# 4 Bench tests of methods and their implementation in AURIGA

### *4.1 A practical implementation of Wiener-Kolmogorov filter*

*From continuous to discrete time; the ARMA algorithm; signal and noise after whitening and WK filters; full sampled (4.9kHz) vs. decimated (70Hz) acquisition; buffered streams and overlap-save method; normalization and calibration*

#### 4.1.1 Overview of the AURIGA data acquisition system (daq)

The AURIGA daq system acquires and archives the signal of the antenna without the usual lock-in down conversion. In fig. 1 we report a schematic of this system.

The antenna output is sampled at 5 kHz with a 23 bit AD converter (HP E1430A) housed into a VXI crate (VXI is an industrial standard bus for electronic instrumentation).

The data from the accessory instrumentation, such as accelerometers, seismometers, electromagnetic probes etc. are sampled at rates between 1 and 200 Hz with a 32 multiplexed channels, 16 bit, AD converter (HP E1413A) housed in the same VXI crate. The thermometers are acquired and controlled by a GPIB interface. The UTC is acquired by the synchronization apparatus. A dedicated UNIX workstation (SUN Spark 10) reads out the converted data from MXI, GPIB and RS232 interfaces and feeds them first to a 9 GBytes hard disk as a safety buffer, then to a 35 GBytes cassette, to the on-line analysis workstation (DEC Alpha) and finally to a shared memory provided for the on--line monitoring of the raw data. To avoid dead times due to system failures or to system calibrations the acquisition chain has been completely duplicated.

The AD converter HP E1430A provides very high performances and flexibility for digitalizing band limited analog signals in a computer compatible format [20]. The normal ADC sample rate is 10MHz and we make use of the external GPS referenced clock to keep a μs synchronization with the UTC. To eliminate higher frequency components the analog signal is filtered by an analog low-pass filter, which rejects signals above 5MHz by at least 110 dB of attenuation. Then the signal is digitized by a 23-bit ADC, so that the effect of finite quantization levels can be completely ignored, leaving only two error sources: linearity error and electronic white noise. The measured total noise can be expressed as -137 dBfs/Hz. This corresponds, for the 2kHz bandwidth of the AURIGA signal to a signal to noise ratio of -106.4 dBfs; this dynamic range should be enough even as a quantum limited electronic chain of the AURIGA detector would be available. The samples of the signal are processed by a Digital Signal Processor (DSP), which provides a cascade chain of digital low-pass filters, each of which reduces the bandwidth of a factor 2. This is sufficient to avoid any aliasing due to the subsequent data decimation of a factor 2. For the AURIGA output we decided to use 11 of these filters reducing the sample rate to $10\text{MHz}/2^{11}=4882.8125$ Hz. The HP E1430A stores the sampled data in the onboard 8 MBytes FIFO memory which has been divided into 64 kBytes blocks. When a data block is ready an interrupt signal (IRQ) is generated. The IRQ notifies to the read out process in the acquisition workstation that a data block is ready.

Thus, the data from the FIFO memory are directed toward a VXI register and read through the MXI bus (see fig. 1). The same IRQ is sent to the GPS synchronization apparatus to date the block-ready event. The IRQ generation mechanism has an intrinsic jitter of 0.1μs while the IRQ propagation lines introduce a fixed delay in the time associate to each data block of (1104.4± 0.4)μs.

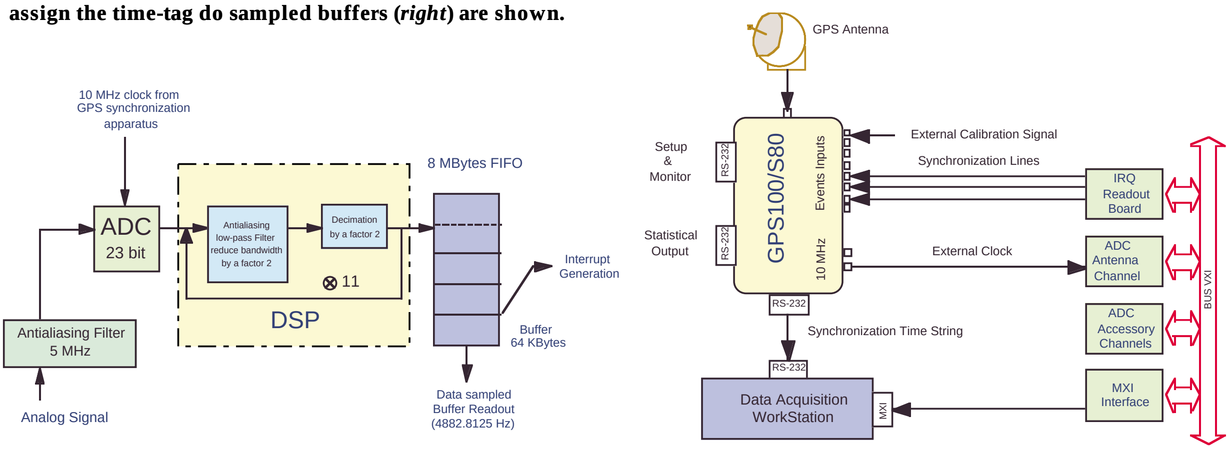
To take into account of the effect of antialiasing filtering, the HP E1430A transfer function have been measured (amplitude and phase) in the range of the detector mode frequencies

900÷950 Hz. We found that the transfer function is dominated by the digital antialiasing filters and that it is flat in amplitude within 1/1000. The phase shift has a well understood frequency dependence that can be translated into a fixed delay of (875.6± 0.1)µs for the AURIGA signal which is a band limited signal with a carrier frequency of ~920 Hz.

Nowadays the Universal Time can be conveniently obtained by the Global Positioning System (GPS). It consists of 24 satellites orbiting the Earth in 12 hours orbits and equipped with atomic clocks, which are continuously synchronized with the UTC. This system provides the so-called GPS time that coincides with UTC with an accuracy of 0.1 µs [18]. The synchronization apparatus GPS100/S80 of the AURIGA antenna has been constructed by ESAT [19] on the specific requirements of our daq system. A schematic of this apparatus and its integration in the AURIGA data acquisition system is shown in Figure 32. The internal oscillator of the GPS100 clock is a 10 MHz VCO with a frequency stability of $10^{-10}$ $day^{-1}$. A dedicated CPU handles the GPS signals and continuously corrects the phase drifts of the internal oscillator so that the phase error of the local second is kept within 0.1µs with respect to UTC. The calculus of the phase correction is based on statistical algorithms that limit the typical jitter of the GPS signal down to ~100ns. The GPS100/S80 is equipped with three RS232 interfaces, which are devised to monitor and setup the GPS receiver, to collect the statistical information about the clock operation and to send to the acquisition workstation the synchronization strings. The S80 board allows to dat up to 8 different events with a total rate limited to 50 events per second by the readout system of the RS232 interface. To synchronize the data flow with UTC we use the IRQs generated by the AD converters when a data block is ready for the data acquisition workstation.

These signals, captured by the IRQ readout board, are sent to the S80 event board which returns to the workstation an ASCII string containing the time of the interrupt generation; then the data acquisition software associates this string to the corresponding data buffer.

**Figure 32 – The fast data acquisition system is the core of the AURIGA sub-millisecond timing system. The schematics of the digital signal procesing unit (*left*) and of the synchronization apparatus GPS100/S80 of logic of the trigger control to assign the time-tag do sampled buffers (*right*) are shown.**



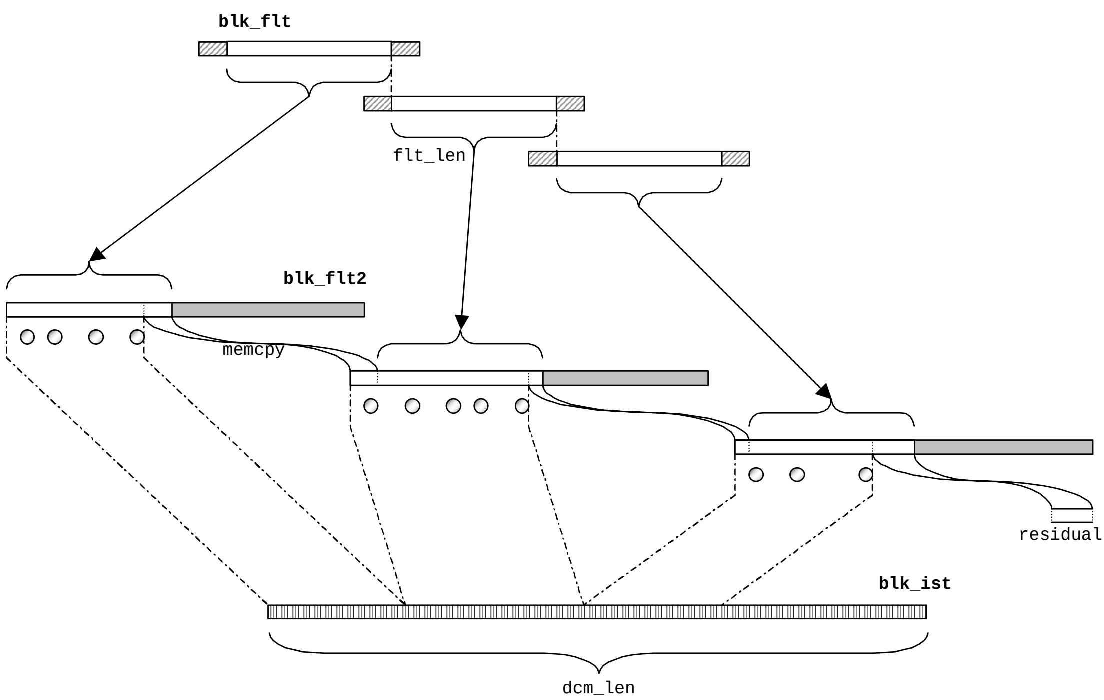
### 4.1.2 Overview of the AURIGA data analysis system

The software developed for AURIGA data analysis [36] is based on non-commercial routines written in C-language, using common routines generally available to ANSI C compilers, and a few the numerical recipes in C, but the final build is made by the Artifex commercial software[16] especially to break the code in different processes capable of running in a network environment, on different machines, and using separated resources (thus limiting the consequences of a runtime fault on a single peace of software).

The whole AURIGA daq-readout and control system was modeled by means the CASE Artifex, which is based on the Protob formalism (allowing the definition of object oriented models) [17]. The model has been then simulated to validate its logical consistency and then translated (with automatic code production) into AURIGA processing unit. The acquisition software is divided into 6 parallel processes: "*Read out antenna and auxiliary channels*", "*Read out UTC*", "*Read out temperatures*" which acquires the digitized signals from ADCs, GPS apparatus and GPIB instruments, "*Collect data*" which collects, formats and writes to disk the data coming from the readout processes, "*Write to tape*" which copies the data files from the disk to the cassette and "*Send data to analysis*" which sends the data buffer to the analysis workstation via TCP/IP. Software interfaces based on Labview (National Instruments) and Paw++ (CERN) allow to setup and control the daq system and to monitor the raw data.

The basic unity input in the analysis system is a raw data block, (**blk_raw**) which contains the signal from the ADC sampled at 10MHz/$2^{11}$≈4883Hz, and sent through a TCP/IP port to the analysis workstation from the acquisition Workstation, or from an offline software, which in this case reads the data archived from a DLT tape drive.

**Figure 33 – An excerpt from AURIGA data flow control in the online analysis. (*top, each program cycle from left to right*) The data buffers coming from the WK filtering routine are partially overlapped in time due to the necessity of scratch removal. A secondary buffer (blk flt 2) helps keeping time contiguity for next event search (*center*), which starts as soon as there are new data ready. When an event (*schetched as circles*) is found within a decorrelation length to the end of the temporary queue, event search is stopped and the residual data are recycled at the beginning of next cycle of event search. Simlessly with event search, decimated data are saved into the input buffer of next step (whitening filter, calculation of X, kurtosis, autocorrelation test statistics, and so on).**



12 raw blocks are collected and filtered with the WK filtered implemented as an ARMA algorithm to create a *filtered data block* (**blk_flt**). It is worth to notice that due to the recursive nature of equations describing the ARMA model we have to set to arbitrary values the initial conditions of the recursive part of the ARMA filter algorithm. As a consequence, the content of the filtered block is the superposition of the correct output of the WK filter fed by the input samples of the raw block, plus two damped spurious signals, one at the beginning and one at the end of the block, corresponding to the homogeneous solutions of the named recursive equations. The initial and final parts (or *scratches*) of the filtered data block have therefore to be removed, for a length (*scretch length*, **scr_len**) equal to several times the characteristic time of the WK filter (for our purposes, 2 bck_raw is sufficient). The remaining **flt_len** (*filtered length*) data are sent to a temporary buffer, and data contiguity is preserved by recycling and overlapping the raw data at the extremes of the filter buffer (see Figure 33). This is a standard tecnique known as *overlap-save method* [15].

In this way, event search is not performed on scratch areas, and its output is completely independent to data buffering. As filtered data are produced, they are sent to a secondary stream after *decimation*, which is simply the process of taking one sample every *RST*, where $(2 \cdot RST \cdot ST)^{-1}$ is the desired bandwidth after decimation (for instance, *RST*≈70 corresponds to 35Hz). The decimated samples are grouped for a length **dcm_len** (*decimated length*), and sent to the whitening filter for filtered data ($L \circ W^{-1}$), which is applied for convenience in the frequency domain (we use FFT's for this step). Also during this process scratch areas have to be deleted at the end of the buffer, and the overlap-save method is used again.

After online (or offline) analysis, AURIGA data have undergo an immense reduction in terms of storage resources needs, and the results of the analysis are permanently archived online in a Oracle relational database. [23][31][26]

### 4.1.3 Decimation, sampling accuracy

In 2.1.1 we saw how decimating the data after the Wiener filter has the has the effect of reducing the deviations of the spectral density against the model. Of course, another and more direct effect is sensible data compression. For example, a decimation of a factor 70 folds every 35Hz side bands into the frequency interval 906.6÷941.5, that is enough to store the information about the signal, with a space occupation for data storage which is 70 times less.

This is very similar to the analog procedure used by other detectors, which use lock-ins as signal demodulators and band-limiting, and then acquire those signals at low sampling rate. If the analog oscillator used to trigger the lock-in has a proper absolute timing capability, and if the acquired samples are properly time tagged, then both procedures end up with the same set of data (apart from the fact that from AURIGA long term storage database every past data in a 2kHz frequency range can be retrieved for diagnostic purpose, or for more complex offline filtering). Suppose that we decide to use these decimated data to reconstruct amplitude and phase information of the signal, for example to perform again an event search, maybe after some other filter has been applied (for instance, a tilted-spectrum burst) or the outputs of two detectors has been summed together (see 0). We can convince ourselves with a few examples (see for instance Figure 34) that any event search which thresholds directly on the sampled amplitude of the signal must deal only with very high sampling rate, at least of the order of the carrier frequency of the signal, otherwise signal parameters are going to be estimated with poor accuracy, and the chance of false dismissal is greatly increased.

The reconstruction of the signal $\mathbf{a}(t)$ from the samples $\{\mathbf{a}_i\}$ –if the all the hypotheses of the sampling theorem are satisfied– can be done *exactly* thanks to the following well-known result:

$$\mathbf{a}(t) = \sum_{n=-\infty}^{+\infty} \mathbf{a}_i nT \frac{\sin \pi/T\,(t-nT)}{\pi/T\,(t-nT)} \qquad (4.1)$$

where T –as usual– represents the sampling time.

The resampling of decimated data can be done in principle applying ( 4.1 ), and in practice we do, but in the frequency domain, where the computationally expensive convolution algorithm becomes a simple multiplication component by component. In this case the frequency modulation pattern is trivial, a simple boxed band-pass filter centered in the original frequency position of the decimated signal. Going from the time domain to the frequency domain and the way back is nowadays relatively costless with Fast Fourier

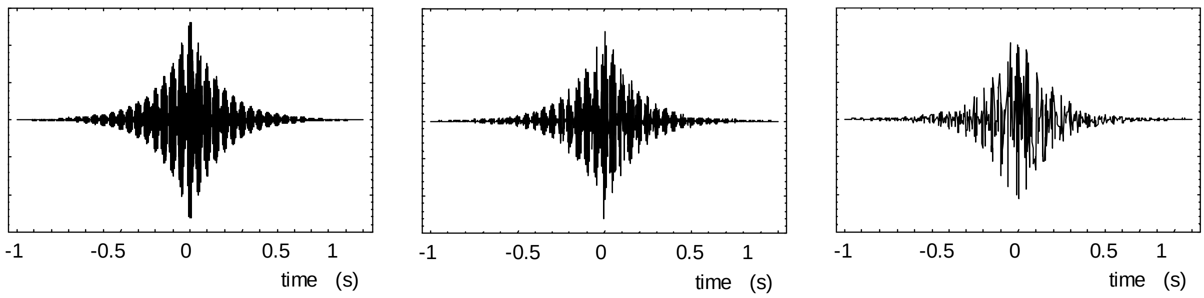
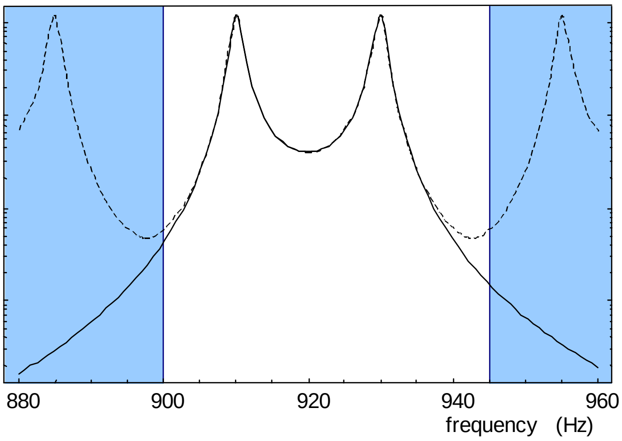



**Figure 34. Effect of decimation on amplitude accuracy: From left to right, the signal of fig …. Sampled at 0.01s, 0.05s and 0.1s. In the latter it is manifest that there is little (if any) hope that the time of arrival or the amplitude are correctly estimated without interpolation of the signal back to the µs accuracy. Note that the decimation offset of was chosen so to maximize somewhat the effect, so the example shows one of the worst cases one can expect. In general, a bias on amplitude estimate of the order of ~5÷10% is typical, and the accuracy in timing is never more than the sampling time itself.**

Transforms (FFT) algorithms. The length of the buffers to which the FFT is applied must be long enough that the corresponding frequency spread about each bin is not compromising the result. As a rule of the thumb, it should be at least equal to the inverse of the bandwidth of the narrower structure in the spectral density function of the buffer, therefore of the order of one second or more for present operating detector configuration. In fact this effect is apparently negligible with respect to the artifacts due to unbalanced oscillations of the function $\left[\sin \pi/T\,(t-nT)\right]/\left[\pi/T\,(t-nT)\right]$ near the borders of the time stretch. In practice we have to throw away at the borders the equivalent to more than $10^4$ decimated samples, or ~$10^6$ final samples, just to keep the residual oscillations a factor ~$10^3$ below the average sample amplitude. The FFT length should be therefore at least two times this value, which corresponds to a time span of a few minutes.

**Figure 35. Signal aliasing in the frequency domain: the original signal is shown as a continuous line, while the dashed line represents the signal restored after the decimation process. Due to the fact that the original signal was band-limited, the deviations in the sensitive region are negligible, but can anyway be (and in fact are) taken into account when comparing the signal with the template.**




With the WK filter output resampled at ~5kHz, the problem of event amplitude accuracy is cured. However for a good timing, we ought to reconstruct the signal until the sampling time is of the same order of magnitude of the desired accuracy, that is to the μs level. But a faster shortcut is available this time. We are dealing with a narrow band signal sampled at more than *twice* its Nyquist frequency. Therefore, there are for sure 4 or 5 samples for each period of the carrier wave. We can then interpolate it by computing the parameters φ, ω, A (>0) of the sine arc

$$y(t) = A\cos(\omega t + \varphi) \tag{4.2}$$

passing through the three consecutive points $\left\{(t_n, a_n) \middle| t_1 < t_2 < t_3\right\}$ about the extremes of the filtered data. From the couple $(\varphi, A)$ we get time of arrival and amplitude; while ω is a free parameter representing a sort of instantaneous estimate of the carrier frequency[17].

This shortcut is made possible because the sampling time T is about five times less than the expected carrier period. Therefore, either of the two following conditions

1. $(a_1 > a_0) \,\&\, (a_2 \le a_1) \Rightarrow$ convex triple
2. $(a_1 < a_0) \,\&\, (a_2 \ge a_1) \Rightarrow$ concave triple

are necessary and sufficient to state that the signal reaches an extreme at a time $t_{ext}$ somewhere between $t_1$ and $t_3$.

Suppose that the points $\left\{(t_n, a_n)\right\}_{n=1,2,3}$ constitute a convex triple. We have to solve the system

[17] It could be used as a redundant diagnostic test, comparing with the result of the frequency measurements performed after digital lock-in.

$$\{a_n = A\cos(\omega t_i + \varphi)\}_{n=0,1,2} \tag{4.3}$$

It is not restricting to put $t_0 = 0$, as it constitute just a trivial change of time coordinates. Substituting $t_n = nT$, the previous system reads

$$\begin{cases} a_0 = A\cos\varphi \\ a_1 = A\cos(\omega T + \varphi) = A[\cos(\omega T)\cos\varphi - \sin(\omega T)\sin\varphi] \\ a_2 = A\cos(2\omega T + \varphi) = A\left[\left(2\cos^2(\omega T) - 1\right)\ \cos(\varphi) - 2\sin(\omega T)\cos(\omega T)\sin\varphi\right] \end{cases} \tag{4.4}$$

with the following constraints:

$$\cos(\omega T) > 0; \qquad \sin(\omega T) > 0; \qquad -\tfrac{3}{2}\omega T \le \varphi < -\tfrac{1}{2}\omega T\,. \tag{4.5}$$

Let us consider first the case $\varphi \neq -\pi/2$. We can define:

$$B = \sin(\omega T)\tan\varphi; \quad C = \cos(\omega T). \tag{4.6}$$

The system ( 4.4 ) can be written as

$$\begin{cases} y_0 = A\cos\varphi \\ Y_1 \equiv \dfrac{y_1}{y_0} = C - B \\ Y_2 \equiv \dfrac{y_2}{y_0} = \left(2C^2 - 1\right) - 2BC = 2Y_1C - 1 \end{cases} \tag{4.7}$$

Solving for C and B, we get

$$C = (1 + Y_2)/2Y_1; \quad B = \left(1 + Y_2 - 2Y_1^2\right)/2Y_1 \tag{4.8}$$

hence

$$\omega T = \arccos\left(\frac{a_0 + a_2}{2a_1}\right); \ \varphi = \arctan\frac{a_0 + y_2 - 2a_1^2/a_0}{2a_1\sin(\omega\ \ T)}; \ A = \frac{a_0}{\cos\varphi}; \ t_{ext} = \ t_0 + T - \varphi/\omega \tag{4.9}$$

where in $t_{ext}$ the original offset value $t_0$ has been restored.

In the case $\varphi = -\pi/2$, i.e. $y_0 = 0$, equations ( 4.9 ) have to be substituted by

$$\omega T = \arccos\left(\frac{a_0 + a_2}{2a_1}\right); \ \varphi = -\frac{\pi}{2}; \ A = \frac{a_1}{\sin\varphi}; \ t_{ext} = \ t_0 + T + \frac{\pi}{2\omega} \tag{4.10}$$

Next triple has to be of course concave. The values of $A$ and $t_{ext}$ are still provided by ( 4.9 ) and ( 4.10 ), but with the substitution $a_i \to -a_i$. Next triple is again convex, and the whole pattern repeats.

The previous results can be summarized as follows:

$$\omega T = \arccos\left(\frac{a_0 + a_2}{2a_1}\right); \ \varphi = \arg\left[a_0^2 + a_0a_2 - 2a_1^2 + i2a_1a_0\sin(\omega T)\right];$$

$$t_{ext} = \ t_0 + T - \frac{\varphi}{\omega}; \quad A = \begin{cases} \left|\dfrac{a_0}{\cos\varphi}\right| & if \quad \cos\varphi \neq 0 \\ \left|\dfrac{a_1}{\sin(\omega T)}\right| & if \quad \cos\varphi = 0 \end{cases} \tag{4.11}$$

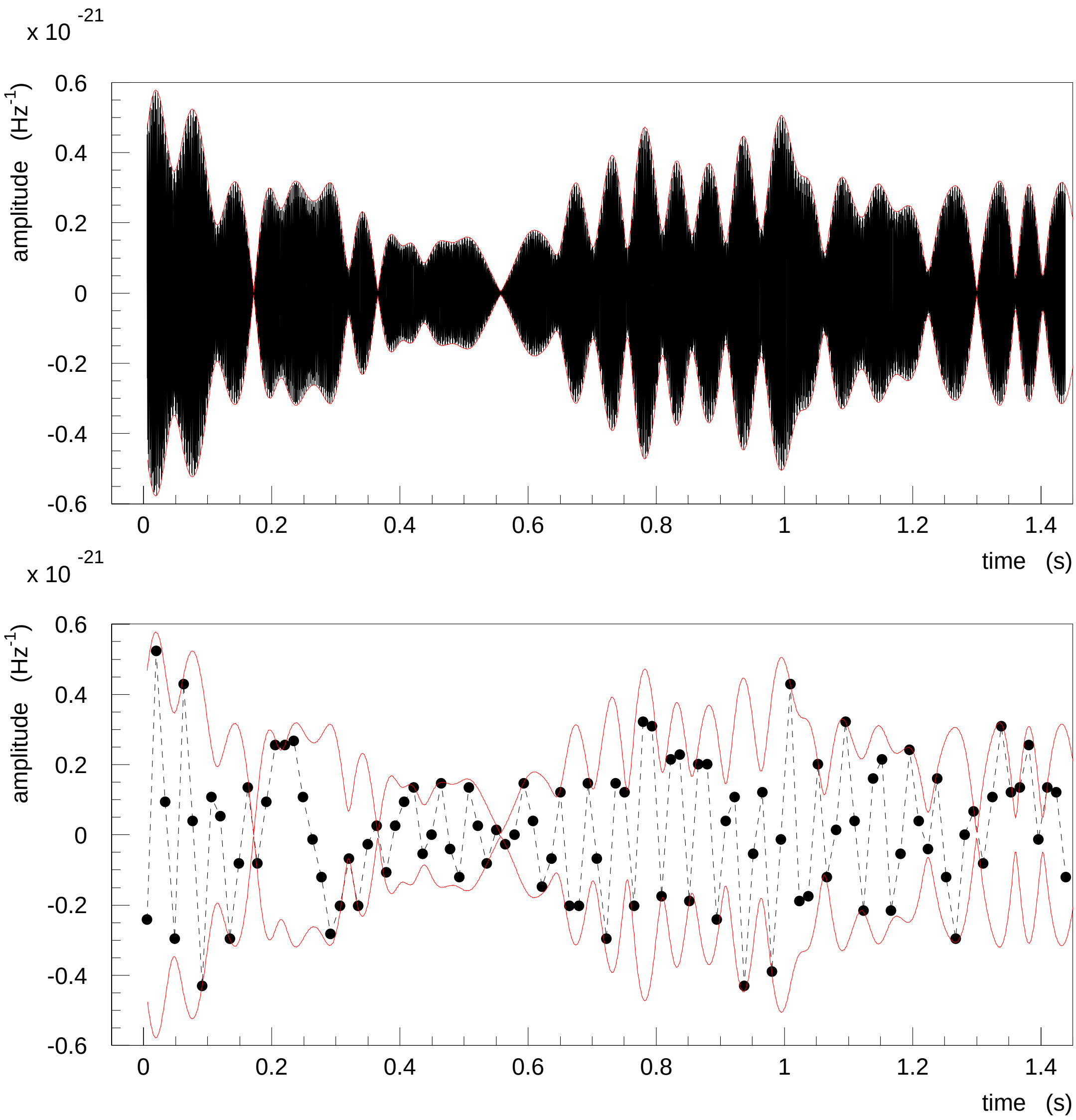



**Figure 36. The original decimated data of a short stretch of noise (*lower*) and the same stretch after full 4.882kHz resampling (*upper*) are compared with the correlated output of the peak interpolation routine (see p. 69-70). While the gain in amplitude accuracy just by resampling is outshining, the further interpolation down to μs scale is needed to recover phase information.**

### 4.1.4 Digital signal processing in the time domain: the ARMA model

To afford in real time data processing with the WK filter while keeping relatively small sampling time (which imply a heavy data stream), a fundamental requirement to the filter algorithms is a clever implementation in order to reduce the number of computational steps, and possibly also the amount of memory to store the temporary results. Implementing ( 1.26 ) literally would imply estimating[18] $R_{ij} = \sum_{k=1}^{m} x_{i-k} x^*_{j-k}$ and then performing a matrix inversion. In order to reduce the error of this estimate we need to choose $m$ as large as possible –and it tends to be *very* large[19] a number.

A better choice would be using FFT and multiplications in the frequency domain instead of convolution in the time domain. The FFT still increases with increasing N, but the dependency is only logarithmic.

An ARMA (Auto-Regressive & Moving-Average) algorithm has the good property that the number of operations to be performed is small and do not depend on the length of the input data stream, or in any time constant of the detector, other than the sampling frequency.

The ARMA takes its name from two opposite operations on the data, a moving average on the input stream

$$y_i = \sum_{k=0}^{m} c_k x_{i-k} \qquad (4.12)$$

and a recursive algorithm on the output stream

$$y_i = \sum_{k=1}^{m} c'_k y_{i-k} \qquad (4.13)$$

These two equations allow building just *causal* filters. However, linear combinations of future data of the stream can be considered as well, to build anti-causal filters:

$$y_i = \sum_{k=-m}^{1} c''_k x_{i-k} \qquad (4.14)$$

$$y_i = \sum_{k=-m}^{1} c'''_k y_{i-k} \qquad (4.15)$$

There isn't a unique recipe to translate the transfer function or the WK filter in the discrete time domain, as there is always a certain amount of approximation involved, whose importance depend on the specific problem. As an interesting example we shall briefly describe the *pole-zero mapping* procedure.

First of all, let us recall the definition of the z-transform of a set of data

$$x(z) = \sum_n x_n z^n \qquad (4.16)$$

This transform resembles the Laplace transform, with the identification $z \leftrightarrow e^{sT}$, where T is the sampling time. This relation shows that the imaginary axis of the s plane is mapped onto the unitary circle of the z-plane, the positive real axis semiplane into the plane outside

---

[18] It's a rough estimate, using the approximation $\mathbf{x}_i \simeq \mathbf{n}_i$, which is valid only in the limit of weak or no signals.

[19] A 0.2ms sampling time would mean N≈5000 just for a 1s long stretch of data. But the decay time of a cryogenic resonant detector is of the order of 1000s.

the circle, etc. Many properties of the Laplace transform readily translate for the z transform. In particular, multiplication by $z^{-1}$ is similar to shift backward in the time domain by T, i.e.

$$x_{n-1} \quad \leftrightarrow \quad x(z)z^{-1} \tag{4.17}$$

This gives us the ability to translate immediately the AR and MA equations from the discrete time domain to the z-transform. Thus, the main task to project a discrete filter is to give a recipe for writing the transfer function of the desired linear system as z-transform.

We shall now briefly overview the pole-zero mapping procedure, as a complete treatment of the digital filter synthesis is out of the scope of this work. Suppose that we are given a rational transfer function, i.e.

$$\mathrm{T}(\omega) = \frac{\mathrm{Q}(\omega)}{\mathrm{P}(\omega)} = \frac{\prod_{k=1}^{Q}(s-q_k)}{\prod_{k=1}^{P}(s-p_k)} \tag{4.18}$$

with $Q \le P$. This function can be split in a cascade (product) of simpler systems, of these two basic kind:

$$\mathrm{T}_1(\omega) = \frac{1}{(s-p)}; \quad \mathrm{T}_2(\omega) = \frac{(s-q)}{(s-p)} \tag{4.19}$$

Let us focus on $T_1$. If we perform the substitution $s \leftrightarrow z$, $p \leftrightarrow z(p) \equiv e^{pT}$, we got in the z domain $\left(z-e^{pT}\right)^{-1}$. Note that

$$\frac{1}{(s-p)} = \frac{Te^{pT}}{\left(e^{sT}-e^{pT}\right)} \frac{\left(e^{(s-p)T}-1\right)}{(s-p)T} \tag{4.20}$$

If $s$ is chosen in a small neighbor around $p$, then $\left|(s-p)\big/\left(e^{(s-p)T}-1\right)\right| \to 1$. Defining

$$\mathrm{T}_1'(z) = \frac{Te^{pT}}{\left(z-e^{pT}\right)} \tag{4.21}$$

we derive from ( 4.22 ) that the integrals of $T_1'(e^{sT})e^{st}$ and of $T_1(e^{sT})e^{st}$ on a circular path around $p$ –i.e. the residuals– are identical, i.e.

$$y_n' \equiv \oint_p e^{sTn} T_1'(e^{sT}) = \oint_p e^{sTn} T_1(s) = y_n \tag{4.22}$$

On the right side of this equation we are performing the inverse Laplace transform of $T_1(s)$, *but* on the left side we are just considering the residual of $e^{sTn}T_1'(e^{sT})$ on a single pole $p$, while $T_1'(e^{sT})$ has a countable infinity of poles at $\{p+2\pi ik\}_{k\in\mathbb{N}}$, each one solution of the

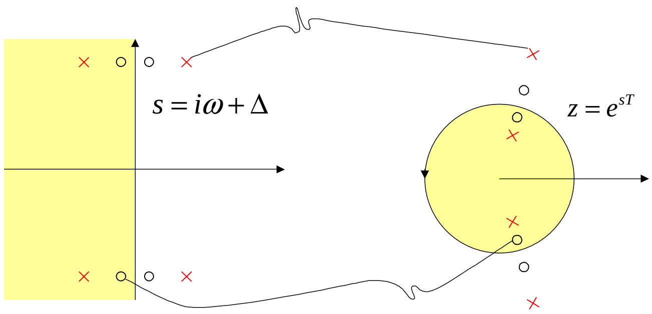



**Figure 37. Pictorial representation of the poles and zeros mapping procedure: after the Laplace transform, the poles and zeroes of the function are identified in the complex *s*-plane (*left*) and mapped by a complex exponential to the *z*-plane (*right*). This transformation maps the causal half-plane in the interior of the unitary circle. The inverse *z*-transform is performed with by integration along a circular path outside or inside the unitary circle, depending on the time direction.**

equation $e^{sT} = z(p)$. Therefore, the inverse Laplace transform of $T_1'(e^{sT})$ is $\sum_{k\in\mathbb{N}} \oint_{p_k} e^{sTn} T_1'(e^{sT}) \neq y_n$, as we expect, because $T_1(s)$ and $T_1'(e^{sT})$ are not identical. Another way to revert $T_1'(e^{sT})$ to the discrete time domain is performing an *inverse z-transform*, which avoid the problem of aliased solutions:

$$y_n = \oint_z z^n T_1'(z) \tag{4.23}$$

Finally, we can write down the discrete time algorithm for the subsystem $T_1$:

$$\tilde{y}(\omega) = \mathrm{T}_1(\omega)\tilde{x}(\omega) = \frac{\tilde{x}(\omega)}{(s-p)} \quad \leftrightarrow \quad y'(z) = T_1'(z)x'(z) = \frac{x'(z)Te^{pT}}{\left(z-e^{pT}\right)} \tag{4.24}$$

hence

$$\begin{aligned} \left(z-e^{pT}\right)y'(z) = x'(z)Te^{pT} \quad &\Rightarrow \quad y'(z) = z^{-1}y'(z)e^{pT} + z^{-1}x'(z)Te^{pT} \\ &\leftrightarrow \quad y_n' = e^{pT}y_{n-1}' - x_{n-1}'Te^{pT} \end{aligned} \tag{4.25}$$

The last line is the result we needed, a single-order causal ARMA model for the single-pole low-pass transfer function $T_1(s)$.

A similar result can be obtained for $T_2(s)$, but with a subtlety:

$$\frac{(s-q)}{(s-p)} = Te^{(p-q)T}\frac{\left(e^{sT}-e^{qT}\right)}{\left(e^{sT}-e^{pT}\right)}\frac{\left(e^{(s-p)T}-1\right)}{(s-p)T}\frac{(s-q)T}{\left(e^{(s-q)T}-1\right)}$$

Now, to follow the same procedure, we have to observe that

$$\frac{(p-q)T}{\left(e^{(p-q)T}-1\right)} = \frac{1}{1-\frac{1}{2}(p-q)T+\ldots} \approx 1 + O\left(\|p-q\|T\right) \approx 1$$

This happens to be a correct approximation in our case, as $\|p_k - q_k\|T \approx \Im q_k T < 10^{-3}$

These two kind of filters can be used as building bricks for more complicated rational filters, by applying the them in cascade, one step for the causal part of the filter, another for the noncausal part, and cycling over all distinct poles of the system. In the actual implementation, the degree of the single step is 2 because the complex conjugate pairs are applied in a single step In the following table are reported the coefficients of the ARMA algorithm for the filter ( 1.45 ).

$$\mathrm{W}(\omega) = \sigma_{\mathbf{a}}^2 S_0^{-1} \prod_{k=1}^{2} \frac{(i\omega - p_k)(i\omega - p_k^*)}{(i\omega - q_k)(i\omega - q_k^*)} \frac{-i\omega}{(i\omega + q_k)(i\omega + q_k^*)}.$$

| | **causal** | **anti-causal** |
|---|---|---|
| AR | $y_i = \sum_{k=0}^{m} c_k x_{i-k}$ $\begin{cases} c_0 = 1 \\ c_1 = -2e^{T\Re p}\cos(\Im pT) \\ c_2 = -2e^{T\Re p} \end{cases}$ | $y_i = \sum_{k=-m}^{1} c_k'' x_{i-k}$ $\begin{cases} c_0'' = 1 \\ c_1'' = 1 \\ c_2'' = 0 \end{cases}$ |
| MA | $y_i = \sum_{k=1}^{m} c_k' y_{i-k}$ $\begin{cases} c_1' = -2e^{T\Re q}\cos(\Im qT) \\ c_2' = -2e^{T\Re q} \end{cases}$ | $y_i = \sum_{k=-m}^{1} c_k''' y_{i-k}$ $\begin{cases} c_1''' = -2e^{T\Re q}\cos(\Im qT) \\ c_2''' = -2e^{T\Re q} \end{cases}$ |

### 4.1.5 Simulations: event search, $\chi^2$, etc.

The first stable release of the data analysis software (1.06) was carefully tested in 1998 to check any inconsistency at least during normal stationary operation of the detector. The code already implemented pre- and post- filter parameter following, and was subsequently subject to minor changes and bug fix, especially to the adaptive procedures. But the present validation procedure removes all highly non-stationary periods when selecting data to be exchanged to the scientific community, so there is little difference also in this respect between the results obtained by various sub-versions.

It is worth to mention here the standard tests that where performed to validate the code. The simulations were based on a Gaussian stochastic process generated by an ARMA model of the detector noise, and therefore with perfectly known parameters (though they were continuously estimated and updated by the analysis). The time of arrival, amplitude and $\chi^2$-values were tested with SW-events. The event generation and summation here was technically performed with a procedure different from that described later (see 4.3). The events were generated at the level of the raw data blocks, in a simulated data acquisition environment, to check all steps from the acquisition-analysis network communication system to the final event production. The events were injected in the primary data stream (in fact the only one, in this version of the analysis) with a delay between them well above the de-correlation time and with final SNR of 34.2. As a consistency check, the background event statistics was also examined.

The conclusions after the tests were the following:

- The LCK (two lock-ins on the raw data and PSD generation) and AAN processes (WK filtering and event search) are as expected the most time-consuming, each one using about the half of the CPU computation time. They could be made run safely at $\times 14$ of the online data acquisition rate on a Digital ALPHA Station with clock speed of 400MHz.
- There is no sign of timing inconsistency[20]. The sharp cut at 3 times the Wiener time is due to the de-correlating procedures.
- Figure 38 shows the histograms of time delay between events with different selections on SNR. For SNR>4 the decay is fitted by an exponential, with a decreased background event rate about a factor 100 every unit step in SNR.
- The events in the control group have a rms amplitude deviation of 1.03±0.025 in SNR units, that is a value compatible with 1.
- The control group phase error histogram is fitted by a Gaussian distribution, as it should, with a standard deviation equal to 179.1±4.4μs/SNR against a theoretical value $1/\omega_0$=173μs/SNR (see 1.3.3). The peak error distribution is grossly Gaussian, with a standard deviation $\sigma_k$=1.0±0.1 (in peak number units). Applying the formula ( 1.55 ) (in the approximation $1/\tau_{Wiener} >> \omega_*$, which fits our case, we obtain an expected value of $\sigma_k$=0.95, not too far from what we found.
- The $\chi^2$ histogram for high SNR events agrees with the $\chi^2$ histogram of background events.

[20] Due to bugs in the buffered architecture of the previous test-type data analysis, sometimes on entire filter buffer –about 25s long– was lost. An tail in the event delay distribution of more than 25s was a strong clue to the identification of the problem, that is completely resolved in the production-type analysis.

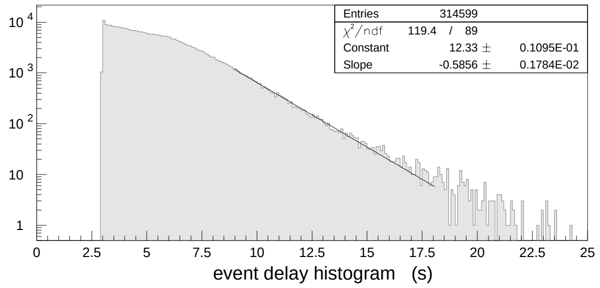
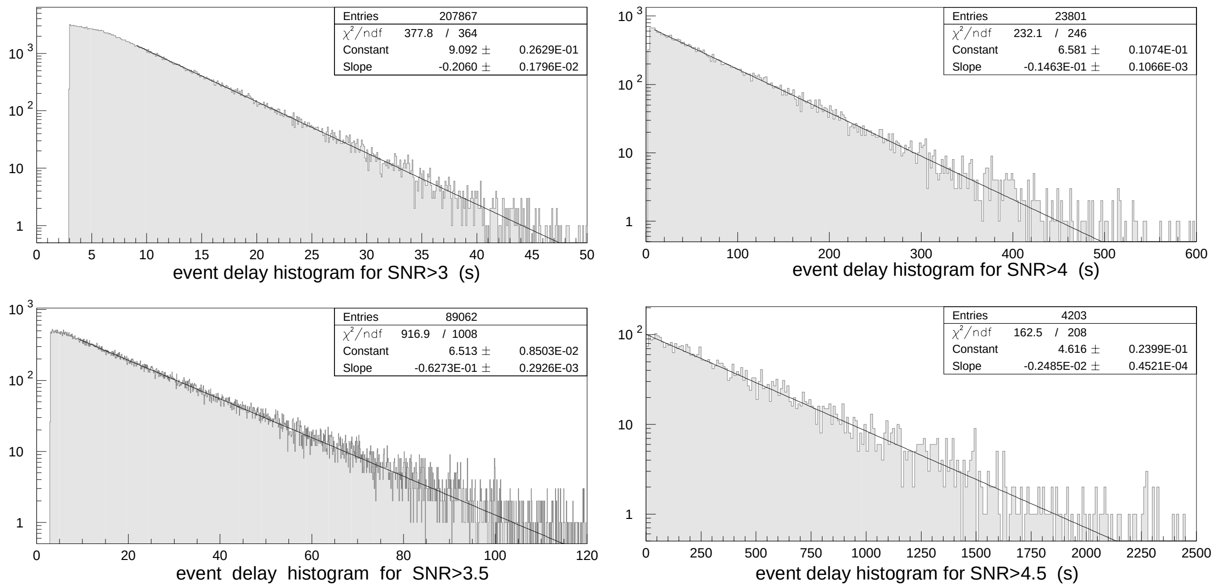
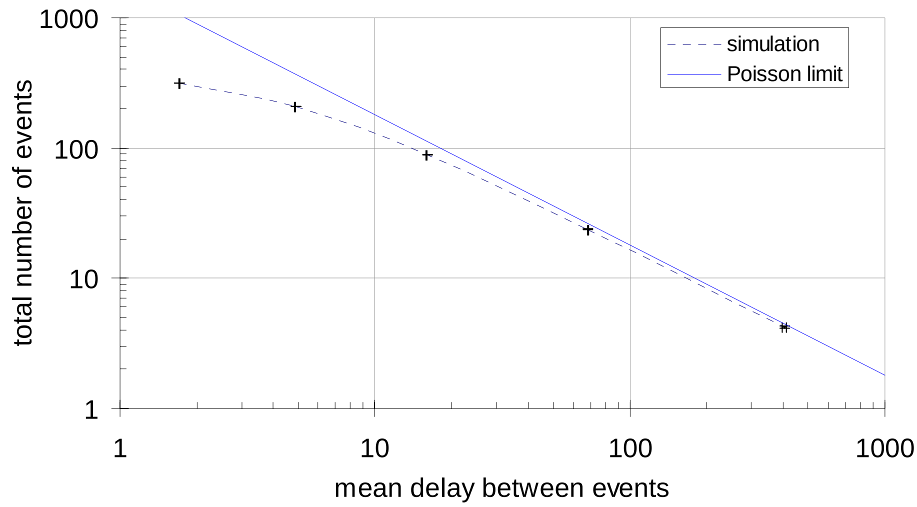



**Figure 38 – It is a good practice to check the histogram of delay times between background events for any inconsistency in the event search algorithm. (*a,b,c,d,e*) The sharp cut below 3 sec is due to the de-correlation algorithm, that allows for only one event in a time interval of 3tw. (*f*) The number of events as a function of the mean distance of the events, probed by selecting subsets of the event lists with varying the SNR threshold. The continous line reresents the behavour of a real poisson process, and it is reached at high SNR, when the de-correlation time is negligible with respect to the mean delay among events.**




| Simulation summary | |
|---|---|
| $\nu_0$= 920.8 Hz | $N_{noise}$= 314700 |
| $\nu_*$= 9.05 Hz | $N_{signal}$= 2577 |
| $\nu_-$= 911.75 Hz | SNR= 34.174 ±0.67 |
| $\nu_+$= 929.84 Hz | $\sigma_{SNR}/\sigma_{FILT}$= 1.019 ±0.013 |
| $\Delta_-$= 0.9 mrad/s | $\sigma_\phi$= 179.1 ±4.4 µs/SNR |
| $\Delta_+$= 1.2 mrad/s | $(\sigma_\phi)^{th}$= 172.8 |
| $\Delta^{opt}_-$= 1.0 rad/s | $\sigma_k$= 0.985 ±0.015 |
| $\Delta^{opt}_+$= 1.0 rad/s | $(\sigma_k)^{th}$= 0.948 ±0.019 |

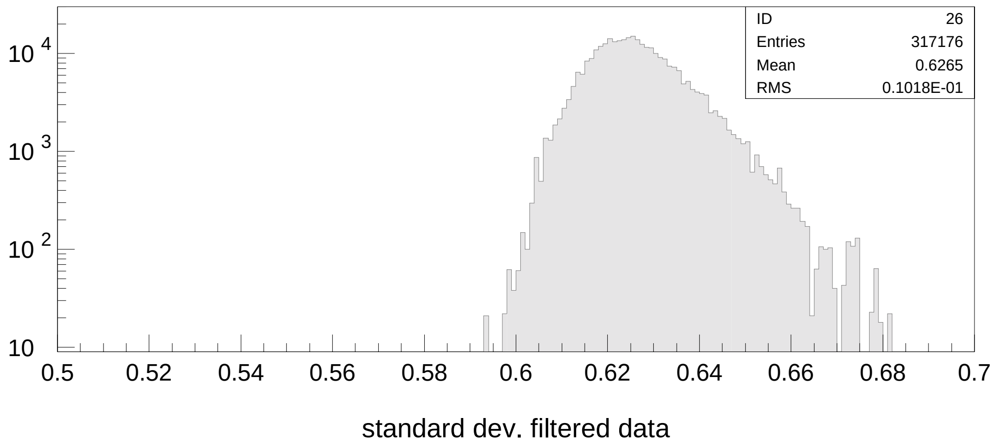



**Figure 39 – Histogram of the buffered estimates of thefiltered data rms, $\sigma_{FILT}$ (the abscissa scale is in arbitray units). Because the variations in the estimates of $\sigma_{FILT}$ are mitigated by a moving average, the data plotted here are partially correlated.**

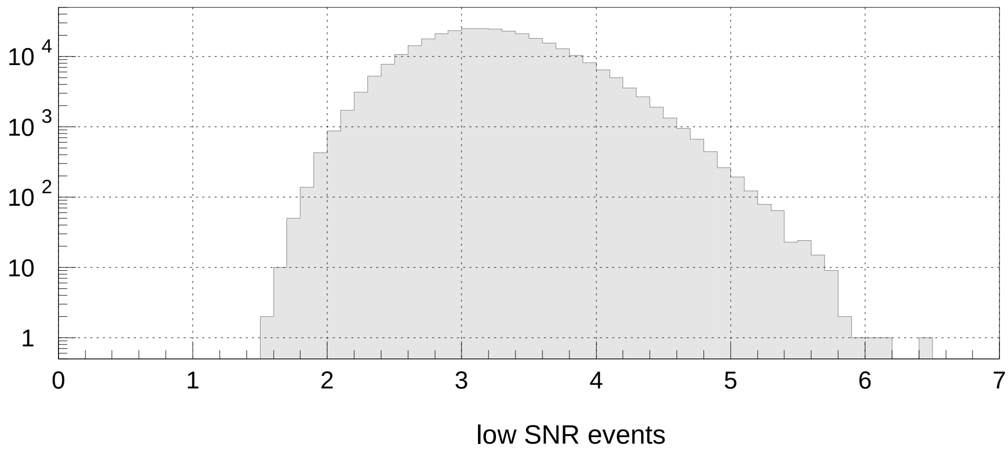



**Figure 40 – Distribution of background events. The tail at SNR>5 is important for the performance of the detector in a coincidence analysis, as it determines the ground false alarm probability achievable in case the noise is modelled by a purely Gaussian stochastic process.**

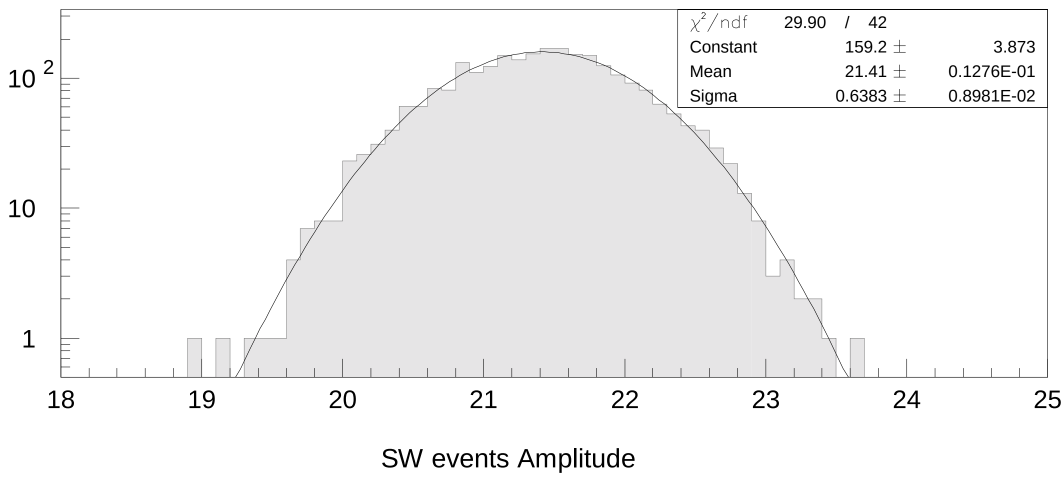



**Figure 41 – Amplitude deviation for SW events generated by an ARMA model of the detector impulse response directly into the raw data stream. The amplitude is measured in the same (arbitrary) units of Figure 39 .**

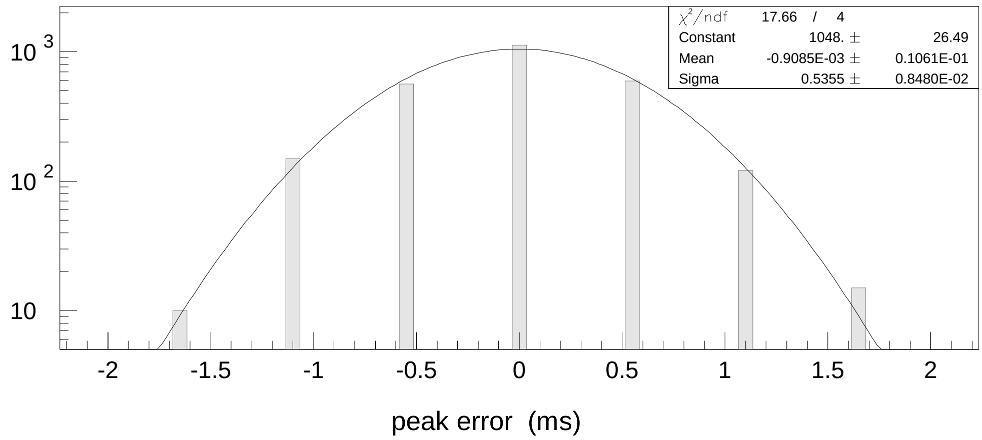



**Figure 42 – Histogram of time deviates for sent SW events (SNR=34), which is clearly peaked around integer multiples of half the inverse beating frequency. The Gaussian fit (see 1.3.3) is rather good, though it is a rather coarse approximation, as the ratio between the beating time and the Wiener time is 1:10, which is not *much* less then 1.**

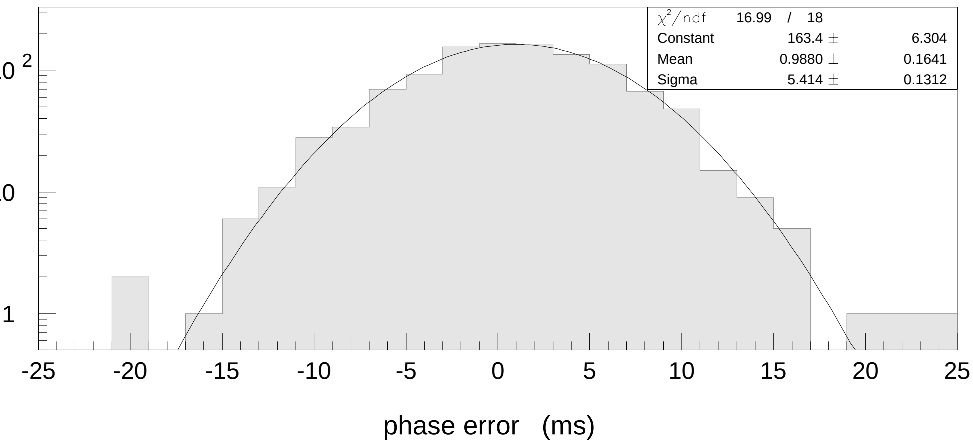



**Figure 43 – Phase error histogram.**

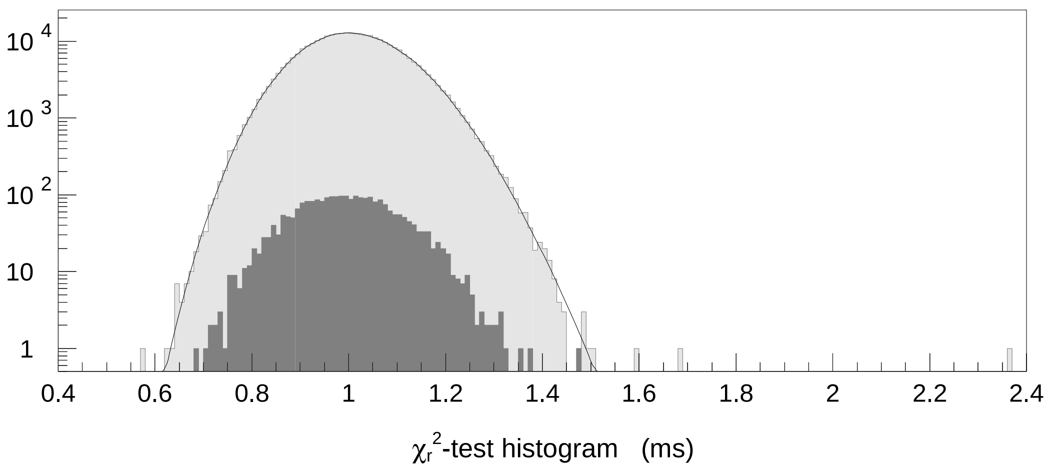



**Figure 44 – The reduced $\chi^2$ histogram for background events is perfectly fitted by the theoretical prediction (a part for an outlier with SNR ≈ 3.5 e $\chi^2 \approx 2.4$). The histogram of events with SNR=34 is somewhat boroader, even if still centerd in 1.**

### 4.1.6 A note on detector calibration

The AURIGA detector mounts a *calibrator*, which is a detuned electromechanical capacitive transducer, placed on the face opposite to the one used to extract the signal. It provides a way to excite the bar with short mechanical bursts that mimic a GW signal. However, during the first cryogenic run it was not possible to excite the system reliably through the calibrator port, because when its electrical port is connected to an external signal generator, the system noise performance were always awfully spoiled.

The AURIGA detector has also a calibration coil between the cryogenic transformer and the pick-up inductance of the SQUID, whose electrical parameters are known, allowing us to inject in the system a well-determined amount of energy. The comparison with the signal received by the ADC give us the ability to convert the measured output voltage in units of strain amplitude. This calibration procedure is model independent. The only assumption is that the system is described by coupled harmonic oscillators. [37]

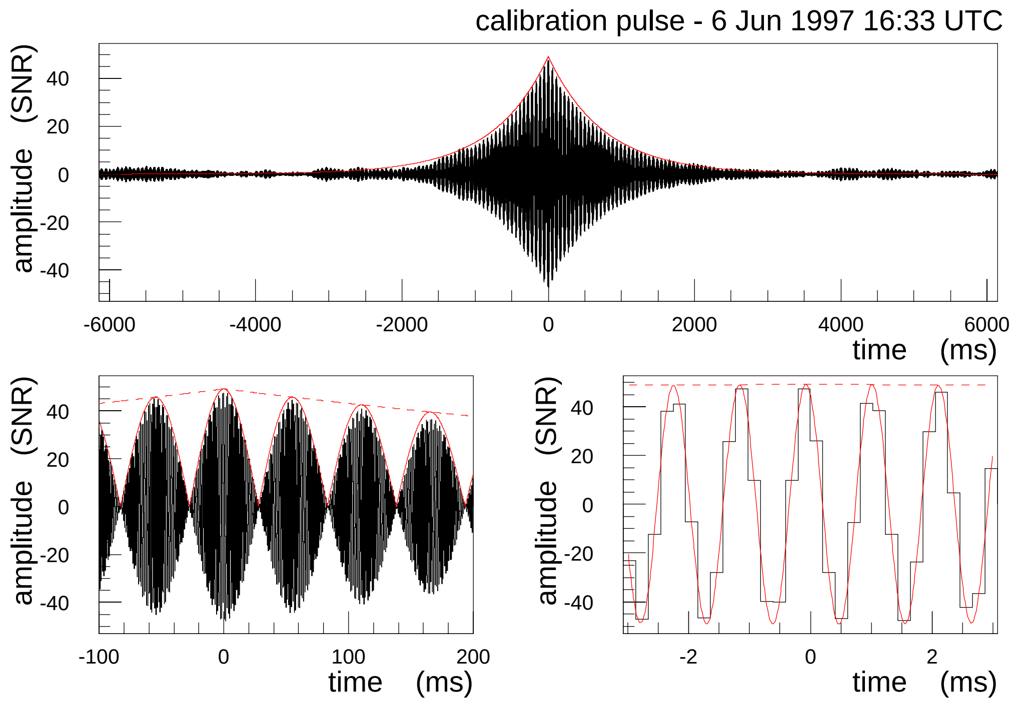



**Figure 45 – a,b,c: A high SNR *hardware* calibration pulse in AURIGA normal operation data pass the $\chi^2$ -test succesfully, demonstrating the validity of the model for the transfer function of the detector. A good match with the model (*red line*) is found at different scales.**

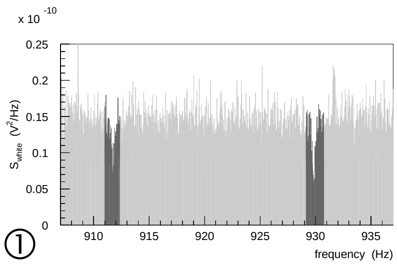
### 4.1.7 Simulations: adaptiveness and parameter recovering

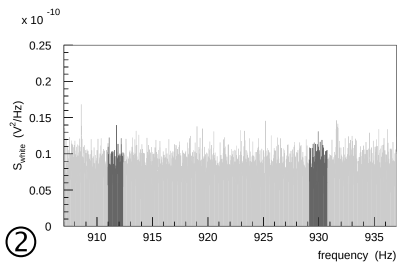
The adaptive estimate procedure of the WK filter parameters has been subdue investigation by simulating with an ARMA model the normal stochastic process described by ( 1.43 ). The analysis was initialized with parameters differing from those used by the simulator, up to a factor 50%, which is still considered in the range of “small parameter correction”.

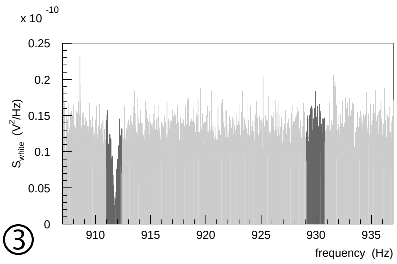
In Figure 45 you can see a few sample images of the whitened data PSD before the first update. The parameter correction to apply for the next hour is computed form numerical lock-ins picking up the energy content of the dark shaded areas. After the first update, the biggest difference observed between the parameters in the simulation and the estimates was of 4%. This figure is well within our needs: from Figure 13 we know that even a variation of ±20% in the bandwidth parameters correspond to only a ±2% variation of the SNR.

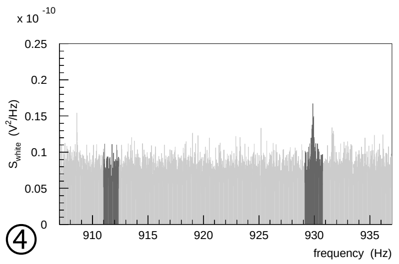
This should guarantee that in good operation periods all biases in the physical quantities associated with events are negligible (provided that everything else is also working, and the update transitins are not freezed).

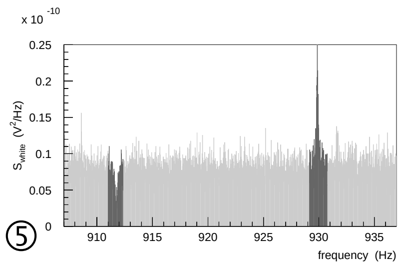
**Figure 46 – ①②③④⑤ – Whitened data PSD estimates in the hour just after the start of the simulation run, before the first update. The corresponding initial parameters are listed in the table below. The darker color put in evidence the frequency span picked up by the adaptive routine to estimate the new parameters.**




| | **Simulation summary** | | | |
|---|---|---|---|---|
| | true values: $\Delta^{opt}_{-} = \Delta^{opt}_{+} = 0.660$ rad/sec | | | |
| | input parameters | | next hour estimates | |
| | $\Delta^{opt}_{-}$ (rad/sec) | $\Delta^{opt}_{+}$ (rad/sec) | $\Delta^{opt}_{-}$ (rad/sec) | $\Delta^{opt}_{+}$ (rad/sec) |
| ① | 0.778 | 0.811 | 0.684 | 0.651 |
| ② | 0.660 | 0.622 | 0.673 | 0.651 |
| ③ | 1.000 | 0.622 | 0.680 | 0.654 |
| ④ | 0.680 | 0.550 | 0.665 | 0.646 |
| ⑤ | 0.820 | 0.500 | 0.654 | 0.667 |

### 4.1.8 Bench test: $\chi^2$ test performances on stimulated events

We shall briefly report here of the first event characterization performed on a real full-sized detector, albeit operating at *room temperature*, i.e. with little –if any– sensitivity to astrophysical sources. Despite this, as long as we are concerned with signal analysis, the system is a noisy equivalent of the ultracryogenic detector, with a lower mechanical quality factor (~1000 instead of ~$10^6$) but with the same post-filtering bandwidth. Therefore it serves perfectly as a "bench test". In fact, there are many advantages in working at room temperature: the mechanical relaxation time is shorter (~1s instead of ~1000s), so it recovers quickly after a high SNR event, and it is possible to collect a vast statistical sample in a few days. Moreover, lacking any cryogenic maintenance activity, it is easier to operate, and is less suffering non-stationarity.

1) *Detector description*

The room temperature detector has already been used for signal timing accuracy measurement [10]. It shares many features with the cryogenic detector: the Al bar, the U-shaped Cu cable suspension and the acquisition system are duplicates of those used in the real detector. A cantilever and lead-rubber piles provide for further mechanical attenuation (-150db overall). The voltage across the capacitive transducer is fed to a low noise (0.5nV·$Hz^{-1}$) FET preamplifier followed by a commercial amplifier. As the cryogenic AURIGA detector, the bench test mounts a *calibrator*, that is a mechanical actuator by which test signals can be sent to probe the transfer function of the system.

2) *Signal generation*

To produce large amounts of identical spurious e.m. signals in the room temperature detector we used a coil placed over the cable bringing the signal from the transducer to the amplifier. A short current impulse in the coil induced a wide band signal in the amplifier, superimposed to the signal from the detector. The experimental setup illustrated in Figure 50 allows us identify precisely the time in which the impulse was sent, and this triggered the selective extraction of the filtered data around the event.

The initial pre-filtering parameters $\omega$ and $\Delta\omega$ for the WK filter were directly estimated fitting as a function of frequency both the output PSD and the real part of the complex impedance seen from the amplifier –which should differ only by a factor $2k_BT$ if we assume that the back action of the amplifier doesn't contribute. Indeed the two measures agreed, and also showed that the noise source was "thermal" within 10%.

The optimum post-filtering bandwidth $\Delta\omega_k^{opt}$ started from a guess value, and was then fine-tuned by means of impulsive signals from the calibrator. That is, the optimum value was chosen so to minimize the $\chi^2$ for these events.

The system was stable enough that the filter parameters were not changed during the following measures.

We verified that for calibration pulses (Figure 47) with SNR as high as 35 the **X** statistics assumed values compatible with 1. Electromagnetic spurious events generated before the amplifier (see Figure 48) were analyzed in two sets of measurements, at fixed and variable SNR.

a) *fixed SNR*

We set the amplitude of the pulse generator so to give a SNR of ~48. Then we sent a series of 10300 events with a delay of 10s. The SNR correctly distribute with a dispersion that substantially agrees with 1, the small deviation being explained as the limit in reproducibility of the signal. The experimental distribution of $\lambda$ was compared with that obtained with a simulation, where the noise PSD of the detector was modeled with an

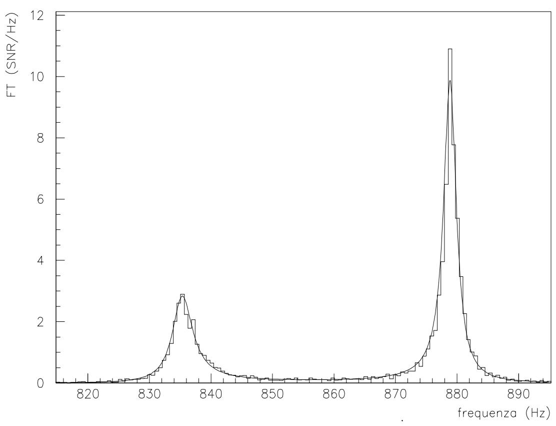



**Figure 47 – Fourier transform of a calibration event after WK filter in the room temperature detector. The smooth, continuous line is what is expected for a gravitational event of same amplitude filtered with same noise parameters.**

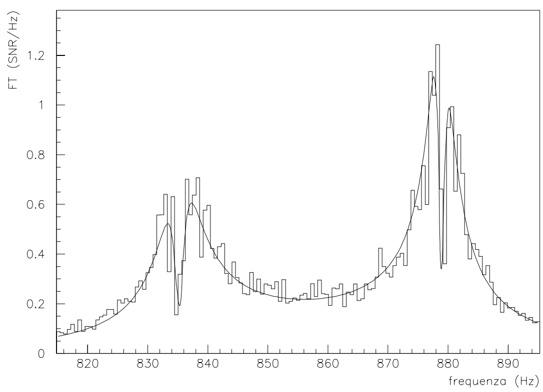



**Figure 48 – Fourier transform of a spurious event generated by e.m. impulse before the amplifier in the room temperature detector. The superimposed smooth line is what one expects from an event with exactly flat spectrum before filtering.**

ARMA equivalent of ( 1.43 ) fed with the measured parameters, and the spurious events were modeled by an exactly flat spectrum (which is justified by the goodness of the fit in Figure 49). The qualitative agreement is good.

*b) variable SNR*

The amplitude of the impulse generator was modulated between SNR=0 and SNR=50 by a triangular wave with period of 10000s, each pulse being sent 20s later than the previous. Background events at small SNR ($10^3$ per hour) were filtered out by time of arrival selection. The distribution shown in Figure 49 fits on average the scaling law of ( 1.34 ). For SNR>10 the vertical spread scales with SNR on the same manner, that is the distribution function of λ is independent on SNR.

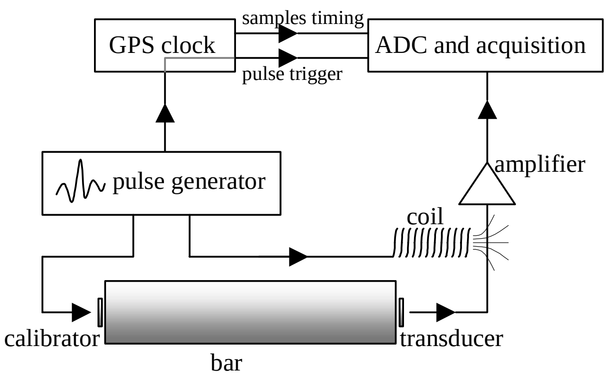



**Figure 50 – Layout of the experimental setup for the room temperature detector. A programmable pulse generator was used to excite either the bar through the calibrator or the amplifier line through an inductive magnetic coupling. It also sent a TTL signal to GPS clock, which passed trigger info to the acquisition system.**

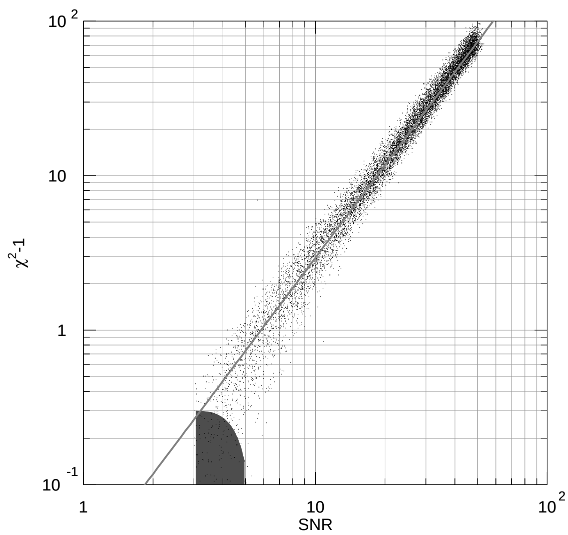



**Figure 49 – $\chi - 1$ vs SNR plot for spurious e.m. impulsive events. The distribution follows a quadratic scale law, as in ( 1.34 ), with an average λ=0.02935 (thick line). The shadowed gray area is a placeholder for background 'thermal' events.**

### 4.2 Simple time-frequency analysis

*The importance of tapering in spectral analysis: overview of the problem and implementation in AURIGA analysis.*

*A new wide band noise estimator, when associated with appropriate tapering in PSD estimates,r result both more accurate and more precise of simple digital lock-ins.*

*SDFT Spectrograms as a 'dynamic' representation of data are particularly useful to track the behavior both in amplitude and frequency of narrow band noise, and to get information on the status of the system at a glance.*

#### 4.2.1 Tapering

Data tapering proves to be a necessary step when amplitude of the spectral lines of the signal to be analyzed spans many orders of magnitude (up to 10) in a few decades. We will show how it can improve both identification of spurious spectral lines in the reduced band of the detector and give the correct estimate for the wide band noise of the SQUID amplifier.

*Tapering* consists in smoothing to zero at its ends the data buffer used for Discrete Fourier Transform (DFT), and this is done by multiplying the samples $\{x_i\}_{i=1...N}$ of the buffer by a properly normalized *windowing function* $\{w_i\}_{i=1...N}$. In the frequency domain this operation corresponds to convolve the discrete Fourier transform (DFT) of the data set with the sampled DFT of the window function (also called its *kernel*).

Even when no tapering is done, in fact a "rectangular" window is implicitly used. The main consequence of all kinds of windowing is *spectral leakage*: the content of a frequency bin is spread among many bins –in principle all bins– according to the amplitude of the kernel centered on the bin. This makes a signal at one frequency appear also in nearby bins, and noise bandwidth to be larger than one single bin.

Another common effect is *scalloping loss*, that is the attenuation of the spectral content seen at one frequency bin relative to a small bandwidth signal placed somewhere between two bins. In the worst case, it is just the ratio of the kernel value one half a frequency sample off the center, to its value at the center frequency.

Except for rectangular window, there is also an effect called *coherent power gain*: part of the signal near the ends of the time ordered data set is lost because of tapering. In fact, a short pulse placed at one ends is completely lost, while the energy of a sinusoidal wave with period much smaller than window length is attenuated of a factor $\sum w_i^2$. To avoid at least the latter effect, we normalize the window so that this sum always adds up to 1.

Suppose that we deal with a stochastic process instead of a deterministic signal. We estimate its PSD S(ω) computing the *periodogram* from a block of N samples:

$$\hat{S}_W(\omega) = \Delta t \left| \sum_{n=1}^{N} w_n x_n e^{-2\pi n i \omega \Delta t} \right|^2 \qquad (4.26)$$

Δt being the sampling time. It can be shown that this estimate has an average value

$$E\left\{\hat{S}_W(\omega)\right\} = \int_{-\Omega}^{\Omega} \left| W(\omega - \omega') \right|^2 S(\omega') d\omega \qquad (4.27)$$

where Ω is the Nyquist frequency. There are two interesting cases: S(ω) is a flat wide band noise or S(ω) is a narrow peak. In both cases the normalization that preserve the integral of S(ω) is $\int_{-\Omega}^{\Omega} |W(\omega')|^2 d\omega' = 1$, which is equivalent to $\sum w_i^2 = 1$.

The variance of a single estimate is[21]

$$\mathrm{var}\left\{\hat{S}(\omega)\right\} \approx \hat{S}^2(\omega) \tag{4.28}$$

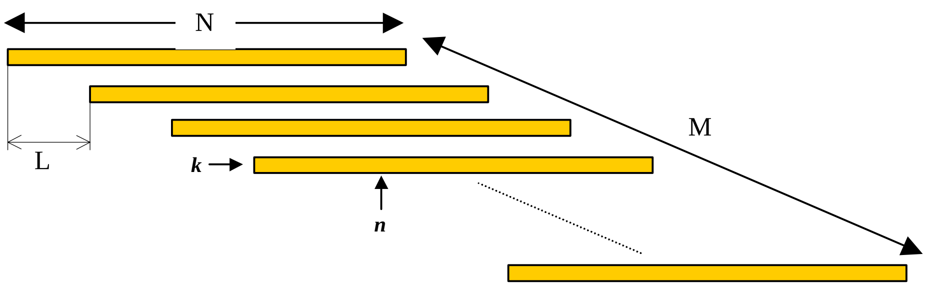
If we take the average of *M* periodograms the variance reduces by a factor 1/M.

The choice of the window to apply depends on many considerations relative to the PSD to be estimated, if it is a signal or a stochastic process, whether it has high amplitude peaks or has a smooth spectrum. The kernel of all commonly used windows appear as a central lobe with a non-null bandwidth, and a series of side lobes, with at least one order of magnitude attenuation relative to the main one, usually extending over all frequencies of the spectrum. For tasks like peak discrimination and wide band noise level estimate it is necessary reducing spectral leakage, hence windows with high side lobes attenuation are to be preferred. The bare rectangular window has a -13dB attenuation with the first null at 1 bin from the center frequency. A *4-terms Blackman-Harris* (BH4) window, given by

$$w_i = 0.35875 - 0.48829 \cos\left(\frac{2\pi i}{N-1}\right) + 0.14128 \cos\left(\frac{4\pi i}{N-1}\right) - 0.01168 \cos\left(\frac{6\pi i}{N-1}\right) \tag{4.29}$$

has a side lobe attenuation of -92dB and a spectral leakage essentially due to the central lobe alone, which spreads the spectral power over about 4 bins. To restore original frequency resolution we take a window length 4 times larger. Of course, this would help also with a rectangular window, but, because of slow roll-off of the side lobes, spectral leakage would be still sensitive at far more than 4 bins.

With a 4·N long periodogram, the variance is 4 times larger, given a fixed number N·M of samples. However, we can do better than this, by *overlapping* the periodograms. A 75% overlapping restores the number M of blocks, but the variance hasn't got a mere factor 1/M, because of correlation between consecutive blocks. Yet due to tapering the correlation isn't




complete, and so there is anyway a good incoming. Welch suggested this combined effect as an improvement of the simple Bartlett method of periodogram averages with no tapering. We refer to it as *Welch's overlapped segment averaging* (WOSA)

There isn't a simple expression for the variance when blocks overlap, anyway we can often do these simple assumptions: $E\{\hat{S}(\omega)\} \approx S(\omega)$, S(ω) is locally constant (this one perhaps isn't so good on the neighbor of a narrow peak) and we do not look to ω~0 or ω~Ω. Then it can be show that

[21] This approximate solution does not take into account spectral leakage, i.e. it holds exactly only in the limit $W(\omega) \approx \delta(\omega)$.

$$\mathrm{var}\left\{\hat{S}(\omega)\right\}=\frac{\hat{S}^2(\omega)}{M}\left(1+2\cdot\left|\sum_{0<n<(N-kL)} w_n w_{n+kL}\right|^2\right) \tag{4.30}$$

where 0<L<N is the number of samples that do *not* overlap between successive blocks.

A numerical calculation with a BH4 window and 75% overlapping gives for the last factor a value of 1.668.

Because of the large central lobe of BH4 the frequency resolution was scarcely increased, so we can smooth the spectrum by taking the average of 4 nearby bins, and then down sampling a factor 4. In this way we reduce another factor 4 the variance, and return to the number of frequency bins we started from. That is:

- we have succeeded in removing (almost) completely spectral leakage;
- standard deviation of the single frequency bin *decreases* by 40%;
- computation time is 4 times longer (using FFT algorithm);
- no change in DFT output length and resolution.

The method was implemented with a C code and tested using a sine wave signal. The bilateral spectral power of a sine wave of unitary amplitude is 0.25, so we were able to check that the correct normalization was used. Three tests of the algorithm are shown for comparison in Figure 51, using a rectangular window (*red*), a simple 3 terms Blackman window (*blue*) and finally BH4 (*black*), before (*upper plot*) and after (*lower plot*) applying frequency smoothing and down sampling.

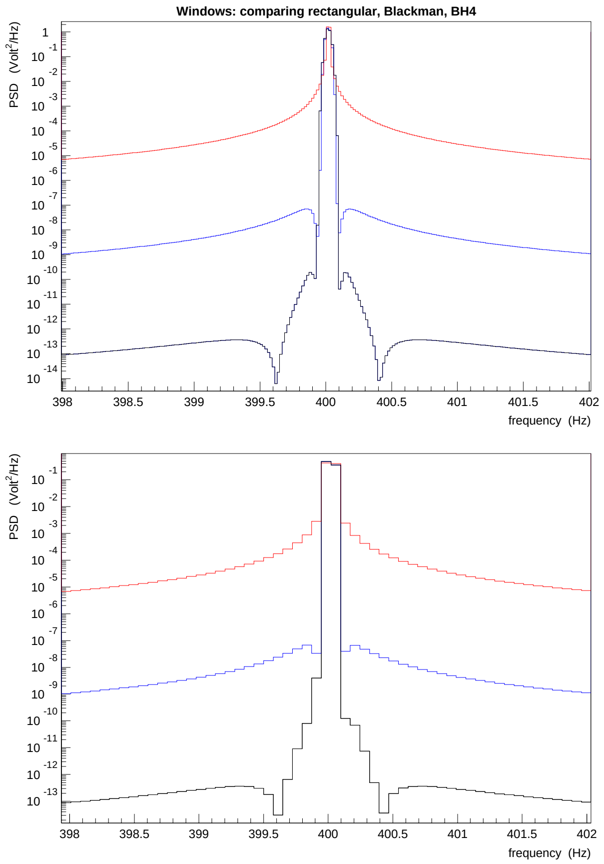
Next we show how the change from Bartlett spectral estimate to WOSA affects a typical hourly PSD of AURIGA raw data (N=65536, M=268). In Figure 52 we compare the results of traditional style estimate (*red*) and a 4 times longer DFT with no tapering. We notice that we missed nearly a factor 2 in wide band noise estimate at 800Hz due to the side lobes of the peak forest at frequencies below 200Hz. The enhancement near the 911Hz mode reveals a small peak at 212.8 Hz. If we now change for BH4 (Figure 52), we see that peaks at low frequency no longer merge at all. The WBN level at 800Hz cuts another 15%, and there is evidence for a bump at 909.5Hz. As Figure 51 clearly shows, the whole improvement of Figure 53 can be entirely explained by the fact we adopted a BH4 window with total side lobe suppression, while in Figure 53 it was due to further roll off because of an increased DFT length. By the way, this means that the residual (i.e. >2 bins) spread of the peaks in the PSD are not an artifact, but reflect the intrinsic bandwidth of the spectral lines.

**Figure 51** (*right*) – (*a*) Frequency representation of a 400Hz sine wave signal as seen with a direct FT of a raw data buffer of AURIGA acquisition software (*red*), with a four times longer buffer (*blue*), and with the latter but after a BH4 tapering (*black*). (*b*) Same thing, but after averaging each four bins.

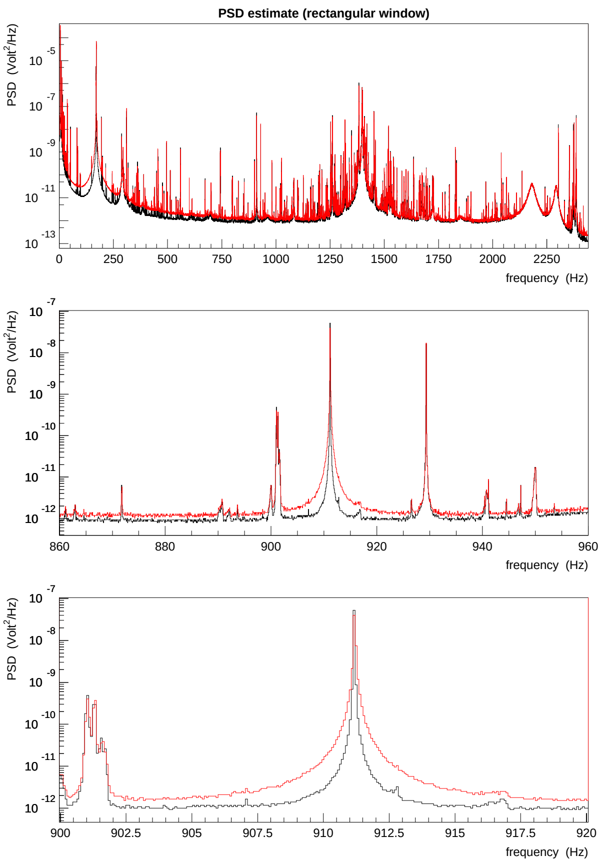
**Figure 52** (*bottom left*) – (*a, b, c*) PSD estimate of AURIGA raw data noise at three different scales, using direct FT on normal sized (*red*) or four times sized raw buffers (*black*).

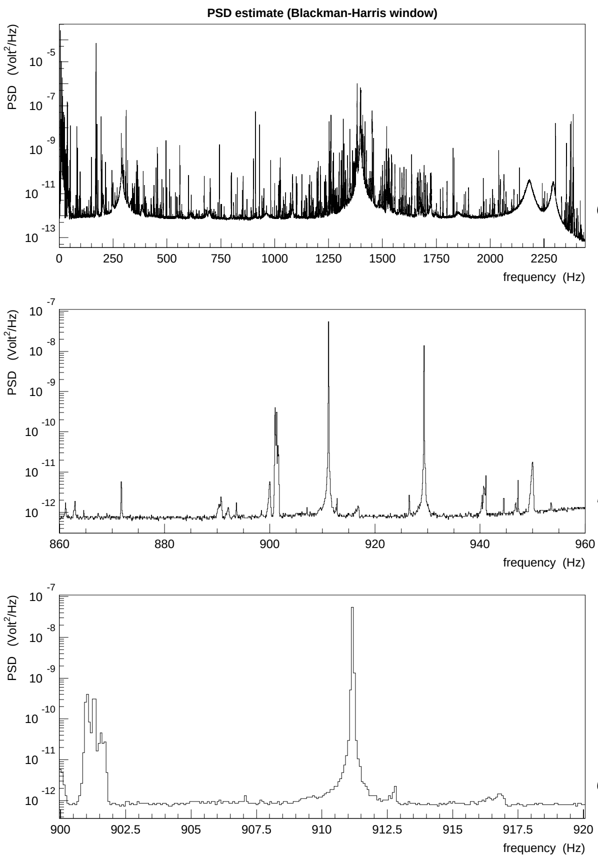
**Figure 53** (*bottom right*) – (*a, b, c*) PSD estimate of the same data of previous figure, but performing BH4 tapering before FT on four times sized raw buffers.

a)
b)




a)
b)
c)




a)
b)
c)

### 4.2.2 Wide band noise level estimate

There are two main tools for wide band noise (WBN) level estimate:

1. a lock-in with central frequency in a region far from peaks or tilts;
2. some kind of fit on the entire PSD estimate.

Method 1 is delicate when handling a WBN superimposed to many powerful sources of narrow-band noise, with high dynamical range in the frequency domain. In fact, if we don't choose the proper kernel (i.e. one with large attenuation at sidelobes) for the band-pass filter, the output of the lock-in is likely to be useless. A Von Hann, Blackman or Blackman-Harris window is indeed to be preferred over a simple Lorentzian.

An example of method 2 is derived from the observation that the most part of AURIGA spectrum bins belongs to WBN. So a projection of the values of the spectrum samples should be peaked around WBN level.

The shape of this peak is a $\chi^2$-like distribution with $n$ degrees of freedom

$$P(S) \propto S^{\frac{n}{2}-1} e^{\frac{-nS}{2\sigma^2}} \qquad (4.31)$$

where $n$ depends on the number of periodograms used (corrected for correlation, if WOSA was used) and how many bins were grouped for PSD smoothing. Typically $n$= 2 (we added two squared components of complex numbers) ×10 periodograms ×4 bin averaging ×0.6 (correction for correlation) ≈48, enough high a value to allow for some approximations that will make the fit simpler. First assume that PSD projection is done with bin spacing given by a geometrical succession. This means to make the substitution $S = \sigma' e^y$. The distribution in the new variable $y$ is approximately normal. In fact, suppose we've chosen σ'≈σ, so that $y$ takes values almost 0 within the width of the distribution P(S), that is ~$\sigma \cdot 48^{-1/2}$ ≈ σ/7. Then we can expand $e^y$ stopping at terms quadratic in y:

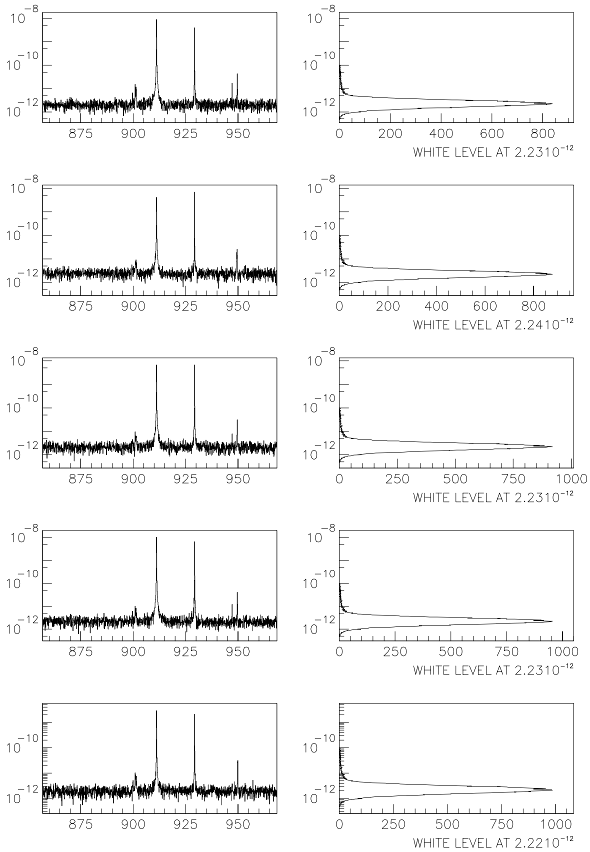



**Figure 54 – White noise level estimate at work. From top to bottom, five consecutive spectral estimates (left side), each one obtained averaging 10 periodograms, with the logaritmic projection of the bin content.**

$$P(y) \propto \exp\left(\frac{n}{2}y - \frac{n\sigma'}{2\sigma}e^y\right) \approx$$

$$\approx \exp\left[\frac{n}{2}y - \frac{n\sigma'}{2\sigma}\left(1 + y + \frac{y^2}{2}\right)\right] \propto \exp\left[-\frac{n\sigma'}{4\sigma}\left(y + 1 - \frac{\sigma}{\sigma'}\right)^2\right] \approx \exp\left[-\frac{n\sigma'}{4\sigma}\left(y - \ln\frac{\sigma'}{\sigma}\right)^2\right] \qquad (4.32)$$

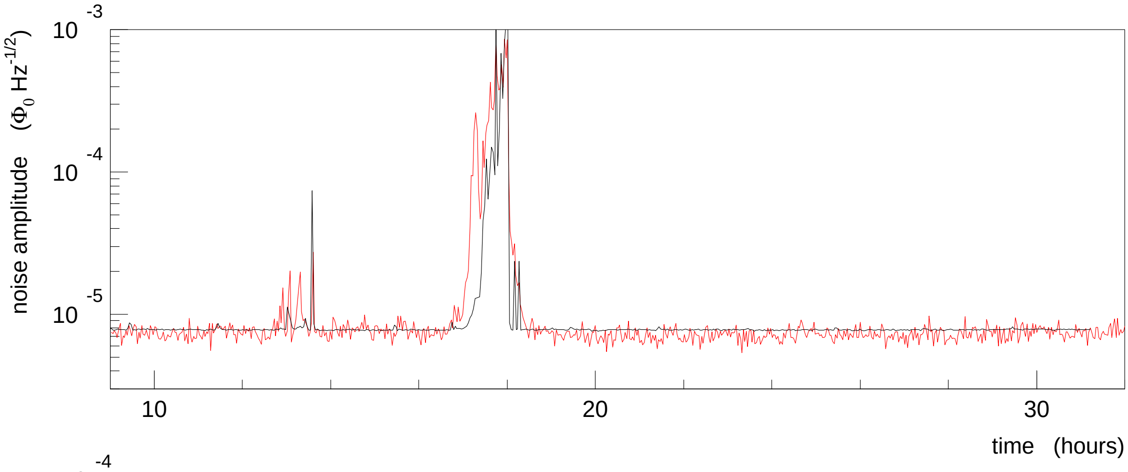
Note that the last expression admit shifting the value of $y$ while preserving the normal distribution, so in fact σ'≠σ, is not really a restricting condition. However, a rough estimate of σ could be useful to help choosing a proper interval and binning for the projection.

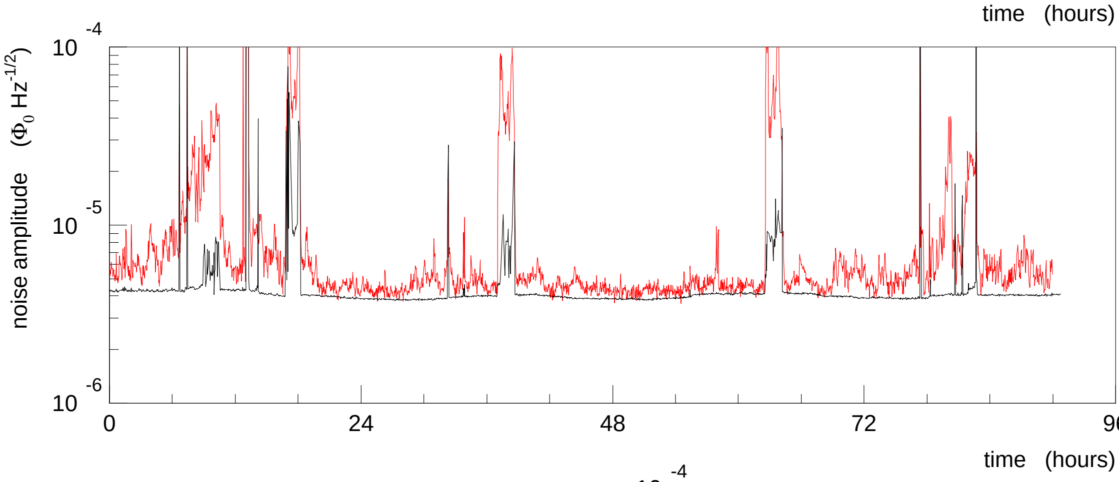
In conclusion, the short recipe is:

- (roughly estimate σ and) project the PSD
- perform a Gaussian fit
- insert the resulting offset $\bar{y}$ back into $S = \Sigma\, e^{y}$ to get the WBN level.

The frequency resolution of standard AURIGA PSD is 1/12 Hz, hence fitting 100Hz near the detector resonance frequencies (890÷950Hz) should give about independent 1000 samples for the projection. Hence, the resulting precision in the estimate is likely to be ~1‰ (the accuracy due to Gaussian approximation is of the same order of magnitude). It is interesting to remark that a goodness-of-the-fit test was implemented and it provides a good way to remove at least part of the outliers in the time sequence.

Both methods were preliminarily applied for comparison on short data stretches taken from AURIGA acquisition system and the results seems agree in quiet days (Figure 55) but when the old lock-in method (with exponential kernel) was experiencing rough changes up to a factor 2 (Figure 56), the projection method affirms in a pretty robust way that it is indeed rather stable (just a slight day/night effect), with a precision consistent with the estimate given above.



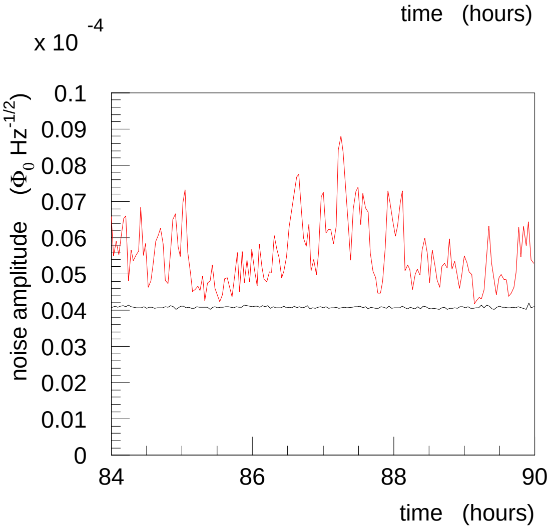




**Figure 55 (*top*) – A stretch of data starting at UTC 9h30' 7 Aug 1997, representing two different estimates of AURIGA wide band noise level: (*black*) a Gaussian fit on a logaritmic projection of the spectrum in the range 860÷960Hz (see text), and (*red*) the output of the average output of a lock-in with a time constant of 1s and reference frequency 880Hz.**

**Figure 56 (*above and right*) – Same as previous figure, but on a data stretch starting at UTC 0h 23 Apr 1999.**