



SAPIENZA
UNIVERSITÀ DI ROMA

Update on simulation for OF trigger

BULLKID analysis meeting

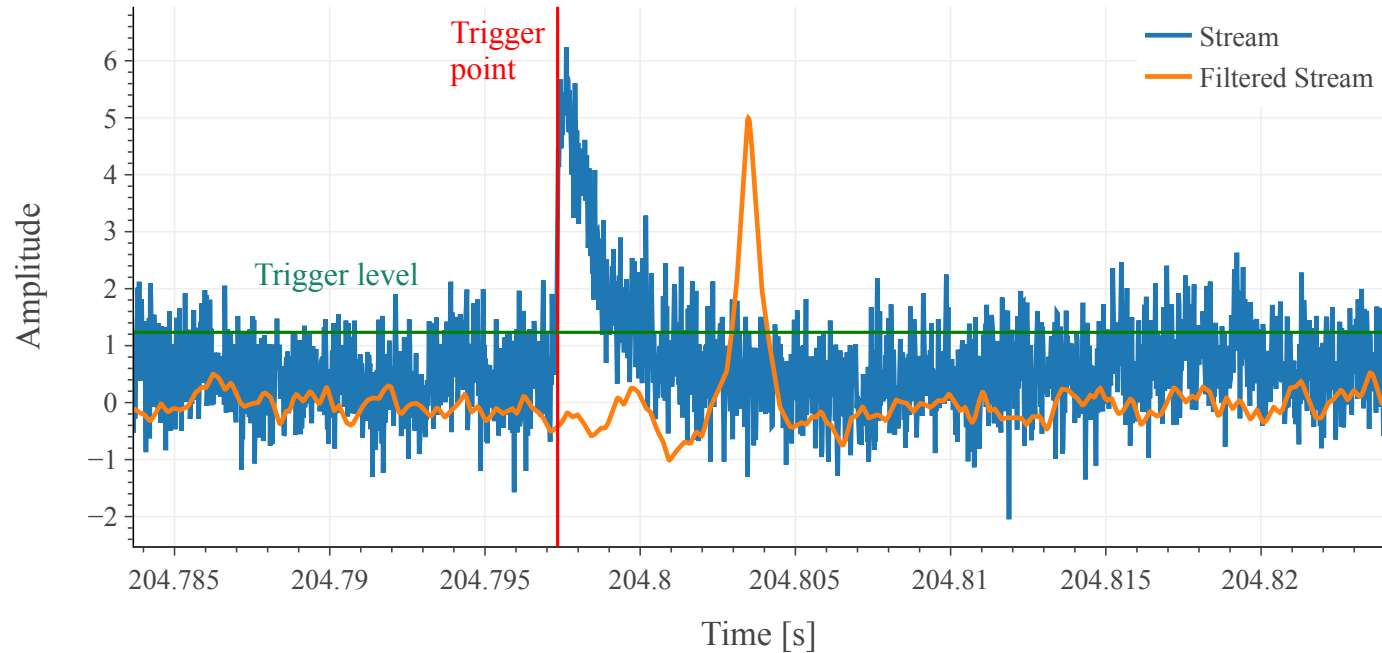
18/03/2026

Matteo Cappelli

OF trigger

We use a trigger based on the Optimum Filter (OF) to lower the **trigger threshold**

- Filter the stream with the OF
- Trigger on the filtered stream



OF trigger

Filtering in the frequency domain

$$v_{\text{flt}}(t) = \int_{-\infty}^{\infty} df e^{i\omega t} \tilde{H}(f) \tilde{v}(f)$$

$$v_{\text{flt}}[k] = \sum_{j=0}^{N-1} e^{i2\pi \frac{k}{N} j} \tilde{H}[j] \tilde{v}[j]$$



Convolution in the time domain

$$v_{\text{flt}}(t) = \int_{-\infty}^{\infty} dt' H(t - t') v(t')$$

$$v_{\text{flt}}[k] = \sum_{j=0}^{N+1} H[(k - j + N) \% N] v[j]$$

Being H the filter and v the waveform to filter. The OF trigger in the new electronics will work on time domain data, we will need to do the **convolution** in the time domain.

Problems:

- Resources: for each channel, $O(2000)$ in the final experiment, we need **N weights** for the convolution, being N the **window length**
- Time: the filtered sample k needs $N/2$ points in the past and $N/2$ points in the future (or N points in the future) to be computed

Solution investigated:

- **Downsample** the stream to reduce N

New simulation

The simulation to see how much we can downsample the stream has been **modified**
Simulate pulses over a noise stream and then trigger with the OF trigger implemented in Diana

Simulation parameters:

Stream time: 6656 s

Sampling frequency: 50 kHz

$\tau_r = 100 \mu\text{s}$ (rise time)

$\tau_d = 1 \text{ ms}$ (decay time)

Window length = **512 points** (10.24 ms)

Noise contributions: flat + 1/f + 1/f²

σ noise (non filtered) = 1

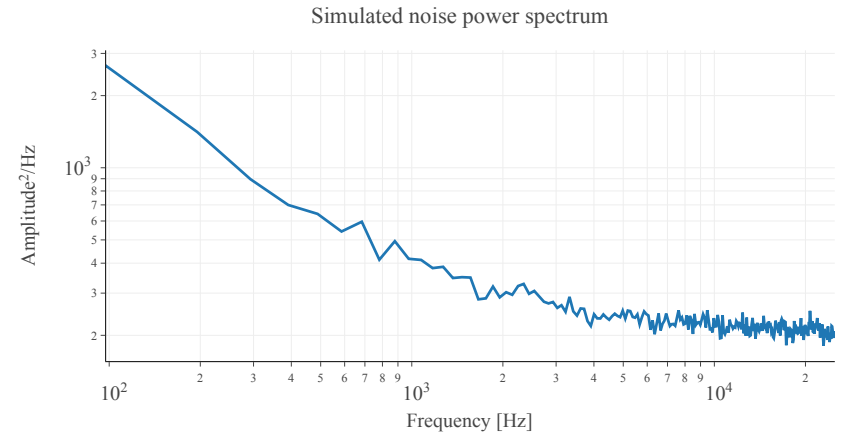
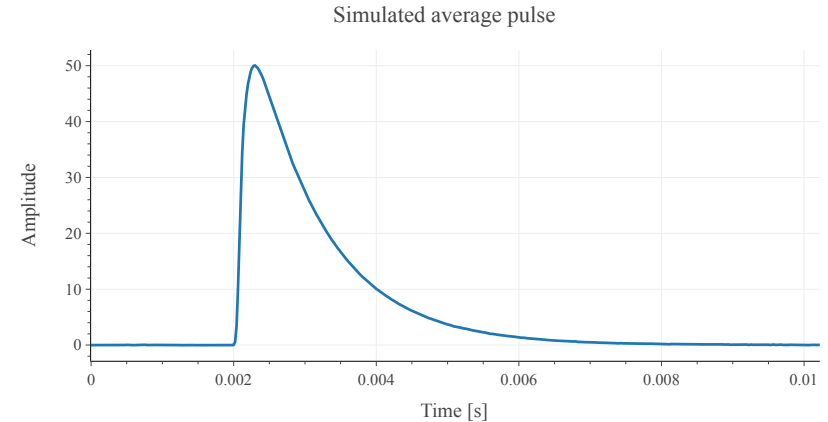
Pulse generated with the following amplitudes:

[0.3, 0.6, 0.8, 1, 1.1, 1.2, 1.25, 1.5, 1.7, 1.9, 2, 2.5, 5]

Downsample the stream at factors: [2,4,8,16,32,**64**]

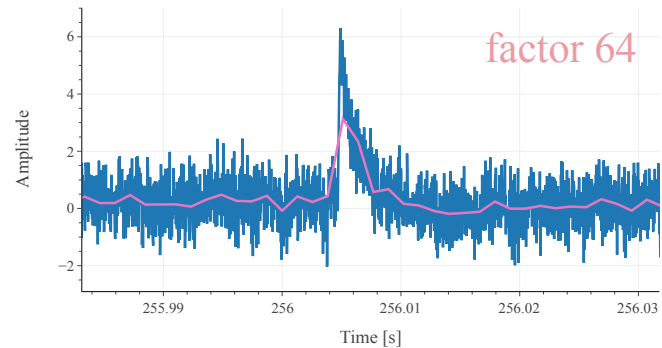
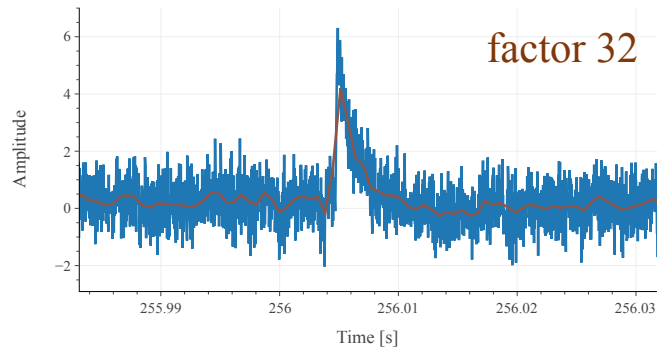
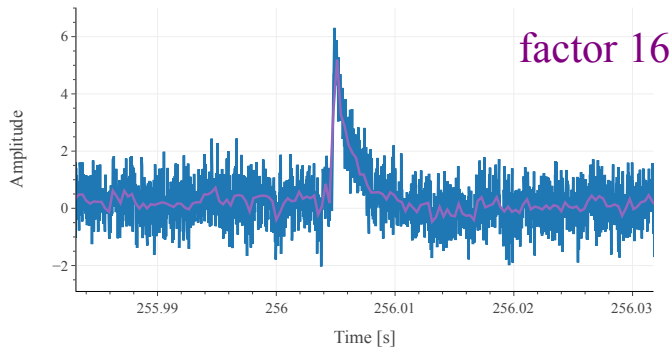
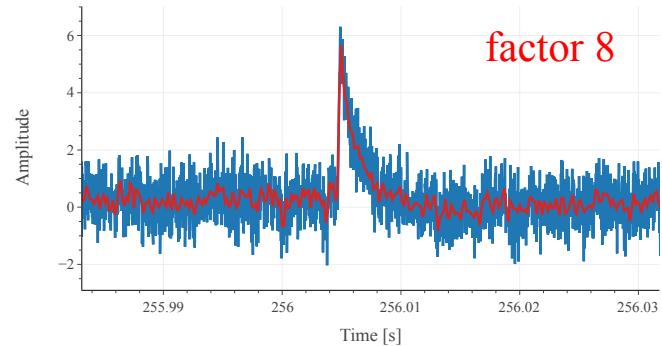
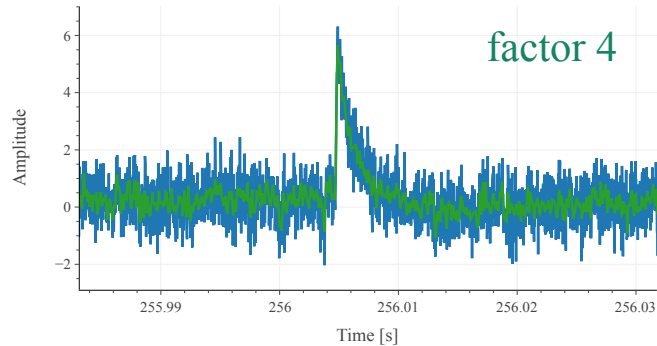
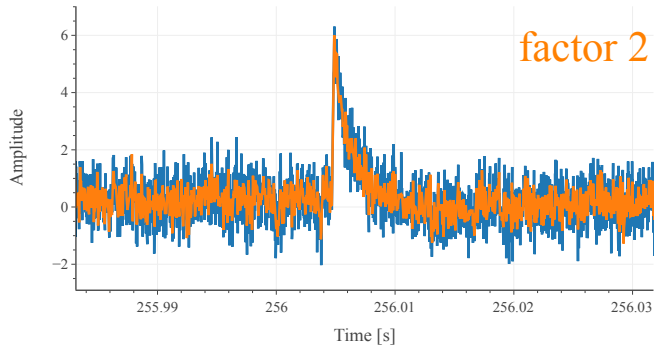
Pulse model:

$$f(t) = \Theta(t - t_0) \left[-e^{-\frac{t-t_0}{\tau_r}} + e^{-\frac{t-t_0}{\tau_d}} \right]$$



Downsample streams

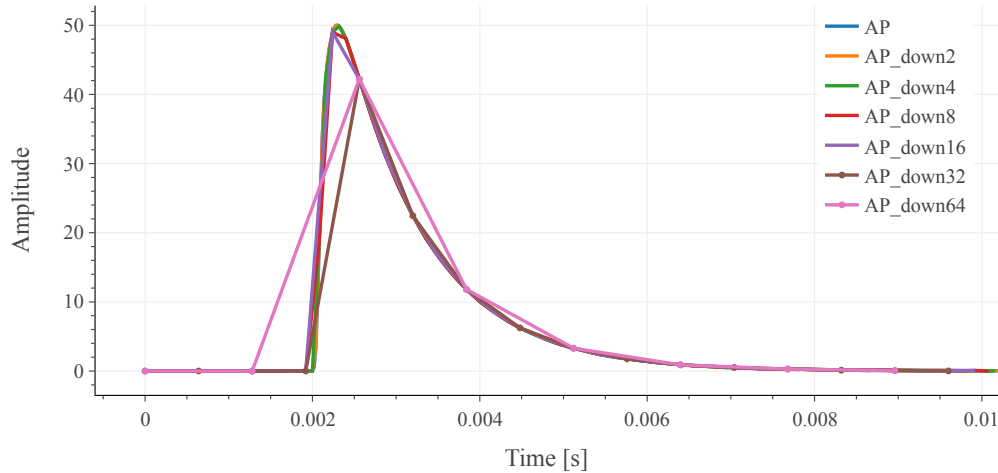
Downsampling the stream using `scipy.signal.decimate`, that includes an antialiasing filter.



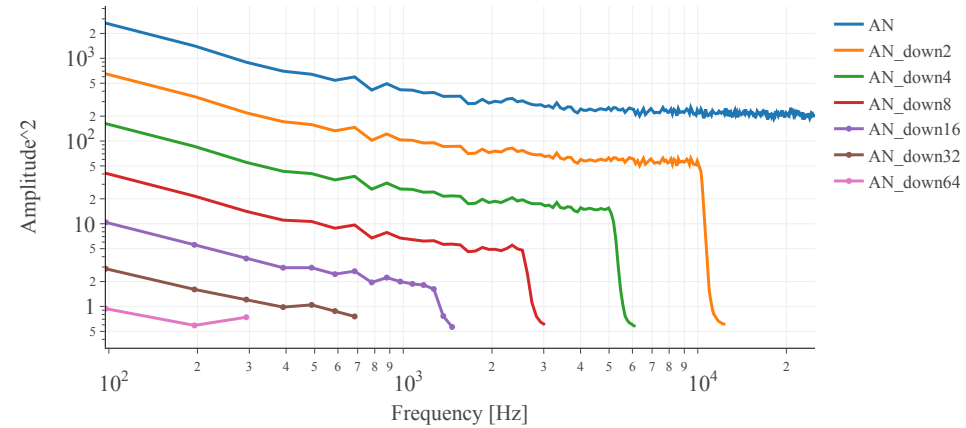
Filters

For each downsampling factor, we build a different optimum filter used for triggering, with the template pulse and the noise power spectrum (NPS)

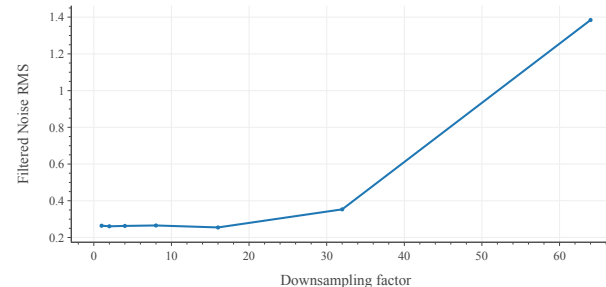
The template pulses are downsampled taking less points on the original template pulse, without antialiasing



The NPS are re-computed from the downsampled noise traces



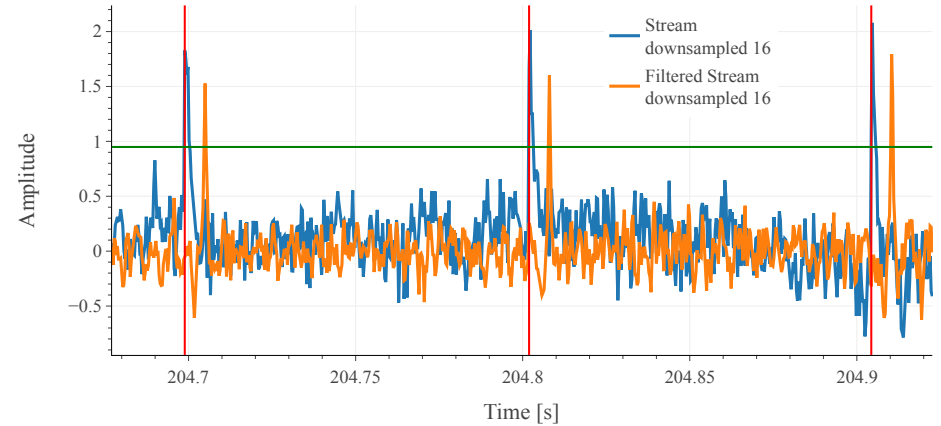
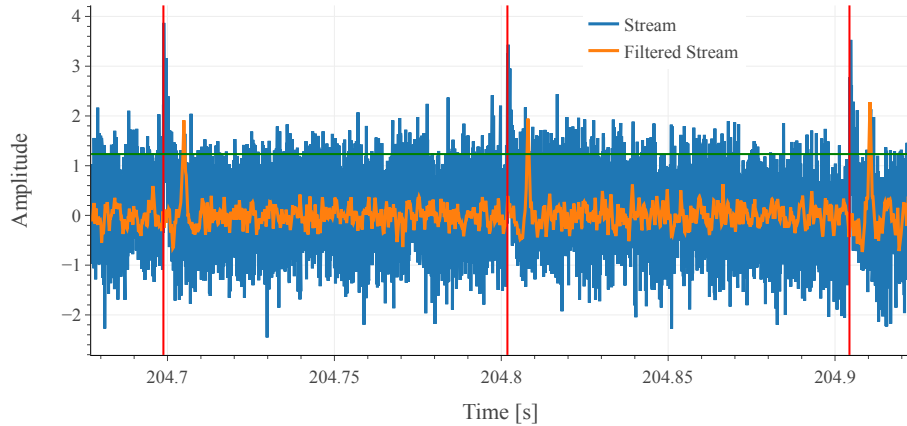
The RMS of the filtered noise changes appreciably only from downsampling = 32



Filter and trigger streams

For each downsampling factor, filter and trigger the streams with the corresponding filter.

Trigger points and trigger level.



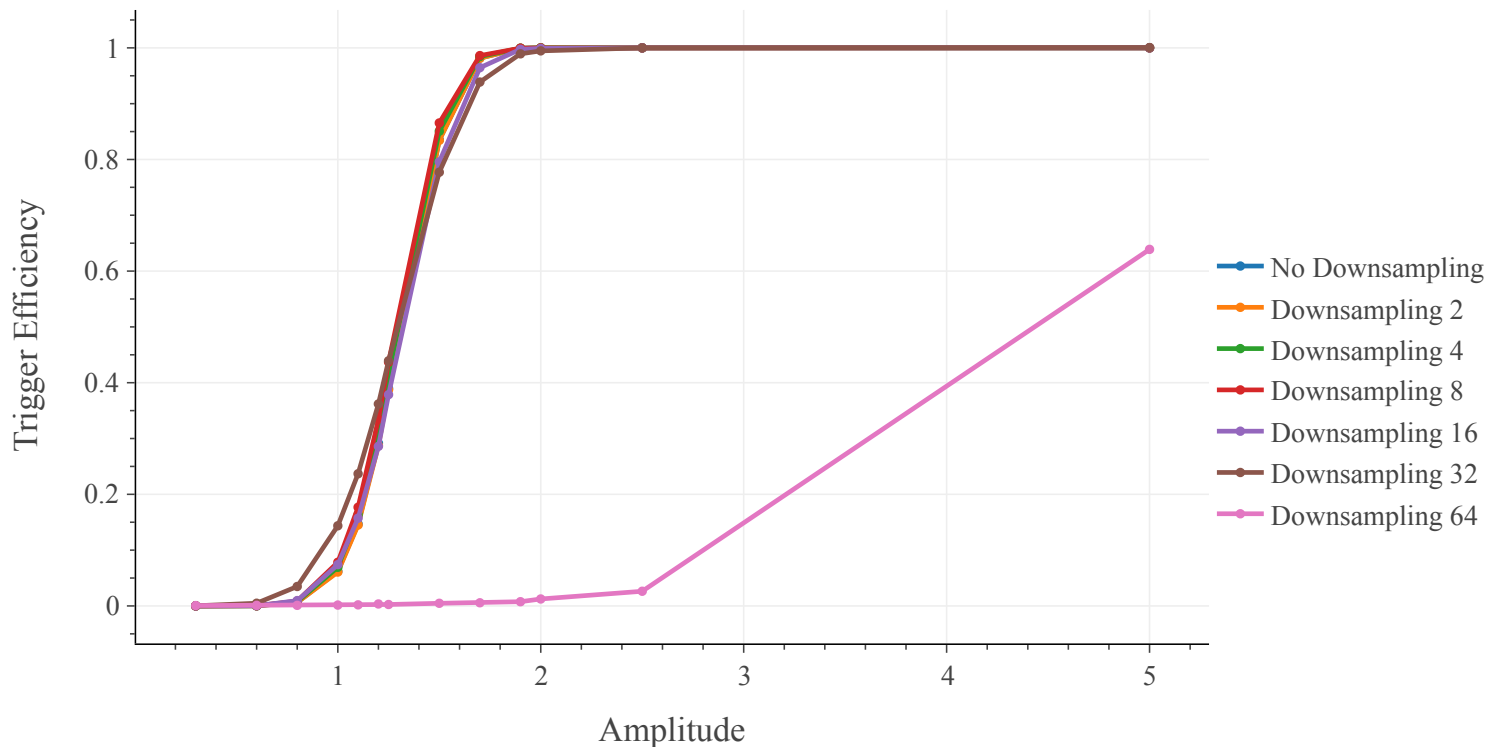
Downsampling factor	Threshold [σ]
1	5
2	5
4	4.9
8	4.7
16	4
32	2.9
64	2.5

In order not to lose events, we need to decrease the **trigger threshold** the more we downsample.

Trigger efficiency

The efficiency curves are now aligned to have the **same trigger efficiency** for every downsampling factor (except for the last one).

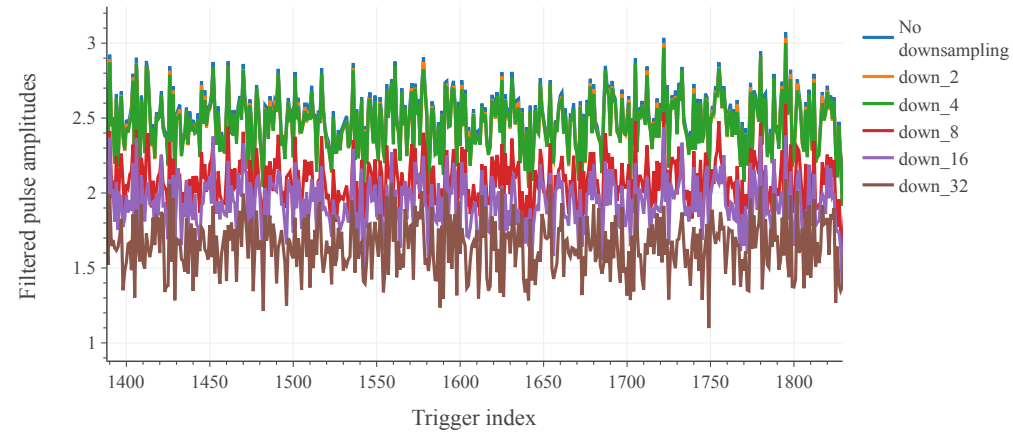
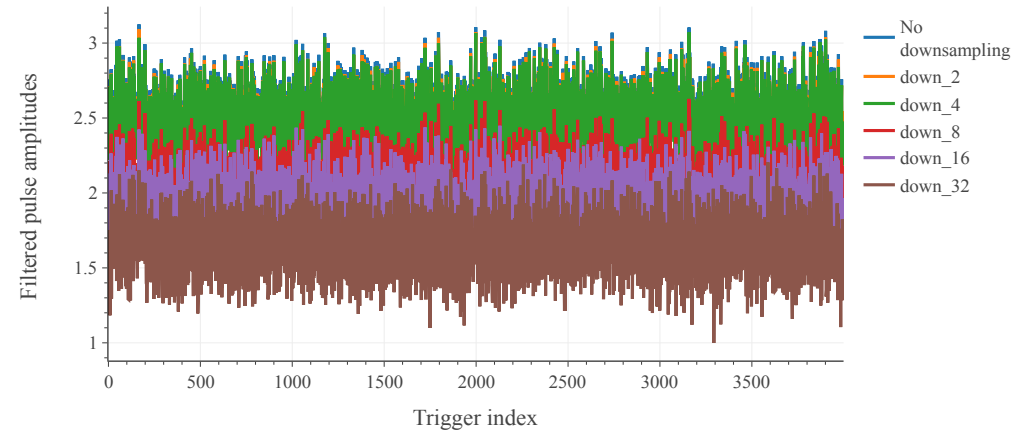
What about false positives? (they could increase due to the low threshold)



Rate of false positives

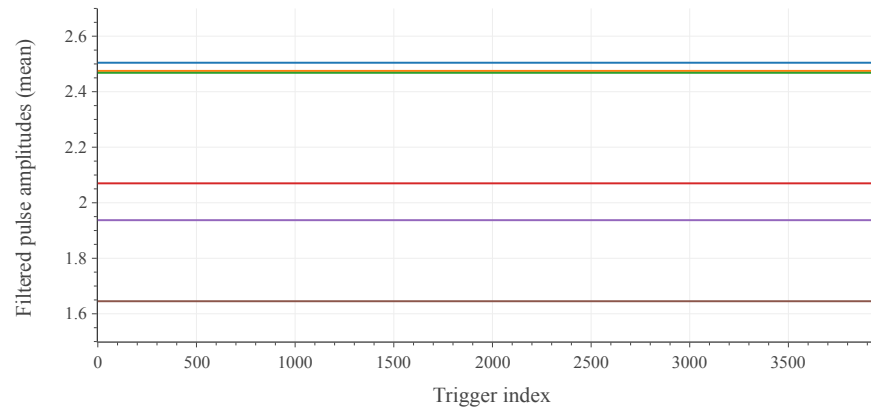
Why the rate of false positives does not increase when triggering at low threshold on the downsampled streams?

Reconstructed amplitudes (the injected one is 2.5):



The reconstructed amplitude of the filtered pulses decrease with the downsampling.

The noise RMS, on the other hand, does not change so much.

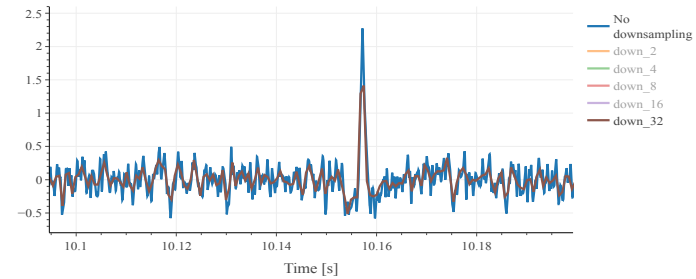
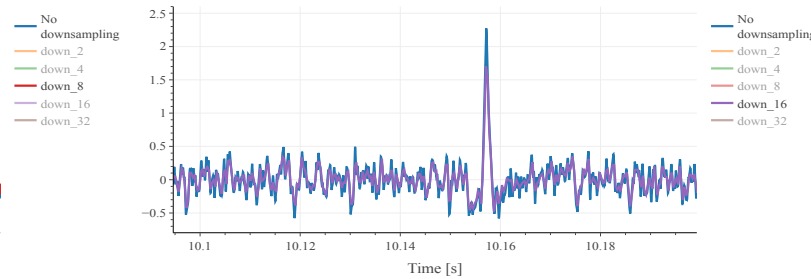
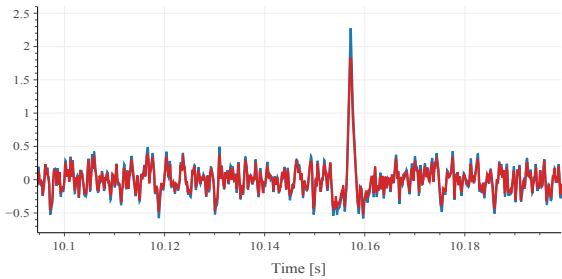
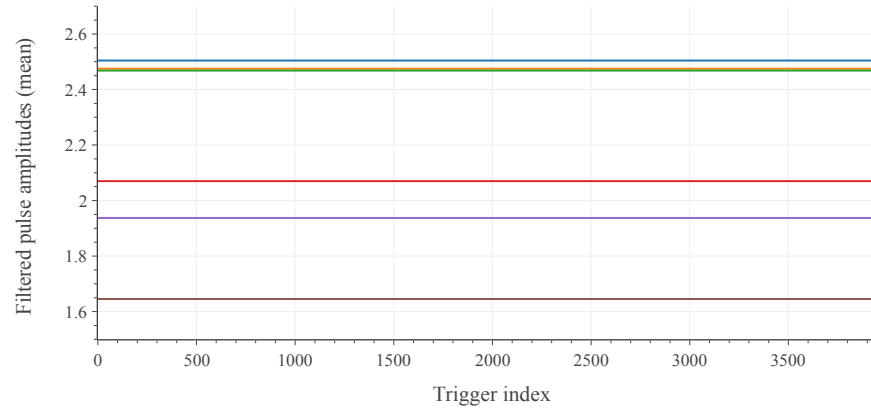


Rate of false positives

Why the rate of false positives does not increase when triggering at low threshold on the downsampled streams?

The reconstructed amplitude of the filtered pulses decrease with the downsampling.

The noise RMS, on the other hand, does not change so much



Conclusions

Results

- Window length was decreased to 512 points and downsampling extended to a factor 64
- We can **downsample up to a factor ~8-16** lowering the threshold without losing efficiency and **without introducing false positives**
- We can work with window length of 512 at 50 kHz. With downsampling at factor 16 this means **just 32 points** on the window (and **32 filter coefficients**)
- To be implemented on FPGA as an hardware online trigger (with the help of Timo). Downsample + FIR filter (?)

To do:

- See if the antialiasing when downsampling the stream is needed.
- Think about ways to reduce computational costs (IIR, math)
- See if a single filter for every channel is sufficient



SAPIENZA
UNIVERSITÀ DI ROMA

thanks for the attention
