CNRS

Centre National de la Recherche Scientifique

INFN

Istituto Nazionale di Fisica Nucleare

VIRGO

An utility for VIRGO data simulation, i.e. how to build noise data from the knowledge of the spectrum

Code: VIR-NOT-ROM-1390-090

Issue:1

Date:May 97

Authors S. Frasca - M.A. Papa

VIRGO * A joint CNRS-INFN Project
Project Office: INFN-Sezione di Pisa * Via Livornese, 1291-56010 San Piero a Grado, Pisa.Italy.
Secretariat: Telephone.(39) 50 880 327 or 880 352 * FAX.(39) 50 880 350 * e-mail virgo@Pisa.infn.it

An utility for VIRGO data simulation. i.e. how to build noise data from the knowledge of the spectrum

S.Frasca M.A. Papa

22-24 april 1997

1 Introduction

In filter implementation. the very first step after having written the code consists in checking errors. i.e. that the code runs well. that the structure used to implement the filter is correct. In order to do this check it is not necessary to have a continuos stream of data. nor that the data are a refined simulation of the noise on which the filters will really work. It is enough to have sets of data. say a few hours. representing the main features of the expected noise. Producing these data is not such a difficult task and it can be done in different ways. We have developed a straighforward strategy which is well suited for noise with spectral features such as those expected for the VIRGO interferometer. This utility can be easily adapted to run on any platform. At present it is written in FORTRAN and it has been inserted as an option in the menu of our SNAG statistical environment program (Signal and Noise for Gravitational Antennas) running on VMS operating systems.

2 Noise simulation

As long as signal and noise add up linearly. the so called "reconstructed h" output of the antenna can be modeled as the sum of two independent contributions: $s(t)$ due to the signal and $n(t)$ due to the noise. The former depends on the shape of the gravitational excitation $h(t)$ and on the total transfer function of the antenna + the h-reconstruction technique. The latter depends only on the noise spectrum. In fact. if the noise is a gaussian process it is well known from elementary statistics that it is completely determined in terms of its mean and autocovariance. The mean of our process is zero. thus the only information needed to construct the process is encoded in its spectrum $S(j\omega)$. In particular. the noise $n(t)$ can be generated by feeding a

/

gaussian zero mean white noise into a system with transfer function $N(j\omega)$. such that $|N(j\omega)|^2 = S(j\omega)$. The phase information is irrelevant. and **any process generated by a system $N'(j\omega) = e^{j\theta(\omega)}N(j\omega)$ would be undistinguishable, whatever the value of $\theta(\omega)$.**

2.1 Wideband noise

In principle the procedure is trivial: fixed a sampling time. and given an array of n real numbers, N_i, representing the spectrum from frequency zero to the Nyquist frequency, one must produce a set of n complex numbers. X_i, which is the FFT of a fictitious real white zero mean gaussian noise. Then one computes $Y_i = X_i \cdot N_i$. anti-FFT and gets an array of real numbers which represent the time samples of the noise one wanted to simulate. Obviously the arrays will really have $2 \times n$ components, but these can be built from the first n. in fact for the spectrum and for the array Y_i they are the complex conjugate of the first n. In practise this procedure should be iterated many times in order to produce large sets of data and to avoid having to compute too large FFT's. But when the data are divided in chunks, some care must be put in how the different chunks are connected, due to the fact that the FFT's of the white noise data chunks do not come from a continuos time stream but are computed separatly in the frequency domain. We have obtained a smooth connection by the following windowing procedure. If Y_i^m is the ith data of the mth chunk then $\mathcal{G}_i$ of the properly windowed stream is

$$\mathcal{G}_i = \frac{1}{A_i}(Y_i^{m-1} \cdot W_i^{(1)} + Y_{i+n}^{m} \; W_i^{(2)}) \tag{1}$$

with $1 \leq i \leq n$ and

$$\begin{cases} W_i^{(1)} = \dfrac{2i}{n} \\ W_i^{(2)} = 1 - W_i^{(1)}. \end{cases} \tag{2}$$

$$A_i = \sqrt{W_i^{(1)^2} + W_i^{(2)^2}}$$

is a normalization factor for the variance of the different samples. Other windowing options than that of eq. 2 can also be used:

$$\begin{aligned} W^{(1)'}_i &= W_i^{(1)^2} \\ W^{(2)'}_i &= (1 - W^{(1)}{}_i)^2 \end{aligned} \tag{3}$$

or

$$\begin{aligned} W^{(1)'}_i &= \sqrt{W_i^{(1)}} \\ W^{(2)'}_i &= \sqrt{1 - W_i^{(1)}} \end{aligned} \tag{4}$$

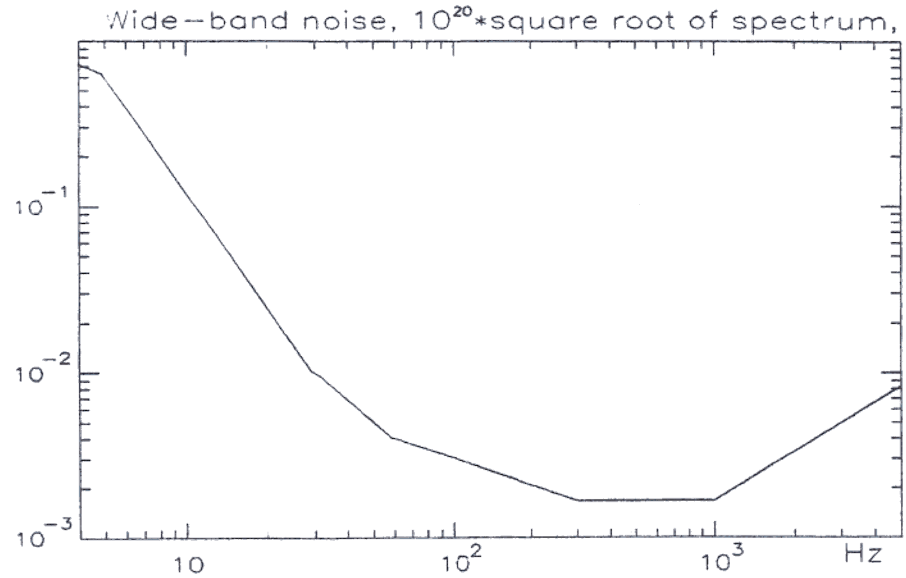

Figure 1: Wide band noise N real array used to simulate the data.

or

$$\begin{cases} W^{(1)\prime}_i = \dfrac{1+\cos\left[\pi(1-W_i^{(1)})\right]}{2} \\ W^{(2)\prime}_i = \dfrac{1+\cos\left[\pi W_i^{(1)}\right]}{2} \end{cases} \tag{5}$$

or

$$\begin{aligned} W^{(1)\prime}_i &= e^{-(1-W_i^{(1)})^2}\sqrt{W_i^{(1)}} \\ W^{(2)\prime}_i &= e^{-W_i^{(1)2}}\sqrt{1-W_i^{(1)}} \end{aligned} \tag{6}$$

or

$$\begin{cases} W^{(1)\prime}_i = W_i^{(1)\alpha} \\ W^{(2)\prime}_i = (1-W_i^{(1)})^{\alpha}, \end{cases} \tag{7}$$

with α as a free parameter. All these options have been implemented in our routine.

2.2 Narrow band noise

For a chunk a typical length is $n = 2048$, that, sampled at 10 kHz corresponds to less than 0.25 seconds of data. On the other hand, there exist peaks in the expected noise spectrum of the VIRGO antenna with correlation time which is much longer. For example, the violin modes contributing to the thermal noise at frequencies $\nu^{(c)}(n) = n\cdot$

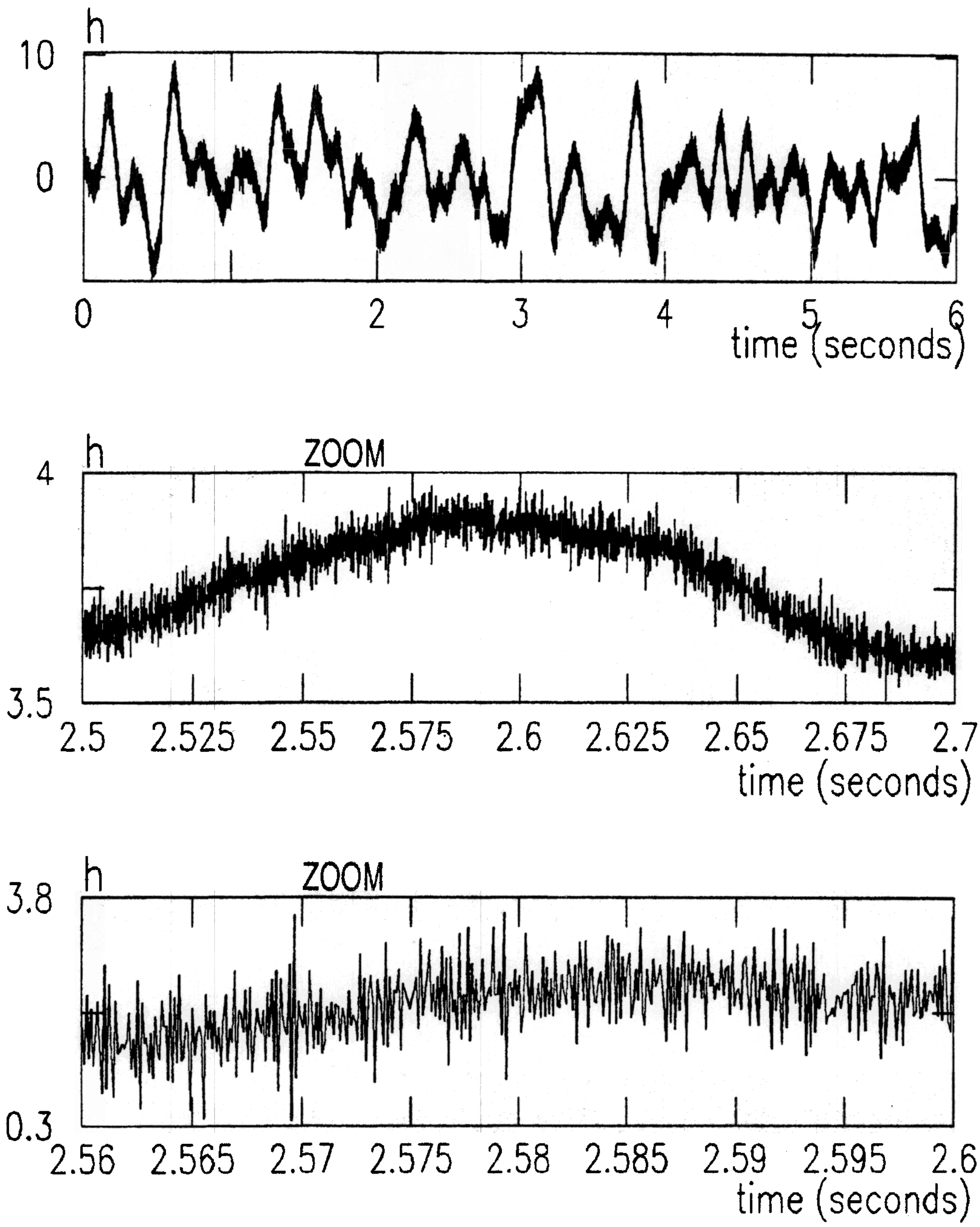

Figure 2: The wide band noise simulated data. zoomed in order to see the high frequency structure.

327 and $\nu^{(f)}(n) = n \cdot 308.6$, with Q factors of the order of 10^5, have τs which range from ~ 1 second to ~ 100 seconds, which are much longer than the durantion of a chunk. The problem can be more easily rephrased by saying that the spectral resolution is not high enough for the bandwidth of the peaks. The solution we have adopted is to sum the noise contribution coming from the peaks directly to the complete time series produced by the wide band noise. Each peak produces a real gaussian second order process which can be built by taking the real part of a complex first order process y_i

$$y_i = x_i + w y_{i-1} \tag{8}$$

x_i is a white gaussian complex noise process, w a complex number

$$w = e^{i\vartheta} e^{-\frac{\Delta t}{\tau}} \tag{9}$$

and Δt the sampling time. Only three input parameters are needed in order to produce the data 8:

- the frequency of the peak $\nu_0 = \dfrac{\vartheta}{2\pi\Delta t}$
- the band-width of the peak $\Delta\nu = \dfrac{1}{2\pi\tau}$
- the peak height in the spectrum $\sigma_x^2 = S_{peak}\dfrac{2\pi(1-|w|^2)}{\tau}$

In table 2.2 the frequencies, the time constants and the height of the peaks (in $\dfrac{1}{\sqrt{Hz}}$) used for the data shown in figs. 3 and 4, are shown.

The method outlined above is straightforward and its reliability is largely independent of the features of the spectrum, for example on how many peaks it has and on where they are placed. This is due to its modularity which does not make it necessary to adjust the parameters of the simulation in order to obtain the desired behaviour as it would happen, for example, with full AR simulations, where the number of needed coefficients is closely connected to the required spectral features. This point is made even more delicate by the fact that for the typical expected VIRGO spectrum a lot of power density is expected at low frequencies: up to five or six orders of magnitude above the best sensitivity, thus making it difficult to optimize the simulation performance while using a single algorithm. Moreover computational time constraints may become relevant in large number AR coefficient simulations because in order to follow such a large dynamical range double precision calculations may be necessary. Finally, we remark that problems related to a large dynamical range would show up in any data handling stage thus making it advisable to use some whitening technique on the "h-reconstructed" data and work with a flatter spectrum, which, by the way, is also the first step in matched filtering procedures.

ν_0 (Hz)	τ (s)	$\sqrt{S_{peak}}\ \frac{h}{\sqrt{Hz}}$
6.7	7.8	30.0
7.12	7.2	20.0
327.0	4.9	16.0
308.6	5.1	12.0
654.0	2.4	8.03
617.2	2.6	6.02
981.0	1.6	4.92
925.8	1.7	3.69
1308.0	1.2	3.37
1234.4	1.3	2.52
1635.0	0.97	2.47
1543.0	1.0	1.85
1962.0	0.8	1.90
1851.6	0.86	1.43
2289.0	0.69	1.52
2160.2	0.74	1.14
2616.0	0.61	1.24
2468.8	0.64	0.93
2943.0	0.54	1.04
2777.4	0.57	0.78
3270.0	0.49	0.88
3086.0	0.52	0.66
3597.0	0.44	0.76
3394.6	0.47	0.57
3924.0	0.41	0.66
3703.2	0.43	0.50

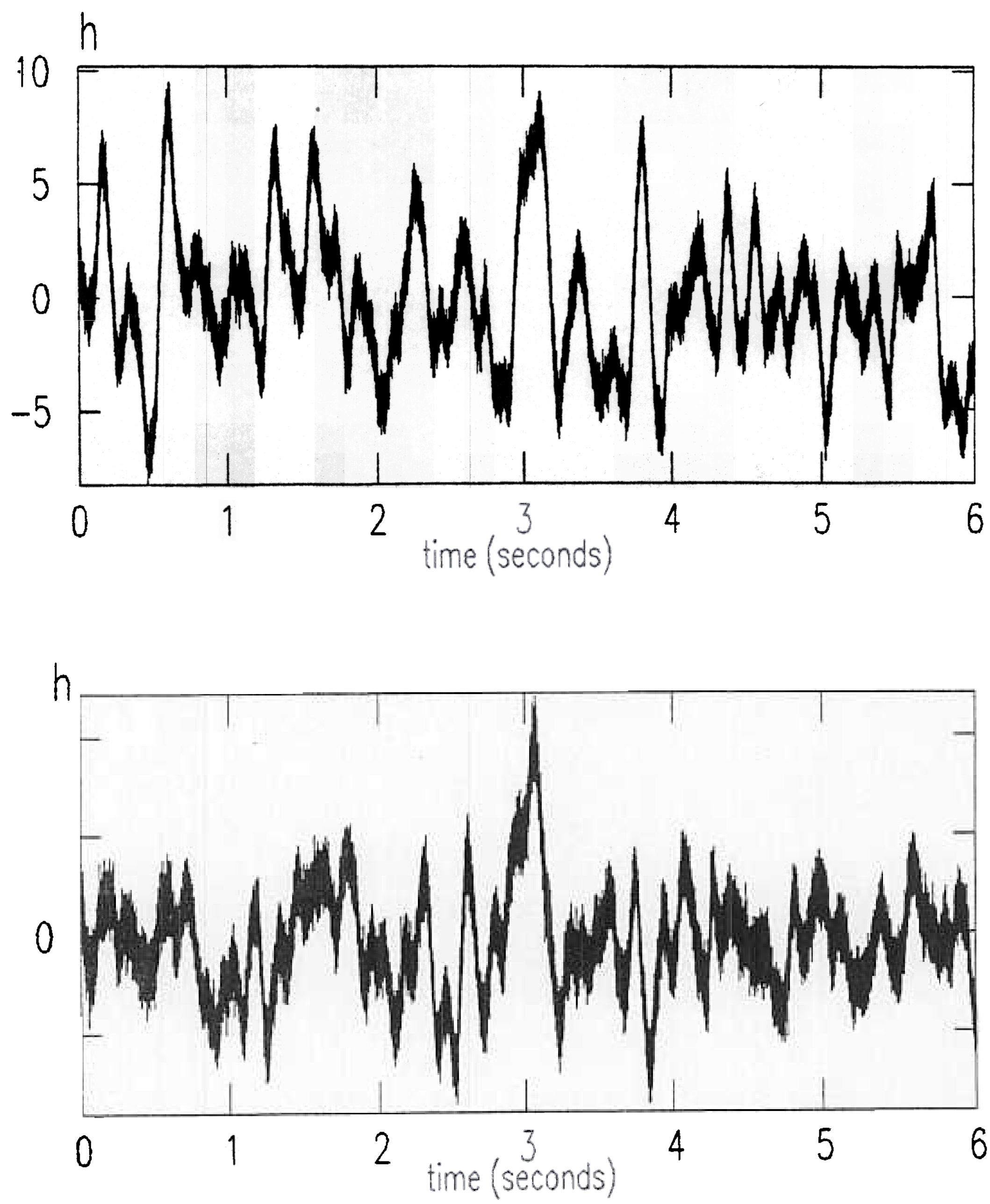

Figure 3: Data from the noise spectrum not including (upper) and including peaks (lower). In the latter case the high frequency contribution is more relevant.

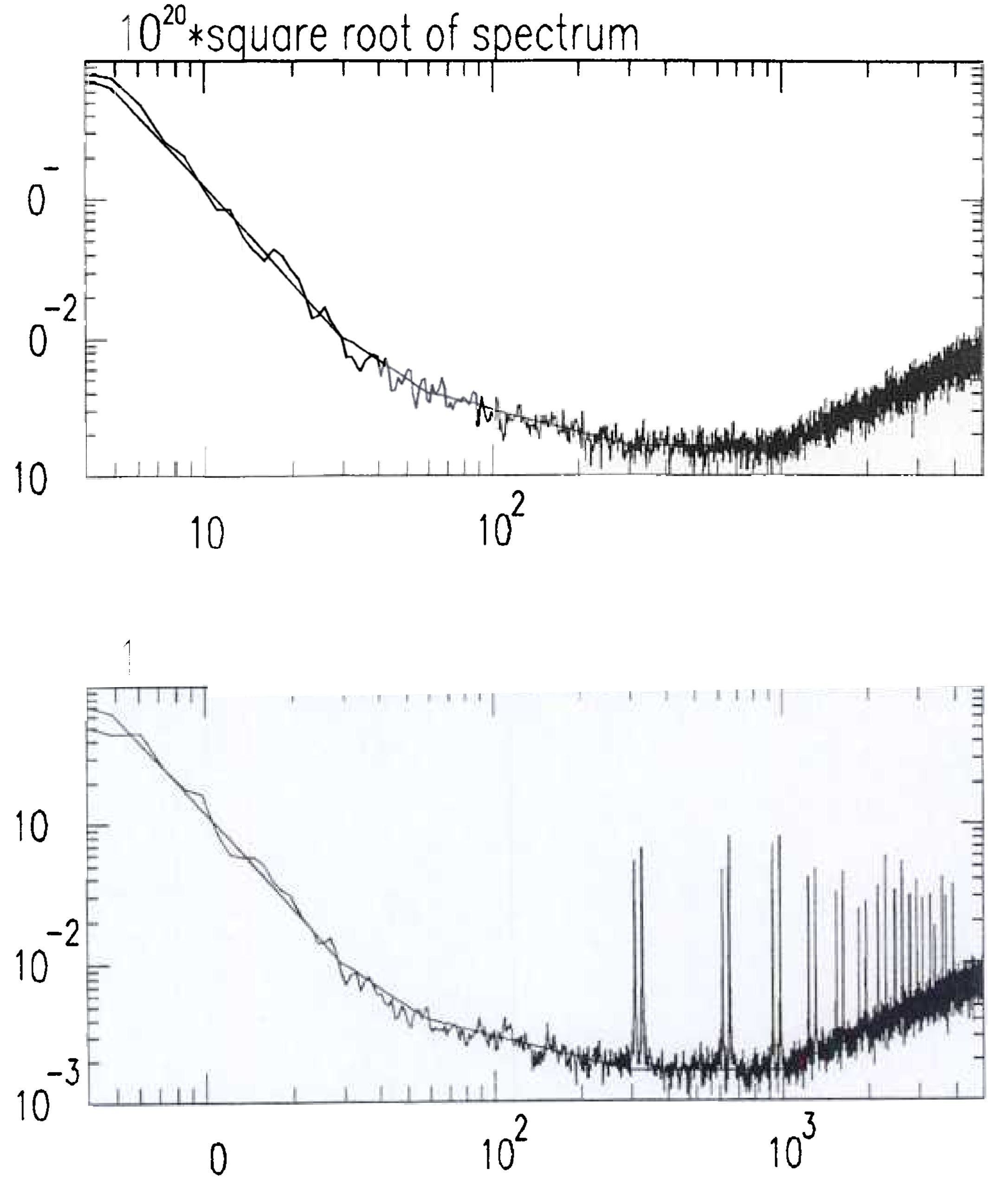

Figure 4: The noise obtained from the simulated data, without and with peaks, superimposed on the expected —N(jw)— used to produce the data. Also a couple of peaks at lower frequencies are present, as it can be seen from table 1, but they do not appear in the spectrum due to its low resolution.