MEMO

Author: J.E Campagne
PAON2/Filtering/17.12.12

Diffusion: R. Ansari, P. Colom, Ch. Magneville, J.M. Martin, M. Moniez, A. S. Torrentò

Subject: Attempt to reduce RFI and other electronic noise with front-end filtering just before visibility computations.

1 Introduction

The DAQ of the PAON2 and later extension of the array is foreseen to take data 100% of the time with online FFT & visibility computations. In order to keep as much as possible only the valuable data, it is investigated to implement a low level filter against RFI and electronic noises. This filtering should be as fast as possible and is intended to validate all the "*paquets*" [1] with the Frame Counter. That is to say, it rejects all the paquets corresponding to the polarization channels if at least one is suspicious, and in counterpart, if all the paquets with the same FC are validated then the computation of all the visibilities (auto & cross correlations) are proceeded.

This filtering procedure cannot follow the algorithm used in the Cluster data processing (Amas/Abell85/21.11.11) which was gathering a time window of paquets to compute a median[2] of the power along the time axis, one per frequency bin. Not only the median computation takes times as it is at least a O(n)-algorithm, but as a matter of principle it is easy to realize that it breaks the time coherence between paquets and so it makes the visibility computations rubbish.

2 Filtering method

A typical power spectrum of a single paquet is shown on Figure 1 (left). It is clear that this spectrum is varying according to the system temperature but first depend on the electronic chain frequency response. To absorb the later dependence, I use a generic response function for each channel (hereafter called "gain") based on a mean of power spectra taken during a quiet run and that I have median-filtered in frequency. Technically, the gains are normalized such that their sums over the 4096 frequency bins are equal to 1. The result of the normalization by the gain is shown on Figure 1 (right).

[1] "*paquet*" is used to nickname a minimal structure of data consisting of a header, then 2x2x4096 2-byte values for the real and imaginary part of FFT coefficient for 2 channels and finally a trailer.

[2] This is different from a median filter as it computes in one shot the median of a complete distribution and not in a list of values resulting of a median computed with a sliding window…

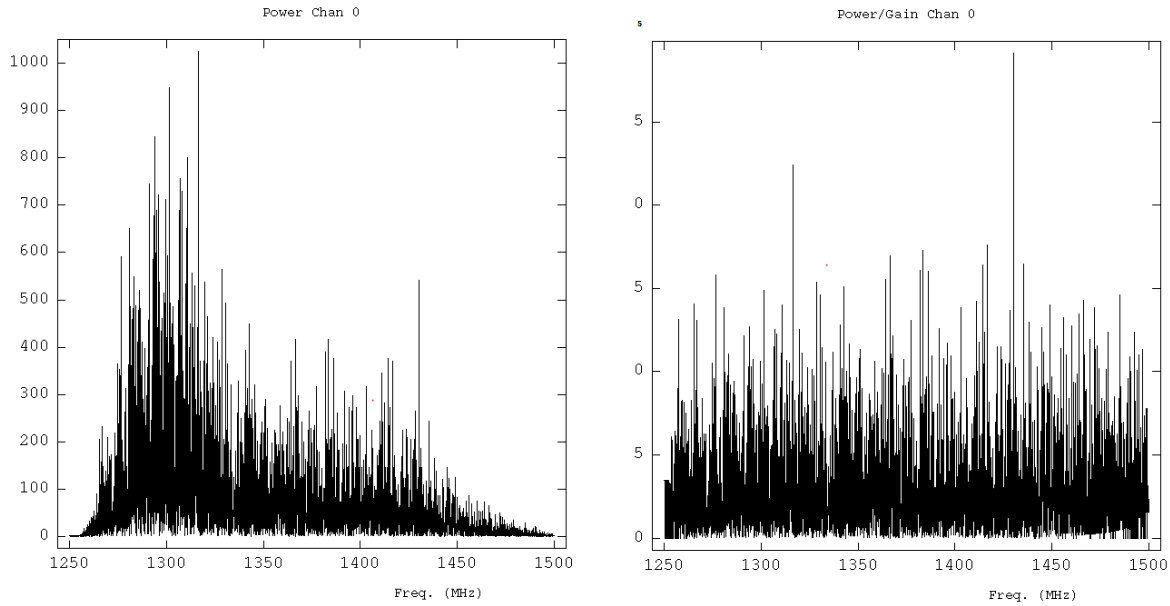

Figure 1 Example of single power spectrum (left) and with gain normalization (right). The gains have been established with the noise run taken 2012-11-21.

Once the power spectrum of a given paquet is obtained, I can use two statistical tests (also called "filter") two detect possible occurrence of RFI and electronic noise. Both use the empirical mean (m) and sample variance (s) computed with the power value (val_i) in each frequency bin except the first one:

$$m = \frac{1}{N_f - 1}\sum_{i=1}^{N_f-1} val_i$$
$$s^2 = \frac{N_f - 1}{N_f - 2}\left[\frac{1}{N_f - 1}\sum_{i=1}^{N_f-1} val_i^2 - m^2\right] \qquad \textbf{Eq. 1}$$

The first filter uses the following property in presence of white noise: *val* is a random variable consisting of the sum of two Gaussian variables squared. As a consequence, it follows a χ^2 distribution with 2 degree of freedom, that is to say an exponential distribution. So, in this case there is a relation between the mean and the standard deviation which is simply:

$$\frac{\mu}{\sigma} = 1$$

So, I have set a cut on the deviation from 1 of the ratio of the empirical mean and sample variance computed according to as:

$$-5\frac{1}{\sqrt{N_f - 1}} < \frac{m}{s} - 1 < 5\frac{1}{\sqrt{N_f - 1}} \qquad \textbf{Eq. 2}$$

It is clear that when a RFI occurs in a frequency domain (ex. 1361MHz) the empirical sample variance is most affected and so the "m/s" ratio tends to be lower than unity. Figure 2 shows an example of the distribution of the "m/s" ratio on real data with a Gaussian fit superimposed. This shows a quite good agreement with a white noise process and the excess at values below 0.93 is probably the sign of RFIs.

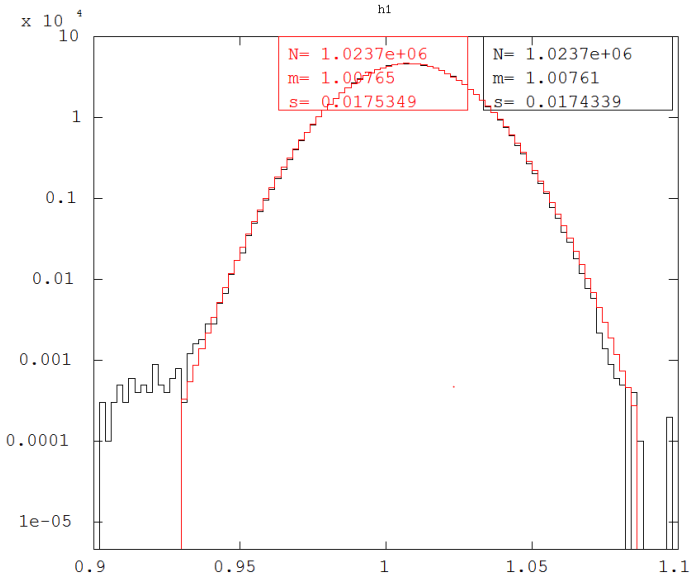

Figure 2 Example of distribution of the ratio of the empirical mean and sample variance (see text). The red histogram is the result of a Gaussian fit on the black data histogram. This shows a quite good agreement over several decades. The lower values of this distribution are the signal of extra power coming probably from RFIs.

The necessity of a second filter has been motivated by the needs to discard paquets affected by the so-called "hyperfine lines". These "lines" were first discovered in Cluster data (see ref. Nançay/Amas/16.03.12 and Nançay/Amas/04.04.12). They may be explained (t.b.c) by the non linearity of the ADCs. Such lines ($\ll$ 1MHz) are considered to be the sign of extra power entering in the ADC. For the moment, probably by ignorance, we want to discard them. But they are not filtered by the first test as can be show on Figure 3.

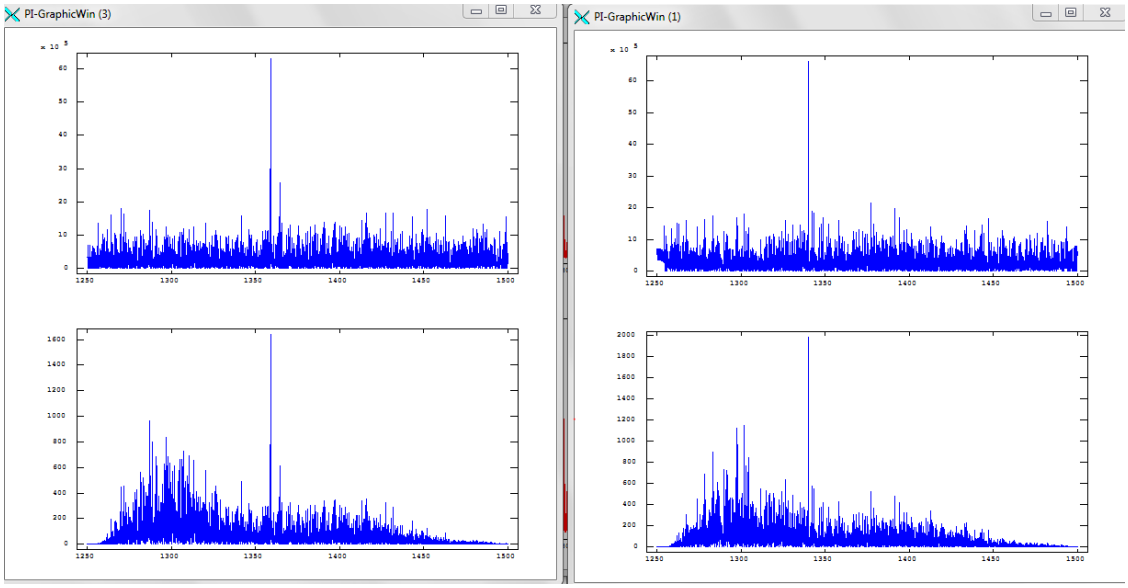

Figure 3 Example of "hyperfine lines" occurring in the power spectra (bottom) and in the normalized spectra (top) at two different moments of a run on the same polarization channel (ie. 0).

To tackle this job, I find relevant to scan the $N_f - 1$ values (val_i) used to compute the empirical mean and sample variance and make a loose cut on the deviation of val_i from "m" ignoring the exponential distribution:

$$-20s < val_i - m < 20s \qquad \textbf{Eq. 3}$$

The cut at "20s" is used to extract only the clear pathological cases. Sometimes more than 1 line could be interpreted as pathological as can be seen on Figure 4.

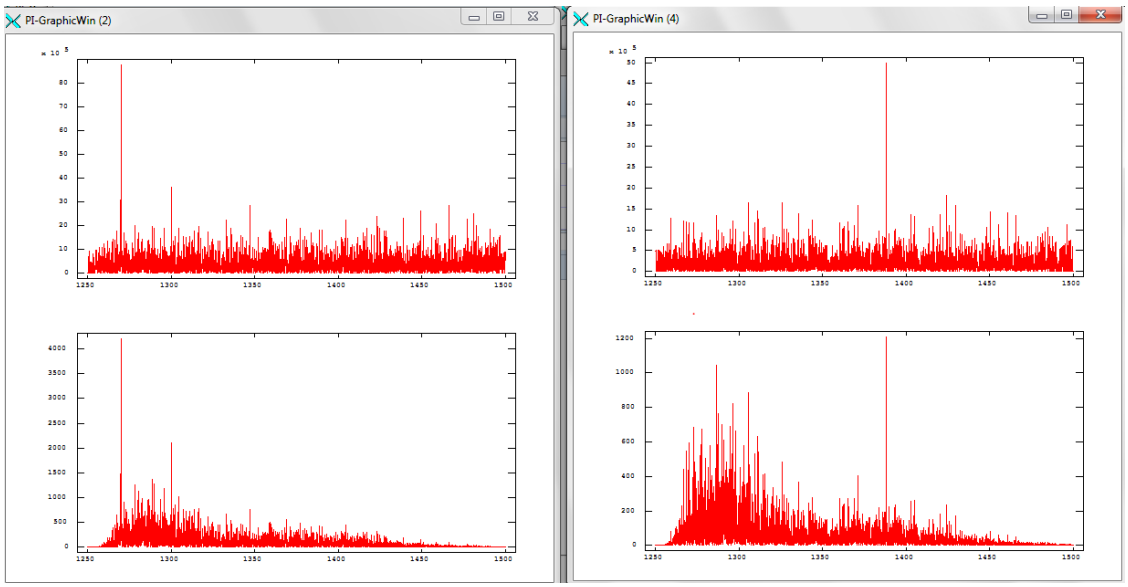

Figure 4 Same kinds of plots as on Figure 3 but for channel 1. One can see on top panels also that more than 1 line may be considered as suspect.

3 Results on a noisy run

To investigate the relevance of the two filters presented in the previous section (Eq. 2 and Eq. 3), I have selected a run taken 21th Nov. 2012 between 19:22 and 21:22l local time. The 2 dishes were pointing towards CygA transit on 13th Nov. 2012. So, I expect that the noise run is not affected at least by CygA.

Figure 5 shows the time evolution of the auto-correlation (power) of channel 0 (polarization West Vertical) in four frequency bands (mean of 1MHz width). The horizontal axis is the mean time tag computed on 1024-paquets samples used to compute the visibilities (~1sec sampling). In black is presented the time evolution without filter while in blue only the first filter is applied and in red the second filter is added. In the same conditions, on Figure 6 is shown the evolution of the cross-correlation between channels 0 and 1.

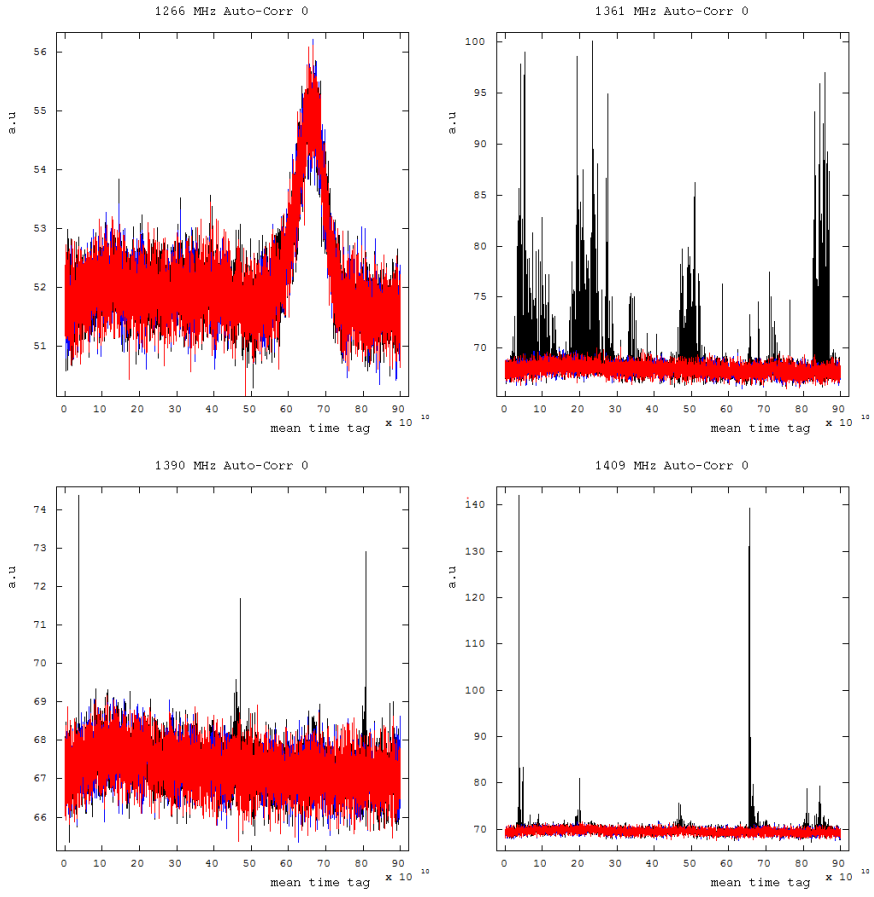

Figure 5 Time evolution of the auto-correlation of channel 0 considering no filtering (black) or filtering (blue and red; see text).

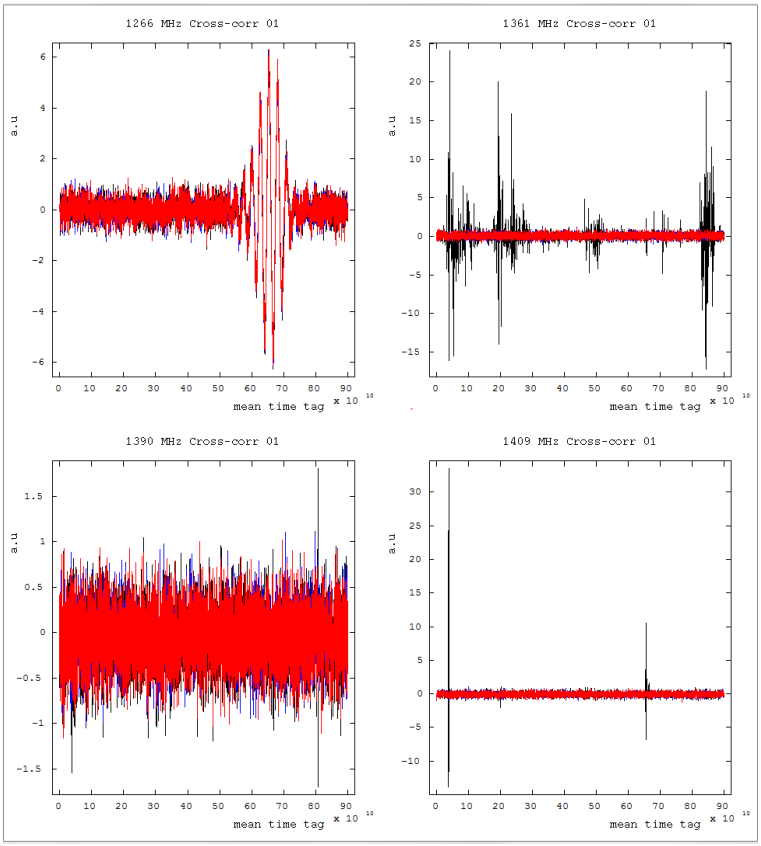

Figure 6 Time evolution of the cross-correlation of channels 0 & 1 in the same conditions as Figure 5.

First of all the filtering are quite effective against RFIs and do not disturb a possible radio source signal. Secondly, there is very little difference when applying the "hyperfine line" filter compared to the "white noise motivated" filter, but this is only the sign that the 4 bands presented are not so much affected by the ADC non linearity. In terms of the number of paquets rejected, this is very dependent on the frequency band in a given run, and might change run to run, so it is not a relevant parameter.

For sure with the loops over the frequency bins, the computations of the empirical mean and sample variance are the heart of the two filters. So, instead of the usual Eq. 1 algorithm, I have used a "one pass" algorithm based on the following recursive equations[3]:

$$\begin{aligned} \mu_k &= \mu_{k-1} + \frac{x_k - \mu_{k-1}}{k} \\ q_k &= q_{k-1} + (x_k - \mu_{k-1})(x_k - \mu_k) \\ \mu_0 &= q_0 = 0 \\ s_N^2 &= \frac{q_N}{N-1} \end{aligned}$$

Eq. 4

[3] http://en.wikipedia.org/wiki/Standard_deviation#Rapid_calculation_methods. This algorithm may be implemented in Sophya framework.

On a CCIN2P3 machine in batch with nthread equals 2, I get about 2.2 10^6 MegaFlop at 654 MFlops/sec without filtering, 433 MFlops/sec applying the first filter and 361 MFlops/sec applying the two filters.

4 Summary & Outlook

In this MEMO, I have investigated the possible introduction of a filtering procedure on the paquets as front-end of the visibility computations. I have shown that relying on two simple statistical tests we can master some RFIs and electronic failures. I propose to implement these filters in the online DAQ program to keep only valuable data during long period of observations (several hours). By these long observations, we will judge of some possible drift that may originate from the “gains” determinations.

The filtering algorithm may be also implemented in the NRT BAO online program for future observations that will occur in the first quarter of 2013.

Concerning post-processing filtering, a demonstration has been reported by the author (not in this MEMO) of the usefulness of median computations for each frequency bins on time windows gathering the visibilities. This study will be pursued further.