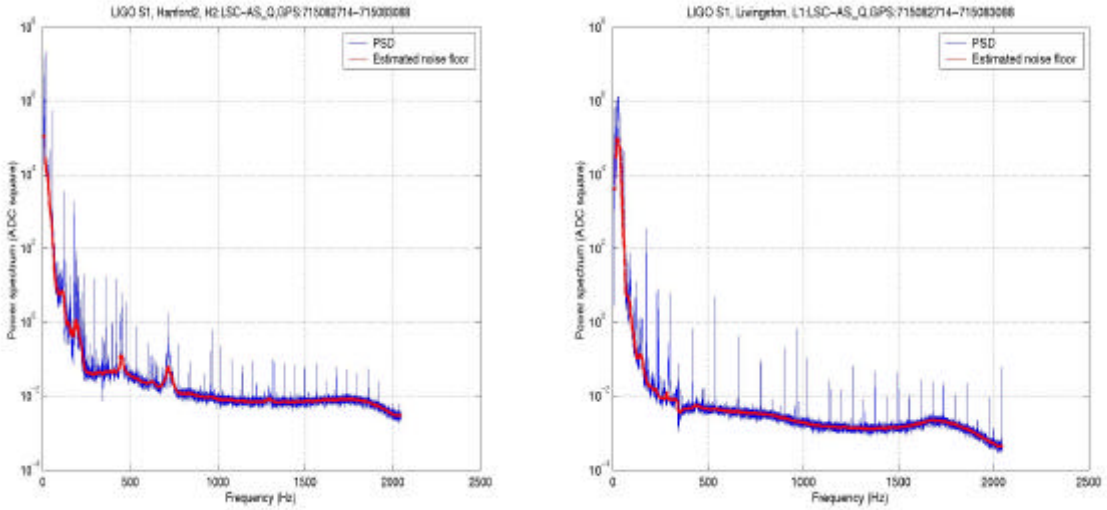

Figure 2. PSD of the lowpassed and resampled data from H2 and L1. The red curve shows the estimated spectral floor by the method of Running Median [1].

From the noise floor estimate $S(f)$, an FIR filter is constructed (using the least squares design technique) with a frequency domain transfer function that is $1/\sqrt{S(f)}$. This time domain filter is then used to filter the entire data stretch. If the data noise floor was stationary, the filtered noise floor should be a flat PSD for not only the segment that was used to construct the filter but for all other segments. Figure (3) shows the PSD of data after application of the whitening filter.

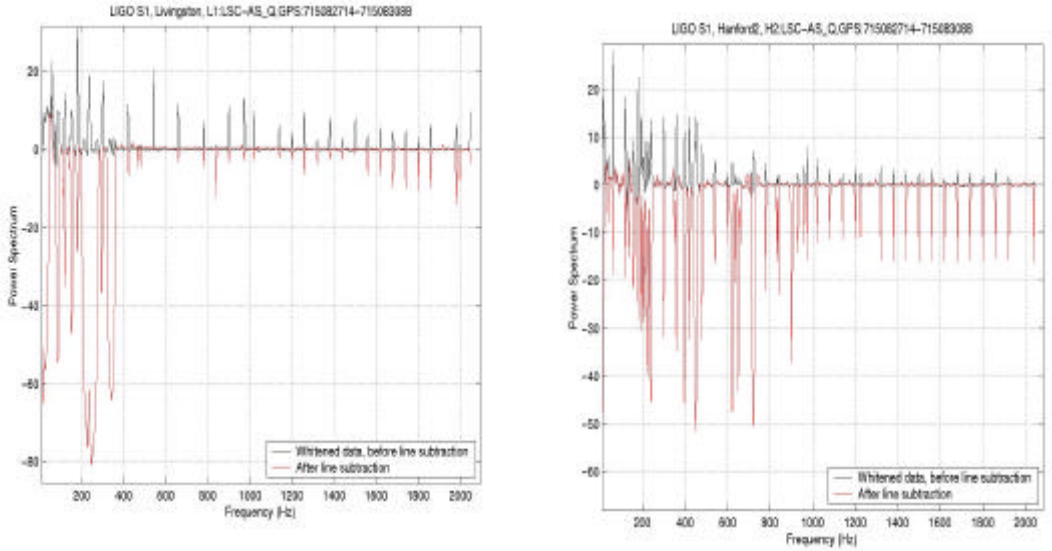

Figure 3. This figure shows the whitened and cleaned data from H2 and L1. The whitening is performed by constructing an FIR filter using the frequency domain transfer function based on the Running Median estimate of the spectral noise floor.

Whitening of interferometric data using adaptive filters has been discussed extensively in [2,4]. The adaptive whitening filters in [3] were based on time series modeling of the data using AR and ARMA models but the model fits were of an extremely high order (~1000) since line features were not removed. (It is also incorrect to include line features such as power lines into an AR or ARMA time series model since they may not meet the assumptions behind an AR/ARMA model.) In our case, we first directly get an estimate of the noise floor using the running median and then construct a time series model (the FIR filter design phase) that fits the noise floor. This yields a much simpler whitening filter. This way of obtaining the whitening filter can be easily made adaptive but we do not address it further here.

Line removal

Strong and steady line features in the PSD are tabulated in terms of their central frequencies and bandwidth at the noise floor level. A high order FIR notch filter is constructed using Matlab [6] with notches positioned at the central frequencies. Sometimes lines are very closely spaced and it would require an extremely large filter order to eliminate each one individually. Instead we eliminate the entire band containing such sets of closely spaced lines. Figure (5) shows the resulting PSD after notch filtering the whitened time series.

Why don't we use an arbitrarily high order filter in order to make very narrow notches and thus avoid taking out wide bands? The constraint comes from the fact that the notch filter also *broadens* transients in its input. So, if the rate of transients is high and/or if they have large amplitudes, the filtered output may show non-stationarity simply due to a high rate of broadened transients (see section on effect of transient broadening) overlapping with each other. Thus, there is a trade off between the order of the notch filter used and the prevention of false slow non-stationarity due to strong and/or high rate transients.

Tracking Non-Stationarity

The time series *c(k)* is an estimate of the whitened noise floor. For a stationary noise floor, the variance of c(k) would be a constant. For a non-stationary noise floor however the variance of c(k) would change in time. This variation could be caused by either a change in the shape of the noise floor or a change in its overall level or both. Thus, the simplest possible test for non-stationarity of the noise floor is then to estimate the variance of c(k) using a suitably defined estimator over a moving window in time. The resulting time dependent variance estimate can then be scrutinised for the presence of *change points* [7].

The standard estimator of variance is the MLE estimator [18]. However, the use of this estimator is not appropriate for the S1 data. This is because the cleaned time series still has transients present in it. Since a running variance estimate using fixed length block averages is essentially an FIR filter acting on the square of the data it can be affected significantly by transients especially if they are grouped closely.

In order to mitigate this problem we estimate the second moment of the cleaned time series by again using a running median on $c^2(k)$. Figure 4 illustrates the difference between running mean and running median estimation in the presence of transients. The running median is a more effective smoothing technique in this case because it eliminates the transients.

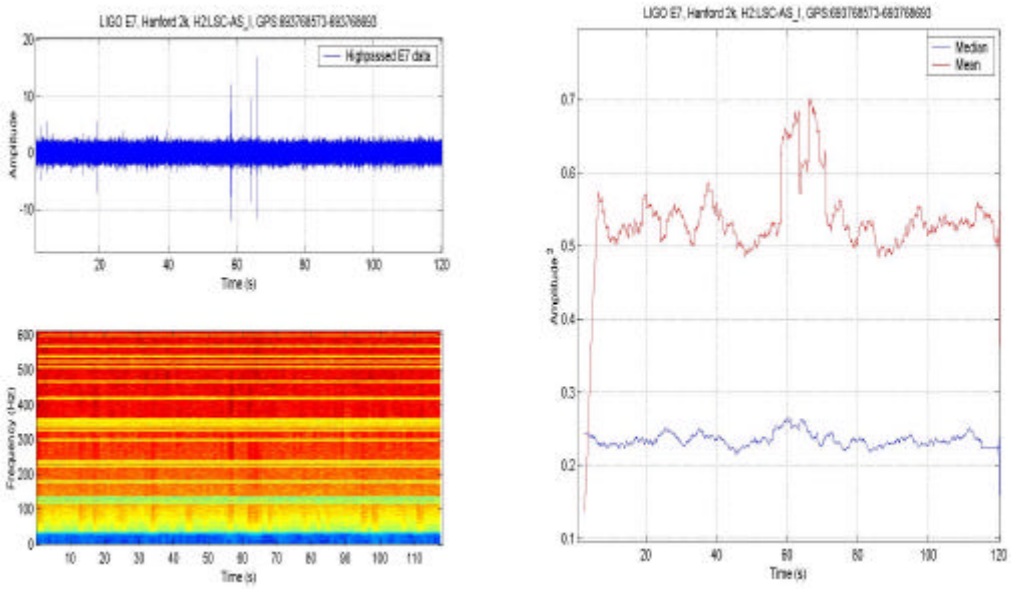

Figure 4. This figure illustrates the difference in the performance of Running Median as a smoothing technique as opposed to running mean. The specific example is drawn from a stretch of LIGO E7 data from the channel H2:LSC-AS_Q. The timeseries and the spectrogram on the left shows the presence of three very strong transients. The figure on the right shows that while the running mean retains much of the transient signature, the running median has smoothed it out. The offset seen in the figure is because the quantities are not normalized.

Simulation

We have used the running median estimate of the second moment in order to supress the effect of transients. However, the price paid for this is that the probability density for the resulting estimate cannot be computed in a simple way. This prevents us from easily computing the threshold for non-stationarity detection that corresponds to a given false alarm probability. We must, therefore, take recourse to Monte Carlo simulations in order to calculate detection thresholds.

The running median of $c^2(k)$ is an *estimate* of the second moment of the noise floor. If the original noise floor had zero mean then the second moment coincides with the variance of the noise floor. A "quiet" part (i.e., visually free of transients) of $c(k)$ is used to obtain an estimate of the standard devitaion s. A pseudo-random stationary white noise is generated with marginal density $N(0,s)$ (Gaussian with zero mean and variance s^2). The simulated noise is then processed using the same steps as the real data.

Figure (5) shows the running median plots for the S1 data and for 50 realizations of the simulated noise.

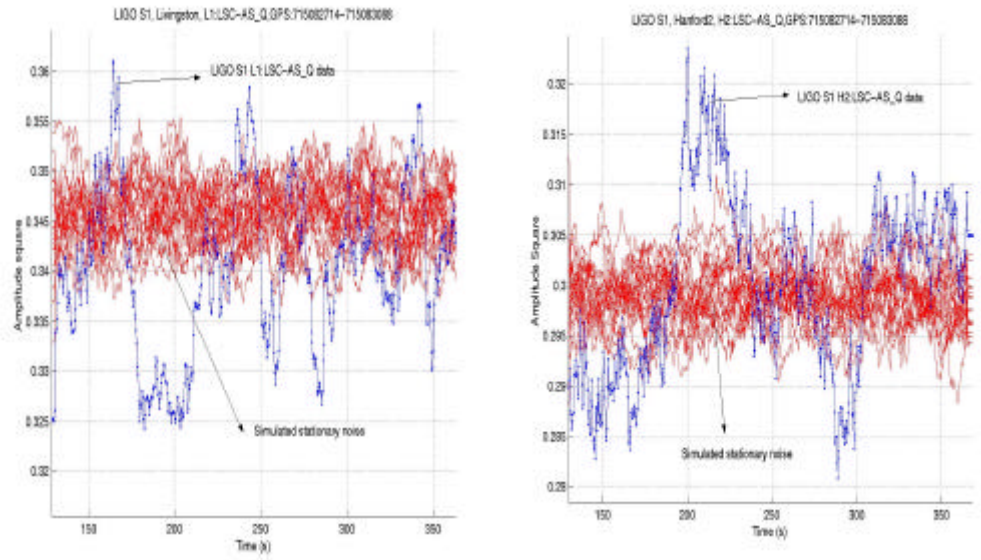

Figure 5. Running median discriminator with S1 data and simulated stationary white noise. The red curves gives an approximate estimate of the spread expected from stationary white noise with same variance. Points where the interferometric data exceeds this bound are indicative of departure from stationarity.

"Block length" as a parameter

The block length over which the Running Median is computed can be kept as a flexible parameter. This allows the investigator to chose the time scale over which (s)he wishes to study the presence of drifts. Figure 6 illustrates this point. The Running Median is shown against the spectrogram for H2. This particular data stretch is known to have non-stationarity (known as 'breathing of the noise floor' [5]) over a timescale of typically 0.5 sec. By suitably adjusting the block length, the Running Median picked up this behaviour.

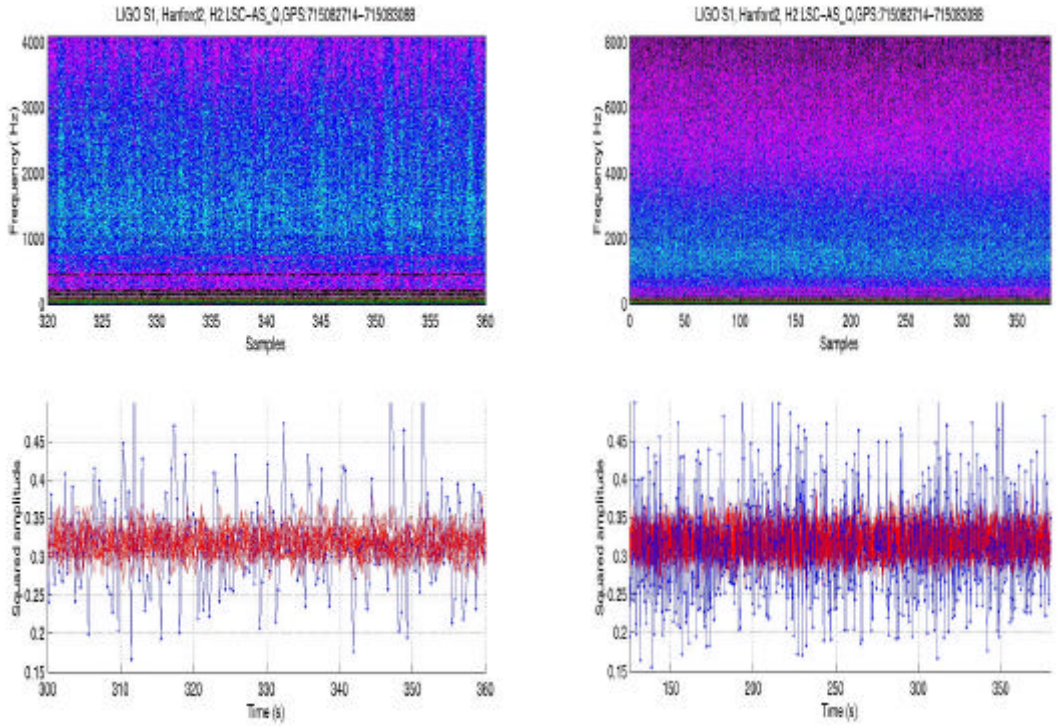

Figure 6. Left top : Spectrogram of the H2 data used in this example. The 'breathing noise floor' is zoomed in. Left bottom : The Running Median estimate of the second moment reflects the same variation. Right top and bottom : Same as the corresponding left figures on a longer time scale. The non-stationarity is very evident here.

Effect of transient spreading

Since the S1 data is infested with a very high rate of transients, it is important to test the impulse repsonse of the high order filters used in the data preprocessing to ensure that the impulses do not spread beyond the block length used for the running median computation.

A timeseries was generated with an impulse of same amplitude as the highest transient in the S1 data. The impulse response of the same series of filters is shown in figure (7). The percentage spread (computed by taking the ratio of the number of samples of the post-processed impulse to the block length of the Running Median) of the impulse for the highest transients is about 4.8.

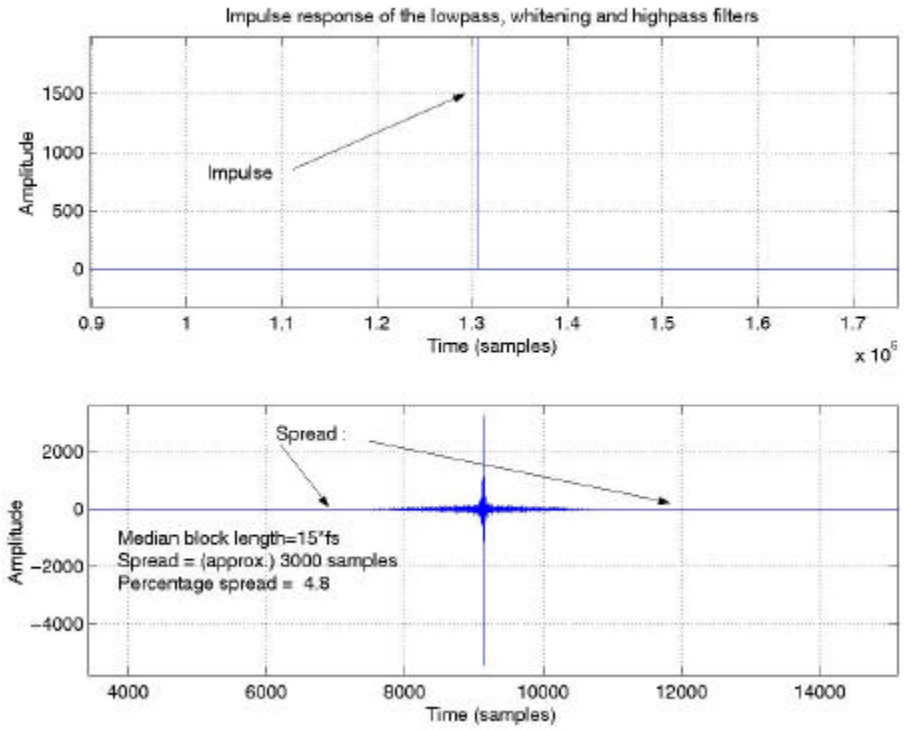

Figure 7. Impulse response of the set of filters used in the pre-processing of the data.

In the next set of simulations, equally spaced transients of the same magnitude were imposed on S1 data to see if the running median differed from the one when there was no transient. The result is illustrated in figures (8) and (9). The graphs show no apparent difference between the two cases mentioned above. The implication is that the transients do not introduce any artificial non-stationarity in the data and hence do not affect the Running Median estimator as a discriminator of non-stationarity.

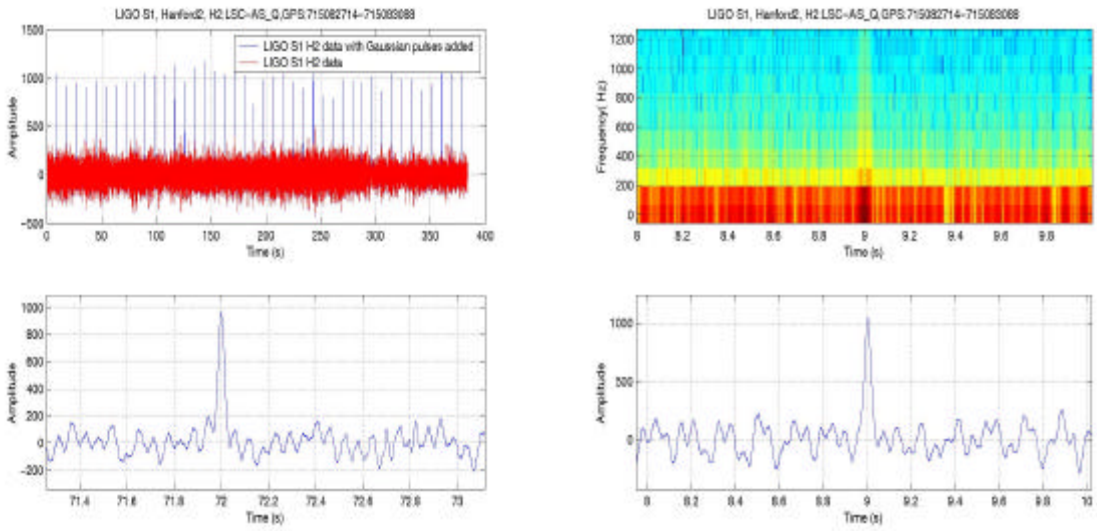

Figure 8. This figure shows Gaussian pulses added to the S1 data from H2. The figue on the left shows the time series, while the one on the right shows how these added transients appear in the spectrogram.

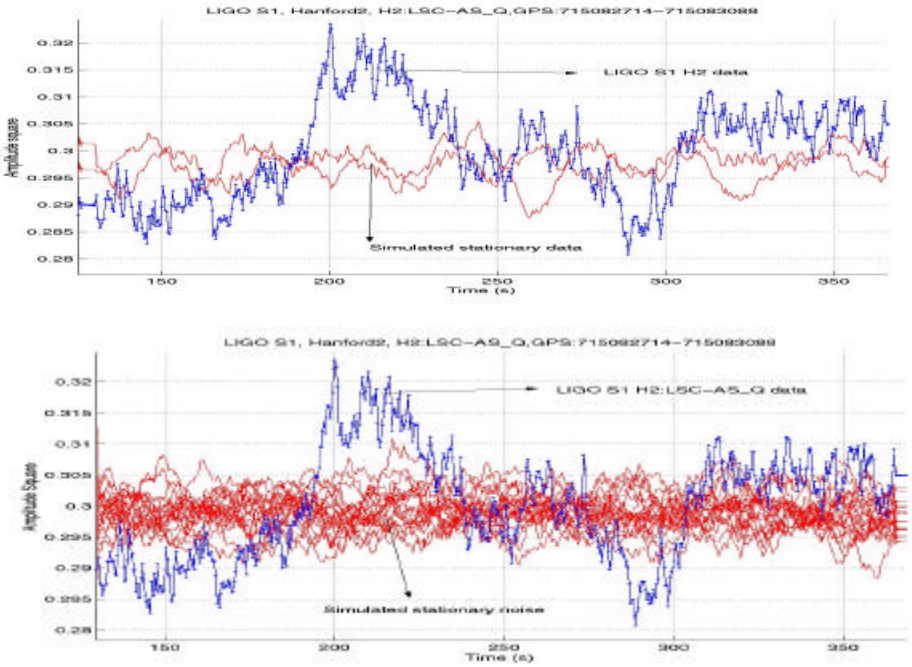

Figure 9. The top shows the Running Median for H2 data with transients added, while the bottom shows the same before the addition of transients.

Validation I : MNFT on simulated stationary white noise

In order to validate the performance of MNFT, we have simulated stationary white noise and processed it exactly the same way as have been done with the real data. The frequencies corresponding to the line frequencies in the real data were subtracted from the simulated data as well so that the simulation is also subjected to same filters as we have used in the real data. Figure 10 illustrates the results. It is clear from the figures that the test data *does not* venture out of the bound set by the stationary white noise spread, thus validating the method.

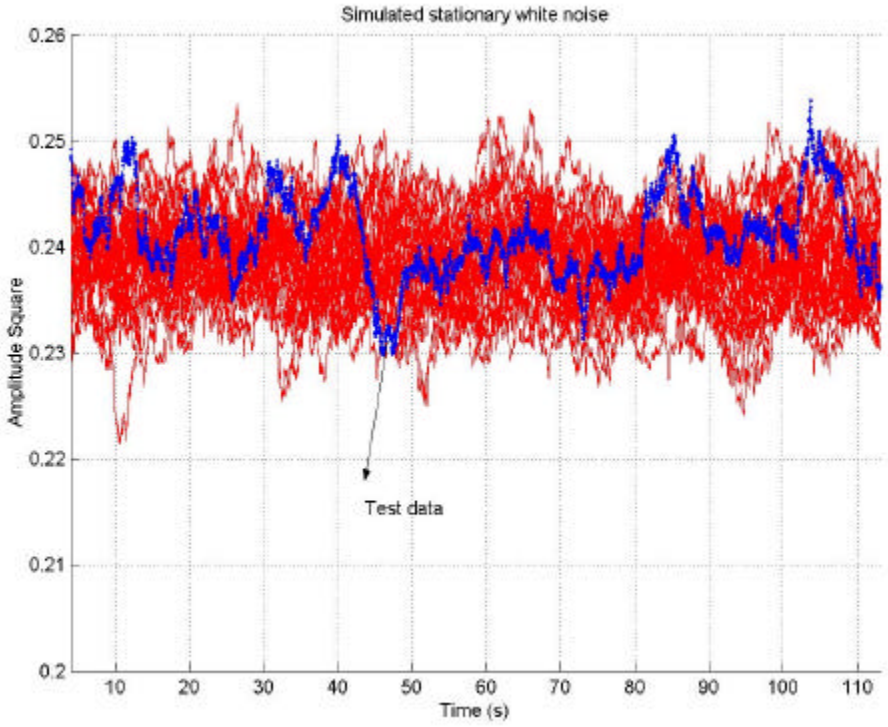

Figure 10. The blue curve shows the stationary test data imposed on 50 realizations of simulated stationary white noise. As should be expected, the test data *does not* exceed the stationary white noise spread.

Validation II : MNFT on simulated non-stationary data

Simulated nonstationary data was produced by a time varying IIR filter operating on a white Gaussian random process with unit variance. The filter coefficients are independent stochastic processes with specified means. Figure 11 shows a section of the non-stationary noise generated by the above method. Figure 12 shows the Running Median curve obtained by MNFT on this data. Contrary to what is seen in figure 10, here one can see the expected departure from the level of stationarity (depicted by the red curves). This confirms the efficiency of MNFT as a detector of non-stationary behaviour in the data stream.

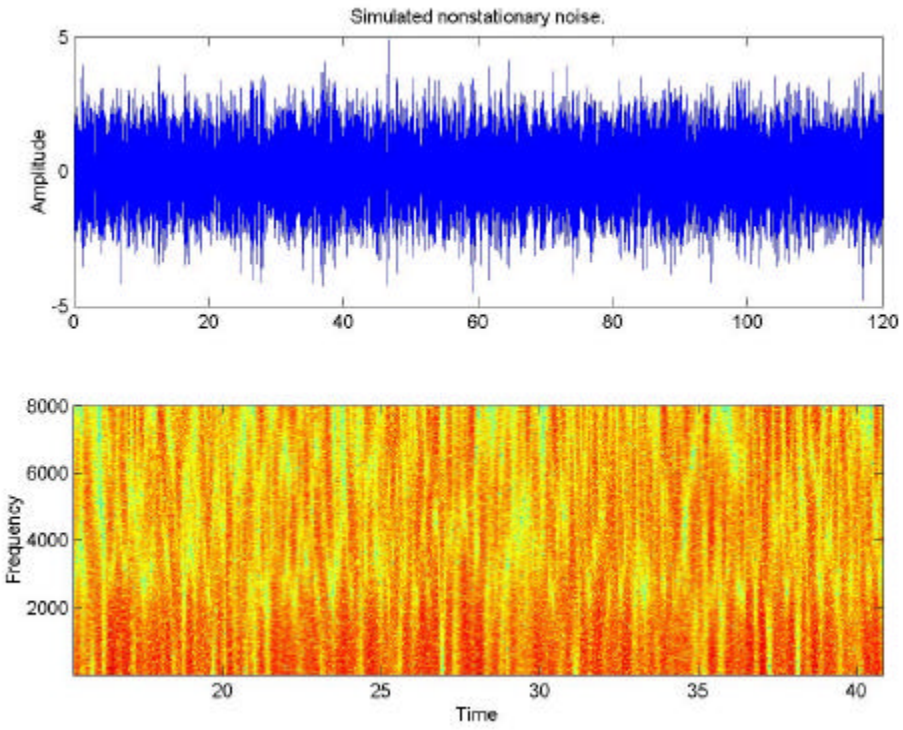

Figure 11. A section of the simulated non-stationary timeseries (top) and the corresponding spectrogram (bottom). The spectrogram is zoomed in to show the presence of non-stationarity clearly.

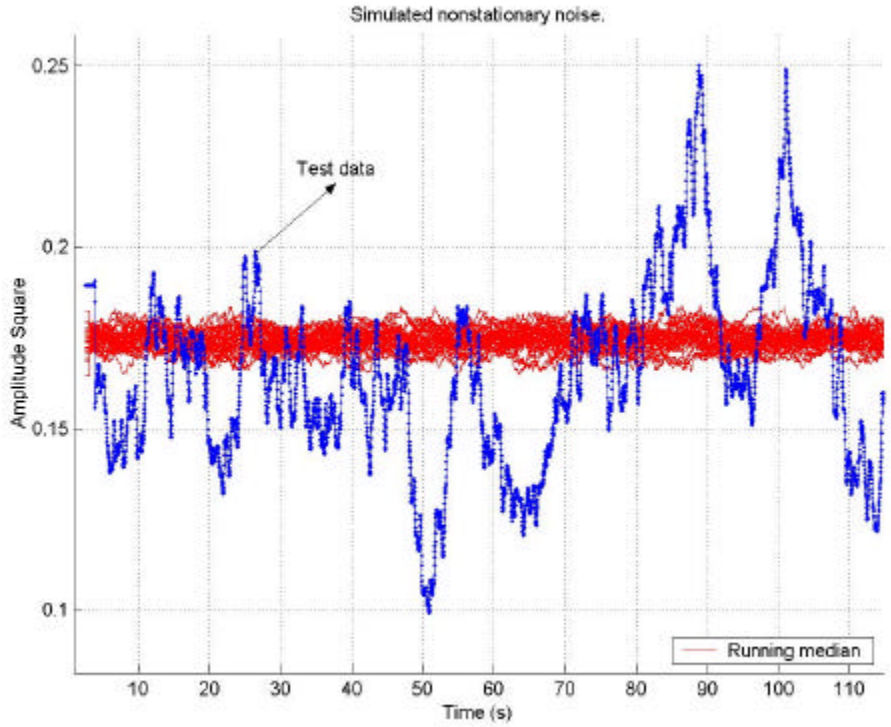

Figure 12. The Running Median of the simulated non-stationarity data. As expected, the departure from stationary noise level is clear. This validates the method as an efficient one to detect non-stationarity present in a timeseries.

Open Issues

1. *Use of other line removal methods to suppress line features before notch filtering* :

Using a suitable line removal method [11-14] as a pre-filter will be useful to reduce the order of the notch filter. Most of these methods are geared towards removal of technical noise of a particular kind e.g. violin modes, power lines etc. and works with assumed models. Median based line tracker (MBLT) [1] is a useful tool which removes all lines without having to assume any specific model and has the advantage of being immune to transients. The methods mentioned above typically suppress the high dB lines by more than 30 dB. Some work has been done along these lines, but is subjected to further testing. MBLT is computationally expensive on a single machine. Work is underway to include MNFT under the Detector Characterization Robot (DCR) [7,8] in a cluster environment. This is expected to speed up the process greatly and is estimated to be able to track the slow drift online.

2. *Setting thresholds using a single simulated stationary timeseries :*

Since the simulation process consists of generating stationary white noise with a given variance, it would be more natural to simulate a single timeseries typically ~50 times the length of the given data (equivalent to 50 different realizations), histogram it and get an estimate of its variance that can serve as a measure of threshold. The exact threshold to be determined is of course based on the requirements of the specific analysis one is looking at. Work is in progress.

Comments and future direction

The study yields interesting results and can find many applications wherever a suitable indicator of non-stationarity in the data is required in detector characterization [15] and data chracterization [16] for setting upper limits for different astrophysical sources. For example, this method could lead to construction of an epoch veto, a monitor [17] or a warning system that clicks whenever the data exceeds the threshold.

One of the important questions relates to obtaining the tolerance of astrophysical searches to non-stationarity. This clearly depends on the search that is being performed. The degree of tolerance can be calculated by assuming an appropriate model for the type of non-stationarity noticed in the data and plugging in the modeled noise to a search algorithm to obtain the necessary sensitivity and tolerance to the

level of departure from an ideal stationary noise. Work along these lines is in progress.

Acknowledgement

SM wishes to thank LIGO project for use of S1 data that has been used to demonstrate the results that emerge from this methodology.

References

1. S.D.Mohanty, in Proc. Fourth AMALDI, Edited by D. Blair, Class. Quantum Grav. **19** (2002), 1513-1520
2. E. Chassande-Mottin and S.V. Dhurandhar, Int. J. Mod. Phys. **D9** (2000), 275-279
3. E. Cuoco et al., Phys. Rev. D **64** (2001), 122002
4. G. Cella, E. Cuoco and G.M. Guidi, gr-qc/0210083 (2002)
5. LSC External Triggers Burst Upper Limit group, LIGO Tech Doc. LIGO-T030050-00-Z, *Triggered search analysis of S1data associated with X-ray flash XRF020903*, Caltech, Pasadena (2003)
6. MATLAB, Mathworks Inc., (2002)
7. S.D. Mohanty and Soma Mukherjee, in Proc. Fourth AMALDI, Edited by D. Blair, Class. Quantum Grav. **19** (2002), 1471-1476
8. Soma Mukherjee, LIGO Technical document LIGO T-030018-00-Z, Caltech, Pasadena (2003)
9. http://www.aei.mpg.de/~yousuke/alsc/readme.ps.gz
10. Soma Mukherjee, LIGO Technical document LIGO T-030052-00-Z, Caltech, Pasadena (2002)
11. L.S.Finn and Soma Mukherjee, Phys. Rev.D, **63** (2001)
12. A.M. Sintes and B.F. Schutz, Phys. Rev. D, **58** (1998)
13. B.Allen et al., gr-qc/9909083 (1999)
14. L.S.Finn et al., in Proc. Third AMALDI, Edited by S. Meshkov, (AIP:New York) (2000), 451
15. http://www-mhp.physics.lsa.umich.edu/~keithr/lscdc/home.html
16. http://www.ligo.caltech.edu/%7Eajw/bursts/bursts.html
17. http://www.ligo.caltech.edu/~jzweizig/DMT-Project.html
18. A. Stuart and J.K. Ord, Kendall's Advanced Theory of Statistics, (Edward Arnold: London) (1994)