

Median Based Line Tracker (MBLT): Model independent and transient resistant removal of line noise from Interferometric data

Soumya D Mohanty

Albert Einstein Institute, Max Planck Institut für Gravitationsphysik, Am Mühlenberg 1, Golm D14476, Germany.

E-mail: mohanty@aei-potsdam.mpg.de

Abstract. Removal of narrowband noise features (also called *lines*) of known instrumental origin from a time series is important for improving the performance of algorithms such as those for the detection of transients. We present a new method for removing lines which (i) is not based on any model for the features to be removed, (ii) is designed not to affect transients substantially and (iii) works in the time domain. Property (i) allows lines to be removed irrespective of their physical origin, (ii) ensures that transients remain detectable in the residual after line removal and (iii) means that, unlike Fourier domain methods, line power is not redistributed in the entire frequency band.

PACS numbers:

1. Introduction

Real time series from interferometers and their associated environmental sensors are often contaminated by narrowband noise features or *lines* such as interference from the power supply. Sometimes these features are strong enough to degrade the performance of algorithms such as those that detect transients. For example, transient detection in the frequency domain would suffer because of power leakage from strong lines biasing spectral estimates. Moreover, since these features appear non-stationary over short time stretches, a transient detector would generate excess false alarms.

Thus, it is desirable to “clean” a time series of strong lines before doing further processing on it. Of course, because of estimation errors caused by the broadband noise, the residual will be disturbed from the original in narrow frequency bands. The effect on the broadband noise is not of much concern. However, the effect on *signals* embedded in the time series should be as small as possible.

It is preferable to use *time domain* methods for line removal as opposed to *frequency domain* methods. By the latter we mean methods which work by taking Discrete Fourier Transforms (DFT) of segments of the data, nulling the frequency components around the lines and inverting back to the time domain. This way of removing lines is only an approximation. The DFT spreads the power in a line over all frequencies. Hence, suppression of a subset of the frequency components does not remove all of the line power.

A standard method of removing unwanted line noise is the use of time domain *notch filters* [1] which have (ideally) nulls in their transfer functions at the line frequencies. The disadvantage is that the signal to noise ratio of a transient will also be reduced. The reduction depends on the bandwidth of the transient and also the location of the notches relative to the shape of the transient spectrum.

Other line removal methods such as the Kalman filter [2], CLRT [3] and cross-channel regression [?, 5] first *estimate* the time series of the line to be removed and then subtract it from the data. The estimation is based on a *model* of the line evolution and, hence, only applies to a subset of lines. Thus, (i) applies to *violin resonances* while (ii) and (iii) work for the power line and its harmonics. Besides this, these models do not explicitly take the effect of transients into account which implies that these methods will in turn affect transients.

The discussion above provides motivation for developing a method of line removal which is (i) independent of any model for line evolution, (ii) resistant to the presence of transients and (iii) works in the time domain. We introduce a new method, called the Median Based Line Tracker (MBLT), which is explicitly designed to meet these three goals. In this short paper we describe the MBLT algorithm and a few results. A more detailed report on its performance will be presented in a future work.

The rest of the paper is organized as follows. Section 2.1 describes the MBLT algorithm. A brief discussion of implementation and computational issues is presented in Section 2.2. This is followed, in Section 3 by the presentation of some results obtained using real data. We conclude with a discussion in Section 4.

2. The Median Based Line Tracker

We fix some notation first. Let \bar{x} be the *time series* to be cleaned. Let $F = \{f_i\}$, $i = 1, \dots, M$, be the carrier frequencies of the lines that are to be removed and $B = \{b_i\}$, $i = 1, \dots, M$, be the bandwidths around the carrier frequencies that are allowed to be disturbed (after line removal). In the following, we will assume the sampling frequency of a time series to be 1 Hz without any loss of generality.

2.1. Algorithm

The algorithm works by successively refining the estimate of each line in F through several stages (say, stage 1 to P). Let the estimate in stage k of a line at frequency f_i be $\bar{e}_i^{(k)}$. To obtain the estimate $\bar{e}_i^{(k)}$, the original time series \bar{x} is first modified by subtracting an auxiliary time series $\bar{s}_i^{(k)}$. Let the modified time series be denoted by $\bar{x}_i^{(k)} = \bar{x} - \bar{s}_i^{(k)}$. We explain below how $\bar{s}_i^{(k)}$ and $\bar{e}_i^{(k)}$ are obtained.

The auxiliary series is obtained as ($k \neq 1$),

$$\bar{s}_i^{(k)} = \sum_{\substack{j=1, \\ j \neq i}}^M \bar{e}_j^{(k-1)}, \quad (1)$$

where

$$\bar{s}_i^{(1)} = 0; \quad (2)$$

The estimate $\bar{e}_i^{(k)}$ at stage $k \geq 1$ is obtained as follows.

- (i) Heterodyne $\bar{x}_i^{(k)}$ at the carrier frequency f_i . This produces two time series which we denote by \bar{X} and \bar{Y} ,

$$\bar{X} = \mathcal{L}(\{x_n * \cos(2\pi f_i n)\}) , \quad (3)$$

$$\bar{Y} = -\mathcal{L}(\{x_n * \sin(2\pi f_i n)\}) , \quad (4)$$

where \mathcal{L} denotes a low pass filter with a cutoff frequency of f_i and x_n is the n^{th} sample of \bar{x} .

- (ii) Apply a *running median* to both quadratures to obtain the time series \bar{X} and \bar{Y} . The running median \bar{Z} of a time series Z is obtained as,

$$Z_n = \text{median}(\{Z_j\}) , j = n - m, \dots, n + m , \quad (5)$$

where m is a fixed number. We call $2m$ the *blocksize* for the running median. The blocksize for $\bar{e}_i^{(k)}$ is set to $1/b_i$.

- (iii) The line estimate $\bar{e}_i^{(k)}$ is obtained by modulating a carrier at f_i with the two quadratures \bar{X} and \bar{Y} ,

$$\bar{e}_n^{(k)} = \text{Re}((\mathcal{X}_n + i\mathcal{Y}_n) \exp(2\pi i f_i n)) , \quad (6)$$

where the subscript i has been temporarily dropped for clarity.

The basic idea behind the above algorithm is to estimate a line *after* removing the influence of other strong lines. Removal of other lines is done by subtracting the auxiliary time series $\bar{x}_i^{(k)}$. The successive stages simply refine this process further. The use of median for estimation instead of a mean, say, makes the estimate more robust against transients as we show later.

2.2. Implementation and computational issues

The algorithm described above is a time domain algorithm: the data \bar{x} does not have to be divided into segments to get the estimate. For a practical reason, namely, the necessity to pass arguments to functions as finite dimensional arrays, \bar{x} can be broken into segments. However, by preserving appropriate information about the state of the estimate at the end of one segment, one can initialize the estimate of the next segment such that continuity is preserved across segments. This information consists of the median of the last block and the samples of the last block sorted in ascending order.

Calculating the running median is expensive if every block is sorted (sorting is an $O(n^2)$ operation in the worst case). However, to compute the next sample of the running median, all one needs to do is insert the next sample of the time series into the previously *sorted* block and remove the first element of the block. We have developed and implemented an algorithm along this idea which reduces the number of operations per block to $O(\sqrt{n})$ in the worst case.

MBLT is parallelizable and can be implemented on Multiple Instruction Multiple Data parallel processing architectures such as Beowulf Clusters. At any given stage, each line estimate is obtained independently of others (though all estimates from the previous stage are used). Thus, in each stage different processors can handle the estimation of different subsets of lines.

The entire algorithm is currently implemented as a Matlab [6] code. Running on a stand alone PC, this code itself is ~ 5 times faster (for a single line with 3 stages of refinement) than what is required for online processing. An MPI [7] based C code is currently under development.

The computer memory requirements of this algorithm can be significant. This is because the estimates of all lines in must be held in memory with each estimate having the original sampling frequency. Alternative implementations that require less memory need to be explored.

The algorithm requires a one time external specification of F and B (c.f., Sec. 2.1). This has to be provided by the user. We currently do this by measuring F and B off the Power Spectral Density of a segment of the data.

3. Results

The results presented here should be treated as preliminary. We have used a short (20 sec) stretch of data (the IFO_DMRO channel) from the 1994 November run of the Caltech 40m prototype interferometer. For these results, $F = \{30.718, 39.6, 41.1, 42.6, 52.2, 58.85, 70.00, 79.5, 109.4, 120.0, 139.6, 180.0, 197.82, 209.385, 210.82, 240.0, 246.0, 263.783, 279.155, 300.0, 349.0, 360.0, 418.76, 420.0, 479.95, 488.55, 499.95, 540.0, 558.38, 571.6, 578.66, 581.075, 582.55, 583.77, 594.183, 595.27, 595.914, 597.8, 599.05, 599.921, 605.5, 660, 720, 780, 840, 900\}$, $b_i \leq 1.0$ Hz except for $f_i = 70.0$ Hz where $b_i = 12.0$ Hz and the number of stages used was 3. The data was down sampled to a sampling frequency of 2 kHz because of memory limitations on the machine running this code.

3.1. Line removal

Figure 3.1 shows the Power Spectral Density (PSD) of the original time series and the final residual after subtraction. The suppression is significant for most of the lines, including the violin resonances, except towards the end of the spectrum ($f > 700$ Hz). The poor performance for the latter is under investigation at present. It appears to more of a technical problem than a fundamental one since the time series corresponding to this band shows much better results when analyzed separately.

Figure 2 shows the PSD of the residual for the earlier stages of the algorithm (stage 1 and 2). The progressive improvement in suppression of the lines is evident.

3.2. Effect on transients

Figure 3 shows the the quadratures of a line in the presence of a transient. Here, we have used synthetic data for both the line and the broadband noise. Also shown are the estimates obtained using a running median and a running mean. The blocksize for the latter is empirically chosen to give the same variance of the estimation error as the running median. It is evident that the running median is affected less than the running mean. Correspondingly, after subtraction, more of the transient will be left behind in the running median based residual. Also note that the running mean is more biased in the wings of the transient.

4. Discussion

A new method is presented for the removal of narrowband noise features from a time series. This method, MBLT, works in the time domain, does not require modeling of line evolution and is designed to have a minimal effect on transients.

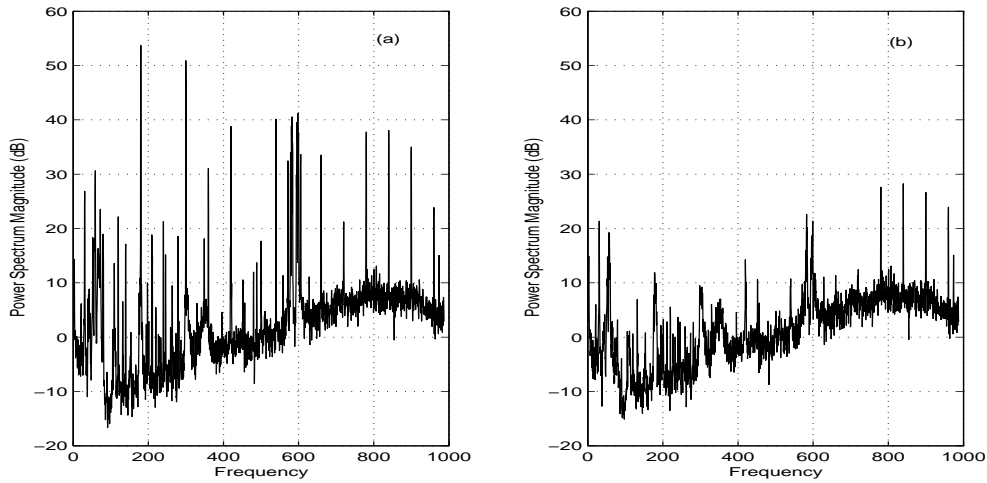


Figure 1. The PSD of (a) the original time series \bar{x} and (b) the final residual after line removal.

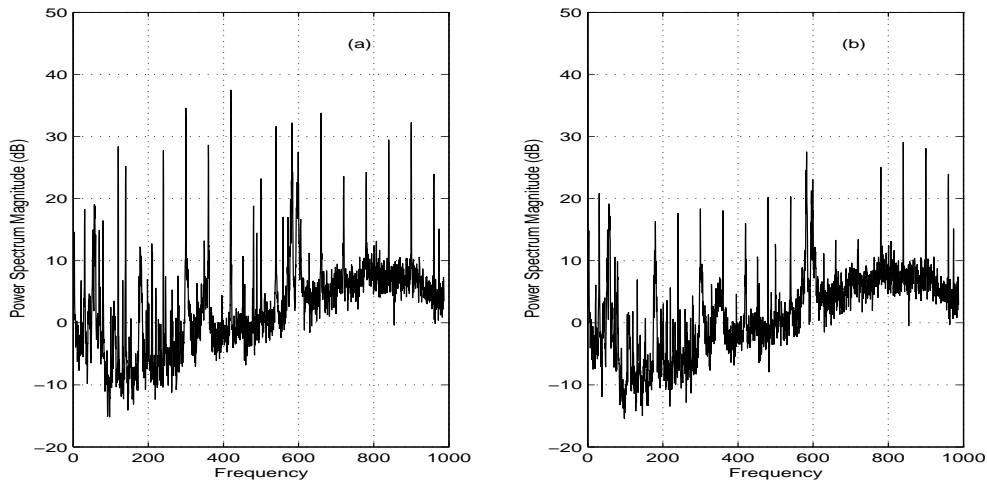


Figure 2. The PSD of the residual after (a) stage 1 and (b) stage 2.

Apart from the main channels, MBLT can be used to clean the auxiliary channels (such as environmental sensors) without the need for any modelling. This allows a better control over the false alarm rate for transient detection in these channels. Transients detected in auxiliary channels can provide anti-coincidence vetoes for events in gravitational wave strain channel and also help in detector diagnostics.

It is possible to improve upon a running median for rejecting transients. One possibility could be to take a weighted average over line estimates computed with different blocksizes. There is also the possibility of merging adaptive filtering methods [8] with an essentially non-linear method like MBLT.

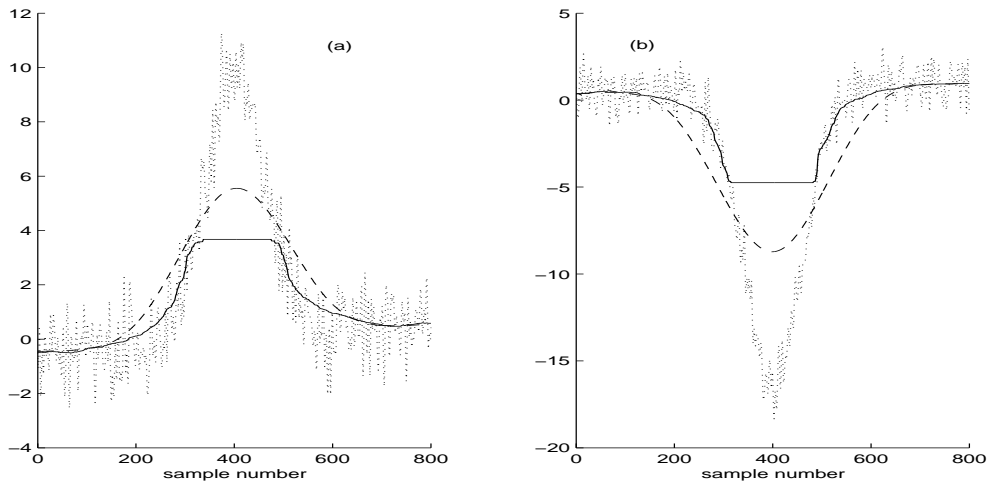


Figure 3. Effect on a transient. (a) The \bar{X} quadrature. (b) \bar{Y} quadrature. In both figures, the faint dotted lines are the quadrature components, the dashed line shows the running mean estimate and the solid line shows the running median estimate.

Acknowledgments

I thank Prof. B. F. Schutz for helpful discussions, especially the ones that led to a clarification of the difference between frequency and time domain line removal. Part of this work was done at the Pennsylvania State University under NSF award PHY 98-00111 and 99-96213. Discussions within the gravitational wave data analysis group at PSU helped the author in understanding several aspects of line removal in general.

References

- [1] J. G. Proakis, D. G. Manolakis, *Digital Signal Processing*, third edition (Prentice Hall), p 343.
- [2] S. Mukherjee, L. S. Finn, *Proc. third Amaldi Conference (Pasadena)*, ed S. Meshkov (New York: AIP) p 362.
L. S. Finn, S. Mukherjee, *Phys. Rev. D* **63**, 062004 (2001).
- [3] A. M. Sintes, B. Schutz, *Phys. Rev. D* **58**, 122003 (1998).
- [4] B. Allen *et al* , Automatic cross-talk removal from multi-channel data, *Preprint gr-qc/9909083*.
- [5] L. S. Finn *et al* , *Proc. third Amaldi Conference (Pasadena)*, ed S. Meshkov (New York: AIP) p 451 .
- [6] Matlab, a technical computing environment for high-performance numerical computations in linear algebra, is a product of The MathWorks Inc.
- [7] <http://www.mpi-forum.org>.
- [8] E. Chassande-Motin, S. V. Dhurandhar, *Phys. Rev. D* **63** 042004 (2001).