INSTITUTE OF PHYSICS PUBLISHING CLASSICAL AND QUANTUM GRAVITY

Class. Quantum Grav. **18** (2001) 1727–1751 www.iop.org/Journals/cq PII: S0264-9381(01)19884-9

On-line power spectra identification and whitening for the noise in interferometric gravitational wave detectors

Elena Cuoco[1,5], Giovanni Calamai[2], Leonardo Fabbroni[1], Giovanni Losurdo[3], Massimo Mazzoni[1], Ruggero Stanga[1] and Flavio Vetrano[4]

[1] Dipartimento di Astronomia e Scienze dello Spazio, Universitá di Firenze and INFN Firenze/Urbino section, Largo E. Fermi 2, 50125 Firenze, Italy
[2] Osservatorio Astrofisico di Arcetri and INFN Firenze/Urbino section, Largo E. Fermi 5, 50125 Firenze, Italy
[3] INFN Firenze/Urbino section, Largo E. Fermi 2, 50125 Firenze, Italy
[4] Università di Urbino and INFN Firenze/Urbino section, Istituto di Fisica, Via S. Chiara 27, 61029 Urbino, Italy

E-mail: cuoco@fi.infn.it

Received 8 December 2000

Abstract

The knowledge of the noise power spectral density of an interferometric detector of gravitational waves is fundamental for detection algorithms and for the analysis of the data. In this paper we address both the problem of identifying the noise power spectral density of interferometric detectors by parametric techniques and the problem of the whitening procedure of the sequence of data. We will concentrate the study on a power spectral density like that of the Italian–French detector VIRGO and we show that with a reasonable number of parameters we succeed in modelling a spectrum like the theoretical one of VIRGO, reproducing all of its features.

We also propose the use of adaptive techniques to identify and to whiten the data of interferometric detectors on-line. We analyse the behaviour of the adaptive techniques in the field of stochastic gradient and in the least-squares filters. As a result, we find that the least-squares lattice filter is the best among those we have analysed. It succeeds optimally in following all the peaks of the noise power spectrum, and one of its outputs is the whitened part of the spectrum. Besides, the fast convergence of this algorithm, it lets us follow the slow non-stationarity of the noise. These procedures could be used to whiten the overall power spectrum or only some region of it. The advantage of the techniques we propose is that they do not require *a priori* knowledge of the noise power spectrum to be analysed. Moreover, the adaptive techniques let us identify and remove the spectral line, without building any physical model of the source that produced it.

PACS numbers: 0480N, 0705K, 0760L, 0540C

[5] Author to whom correspondence should be addressed.

1. Introduction

1.1. Generalities

The detection of gravitational waves represents a major goal in contemporary physics. A world-wide effort has been made in building detectors (especially ground-based long-arm detectors) with enough sensitivity to make astrophysical observations or, in a wider sense, to move into the field of gravitational astronomy [1–3]. The building of these large interferometers is now reaching the phase of data taking: TAMA (Japanese) [4] is already working; GEO (British/German) [5] will begin to take data next year; LIGO (USA) [6] in 2002; VIRGO (French/Italian) [7] in 2003. However, a long test run of the central interferometer in VIRGO is foreseen during 2001, thus leading to a large amount of data to be analysed: although these data are mainly for diagnostic purposes, they provide a very good opportunity to examine analysis techniques. In the following, even though we are referring to the VIRGO antenna, our considerations can, in principle, be applied to all of ground-based interferometric gravitational detectors. Generally speaking, all these detectors are Michelson interferometers with suitable technical additions in order to improve sensitivity (see [1, 8, 9] for exhaustive and up-to-date descriptions of the physics and technology involved in building up interferometric gravitational antennas). A gravitational wave (GW) displaces in a different way the far mirrors in the two arms, thus shifting the interference pattern at the beamsplitter; however, the best models we have for GW (astrophysics) sources are leading to a very small value for the wave amplitude on the Earth [8, 9] requiring for its detection a spectral sensitivity of about 10^{-19} m/$\sqrt{\text{Hz}}$ in a band of about 1 kHz, let us say from few Hz to many hundreds of Hz. However, to reach this nominal sensitivity is only the first challenge. In fact, other not minor challenges arise: the run of the antenna should be continuous because we are searching for rare time-limited events (supernovae bursts, coalescing binary systems) or, in contrast, we have to integrate small continuous signals over long times (e.g. pulsars signals), while a very sensitive monitoring of the environment and instrument noise should be carefully and continuously done; when considering the low value of signal-to-noise ratio (SNR), we are led to foresee that a large amount of data will be handled, a lot of computing power (TFLOPS) will be required, large archiving capacity (TBytes) will be needed for storage and retrieval [10, 11]. Finally, from a preliminary point of view for data treatment, which is indeed the subject of this paper, a major challenge lies in the fact that the noise of GW detectors does not satisfy the simplifying assumptions of white noise, but the expected noise is a coloured broad-band background with some spectral (deterministic) peaks. Moreover, the noise distribution could be non-stationary and non-Gaussian.

1.2. The problem

The large amount of data produced by gravitational wave detectors will be essentially noise and, hopefully, buried in the noise will be the signal we are looking for. As we have already seen, ground-based interferometric detectors are sensitive to a broad range of frequencies (2–3 Hz to more than 1 kHz) in revealing the relative displacement of test masses at the near and far extremities of the interferometer arms due (possibly!) to a GW signal, but unfortunately many other factors can cause a similar displacement. The test masses are suspended from pendular structures in order to isolate them from seismic noise [13], but thermal noise of the suspension chain will cause a displacement of the mass [14]. Also shot noise and radiation pressure from the laser will move the mirrors [1, 15]. The physicists are working on modelling all possible causes of noise in the interferometer giving the apparatus sensitivity curve [16–22]. This curve is limited at very low frequencies (a few Hz and below) by seismic noise; in the middle band

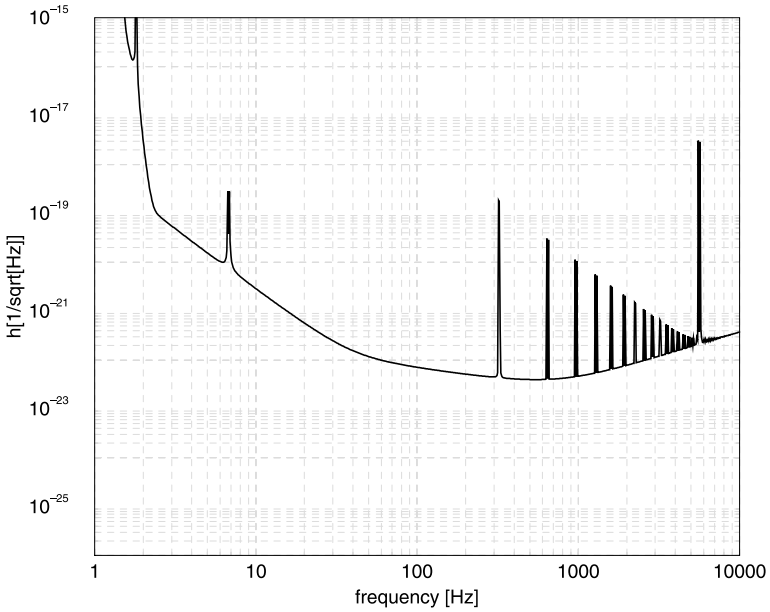

Figure 1. VIRGO sensitivity curve.

by thermal noise and at high frequencies (higher than 0.7–1 kHz) by shot noise. In figure 1 we plot the predicted VIRGO sensitivity curve obtained as incoherent sum of all estimated noise contributions[6] [23]. This curve is characterized by a broad-band noise plus several very narrow peaks due to the violin modes of the suspensions wires, that will make the detection of a gravitational signal in this frequency band very difficult. For this reason efforts have been made in the preparation of the analysis of data for cutting out these resonances [24–26]. It is evident that the analysis of data to detect the gravitational signal requires an accurate knowledge of the noise, which requires a statistical characterization of the stochastic process, evaluating its stationarity and its Gaussian nature, and in the case of local stationarity and Gaussian nature an accurate estimation of the power spectral density (PSD).

The output of the interferometer will be non-stationary over a long period of time, so we must be ready to follow the changes in the PSD. A way to achieve this is to estimate the PSD on a chunk of data at a different interval of time, using classical techniques [29, 33, 36]. We propose to use adaptive methods to follow on-line the change in the feature of the spectrum in such a way as to have the correct curve for the PSD at any desired instant.

If we are able to identify the noise of our detector we can also apply the whitening procedure to the data.

The goal of the whitening procedure is to make the sequence of data delta-correlated, removing all the correlation of the noise. Most detection theory is considered within the framework of a wide-sense stationary Gaussian white noise, but in our problem the noise is surely coloured and, in principle, non-stationary and non-Gaussian features could be present. If we whiten the data, supposing henceforth to be within the framework of a stationary and Gaussian noise, we can apply the optimal detection algorithm [37].

For example, if we assume that the noise data which we are analysing are stationary and Gaussian distributed and we assume we know the waveform of our signal, then the optimal

[6] We would like to thank Michele Punturo from INFN Perugia who gave us this plot.

detection filter matches between our sequence of data and the Wiener filter:

$$M(t,\theta) = \int_{-\infty}^{\infty} \mathrm{e}^{-2\mathrm{i}\pi\nu t} \frac{h(\nu,\theta)}{S(\nu)}\,\mathrm{d}\nu, \tag{1}$$

where $S(\nu)$ is the noise PSD, $h(\nu,\theta)$ is the template of the signal we are looking for and θ denotes the parameters of the waveform. As is evident from (1) the operation of whitening is implicitly done each time we apply the Wiener filter to detect a signal, because we weight the data with the inverse PSD of the noise: in such a way we have 'whitened' sequence to analyse.

Moreover, when we are searching a transient signal of unknown form it is very important to have a whitened noise [11, 12]. The importance of whitening data is also linked to the possibility of reducing their dynamical range [34, 35].

The aim of this paper is to show how to identify the noise PSD and how to whiten the data produced by an interferometric detector before applying any detection algorithm.

In section 2 we underline the advantages of parametric modelling; in section 3 we show the whitening techniques based on a lattice structure. In section 4 we report the application of PSD fitting and data whitening on VIRGO-like simulated data. In section 5 we introduce the theory of adaptive filters based on stochastic and least-squares methods, and its application to VIRGO-like data: for this we compare their performances on simulated data.

2. Parametric modelling

The advantages of parametric modelling with respect to the classical spectral methods are described in an exhaustive way in [29]. Here we want to underline that the kind of analysis we have to perform can take advantage of these methods in two main ways. First of all we can achieve a better resolution in the estimation of the PSD, because we can better use the information contained within the autocorrelation function. In fact, we suppose that the process we are analysing is governed by a dynamical law and we can use the knowledge of the autocorrelation function until a certain lag and then extrapolate its value under the dynamical hypothesis we made. Moreover, we may compress the information from the PSD into a restricted number of parameters and not in the full autocorrelation function. This can help, for example, if we want to create a database of noise sources.

In this context we also want to talk about parametric modelling because it offers the possibility to write down a linear whitening filter and in a fast way to build simulated data on which we can perform our tests of the whitening filter.

We work in the field of rational functions to fit the theoretical PSD. We will show that it is possible to obtain a fit of the theoretical PSD of an interferometer output such as that of VIRGO with an autoregressive moving average (ARMA) or autoregressive (AR) model [27, 28, 30], then we will use the data we can generate in this way, to test the whitening algorithms we propose.

2.1. ARMA and AR models

The procedure to estimate the PSD using parametric modelling is based on three steps:

(a) select the appropriate model for the process;
(b) estimate the model parameters from the given data;
(c) use these parameters in the theoretical power spectrum density for the model.

Once we have the parameters which make the fit we use them to generate noise data to perform our tests.

A general process described by an ARMA(P, Q) model, where P is the number of poles and Q is the number of zeros, satisfies the relation

$$x[n] = -\sum_{k=1}^{P} a[k]x[n-k] + \sum_{k=0}^{Q} b[k]w[n-k] \tag{2}$$

and its transfer function is given by

$$\mathcal{H}(z) = \frac{\mathcal{B}(z)}{\mathcal{A}(z)}, \tag{3}$$

where $\mathcal{A}(z) = \sum_{k=0}^{P} a[k]z^{-k}$ and $\mathcal{B}(z) = \sum_{k=0}^{Q} b[k]z^{-k}$.

The PSD of the ARMA output process is

$$P_{ARMA}(f) = \sigma^2 \left| \frac{B(f)}{A(f)} \right|^2, \tag{4}$$

where σ is the variance of the driven white noise w, $A(f) = \mathcal{A}(2\pi \mathrm{i} f)$ and $B(f) = \mathcal{B}(2\pi \mathrm{i} f)$. An autoregressive process AR(P) is governed by the relation

$$x[n] = -\sum_{k=1}^{P} a[k]x[n-k] + w[n], \tag{5}$$

and its PSD for a process of order P is given by

$$P_{AR}(f) = \frac{\sigma^2}{\left|1 + \sum_{k=1}^{P} a_k \exp(-\mathrm{i}2\pi k f)\right|^2}. \tag{6}$$

Once we have selected the model for our process, we need to find the parameters for this model. The parameters of the ARMA model are linked to the autocorrelation function of the process by the Yule–Walker equations [29]. In the general case of an ARMA process we must solve a set of nonlinear equations while, if we specialize to an AR process, that is an all-poles model, the equations to be solved to find the AR parameters become linear.

The relationship between the parameters of the AR model and the autocorrelation function $r_{xx}(n)$ is given by the Yule–Walker equations

$$r_{xx}[k] = \begin{cases} -\sum_{l=1}^{P} a_l r_{xx}[k-l] & \text{for} \quad k \geqslant 1 \\ -\sum_{l=1}^{P} a_l r_{xx}[-l] + \sigma^2 & \text{for} \quad k = 0. \end{cases} \tag{7}$$

The problem of determining the AR parameters is the same as that of finding the optimal 'weights vector' $\boldsymbol{w} = w_k$, for $k = 1, \ldots, P$ for the problem of linear prediction [29]. In the linear prediction we would predict the sample $x[n]$ using the P previous observed data $\boldsymbol{x}[n] = \{x[n-1], x[n-2], \ldots, x[n-P]\}$ building the estimate as a transversal filter:

$$\hat{x}[n] = \sum_{k=1}^{P} w_k x[n-k]. \tag{8}$$

We choose the coefficients of the linear predictor by minimizing a cost function that is the mean-square error $\epsilon = \mathcal{E}[e[n]^2]$ ($\mathcal{E}$ is the expectation operator), with the error we make in this prediction

$$e[n] = x[n] - \hat{x}[n] \tag{9}$$

and obtaining the so-called normal or Wiener–Hopf equations

$$\epsilon_{min} = r_{xx}[0] - \sum_{k=1}^{P} w_k r_{xx}[-k], \tag{10}$$

which are identical to the Yule–Walker equations with

$$w_k = -a_k \tag{11}$$

$$\epsilon_{min} = \sigma^2. \tag{12}$$

This relationship between the AR model and the linear prediction ensures that we obtain a filter which is stable and causal [29]. It is this relation between the AR process and the linear predictor that becomes important in the building of a whitening filter.

2.2. Durbin algorithm and lattice structure

A method of solving the Yule–Walker equation is the Durbin algorithm [32]. Let us suppose that the process is an autoregressive one of order P.

The strategy of this method is to assume that the optimal $(P-1)$th-order filter has been computed previously, and then to calculate the optimal Pth-order filter based on this assumption. To accomplish the algorithm we perform a loop on the order of the process for $1 \leqslant j \leqslant P$. We initialize the mean-square error as $\epsilon_0 = r_{xx}[0]$ and then we begin the iteration on the loop, introducing the reflection coefficients k_p [36] at the stage p[7], which are linked to the autocorrelation function r_{xx} and to the a_j parameters of the filter at stage $p-1$ by the relation

$$k_p = \frac{1}{\epsilon_{p-1}} \left[r_{xx}[p] - \sum_{j=1}^{p-1} a_j^{(p-1)} r_{xx}[p-j] \right]. \tag{13}$$

So the loop for $1 \leqslant j \leqslant P$ proceeds in the following way:

- estimation of the reflection coefficient

$$k_j = \frac{1}{\epsilon_{j-1}} \left[r_{xx}[j] - \sum_{i=1}^{j-1} a_i^{(j-1)} r_{xx}[j-i] \right], \tag{14}$$

- at the jth stage the AR parameter of the model is equal to the jth reflection coefficient

$$a_j^{(j)} = k_j, \tag{15}$$

- the other parameters are updated in the following way. For $1 \leqslant i \leqslant j-1$

$$a_i^{(j)} = a_i^{(j-1)} - k_j a_{j-i}^{(j-1)} \tag{16}$$

$$\epsilon_j = (1 - k_j^2)\epsilon_{j-1}. \tag{17}$$

- At the end of the jth loop, when $j = P$, the final AR parameters are

$$a_j = a_j^{(P)}, \qquad \sigma^2 = \epsilon_P. \tag{18}$$

[7] Note that p indicates any order of the filter we choose for our model. We can always add a new stage to the filter, having an AR$(p+1)$ model

3. Whitening filter

3.1. Link between the AR model and the whitening filter

The tight relation between the AR filter and the whitening filter is clear in figure 2. The figure describes how an AR process colours a white process at the input of the filter if you look at the picture from left to right. If you read the picture from right to left you see a coloured process at the input that passes through the AR inverse filter coming out as a white process.

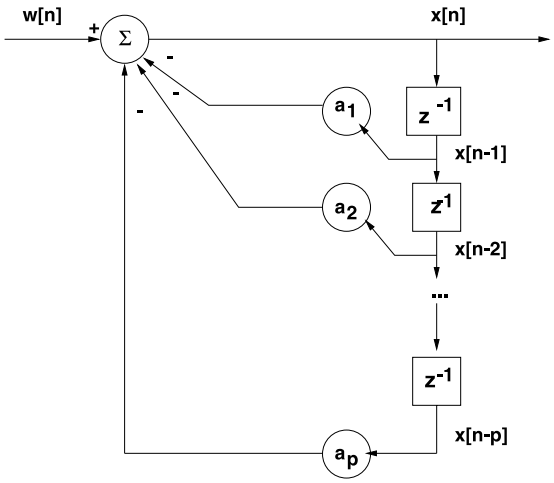

Figure 2. Whitening filter and AR filter.

When we find the P parameters that fit a PSD of a noise process, what we are doing is to find the optimal weights vector that lets us reproduce the process at the time n knowing the process at the previous time P. All the methods that involve this estimation try to make the error signal (see equation (9)) a white process in such a way as to throw out all the correlation between the data (which we use for the estimation of the parameters).

The Durbin algorithm introduces in a natural way the lattice structure for the whitening filter.

We show how the reflection coefficients k_p are used to build a lattice whitening filter. Let us suppose we have a stochastic Gaussian and a stationary process $x[n]$ which we modelled as an autoregressive process of order P. We define the *forward* error (FPE) for the filter of order P in the following way:

$$e_P^f[n] = x[n] + \sum_{k=1}^{P} a_k^{(P)} x[n-k], \tag{19}$$

where the coefficients a_k are the coefficients for the AR model for the process $x[n]$. The FPE represents the output of our filter. We can write the *zeta* transform for the FPE at each stage p for the filter of order P as

$$FPE(z) = F_p^f[z]X[z] = \left(1 + \sum_{j=1}^{p} a_j^{(p)} z^{-j}\right) X[z]. \tag{20}$$

At each stage p of the Durbin algorithm the coefficients a_p are updated as

$$a_j^{(p)} = a_j^{(p-1)} + k_p a_{p-j}^{(p-1)} \qquad 1 \leqslant j \leqslant p-1. \tag{21}$$

If we use the above relation for the transform $F_p^f[z]$, we obtain

$$F_p^f[z] = F_{p-1}^f[z] + k_p \left[z^{-p} + \sum_{j=1}^{p-1} a_{p-j}^{(p-1)} z^{-j} \right]. \tag{22}$$

Now we introduce in a natural way the *backward* error of prediction BPE

$$F_{p-1}^b[z] = z^{-(p-1)} + \sum_{j=1}^{p-1} a_{p-j}^{(p-1)} z^{-(j-1)}. \tag{23}$$

In order to understand the meaning of $F_p^b[z]$ let us see its action in the time domain

$$F_{p-1}^b[z]\, x[n] = e_{p-1}^b[n] = x[n-p+1] + \sum_{j=1}^{p-1} a_{p-j}^{(p-1)} x[n-j+1]. \tag{24}$$

So $e_{p-1}^b[n]$ is the error we make, in a backward way, in the prediction of the data $x[n-p+1]$ using $p-1$ successive data $\{x[n], x[n-1], \ldots, x[n-p+2]\}$.

We can write equation (22) using $F_{p-1}^b[z]$. Let us substitute this relation in the z-transform of the filter $F_p^f[z]$

$$F_p^f[z] = F_{p-1}^f[z] + k_p F_{p-1}^b[z]. \tag{25}$$

In order to know the FPE filter at the stage p we must know the BPE filter at the stage $p-1$.

Also for the *backward* error we may write in a similar way the relation

$$F_p^b[z] = z^{-1} F_{p-1}^b[z] + k_p F_{p-1}^f[z]. \tag{26}$$

Equations (25) and (26) represent our lattice filter that in the time domain could be written as

$$e_p^f[n] = e_{p-1}^f[n] + k_p e_{p-1}^b[n-1], \tag{27}$$

$$e_p^b[n] = e_{p-1}^b[n-1] + k_p e_{p-1}^f[n]. \tag{28}$$

Figure 3 shows how the lattice structure is used to estimate the forward and backward errors.

Using a lattice structure we can implement the whitening filter by following these steps:

- estimate the values of the autocorrelation function $\hat{r}_{xx}[k]$, $0 \leqslant k \leqslant P$ of our process $x[n]$;

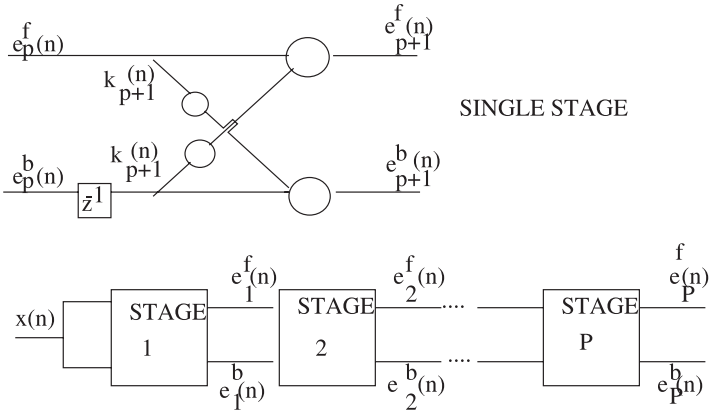

Figure 3. Lattice structure for the Durbin filter.

- use the Durbin algorithm to find the reflection coefficients k_p, $1 \leqslant p \leqslant P$;
- implement the lattice filter with these coefficients k_p, initiating the filter $e_0^f[n] = e_0^b[n] = x[n]$.

In this way the forward error at the Pth stage is equivalent to the forward error of a transversal filter and represents the output of the whitening filter.

The procedure of whitening will be accomplished before applying the algorithms for the detection of a gravitational signal of different wave forms. The level of whiteness of the data needed for the various algorithms could be different. It is important to have a common language and to assign a parameter which characterizes the performance of the whitening filter. We want now to introduce the parameter that lets us quantify the level of whiteness of the data at the output of the whitening filter.

3.2. The 'whiteness' of data: measure of flatness of PSD

The spectral flatness measure for a PSD is defined as [29]

$$\xi = \frac{\exp(\frac{1}{N_s}\int_{-N_s/2}^{N_s/2} \ln(P(f))\,\mathrm{d}f)}{\frac{1}{N_s}\int_{-N_s/2}^{N_s/2} P(f)\,\mathrm{d}f} \tag{29}$$

where the integral is extended in a bandwidth of Nyquist frequency; this parameter satisfies

$$0 \leqslant \xi \leqslant 1. \tag{30}$$

If $P(f)$ is very peaky, then $\xi \simeq 0$, if $P(f)$ is flat than $\xi = 1$.

With the definition (29) the flatness for a process at the output of a whitening filter built with a minimum phase filter (such as the AR filter is) is

$$\xi_e = \xi \frac{r_{xx}[0]}{r_{ee}[0]}, \tag{31}$$

where $r_{xx}[0]$ and $r_{ee}[0]$ are the values of the autocorrelation function of the process before and after the whitening procedure, and ξ is the value of flatness for the initial sequence [29].

4. Results on simulated VIRGO-like noise data

We want to investigate the performance of the Durbin filter in fitting the VIRGO PSD and in whitening the simulated output of this interferometer.

We can simulate the data as an AR or ARMA process [27], by fitting the theoretical PSD as an AR or ARMA model. If we simulate the data as an AR process and fit them with an AR model, the number of parameters we must use will be small. In the real situation the output of the interferometer will not be an AR process. This does not mean that we cannot ever fit the data as an AR process, but that probably we need a greater number of parameters. In order to be closest to the real situation we use an ARMA fit to the theoretical PSD and we performed the tests following the steps:

- *an ARMA fit to theoretical VIRGO-like noise PSD;*
- *generation of noise data with the ARMA parameters;*
- *realization of one noise process;*
- *Pth-order selection for the AR fit to the realization of the noise;*
- *Durbin(P) whitening filter.*

4.1. The VIRGO noise power spectrum

We consider a theoretical curve for a VIRGO-like power spectrum in which shot noise and thermal noise of pendulum, mirrors and violin modes are present:

$$S(f) = \frac{S_1}{f^5} + \frac{S_2}{f} + S_3\left(1 + \left(\frac{f}{f_K}\right)^2\right) + S_v(f) \tag{32}$$

where

$$f_K = 500 \text{ Hz} \qquad \text{shot noise cut-off frequency} \tag{33}$$

$$S_1 = 1.08 \times 10^{-36} \qquad \text{pendulum mode} \tag{34}$$

$$S_2 = 0.33 \times 10^{-42} \qquad \text{mirror mode} \tag{35}$$

$$S_3 = 3.24 \times 10^{-46} \qquad \text{shot noise.} \tag{36}$$

The contribute of violin resonances is given by

$$S_v(f) = \sum_n \frac{1}{n^4}\frac{f_1^{(c)}}{f}\frac{C_c\phi_n^2}{\left(\frac{1}{n^2}\frac{f^2}{f_1^{(c)2}} - 1\right)^2 + \phi_n^2} + (c \leftrightarrow f) \tag{37}$$

where we take into account the different masses of close and far mirrors, being

$$f_n^{(c)} = n\,327 \text{ Hz} \qquad f_n^{(f)} = n\,308.6 \text{ Hz} \tag{38}$$

$$C_c = 3.22 \times 10^{-40} \qquad C_f = 2.82 \times 10^{-40} \qquad \phi_n^2 = 10^{-7}. \tag{39}$$

The difference between far and close masses leads to the presence of double violin peaks, as we can see in figure 4, where we have plotted the spectrum obtained with a sampling frequency $f_s = 4096$ Hz.

We suppose we explore the band of frequencies from 10 Hz to ~ 2000 Hz, where it is most probable to find a gravitational signal, choosing a sampling frequency of 4096 Hz. The low-frequency part of the spectrum has been filtered to cut the tail of the thermal noise. We used a second-order high-pass filter with spectral density [40]:

$$|H(\omega)|^2 = \frac{1}{1 + \epsilon_{pass}^2\left(\frac{\cot(\omega/2)}{\cot(\omega_{pass}/2)}\right)^{2N}} \tag{40}$$

with the following values:

$$\epsilon_{pass} = 1000, \qquad N = 2, \qquad \omega_{pass} = 3\pi. \tag{41}$$

4.2. Data simulation

First of all we make an ARMA fit to the theoretical PSD with the techniques used in [27]. We choose to use $P = 32$ and $Q = 32$ parameters, then we simulated the data in the time-domain using relation (2), with a pre-heating techniques as described in [29].

In figure 4 we plot the theoretical VIRGO PSD and the ARMA(32, 32) fit, while in figure 5 we show the PSD obtained as an averaged periodogram on 50 realizations of the process for simulated data. As is evident the fit is good and we can suppose the time domain data to well represent the expected Gaussian and stationary noise process for the VIRGO interferometer.

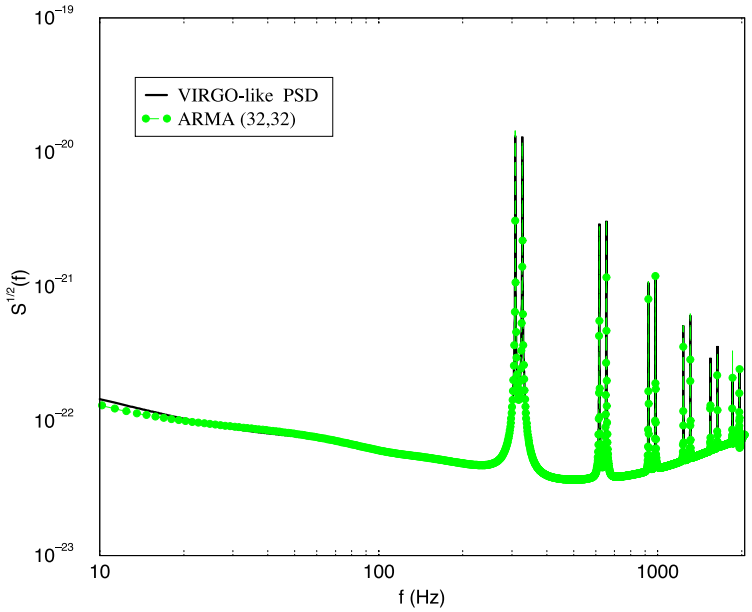

Figure 4. ARMA(32, 32) fit to theoretical VIRGO PSD.

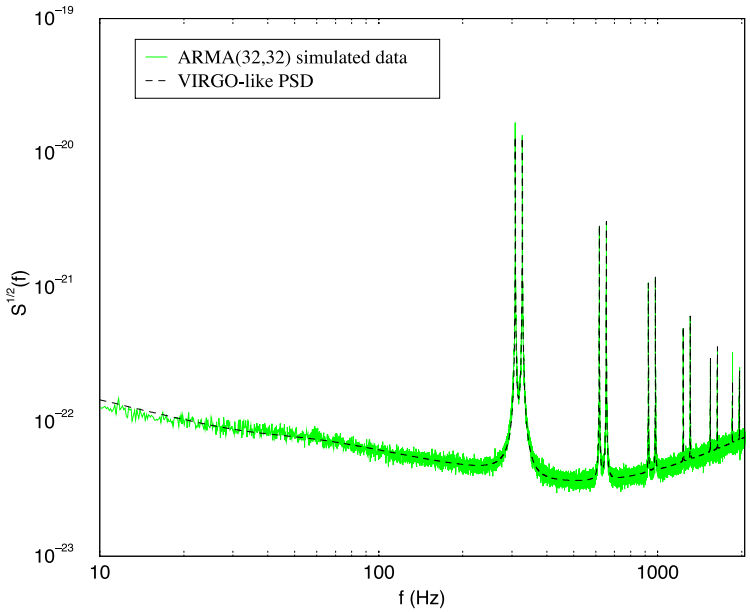

Figure 5. PSD of VIRGO-like simulated data.

4.3. Order selection

The idea of the whitening filter is that the process we analyse is an autoregressive one and that once we have the AR parameters we can use them in the filter of figure 2.

In general we do not know the order of our process, even if we suppose that it is an AR one. If it is an AR of order P, and we use an order $p < P$, the fitted spectrum will be smoother than the original one; if we choose an order $p > P$, there may be spurious peaks in the spectrum. In both cases the whitening will not be good.

If our process is not AR, the number of parameters could be, in principle, infinite. We must then fix a criterion that lets us select the right order of the process, or at least the best one.

We used the classical order selection criteria [29, 33], that is the Akaike information criterion (AIC), the forward prediction error (FPE), Parzen's criterion (CAT) [38] and the minimum description length (MDL) one:

$$\mathrm{AIC}(P) = N \log \epsilon(P) + 2P, \tag{42}$$

$$\mathrm{FPE}(P) = \epsilon(P) \frac{N + P + 1}{N - P - 1}, \tag{43}$$

$$\mathrm{CAT}(P) = \left(\frac{1}{N} \sum_{j=1}^{P} \frac{N - j}{N \epsilon_j} \right) - \frac{N - P}{N \epsilon_P}, \tag{44}$$

$$\mathrm{MDL}(P) = N \log \epsilon(P) + P \log N, \tag{45}$$

where $\epsilon(P)$ is the mean-square error at the order P and N is the length of the data. In the literature the MDL criterion is considered to be the best among them, because it is robust with respect to the length of the sequence, while the others depend a lot on N [33]. Suppose, as in the real situation, we do not have access to the theoretical PSD of our noise process and we want to estimate the best order of the whitening filter. We can use a single realization of process, i.e. a sequence of N-data to estimate the autocorrelation function and apply the selection order criteria to it. The results of these criteria are reported in table 1.

Table 1. Minimum of order selection criteria on a single sequence for 1 min of ARMA simulated VIRGO data.

MDL	AIC	FPE	CAT
338	626	626	681

The best order to whiten the data is given by the MDL criterion [29, 33] which produces of the order of 338 parameters.

This number is an indicative one. We can choose to build a higher-order filter to be sure to have whitened data at the output of the filter. We choose to adopt the number of parameters of the MDL criterion and we test the flatness of the spectrum at the output of the Durbin filter measuring the value of ξ.

All the order selection criteria give an estimation of the number of parameters such that the output of the whitening filter has the maximum value of ξ.

In figure 6 we report ξ versus the number of parameters of the whitening filter. The value of ξ for very low order P is small and, as expected, it increases with the order P until it converges to a plateau around $P \sim 300$. So we deduce that our choice of $P = 338$ is a good estimation of the whitening filter order.

4.4. Results of the Durbin whitening filter on simulated VIRGO-like noise data

In figure 7 we plotted the averaged PSD on 100 realizations of the input noise and of the output of the Durbin whitening filter. The results are good even if there are some residual lines at high frequencies.

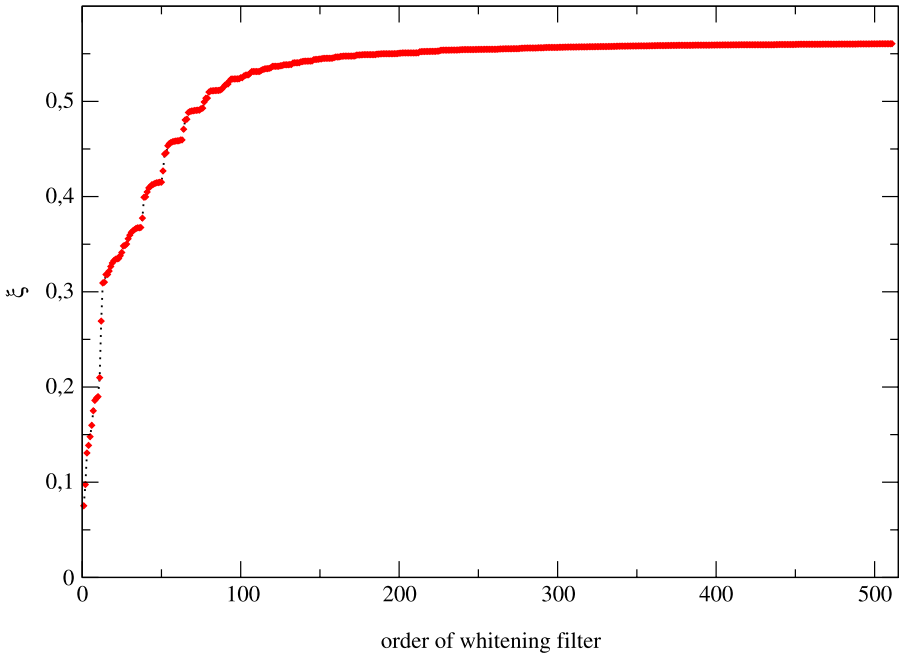

Figure 6. Behaviour of ξ with respect to the order P for the VIRGO-like simulated data.

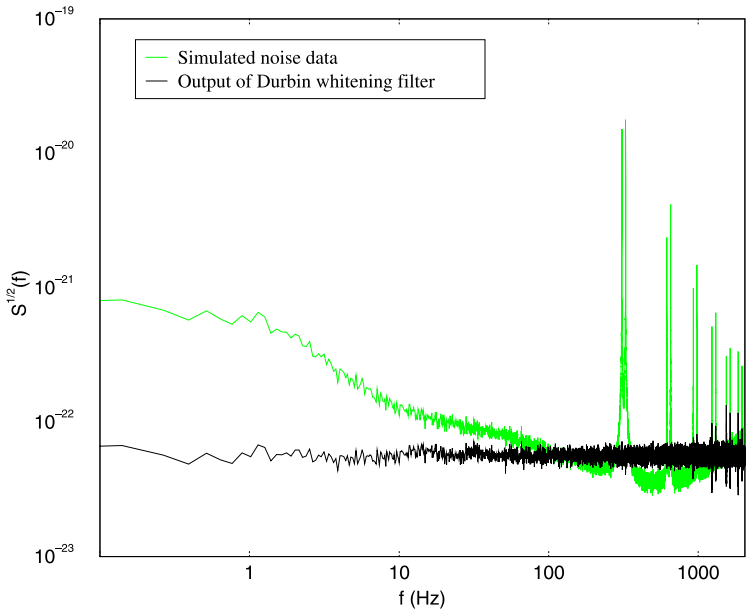

Figure 7. Output of the Durbin whitening filter.

We can perform a better whitening if we take a higher-order whitening filter, but we must pay a higher computational cost itself, because it is proportional to the order P of the filter. The level of whitening we choose to perform depends on the requests for the detection algorithms.

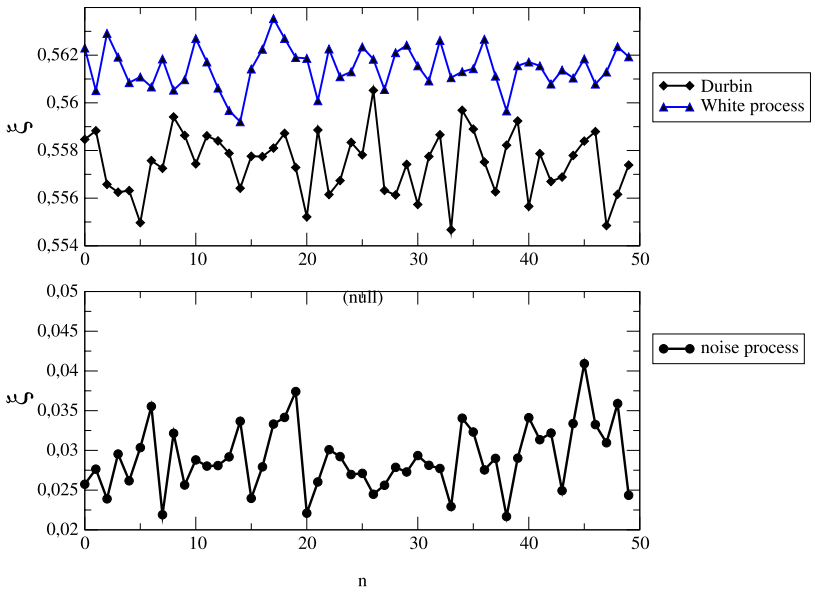

Figure 8. Measure of flatness for the simulated noise process and outputs of whitening filters.

Table 2. Flatness on averaged PSD at the input and the outputs of the whitening filter for VIRGO-like simulated data.

Simulated noise	Durbin	White process
0.050	0.983	0.989

In figure 8 we report the measure of ξ in a set of realizations of the process of simulated noise. The values are reported before any application of the whitening filter and after the application of a 'static' whitening filter. We also estimate the value of flatness for a simulated white noise process to check the goodness of the whitening filters.

As we can see in figure 8, in a single realization the value of flatness is high but not equal to 1, because of the variance of the estimate of the periodogram. On the other hand, in the averaged periodogram the value of ξ is very close to 1 as we can see in table 2.

5. Adaptive filters

We have shown filters which estimate the parameters to be used in the whitening filter from the autocorrelation function or PSD, i.e. these filters use *a priori* information about the statistics of the data to be analysed. Now we want to investigate the behaviour of filters which are self-designing [31, 39]. These filters estimate the parameters directly from the data, adjusting themselves by using as feedback the signal obtained by the minimization of a cost function of the error signal. In figure 9 we report the scheme of an adaptive filter for system identification. In our case the plant is represented by the parameters which fit the PSD of data. The implementation of an adaptive filter follows two steps: the filtering of the input data and the adjustment of the filter parameters with which we process the data to the next iteration.

The filter parameters are updated by minimizing a cost function. The way in which we build this cost function distinguishes the adaptive methods as [31, 32]:

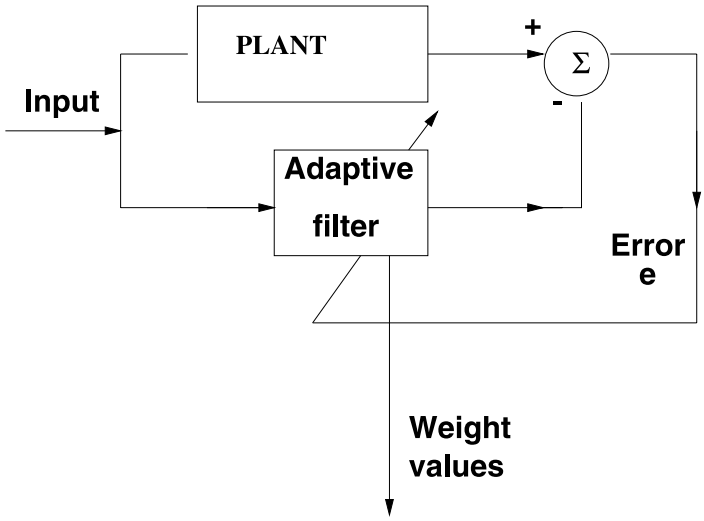

Figure 9. Scheme of an adaptive filter for system identification.

- methods of stochastic gradient,
- least-squares methods.

To the first class belong the algorithms whose cost function is the mean-square error $\mathcal{E}[e^2[n]]$, where $e[n]$ is the difference between the function we desire to find and the output of our filter. We talk of stochastic methods because the cost function is a statistical measure of the error. In the second class, the cost function is the weighted sum of the square errors $e^2[n]$. These methods could be implemented with a block estimation or with a recursive one (recursive least squares). For the block estimation a block of data is acquired and then the least-squares algorithm is applied, while in the recursive case the least-squares methods should be implemented in a recursive way.

In order to be able to obtain the fit to the PSD on-line, we will use only the recursive kind. We used the gradient adaptive lattice (GAL), the recursive least squares (RLS) and the least-squares lattice (LSL) algorithms. The technical details of these algorithms are described in [31, 32].

The next sections of this paper are organized into two parts: in the first part we make a comparison of the GAL, RLS and LSL methods to fit a VIRGO-like noise PSD. For this we simulated the data as an autoregressive process; we consider the parameters we use in the simulation as the true values and we check the capability of the algorithm to converge towards these values. In the second part, we will report the application of the LSL methods as whitening filters on data simulated as an ARMA process, to show that even in the case of a process which is not an autoregressive one, we succeed in fitting it with an AR with a low number of parameters and in obtaining a whitened PSD.

To check the performance of these algorithms we use the following scheme:

- modelling of the VIRGO as an AR process using the Durbin algorithm;
- data simulation of the noise process following the AR relation

$$x[n] = -\sum_{k=1}^{P} a_k x[n-k] + w[n]; \tag{46}$$

- implementation of an adaptive algorithm without any information on the input sequence of data;
- comparison between the estimated PSD and the obtained one.

It is fundamental to test the time convergence of the algorithms to compare them with the typical times of non-stationarities. To this end we measure the number of iterations by which the measured values reach the true values of the parameters. This is done in the case of the AR simulation, where the two quantities are comparable.

If we simulate the process as an AR one, the MDL order selection criterion gives as the best order the value 292. We select this one to perform our simulation and our tests.

6. Application to VIRGO-like simulated data

We applied the GAL method on simulated data to verify its capability in identifying the VIRGO-like noise power spectrum.

The convergence is reached after 2 min of data, but not for all the coefficients, as is evident in figure 10, where we have plotted all the 292 coefficients and have given expanded views of the regions corresponding to $p = 1, 2 \ldots, 50$ and $p = 50, 51, \ldots, 100$. After the first 50 points there is an evident discrepancy between the simulated and the estimated reflection coefficients. This causes the non-convergence of the AR parameters even for the first two coefficients (see figure 11). Even if we use a larger number of iterations, convergence is not reached. This is reflected in the estimate of the PSD, as one can see in figure 12 where the estimated GAL PSD is reported. The violin peaks are reproduced only in a rough way. This is due to the kind of cost function we used to find the reflection coefficient, which is only optimal in a statistical sense and not for the actual value of the error function.

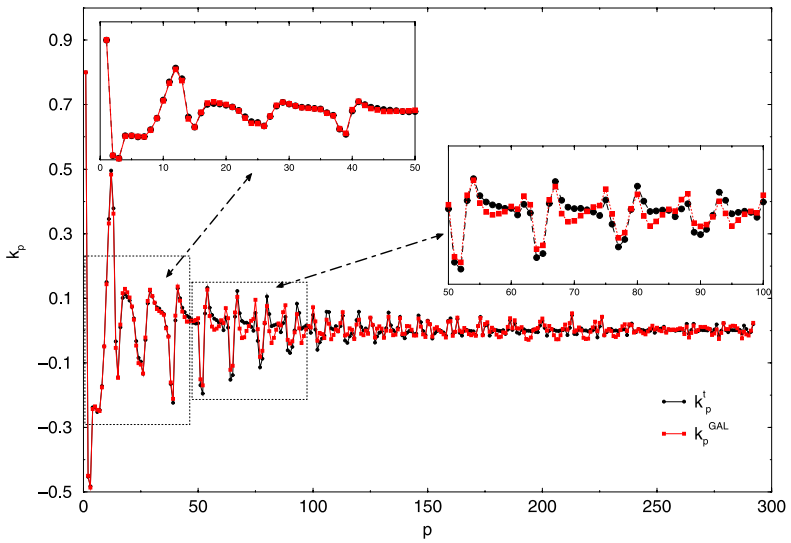

Figure 10. Comparison between the reflection coefficients estimated by the Durbin and GAL algorithms.

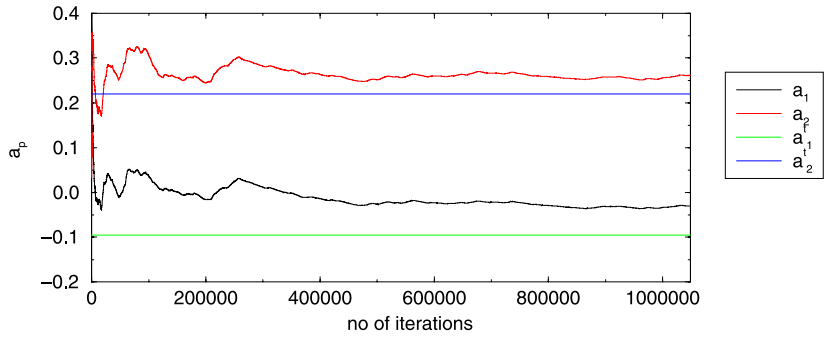

Figure 11. Convergence of the AR coefficients to true values after 4 min of data.

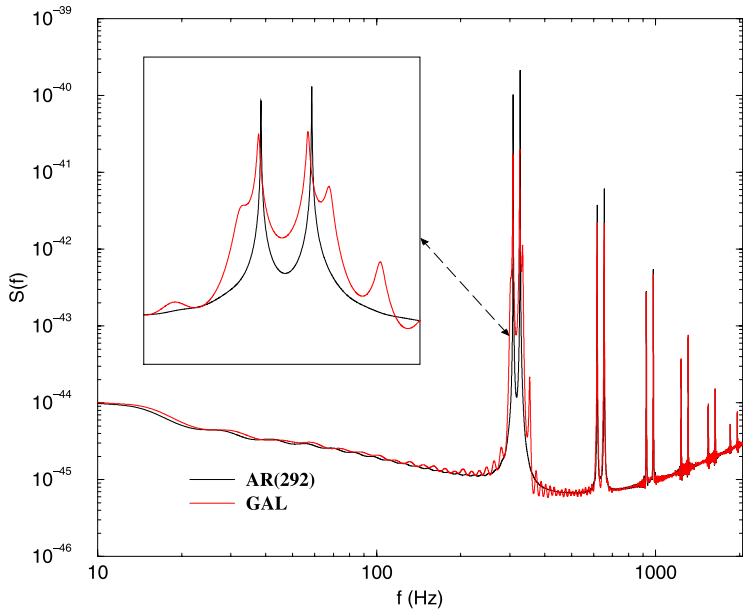

Figure 12. GAL fit to the VIRGO PSD.

7. Least-squares-based methods

The least-squares-based methods build their cost function using all the information contained in the error function at each step, writing it as the sum of the error at each step up to iteration n:

$$\epsilon[n] = \sum_{i=1}^{n} \lambda^{n-i} e^2(i|n), \tag{47}$$

where

$$e(i|n) = d[i] - \sum_{k=1}^{N} x_{i-k} w_k[n], \tag{48}$$

with d being the signal to be estimated, x the data of the process and w the weights of the filter. We introduced the forgetting factor λ that let us tune the learning rate of the algorithm. This

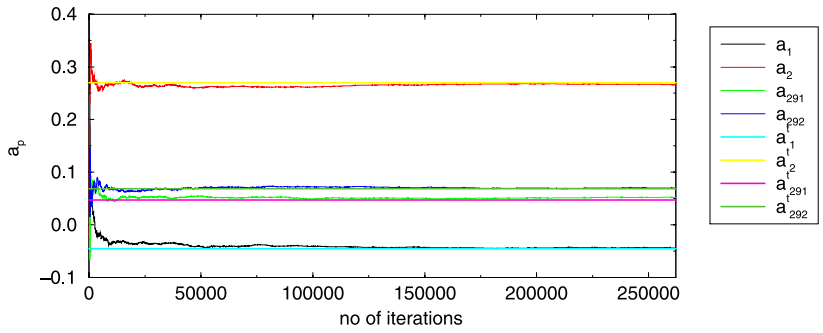

Figure 13. Convergence of the first two and the last two AR parameters for the RLS filter.

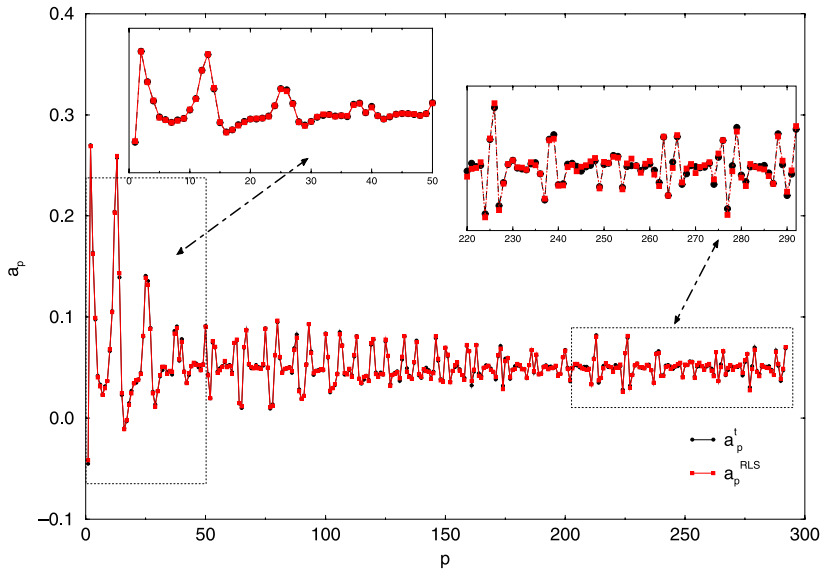

Figure 14. Simulated AR parameters and estimated coefficients using the RLS algorithm.

coefficient can help when there are non-stationary data in the data and we want the algorithm to have a short memory. If we have stationary data we fix $\lambda = 1$.

There are two ways to implement the least-squares methods for the spectral estimation: in a recursive way (recursive least squares or Kalman filters) or in a lattice filters using fast techniques [32]. The first kind of algorithm, examined in [28], has a computational cost proportional to the square of the order of filter, while the cost of the second one is linear in the order P.

7.1. RLS: application to VIRGO noise data

We used about 1 min of data, choosing a sampling frequency of 4096 Hz.

In this transverse filter we update the weights directly, without estimating the reflection coefficients. In figure 13 we report the convergence curves for the first two coefficients a_1, a_2 and for the last two coefficients a_{291}, a_{292} estimated by the RLS algorithms and the

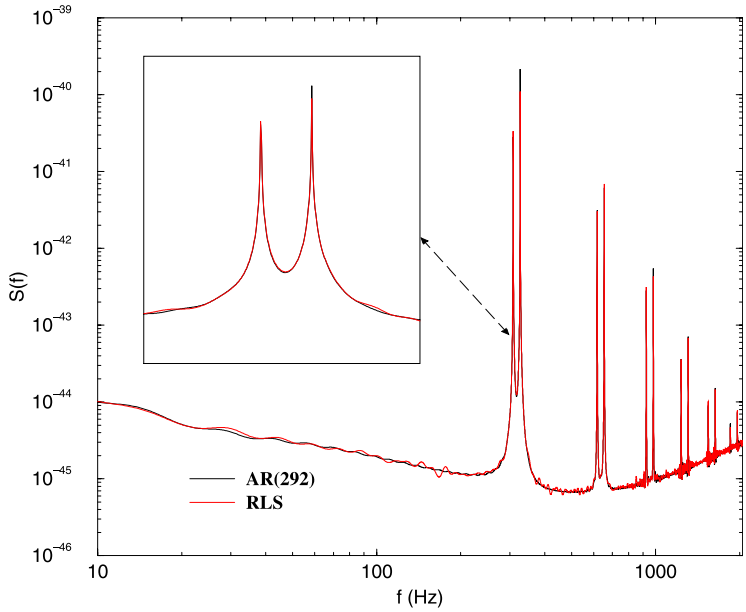

Figure 15. RLS fit to the VIRGO PSD.

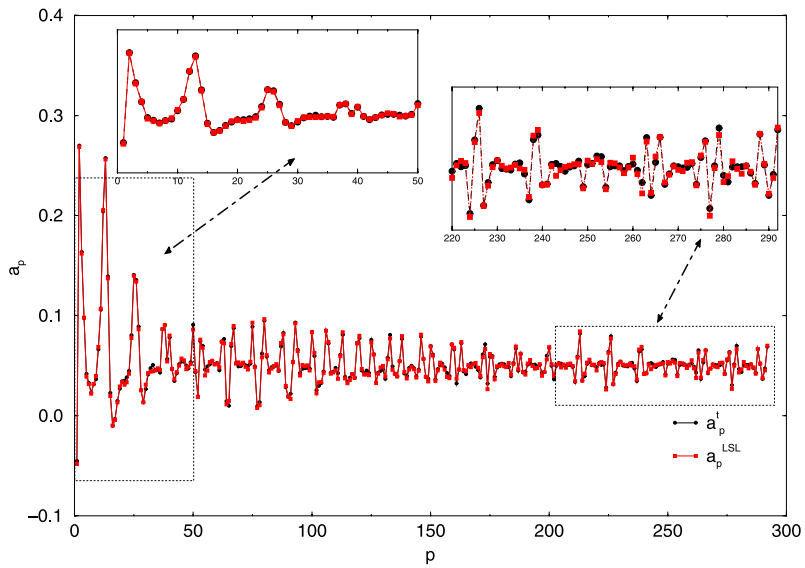

Figure 16. LSL estimated AR parameters.

corresponding 'true' values a_1^t, a_2^t, a_{291}^t and a_{292}^t. The RLS algorithm converges to the true value of the parameters, and its convergence time is of the order of 30 s.

In figure 14 we report the 292 parameters for the AR model estimated with RLS after 1 min of iterations and the corresponding true values. In the insets we can see the first 50 and the last 70 coefficients. There is a small discrepancy in the estimations of the last coefficients,

but this does not affect the fit of the original PSD as is evident in figure 15, where all the spectral features are well reproduced.

7.2. LSL: application to VIRGO-like noise data

The computational cost of RLS is prohibitive for an on-line implementation. Moreover, its structure is not modular as for the GAL algorithm, thus forcing the choice of the order P once and for all. The algorithm with a modular structure like that of the lattice offers the advantages of giving an output of the filter at each stage p, so, in principle, we can change the order of the filter by imposing some criteria on its output. In contrast, the least-square lattice filter is a modular filter with a computational cost proportional to the order P.

We introduced for the LSL filter the forward and backward reflection coefficients. These, in principle, could have different values if the sequence is not stationary, but we simulate the VIRGO-like noise data as a stationary process, therefore $k_p^f = k_p^b = k_p$. Moreover, we use the pre-windowed case $\lambda = 1$.

We always used 1 min of data with a sampling frequency of 4096 Hz. The AR parameters have been estimated from the reflection coefficients using relation (21).

The error e_p^f at the last stage is the whitened sequence of the input data. So at the output of the LSL filter we find the parameter for the estimation of the AR fit to the VIRGO PSD and the whitened sequence of data.

As we expected, the performance is similar to that of the RLS filter and the convergence is reached after about 30 s of data. This behaviour is satisfied by all the coefficients a_p as is evident in figure 16 where we have plotted all the coefficients of simulation and the estimated ones.

We have zoomed in on the first 50 coefficients and on the last 70. As for the RLS filter there is a small discrepancy only for the last coefficients, but this does not affect the spectral estimation as reported in figure 17, where all the violin peaks are well reproduced. In only 1 min of data we succeeded in identifying an AR model with 292 parameters. If we think about non-stationary noise with characteristic time of 1 h, we are certain to obtain the right estimation of the PSD on-line, and the right whitened sequence.

In figure 18 we reported the PSD of the sequence $e_P^f[n]$ obtained as an averaged periodogram on 100 noise simulations.

It is clear that the LSL is a good whitening filter and it offers the advantages, with respect to the Durbin one, of being adaptive and of working without estimating the autocorrelation function before hand from the data.

7.3. LSL statistics

In order to evaluate the goodness of an estimator we verify whether it is an unbiased one and if it satisfies the Cramer–Rao bound which for the AR parameters are [29]

$$\mathrm{var}(\hat{a}_i) \geqslant \frac{\sigma^2}{N}[\boldsymbol{R}_{xx}^{-1}]_{ii} \qquad i = 1, 2, \ldots p, \tag{49}$$

and

$$\mathrm{var}(\hat{\sigma}) \geqslant \frac{2\sigma^2}{N}. \tag{50}$$

We fixed 1 min of data as the maximum length for N and we estimated the bias for the parameters a_p. The statistical quantities have been evaluated as averages on 100 realizations of the process. In figure 19 we report the bias for each AR parameter

$$B(a_p) = \mathcal{E}[\hat{a}_p] - a_p^t \tag{51}$$

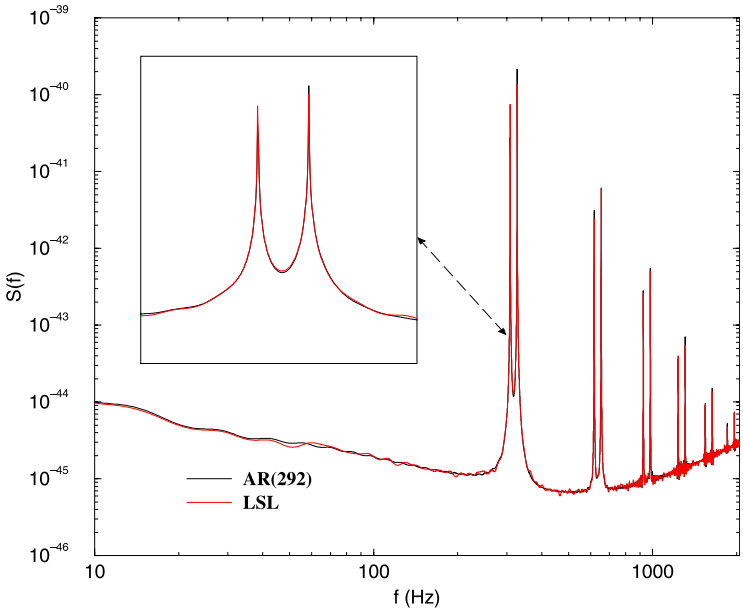

Figure 17. LSL fit to the VIRGO-like noise PSD.

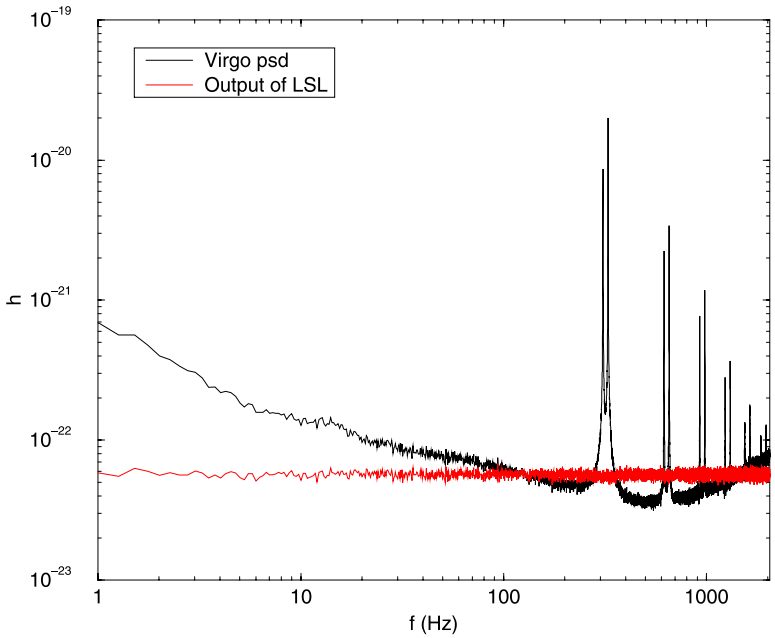

Figure 18. Output of the LSL whitening filter.

estimated at two different times: the first at 8 s of data and the second at 64 s. It is evident that the quantities $B(a_p)$ are equal to zero after 1 min of data.

Now to evaluate the efficiency of the estimator LSL we verify that the variance for the estimated coefficients a_p reaches the Cramer–Rao (equation (49)) limit.

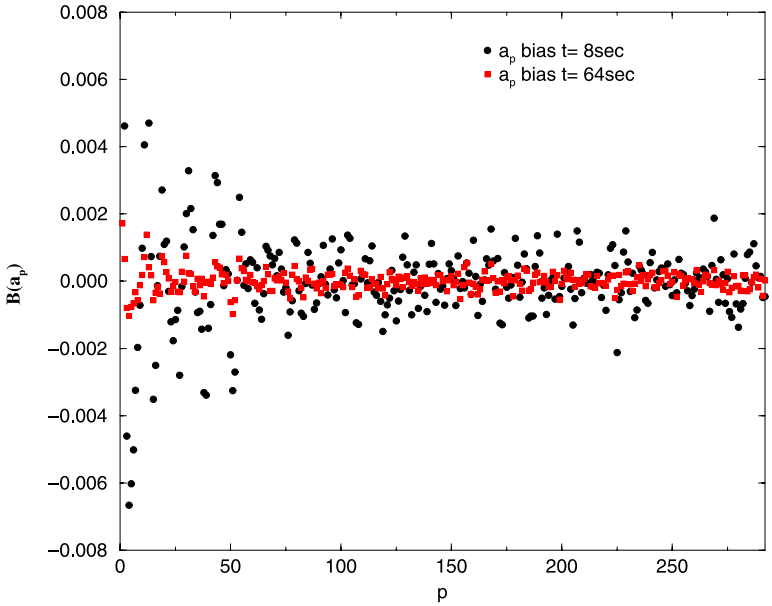

Figure 19. Bias for the AR parameters.

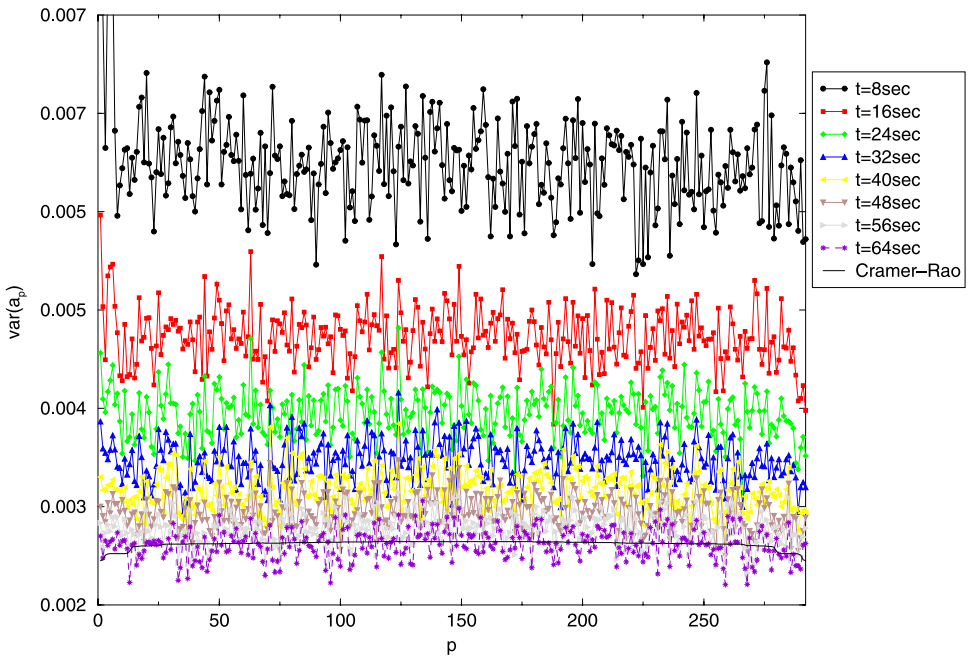

Figure 20. Cramer–Rao bound for LSL parameters.

In figure 20 we report the estimated variance for the coefficients a_p at the output of the LSL filter and the theoretical Cramer–Rao bound. The variance has been estimated at steps of

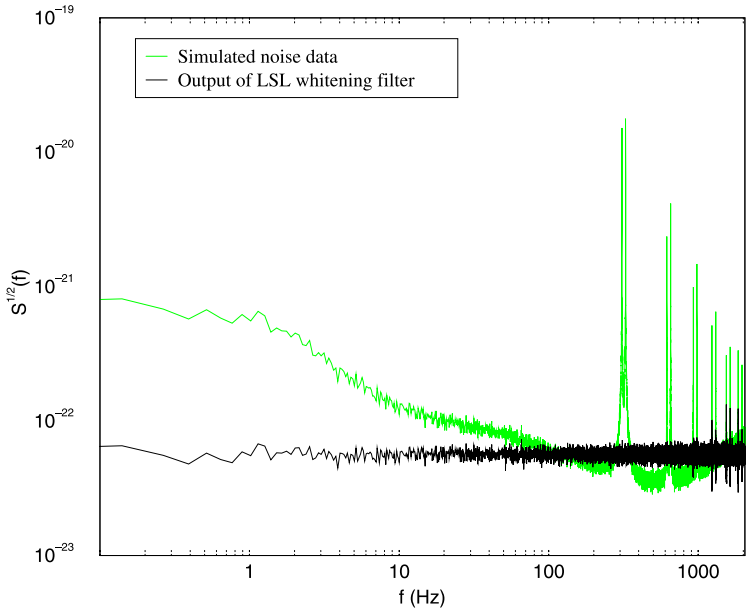

Figure 21. Output of the LSL whitening filter.

Table 3. Flatness on averaged power spectra at the input and the outputs of the whitening filter for VIRGO-like simulated data.

Simulated noise	Durbin	LSL	White process
0.050	0.983	0.984	0.989

time growing until the limit of 64 s. It is evident that its values become smaller and smaller, increasing the number of iterations and that it reaches the Cramer–Rao theoretical limit [28].

7.4. If the noise is not an autoregressive process

Suppose that our process is not an autoregressive one: does the LSL work well in this situation? To verify this we simulated, as in the Durbin case, the data as an ARMA process and tested the LSL on this sequence of data. The optimal order for the AR fit to these data now is 338, so we use this order for the LSL final stage.

In figure 21 we plotted the PSD of the simulated data and at the output of the LSL filter averaged over 100 realizations. It is evident that the LSL also succeeded in whitening the ARMA sequence, even if we use an AR fit. In table 3 we reported the values of flatness for the outputs of the Durbin and the LSL filter.

The values of flatness for the LSL and the Durbin whitening filter are similar, even if it is evident that LSL whitens better than the Durbin filter. We think that this is related to the fact that the adaptive filters do not need the previous estimation of the autocorrelation function; in this way if we make a mistake in estimating the autocorrelation it will not propagate in the estimation of the coefficients.

8. Conclusion

In this paper we have addressed the problem of on-line identification of the parameters which fitted the PSD at the output of an interferometric detector such as the VIRGO one. Moreover, we face the problem of whitening the sequence of data on-line.

In this work we have reviewed the Durbin and LSL whitening algorithms and we reported the results of whitening on VIRGO-like noise simulated data, showing that it is possible to obtain a whitened PSD. We verified that the LSL adaptive algorithm has a better performance with respect to the static algorithm so it could be useful if we are faced with non-stationary data. It is important to note that in selecting the order of the whitening filters a good knowledge of the level of whiteness needed for the different signal detection algorithms is crucial. In fact, we showed that the value of flatness tends to reach a plateau with respect to the number of parameters used in the whitening filters, while the computational cost increases proportionally to the order of the filter.

The procedure of whitening we described is a linear procedure that does not destroy any part of the data. It is a reversible process that can be updated to the level of whiteness we need. It is worth noting that the Durbin algorithm is a 'static' procedure: we suppose we know that we are analysing only noise and we fit the parameters to perform the whitening on the next sequence of data. In this way if in the next sequence of data there is a signal, we do not white the signal even if we can modify its waveform. Note that we have under control the kind of changes we made to the signal, because we know the parameters of our whitening filter. Instead when we use the adaptive algorithms we could in principle let the algorithm also learn the signal buried in the noise, and whiten all the information about it, but the learning time of the algorithm we described is such that only a periodic signal can be captured by this algorithm. In a forthcoming paper we will report the tests we have made on the whitening procedure when applied to sequences of data containing gravitational signals [41].

The goodness of filters such as whitening filters has already been tested on data taken from prototype interferometers with very encouraging results [30, 42].

References

[1] Saulson P R 1994 *Fundamentals of Interferometric Gravitational Wave Detectors* (Singapore: World Scientific)
[2] Schutz B 1999 *Class. Quantum Grav.* **16** 131
[3] Grishchuk L P *et al* 2001 *Phys.–Usp.* **171** N1 3
(Grishchuk L P *et al* 2000 *Preprint* astro-ph/0008481)
[4] Ando M *et al* 2000 *Proc. 3rd E Amaldi Conf.* (New York: AIP Proceedings) p 128
[5] Luck H *et al* 2000 *Proc. 3rd E Amaldi Conf.* (New York: AIP Proceedings) p 119
[6] Coles M W 2000 *Proc. 3rd E Amaldi Conf.* (New York: AIP) p 101
[7] Marion F 2000 *Proc. 3rd E Amaldi Conf.* (New York: AIP Proceedings) p 110
[8] Blair D G (ed) 1993 *The Detection of Gravitational Waves* (Cambridge: Cambridge University Press)
[9] Barone M, Calamai G, Mazzoni M, Stanga R and Vetrano F (ed) 2000 *Proc. Int. Summer School on Experimental Physics of Gravitational Waves* (Singapore: World Scientific)
[10] Owen B J and Sathyaprakash B S 1999 *Phys. Rev.* D **60** 22 002
[11] Pradier T *et al* 2000 *Int. J. Mod. Phys.* D **9** 309
[12] Pradier T *et al* 2001 *Phys. Rev.* D **63** 042002
(Pradier T *et al* 2000 *Preprint* gr-qc/0010037)
[13] Bradaschia C *et al* 1990 *Nucl. Instrum. Methods* A **289** 518
[14] Cagnoli G *et al* 1999 *Phys. Lett.* A **255** 230
[15] Meers B J 1988 *Phys. Rev.* D **38** 2317
[16] Thorne K S and Winstein C J 1999 *Phys. Rev.* D **60** 082001
[17] Hughes S A and Thorne K S 1998 *Phys. Rev.* D **58** 122002
[18] Cella G and Cuoco E 1997 VIR-NOT-PIS-1390-099 *Internal note*

[19] Beccaria M *et al* 1998 *Class. Quantum Grav.* **15** 3339
[20] Cella G 1999 Off-line subtraction of Newtonian noise *Gravitational Waves and Experimental Gravity: Proc. 34th Rencontre de Moriond Conf. (Les Ares, Jan. 1999)*
[21] Cagnoli G *et al* 1998 *Phys. Lett.* A **237** 21
[22] Braginsky V B, Levin Y and Vyatchanin S 1999 *Meas. Sci. Technol.* **10** 598
[23] Cattuto C *et al* 1999 VIR-NOT-PER-1390-51 *Internal note*
[24] Finn L S and Mukherjee S 2001 *Phys. Rev.* D **63** 062004
(Finn L S and Mukherjee S 2000 *Preprint* gr-qc/0005061)
[25] Sintes A M and Schutz B F 1998 *Proc. 2nd Workshop on Gravitational Wave Data Analysis* ed M Davier and P Hello (Gif-sur-Yvette: Editions Frontières) p 255
[26] Chassande-Mottin E and Dhurandhar S V 2001 *Phys. Rev.* D **63** 042004
(Chassande-Mottin E and Dhurandhar S V 2000 *Preprint* gr-qc/0004075)
[27] Cuoco E and Curci G 1997 Modeling a VIRGO-like noise spectrum. Note IVIR-NOT-PIS-1390-095 *Internal note*
[28] Beccaria M, Cuoco E and Curci G 1997 Adaptive system identification of VIRGO-like noise spectrum *Proc. 2nd Amaldi Conf.* (Singapore: World Scientific)
[29] Kay S 1988 *Modern Spectral Estimation. Theory and Application* (Englewood Cliffs, NJ: Prentice Hall)
[30] Cuoco E 2000 Whitening of noise power spectrum. Test on LIGO 40-meter interferometer data VIR-NOT-FIR-1390-145 *Internal note*
[31] Haykin S 1996 *Adaptive Filter Theory* (Upper Saddle River: Prentice Hall)
[32] Alexander S T 1986 *Adaptive Signal Processing* (Berlin: Springer)
[33] Hayes M H 1996 *Statistical Digital Signal Processing and Modeling* (New York: Wiley)
[34] Allen B 1999 GRASP: a data analysis package for gravitational wave detection
[35] Allen B and Brady P 1997 Quantization noise in Ligo interferometer Ligo *Internal note*
[36] Therrien C W 1992 *Discrete Random Signals and Statistical Signal Processing* (Englewood Cliffs, NJ: Prentice Hall)
[37] Zubakov L A and Wainstein V D 1962 *Extraction of Signals from Noise* (Englewood Cliffs, NJ: Prentice Hall)
[38] Parzen E 1977 *An Approach to Time Series Modeling: Determining the Order of Approximating Autoregressive Schemes in Multivariate Analysis* (Amsterdam: North-Holland)
[39] Widrow B and Stearns S D 1985 *Adaptive Signal Processing* (Englewood Cliffs, NJ: Prentice Hall)
[40] Orfanidis S J 1996 *Introduction to Signal Processing* (Englewood Cliffs, NJ: Prentice Hall)
[41] In preparation
[42] Cuoco E *et al* 2001 Noise parametric identification and whitening for LIGO 40-meter interferometer data *Phys. Rev.* D submitted