A new synthesis approach for non-uniform filters in the log-scale: proof of concept

Brigitte Bidegaray-Fesquet* and Laurent Fesquet†
*Univ. Grenoble Alpes, CNRS, Grenoble INP‡, LJK, 38000 Grenoble, France
Email: Brigitte.Bidegaray@univ-grenoble-alpes.fr
†Univ. Grenoble Alpes, CNRS, Grenoble INP‡, TIMA, 38000 Grenoble, France
Email: Laurent.Fesquet@univ-grenoble-alpes.fr
‡ Institute of Engineering Univ. Grenoble Alpes

***Abstract*—We theoretically describe and give the proof of a new way to synthesize filters that are affine in the log–log scale in the frequency domain and are especially appropriate to filter non-uniformly sampled data, and take advantage of a very low number of signal samples and filter coefficients. This approach leads to a summation formula which plays the same role as the discrete convolution for the usual finite impulse response filters.**

Keywords–Non-uniform sampling, filtering, log-scale.

I. Introduction

As the energy is becoming a major issue for most of the autonomous electronic systems powered by batteries or energy harvesters, such as smart sensors and many IoT devices, finding signal processing techniques able to reduce the computational load is crucial. The aim of this work is to provide a new method for synthesizing filters able to drastically reduce the filter coefficient numbers compared to classical filters. Moreover, the approach is coupled to a level-crossing sampling scheme known to spare the signal samples. This two techniques are able to strongly decrease the filter computational load and make this approach relevant for autonomous electronic devices.

For a large class of signals, especially sporadic signals, non-uniform sampling leads to a reduced number of samples, compared to a Nyquist sampling. To have the lowest possible number of samples is an important issue to reduce the power consumption of mobile systems, thus allowing to increase their autonomy and/or reduce their size and weight [1]–[4]. This goal can be achieved combining specific system architectures (e.g. event-driven) and associated specific signal processing theory [5], [6].

Algorithms to filter non-uniformly sampled signals have already been investigated [7]. A first natural approach is to use classical (uniform) filters, and the usual discretization in time of their impulse response. This leads to non-uniform finite impulse response filtering techniques based on interpolation [8] and infinite impulse response filtering in the state representation, using specific schemes for solving ordinary differential equations [9].

Since the impulse response is highly oscillatory, a further work investigates a way to use the frequency response and interpolate it in the frequency domain [10]. In this paper, classical filters (Chebyshev and Butterworth) are used to illustrate the algorithms. The filters are non-uniformly sampled in the time domain using a level crossing scheme to select samples. They are then linearly interpolated, and different ways to do this interpolation are discussed.

We want to go here further in this direction. Instead of interpolating existing filters, we want to define new ones that are explicitly designed to achieve high-order with a very limited number of samples to describe their frequency responses. In Section II, we define how the non-uniform signal is described and how we define a non-uniform filter in the frequency domain. In Section III we give the notations used to define the associated numerical filter and give the main result of the paper, i.e. the filtering formula, which is proved rigorously in Section IV. Section V is devoted to two examples, a toy one namely a frequency-sweep signal and a real one with an ECG signal.

II. Signal and filter representation

We consider an analog signal $s(t)$ in the time domain and an analog filter frequency response $H(\omega)$ in the frequency domain. For these continuous functions the result of the filtering process, $x(t)$, is the convolution of $s(t)$ with the impulse response $h(t)$

$$x(t) = \int_{-\infty}^{+\infty} h(t-\tau)s(\tau)\ d\tau, \tag{1}$$

and h is the inverse Fourier transform of H:

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} H(\omega)e^{i\omega t}\ d\omega. \tag{2}$$

Both the analog signal and the filter frequency response are sampled non-uniformly, and the first result we will have to make explicit Eq. (1) in terms of the samples.

A. Non-uniform signal description

A non-uniformly sampled signal consists of a series of N samples $(s_n, t_n)_{n=1,\dots,N}$. In this representation, s_n is the amplitude of the n^{th} sample and t_n is the timestamp of the sample.

The signal $s(t)$ is approximated by a linear interpolation of the non-uniform samples. Different ways to proceed are described in [10]. In all cases, the approximated input signal $\bar{s}(t)$ is an affine function on the time interval $[t_{n-1}, t_n]$ given by $\bar{s}(t) = a_n + b_n t$, where a_n and b_n are easily computed from the samples.

We address applications where a level crossing scheme is used to capture the samples [11] and where only time delays $\delta t_n = t_n - t_{n-1}$ between two samples are captured thanks to a local clock. All these features have no impact on the description of the filtering algorithm. We do not use here the fact than levels are quantized, and it is easy to pass from the δt_n to the t_n.

B. Non-uniform filter description

We only consider here symmetric filters with respect to the zero frequency. A filter is described by K non-uniformly distributed samples $(H_k, \omega_k)_{k=0,\dots,K}$ which only yields the positive part of the filter frequency response. The negative part can be deduced since a symmetric filter has even amplitude and odd phase.

1) Filter interpolation: In [10] classical filters such as Chebyshev or Butterworth filters are considered. Their filter frequency response is sampled (in the frequency domain) using properly placed thresholds and a linear interpolation is proposed. Since the amplitudes of the samples of the filter frequency response are complex numbers, there are many ways to linearly interpolate. A simple solution is to interpolate in the complex plane

$$H(\omega) = a_k^\omega + b_k^\omega \omega \text{ for } \omega \in [\omega_{k-1}, \omega_k],$$

where a_k^ω and b_k^ω are complex numbers, but this leads to too much attenuation between samples.
The best way, discussed in [10], is to interpolate the modulus and the phase separately, leading to a more complex description

$$H(\omega) = (a_k^\rho + b_k^\rho \omega) \exp(i(a_k^\theta + b_k^\theta \omega)) \text{ for } \omega \in [\omega_{k-1}, \omega_k],$$

where now all the coefficients, a_k^ρ, b_k^ρ, a_k^θ, and b_k^θ are real numbers.

If the thresholds are well chosen, we can describe in this way very precisely already existing filters with a low number of samples, namely lower than the order of the filter. However these classical filters have been designed to be used with classical uniform signals and coefficients in the time domain. They may not be the best suited for non uniform signals. The use of new signal paradigms is therefore associated to the use of specific adapted filters.

Very often filter frequency responses are represented in the log–log scale and in this description show almost affine features in some parts. We shall use this to describe and design new filters.

2) Filter description: the log–log scale: To use the log-log scale, we define the logarithm of the filter frequency response $L(\omega) = \ln(H(\omega))$ and suppose it is piecewise affine on the intervals $[\ln(\omega_{k-1}), \ln(\omega_k)]$:

$$L(\omega) = \lambda_k^0 + \lambda_k^1 \ln(\omega)$$

where

$$\lambda_k^1 = \frac{\ln(H_k) - \ln(H_{k-1})}{\ln(\omega_k) - \ln(\omega_{k-1})} \text{ and } \lambda_k^0 = \ln(H_k) - \lambda_k^1 \ln(\omega_k).$$

Coming back to the original frequency space, for all $\omega \in [\omega_{k-1}, \omega_k]$, we have

$$\begin{aligned} H(\omega) &= \exp(L(\omega)) = \exp(\lambda_k^0 + \lambda_k^1 \ln(\omega)) = \exp(\lambda_k^0)\omega^{\lambda_k^1} \\ &= H_k \omega_k^{-\lambda_k^1} \omega^{\lambda_k^1} = H_k \left(\frac{\omega}{\omega_k}\right)^{\lambda_k^1}. \end{aligned} \tag{3}$$

Here λ_k^0 and λ_k^1 are supposed to be real numbers. The subsequent filter will therefore be necessarily be non-causal. Two more conditions on the coefficients have to be fulfilled to make our approach tractable:

- $\lambda_1^1 = 0$. In the log scale, the zero frequency is pushed toward $-\infty$, the first interval $[-\infty, \ln(\omega_1)]$ is therefore infinite, and a constant is the only possible linear interpolation on such an interval.
- $-\lambda_k^1 \in \mathbb{N}$. The reason for this will be clear in the following computation, in order to be able to give an explicit formulation of the integrals.

The interpolation of any pre-existing filter is therefore not possible as in [10] and we therefore have to build specific filters to apply our approach. The scheme of such a filter with two samples is given in Figure 1.

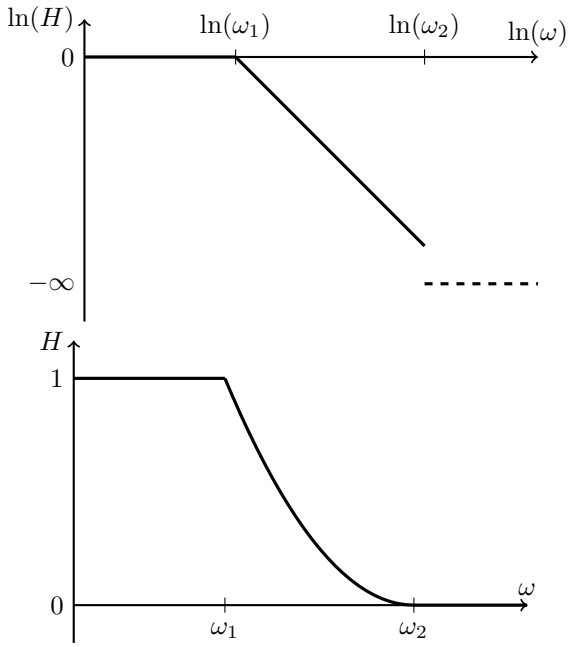

Figure 1. Example of a simple two-sample affine filter in the log-log scale (above) and the usual frequency domain (below).

The filter can be compared for example with a Butterworth filter, and this will be done in the numerical examples. We show an example on Fig. 2. In this example we have used a 10-order Butterworth filter with the same cut-off frequency. On the right plot we see that there are differences at the vicinity of the cut-off frequency, since our approach allows to exactly cut, while it is impossible to design with usual filter design methods. On the left we see the possibility to use different slopes for the log-log filter and define a second frequency beyond which the signal is filtered with an order 15. The log-log filter is described with 3 coefficients, while the 10-order Butterworth filter uses 10 coefficients.

III. The log–log scale filtering algorithm: main result

A. Elementary impulse response

In Eq. (2), the impulse response is defined as an integral over $\omega \in \mathbb{R}$. We can split this integral into K elementary impulse responses: $h(t) = \sum_{k=1}^K h_k(t)$ where (recall the filter

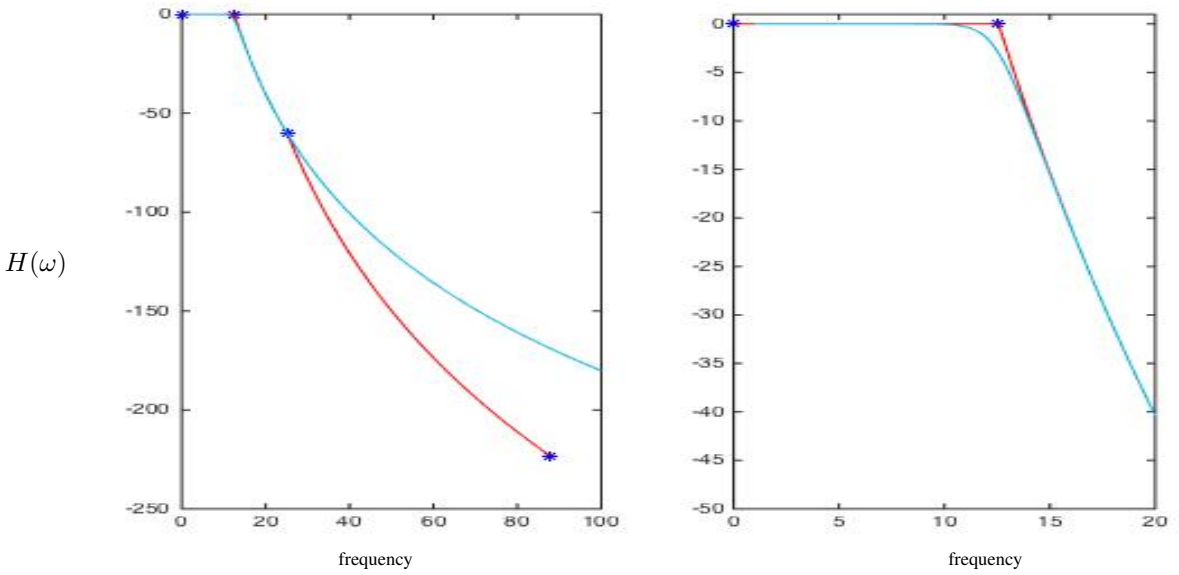

Figure 2. Bode diagramme for the log-log (red) and its "equivalent" Butterworth filter (blue). General view (left), detail (right).

is symmetric)

$$h_k(t) = \frac{1}{2\pi}\left\{ \int_{\omega_{k-1}}^{\omega_k} H(\omega)e^{i\omega t}d\omega + \int_{-\omega_k}^{-\omega_{k-1}} H^*(-\omega)e^{i\omega t}\ d\omega \right\}.$$

Using the explicit formula for $H(\omega)$ on each subinterval given by Eq. (3), we easily obtain

$$h_k(t) = \frac{1}{2\pi}\int_{\omega_{k-1}}^{\omega_k} \left(\frac{\omega}{\omega_k}\right)^{\lambda_k^1} (H_k e^{i\omega t} + H_k^* e^{-i\omega t})\ d\omega.$$

Since we have supposed until now that H_k are real, we simply have

$$h_k(t) = \alpha_k \int_{\omega_{k-1}}^{\omega_k} \omega^{\lambda_k^1}\cos(\omega t)\ d\omega \text{ where } \alpha_k = \frac{H_k}{\pi\omega_k^{\lambda_k^1}}.$$

Since we compute integrals, we notice that $h_k(t) = \alpha_k\left(g_k(t,\omega_k) - g_k(t,\omega_{k-1})\right)$, where

$$g_k(t,\Omega) = \int^{\Omega} \omega^{\lambda_k^1}\cos(\omega t)\ d\omega.$$

B. Elementary contributions

The output signal $x(t)$ defined by (1) is approximated in an affine way by

$$\begin{aligned}\bar{x}(t) &= \sum_{n=1}^{N}\int_{t_{n-1}}^{t_n} h(t-\tau)\bar{s}(\tau)d\tau \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\alpha_k(a_n x_{nk}^0(t) + b_n x_{nk}^1(t)),\end{aligned}$$

where

$$x_{nk}^0(t) = \int_{t-t_n}^{t-t_{n-1}} h_k(\tau)\ d\tau$$

and

$$x_{nk}^1(t) = \int_{t-t_n}^{t-t_{n-1}} h_k(\tau)(t-\tau)\ d\tau.$$

All the elementary contributions have the same form, we therefore define the following difference operator applied to functions $f:\mathbb{R}^2\to\mathbb{R}$

$$\begin{aligned}\Delta_{n,k}(t)f \equiv& f(t-t_n,\omega_k) - f(t-t_n,\omega_{k-1}) \\ &- f(t-t_{n-1},\omega_k) + f(t-t_{n-1},\omega_{k-1}).\end{aligned} \quad (4)$$

We will say that two functions f and f' are equivalent, and denote $f \bowtie f'$

$$f \bowtie f' \qquad \Longleftrightarrow \qquad \Delta_{n,k}(t)f = \Delta_{n,k}(t)f'.$$

Using this notation we obviously have

$$\begin{aligned}x_{nk}^0(t) &= \Delta_{n,k}(t)f_{-\lambda_k^1}, \\ x_{nk}^1(t) &= t\ x_{nk}^0(t) - \Delta_{n,k}(t)\mathring{f}_{-\lambda_k^1},\end{aligned}$$

where for any integer m (recall $-\lambda_k^1\in\mathbb{N}$) we denote

$$\begin{aligned}f_m(T,\Omega) &\equiv \int_T\int^{\Omega}\frac{\cos(\omega\tau)}{\omega^m}d\omega d\tau, \\ \mathring{f}_m(T,\Omega) &\equiv \int_T\int^{\Omega}\frac{\cos(\omega\tau)\tau}{\omega^m}d\omega\ d\tau.\end{aligned}$$

C. Main result

Theorem 1: The approximated output signal can be reconstructed with

$$\bar{x}(t) = \sum_{n=1}^{N}\sum_{k=1}^{K}\alpha_k\Big(a_n x_{nk}^0(t) + b_n x_{nk}^1(t)\Big),$$

where

$$x_{nk}^0(t) = \Delta_{n,k}(t)f_{-\lambda_k^1}$$

and

$$x_{nk}^1(t) = t\ x_{nk}^0(t) - \Delta_{n,k}(t)\mathring{f}_{-\lambda_k^1}.$$

In this formula, the coefficients a_n and b_n stem from affine approximation of the signal and the coefficients α_k and λ_k^1 from the description of the frequency response. The transform $\Delta_{n,k}(t)$ defined by (4) also depend on the signal and filter description *via* the values of t_n, t_{n-1}, ω_k and ω_{k-1}. For integer values of $m = -\lambda_k^1$ the functions

$$F_m(T,\Omega) \equiv m!\Omega^m f_m(T,\Omega),$$
$$\mathring{F}_m(T,\Omega) \equiv \frac{(m+1)!}{m}\Omega^{m+1}\mathring{f}_m(T,\Omega)$$

only depend on the product $X = \Omega T$. The functions $F_m(T,\Omega)$ and $\mathring{F}_m(T,\Omega)$ can be reconstructed as the real part of two functions

$$\begin{aligned} F_m^{\mathrm{c}}(T,\Omega) =& (P_m(X) - iQ_m(X))e^{iX} \\ &+ (iX)^m(-\operatorname{Si}(X) + i\operatorname{Ci}(X)), \\ \mathring{F}_m^{\mathrm{c}}(T,\Omega) =& (\mathring{P}_m(X) - i\mathring{Q}_m(X))e^{iX} \\ &+ (iX)^m(-\operatorname{Si}(X) + i\operatorname{Ci}(X)), \end{aligned}$$

where P_m, Q_m, $\mathring{P}_m$, $\mathring{Q}_m$ are polynomials which can be computed via induction on the index m:

$$\begin{aligned} P_m(x) =& x(m-2)! - x^2P_{(m-2)}(x), \\ & P_{(0)}(x) = 0,\ P_{(1)}(x) = 0, \\ Q_m(x) =& (m-1)! - x^2Q_{(m-2)}(x), \\ & Q_{(0)}(x) = 0,\ Q_{(1)}(x) = 1, \\ \mathring{P}_m(x) =& (m-1)! + xP_m(x), \\ & \mathring{P}_{(0)}(x) = 1,\ \mathring{P}_{(1)}(x) = 1, \\ \mathring{Q}_m(x) =& xQ_m(x), \\ & \mathring{Q}_{(0)}(x) = 0,\ \mathring{Q}_{(1)}(x) = x. \end{aligned}$$

This theorem yields an efficient algorithm to compute the approximated output signal $\bar{x}(t)$. See the Appendix for details on the special functions Si and Ci. The next section is devoted to the proof of Theorem 1.

IV. Derivation of the algorithm

To avoid computation duplication we set

$$f_m^{\mathrm{c}}(t,\Omega) = \int_T \int^\Omega \omega^m e^{i\omega\tau}\ d\omega\ d\tau,$$
$$\mathring{f}_m^{\mathrm{c}}(t,\Omega) = \int_T \int^\Omega \omega^m e^{i\omega\tau}\tau\ d\omega\ d\tau$$

and we will be able to recover f_m^α and f_m^β ($\mathring{f}_m^\alpha$ and $\mathring{f}_m^\beta$) taking the real and imaginary part respectively. For $m \geq 1$, we first compute

$$\begin{aligned} f_m^{\mathrm{c}}(T,\Omega) &= \int^\Omega \int_T \frac{e^{i\omega t}}{\omega^m}\ dt\ d\omega = i\int^\Omega \frac{e^{i\omega T}}{\omega^{m+1}}\ d\omega \\ &= -\frac{T}{m}\int^\Omega \frac{e^{i\omega T}}{\omega^m}\ d\omega - \frac{i}{m}\frac{e^{i\Omega T}}{\Omega^m} \\ &= i\frac{T}{m}f_{m-1}^{\mathrm{c}}(T,\Omega) - \frac{i}{m}\frac{e^{i\Omega T}}{\Omega^m}, \end{aligned}$$

or in other words

$$F_m^{\mathrm{c}}(T,\Omega) = i\Omega T F_{m-1}^{\mathrm{c}}(T,\Omega) - i(m-1)!e^{i\Omega T}. \tag{5}$$

The Ansatz we want to prove reads

$$\begin{aligned} F_m^{\mathrm{c}}(T,\Omega) =& (P_m(X) - iQ_m(X))e^{iX} \\ &+ (iX)^m(-\operatorname{Si}(X) + i\operatorname{Ci}(X)), \end{aligned} \tag{6}$$

Plugging (6) in (5), we easily obtain the recurrence relations Therefore we have

$$P_m(x) = xQ_{(m-1)}(x) \text{ and } Q_m = (m-1)! - xP_{(m-1)}(x)$$

which we can also write

$$\begin{aligned} P_m(x) &= x(m-2)! - x^2P_{(m-2)}(x), \\ Q_m(x) &= (m-1)! - x^2Q_{(m-2)}(x). \end{aligned}$$

The case $m = 0$ is of course not covered by these computations and needed to initialize the recurrence:

$$\begin{aligned} F_0^{\mathrm{c}}(T,\Omega) = f_0^{\mathrm{c}}(T,\Omega) &= \int^\Omega \frac{e^{i\omega T}}{\omega}\ d\omega \\ &= -\operatorname{Si}(\Omega T) - i\operatorname{Cin}(\Omega T), \end{aligned}$$

$$P_0(x) = 0,\ Q_0(x) = 0.$$

We proceed the same way for $\mathring{f}_m^{\mathrm{c}}$.

$$\begin{aligned} \mathring{f}_m^{\mathrm{c}}(T,\Omega) &= \int^\Omega \int_T \frac{e^{i\omega\tau}}{\omega^m}\tau\ d\tau\ d\omega \\ &= i\int^\Omega \int_T \frac{e^{i\omega\tau}}{\omega^{m+1}}\ d\tau\ d\omega + iT\int^\Omega \frac{e^{i\omega T}}{\omega^{m+1}}\ d\omega \\ &= if_{m+1}^{\mathrm{c}}(T,\Omega) + Tf_m^{\mathrm{c}}(T,\Omega), \end{aligned}$$

or in other words (for $m \geq 1$)

$$\mathring{F}_m^{\mathrm{c}}(T,\Omega) = \frac{i}{m}F_{m+1}^{\mathrm{c}}(T,\Omega) + \frac{m+1}{m}(\Omega T)F_m^{\mathrm{c}}(T,\Omega).$$

Thanks to (5)

$$\mathring{F}_m^{\mathrm{c}}(T,\Omega) = (\Omega T)F_m^{\mathrm{c}}(T,\Omega) + (m-1)!e^{i\Omega T}.$$

Using (6), we immediately obtain that

$$\begin{aligned} \mathring{F}_m^{\mathrm{c}}(T,\Omega) =& (\mathring{P}_m(X) - i\mathring{Q}_m(X))e^{iX} \\ &+ (iX)^m(-\operatorname{Si}(X) + i\operatorname{Ci}(X)), \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathring{P}_m(x) &= (m-1)! + xP_m(x), \\ \mathring{Q}_m(x) &= xQ_m(x). \end{aligned}$$

For $m = 0$, we compute

$$\begin{aligned} \mathring{F}_0^{\mathrm{c}}(T,\Omega) = \Omega\mathring{f}_0^{\mathrm{c}}(T,\Omega) &= i\Omega f_1^{\mathrm{c}}(T,\Omega) + (\Omega T)f_0^{\mathrm{c}}(T,\Omega) \\ &= -ie^{i\Omega T}. \end{aligned}$$

	$\omega_1 = \omega_c$	λ_1^1	$\omega_2 = \omega_m$	λ_2^1
Filter 1	4π	0	30π	-5
Filter 2	4π	0	30π	-10

TABLE I. FILTER COEFFICIENTS USED FOR EXPERIMENTS.

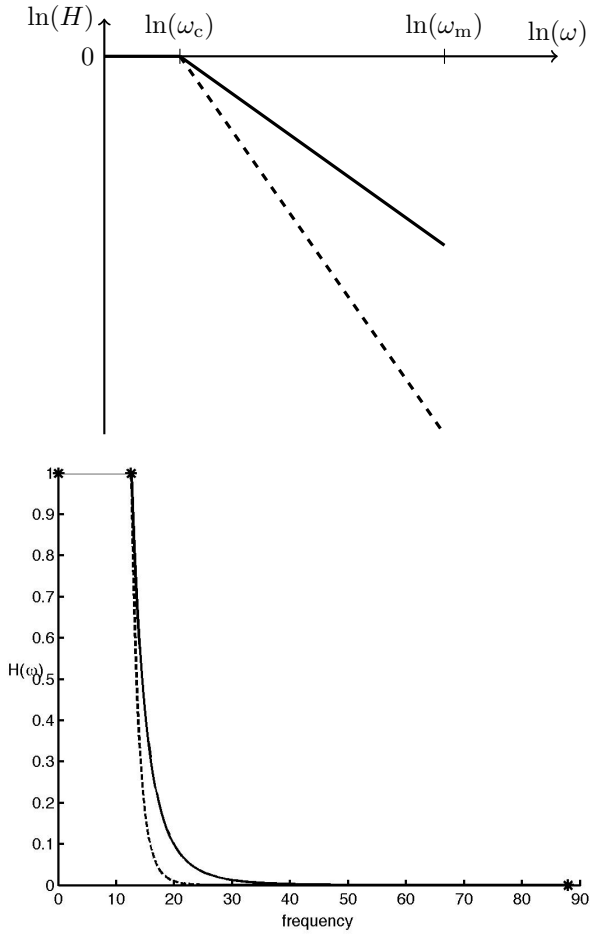

Figure 3. The two filters in the frequency domain, in the log-scale (top) and the linear scale (down). Filter 1 is plotted in solid line and Filter 2 in dashed line.

V. EXPERIMENTS

A. A first example: frequency-sweep input signal

We try two log-log affine filters with only two coefficients (couples (ω_k, H_k) with the same cut-off frequency ω_c and second frequency interval $[\omega_c, \omega_m]$, which implies that $H(\omega) = 0$ for $\omega > \omega_m$. The two filters only differ in the slope on interval $[\omega_c, \omega_m]$. The values are gathered in Table I.

The positive part of these filters are plotted on Figure 3, where the first filter is plotted with a solid line and the second filter with a dashed line. The second filter is steepest and would correspond to a higher order filter if classical filters are considered.

We test the efficiency of the above designed algorithm to filter a frequency-sweep input signal:

$$s(t) = 0.9\ (\cos(2\pi t^2) + 1).$$

The input signal is plotted with a solid line in Figure 4.

TABLE II. FILTER COEFFICIENTS.

k	1	2	3
ω_k	ω_c	$2\omega_c$	$7\omega_c$
λ_k^1	0	-5	-10

The first step of the algorithm consist in producing non-uniform samples out of this input signal. We a 3-bit asynchronous ADC [11] with $[0; 1.8]$ range, that performs cross-level sampling. For our example, this leads to 88 samples, which are plotted with stars on Figure 4.

The input signal is an academic test case that has been chosen to see how the filter treats the different frequencies. This type of signal is not typical of the signals that can benefit at most of our approach which is dedicated to sporadic signals that lead to a very few number of non-uniform samples.

The result of the filtering process is also displayed in Figure 4. The output signal $x_1(t)$ using Filter 1 is plotted with a dashed line. The dotted plot corresponds to $x_2(t)$ obtained using Filter 2. We can see that the signal is correctly filtered. An important issue is the fact that, although their theoretical order are different, the complexity of both filters are about the same since they both use two samples. The filters have been implemented in the MATLAB toolbox SPASS [12]. In this framework, the processing time for both filters is exactly the same.

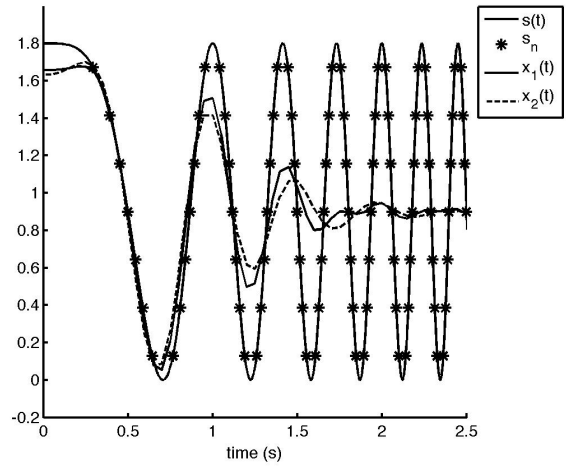

Figure 4. Input and filtered signals.

B. Filtering a real signal: ECG example

In order to easily retrieve the heart frequency in a heart rate monitor, it is suitable to first filter the erratic high frequencies. Therefore we test our algorithm on ECG signals. The result plotted in Figure 5 is obtained using a 2-coefficient filter with cut-off frequency $\omega_c = 100$ Hz.

The 2-coefficient filter in the log-scale yields similar results to uniform FIR filter techniques with a larger number of coefficients. For the comparison we used a 5-order Butterworth filter.

The second column in Table III displays the number or samples obtained through the level-crossing scheme when levels are equally spaced. In this context, the expected number

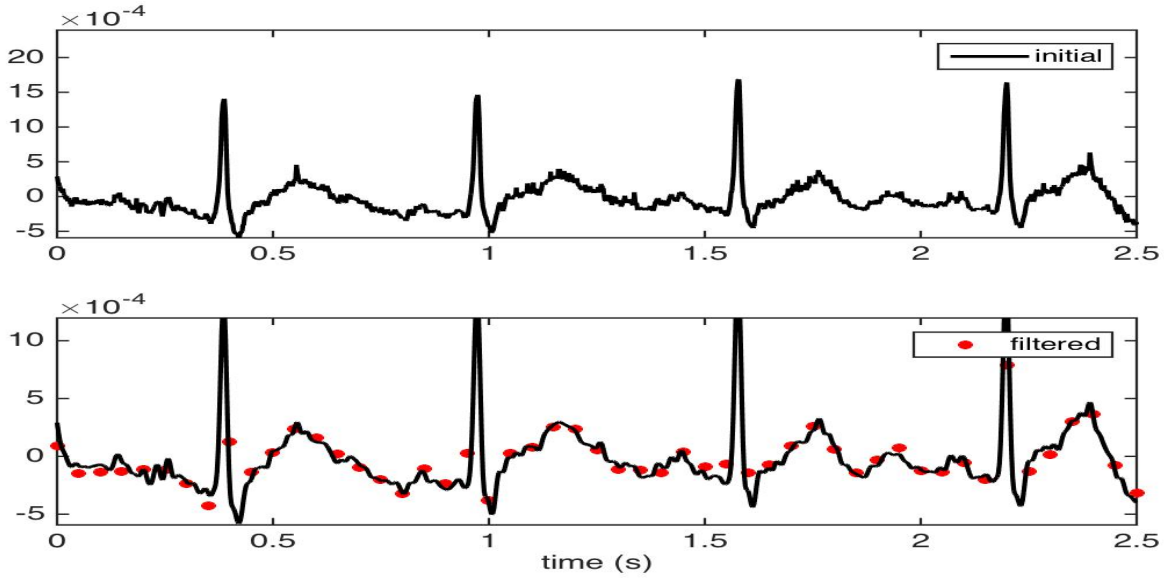

Figure 5. Input and filtered signals computed at regular times. The 16 level log-log filtered signal is plotted above the corresponding Butterworth filtered signal.

TABLE III. ERROR BETWEEN BUTTERWORTH AND LOG-LOG FILTERS.

nb levels	nb samples	ℓ^∞ error	ℓ^1 error
4	584	0.2677	0.0901
8	1377	0.2727	0.0436
16	2414	0.2645	0.0276
32	5081	0.2652	0.0219
64	10029	0.2587	0.0222

of samples has been studied in [13] with respect to the Hölder regularity of the input signal. Here the number of samples is more or less linear in the number of levels.

The two next columns give the norm of the error between the "equivalent" Butterworth filtered signal and the log-log filtered signal. The norms are normalized with respect to the amplitude of the input signal and the duration of the signal. As seen in Section V-A, we do not expect the results to match exactly, for two main reasons. First the filters are not exactly the same: 1) the Butterworth filter filters the signal causally while our is non causal; 2) the filters are quite different in the vicinity of the cut-off frequency; 3) the data are not the same since the level-crossing procedure is more or less a first filtering procedure, that also clips the signal. This last feature explains in particular that the ℓ^∞ error does not really depends on the number of levels. The ℓ^1 error shows that it is nor really necessary to use a lot of levels to obtain a fair result. For the application targeted by non-uniform sampling, a 10% error is very often acceptable. For this application, 8 levels have been chosen with an evenly-spaced positioning of the levels.

One characteristics of our filter is that the times at which the filtered result is computed can be any times. In particular, they are not linked with the input times. We can space them regularly for easier subsequent treatment even if the input signal is non uniform. We can also place then irregularly in a properly, application-targeted way. Here better results can be using the non-uniform input signal times. The reason is that we are then sure to capture the peaks. This is displayed on Fig. 6.

VI. CONCLUSION

We have designed a new class of filters based on a non-uniform, log-log affine description of the frequency response in the frequency domain. This allows to reach slopes that would correspond to high order schemes but with a low number of filter samples. Each processing step being rather complicated these filters are dedicated to specific signals, such as sporadic signals or signals mostly smoothly varying in time, that are also sampled non-uniformly in the time domain. We only describe here non-causal filters. Causal filters will be the subject of a forthcoming paper.

ACKNOWLEDGMENT

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-61 LABX-0025-01).

REFERENCES

[1] J. W. Mark and T. D. Todd, "A nonuniform sampling approach to data compression," IEEE Transactions on Communications, vol. 29, no. 1, Jan. 1981, pp. 24–32.

[2] F. Marvasti, Nonuniform Sampling. Theory and Practice, ser. Information Technology: Transmission, Processing and Storage. Kluwer Academic Publishers, 2001.

[3] K. Guan and A. C. Singer, "Opportunistic sampling by level-crossing," in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007. Honolulu, Hawai'i, USA: IEEE, Apr. 2007, pp. 1513–1516.

[4] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, "A level-crossing sampling scheme for A/D conversion," IEEE Transactions on Circuits and Systems II, vol. 43, no. 4, Apr. 1996, pp. 335–339.

[5] F. J. Beutler, "Error-free recovery of signals from irregularly spaced samples," SIAM Review, vol. 8, no. 3, Jul. 1966, pp. 328–335.

[6] I. Bilinskis, Digital alias free signal processing. John Wiley and Sons, 2007.

[7] D. Poulton and J. Oksman, "Digital filters for non uniformly sampled signals," in Nordic Signal Processing Symposium (NORSIG 2000), Vildmarkshotellet Kolmarden, Sweden, Jun. 2000, pp. 421–424.

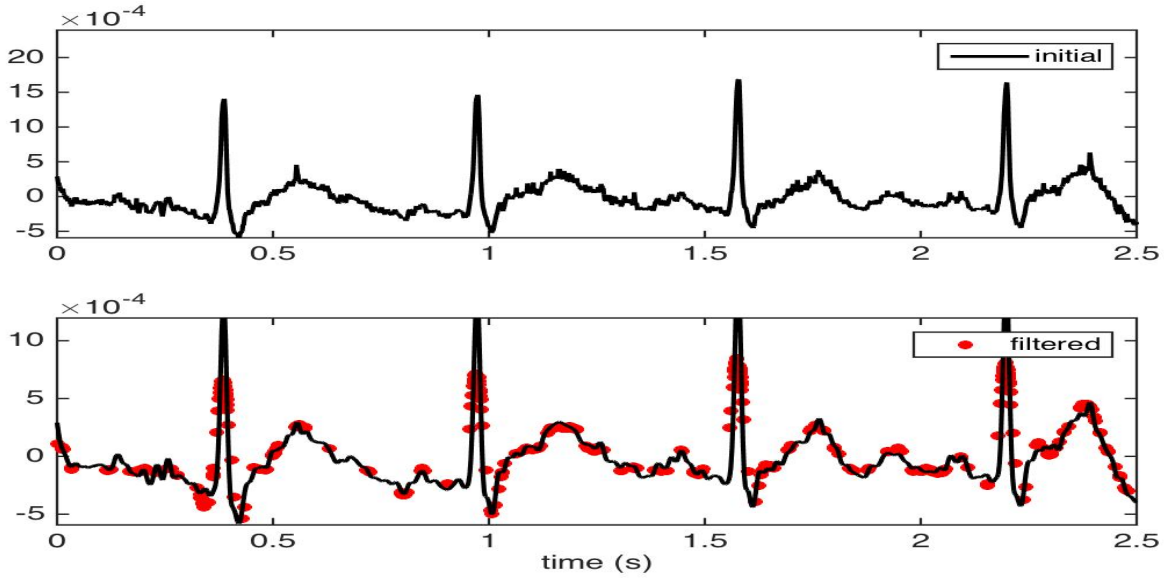

Figure 6. Input and filtered signals computed at sampled input times. The 16 level log-log filtered signal is plotted above the corresponding Butterworth filtered signal.

[8] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, "Asynchronous FIR filters: Towards a new digital processing chain," in 10th International Symposium on Asynchronous Circuits and Systems (Async'04). Hersonisos, Crete: IEEE, Apr. 2004, pp. 198–206.

[9] L. Fesquet and B. Bidégaray-Fesquet, "IIR digital filtering of non-uniformly sampled signals via state representation," Signal Processing, vol. 90, no. 10, Oct. 2010, pp. 2811–2821.

[10] B. Bidégaray-Fesquet and L. Fesquet, "Non-uniform filter interpolation in the frequency domain," Sampling Theory in Signal and Image Processing, vol. 10, no. 1–2, 2011, pp. 17–35.

[11] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "Asynchronous level crossing analog to digital converters," Measurement, vol. 37, no. 4, Jun. 2005, pp. 296–309.

[12] B. Bidégaray-Fesquet and L. Fesquet, "SPASS 2.0: Signal Processing for ASynchronous Systems," Software, May 2010.

[13] B. Bidégaray-Fesquet and M. Clausel, "Data driven sampling of oscillating signals," Sampling Theory in Signal and Image Processing, vol. 13, no. 2, 2014, pp. 175–187.

APPENDIX

We give here general formulae for trigonometric integral functions which are intensively used in this paper, paying particularly attention to define them for all $x \in \mathbb{R}$ and not only $x > 0$ as usual in the literature.
The exponential integral is defined for all $x \in \mathbb{R}^*$ by

$$\mathrm{Ei}(ix) = -\int_{|x|}^{+\infty} e^{i\,\mathrm{sgn}(x)y}\frac{dy}{y} + i\,\mathrm{sgn}(x)\frac{\pi}{2}.$$

In particular we notice that $\mathrm{Ei}(-ix) = \mathrm{Ei}(ix)^*$.
The sine integral is defined for all $x \in \mathbb{R}$ by

$$\mathrm{Si}(x) = \int_0^x \sin(y)\frac{dy}{y}.$$

This is clearly an odd function. It can be expressed for all $x \in \mathbb{R}^*$ in terms of the exponential integral:

$$\mathrm{Si}(x) = \frac{1}{2i}(\mathrm{Ei}(ix) - \mathrm{Ei}(-ix)) + \mathrm{sgn}(x)\frac{\pi}{2}.$$

The cosine integral is defined for all $x \in \mathbb{R}^*$ by

$$\mathrm{Ci}(x) = -\int_{|x|}^{\infty} \cos(y)\frac{dy}{y},$$

which is an even function which can also be expressed in terms of the exponential integral:

$$\mathrm{Ci}(x) = \frac{1}{2}(\mathrm{Ei}(ix) + \mathrm{Ei}(-ix)).$$

We will use in fact need the integral of the cosine cardinal. We classically denote for x positive

$$\mathrm{Cin}(x) = \int_0^{|x|}(1-\cos(y))\frac{dy}{y} = -\,\mathrm{Ci}(x) + \gamma + \ln|x|,$$

where γ is the Euler constant:

$$\gamma = \lim_{n\to\infty}\left(\sum_{k=1}^{n}\frac{1}{k} - \ln n\right).$$

Clearly Cin is an even function with $\mathrm{Cin}(0) = 0$. The exponential integral can be in turn expressed using the sine and cosine integral functions, namely

$$\mathrm{Ei}(ix) = \mathrm{Ci}(x) + i\,\mathrm{Si}(x) - \mathrm{sgn}(x)i\frac{\pi}{2}.$$