



**HAL**  
open science

# Infinite Impulse Response Filters for Nonuniform Data

Brigitte Bidégaray-Fesquet, Laurent Fesquet

► **To cite this version:**

Brigitte Bidégaray-Fesquet, Laurent Fesquet. Infinite Impulse Response Filters for Nonuniform Data. 3rd International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2018), May 2018, Nice, France. pp.67-72. hal-01802340

**HAL Id: hal-01802340**

**<https://hal.univ-grenoble-alpes.fr/hal-01802340>**

Submitted on 9 Jun 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Infinite Impulse Response Filters for Nonuniform Data

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

**Abstract**—Capturing and storing samples only when needed is a way to ensure a drastic reduction of the data to be processed, which is a crucial issue in many applications, such as autonomous and communicating smart devices (Internet of Things). This leads to nonuniform data for which specific processing chains have to be designed. In this paper, we discuss the issues to be addressed to generalize Infinite Impulse Response filters to the nonuniform case. We illustrate the performance of the constructed filters on an electrocardiogram signal, for two ways to obtain the nonuniform samples, and select linear decimation and the bilinear scheme as a good combination for this application.

**Keywords**—Nonuniform sampling; Infinite Impulse Response Filters; Asynchronous and Event-Driven Systems; Electrocardiogram.

## I. INTRODUCTION

For many applications, a way to drastically reduce the number of data in order to make their processing tractable with light platforms such as autonomous and communicating smart devices (often named Internet of Things) is to only capture and store samples when needed. For instance, analog-to-digital converters capture samples only when events occur or signals vary significantly. This induces data that are sampled irregularly in time, also called nonuniform data. The subject on how to sample the right data for a target application is out of the scope of this paper, but is an active research subject [1] [2].

Usual signal processing chains strongly rely on the fact that samples are taken regularly in time, or are the output of decimated regular data. When addressing genuine nonuniform data, we need to rethink the whole processing chain, in particular filtering devices.

Two main directions have been explored for constructing nonuniform filters: 1. the use of existing filters, originally designed for uniform signal and adapted to nonuniform signal [3]–[7]; 2. the design of specific filters [8]. Here we explore the first direction, and more specifically Infinite Impulse Response (IIR) filters. The case of finite impulse filters is somewhat simpler since the coefficients of the filter in the time domain can be directly used and there is only the need to interpolate correctly both the signal and the filter [3]. As will be shown in this paper, IIR filtering has to stem from a continuous representation of the filters in the Laplace domain, then in the time domain, eventually discretized at the sampling times. Other approaches in processing nonuniform sampled signals may also be found, e.g., in [9] or [10].

The design flow presented here has already been studied in [11] and its performances illustrated on a toy signal, namely a superposition of two sine signals. In particular, stability proofs can be found there. In this paper, we are more precise on the

strategies to implement IIR filters on nonuniform data and give a more realistic numerical illustration.

The outline is as follows. In Section II, we describe the filter and the signal representation, and define most of the useful notations for the sequel. We particularly justify the choice of the state equation representation. In Section III, we show various ways to discretize the state equation and recall existing results on stability issues. In Section IV, we explore the filtering of an electrocardiogram (ECG) signal, using two types of sampling, level-crossing and linear decimation.

## II. SIGNAL AND FILTER REPRESENTATIONS

In the usual uniform world, an IIR filter is often represented thanks to a difference equation that links the new output sample to previous samples of both the input and output signals. The coefficients in this difference equation are the feedforward and feedback filter coefficients. This representation of the filter strongly relies on the fact that samples are uniformly spaced. Another representation directly stems from the difference equation using the  $Z$ -transform. Both these representations are not possible to extend to a nonuniform context. A third possible representation is the state representation, which makes use of the representation of the filter in the Laplace domain. In this representation the output signal  $Y$  is simply the product of the filter transfer function  $H$  and the input signal  $I$ :  $Y(s) = H(s)I(s)$ . The filter transfer function can be written as a rational function of the Laplace variable  $s$ :

$$H(s) = \frac{\sum_{j=0}^N \alpha_j s^j}{\sum_{j=0}^N \beta_j s^j},$$

where  $N$  is the filter order.

Coming back to the time domain, this is classically cast as a system of ordinary differential equations

$$\frac{dx(t)}{dt} = Ax(t) + Bi(t), \quad (1)$$

$$y(t) = Cx(t) + Di(t), \quad (2)$$

where the state matrix  $A$ , the command vector  $B$ , the observation vector  $C$ , and the direct link coefficient  $D$  are expressed in terms of the  $\alpha_j$  and  $\beta_j$ ,  $j = 0, \dots, N$ . The advantage of this formulation is that it addresses *a priori* a continuous time, and can be considered at equally spaced time instants, in the case of classical uniform signal, or at irregularly spaced time instants in the nonuniform case, which we consider in this paper.

A uniform signal is described by a series of amplitudes, and the time delay between two samples is implicit, or even more usually renormalized to 1. In the case of nonuniform samples,

time cannot clearly remain implicit and the samples consist of amplitude–delay couples  $(a_n, dt_n)$ , where the delay  $dt_n$  is the time elapsed since the previous sample was taken (see Figure 1). This choice, compared to the choice of amplitude–time couples has two reasons. First, it has been applied to systems in which the samples are captured on the fly thanks to an asynchronous technology where no global clock synchronizes the whole system but where the synchronization is implemented with local handshakes [12]. Second, as we will see, the stability of the scheme depends on the values of the delays and not the time instants themselves.

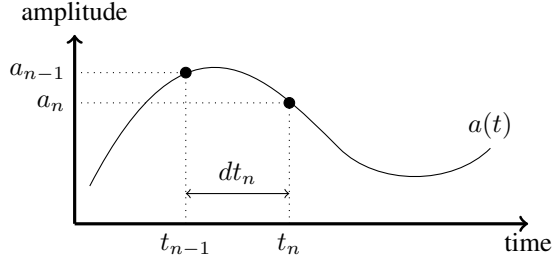


Figure 1. Non-uniform data.

In [11], the samples were collected using a level crossing algorithm. This gives extra information on the signal, but this is not an important feature for the application of a filtering scheme. In the world of nonuniform signals, we have to assume at some point that the right samples have been taken for the targeted application. In view of (2), it seems natural to define and compute the state signal  $x(t)$  and the output signal  $y(t)$  at the same time instants as the input signal. Another choice could be made, but this would necessitate the interpolation of signals and introduce interpolation errors. Hence, to an input signal  $(x_n, dt_n)$  will naturally correspond an output signal  $(y_n, dt_n)$  with the same time delays.

### III. DISCRETIZATION OF THE STATE EQUATION AND STABILITY ISSUES

Once the filter is chosen and written in the state representation, there are various ways to discretize the system. Stability is the main issue to address, as well as the ability to implement efficiently the algorithm in an autonomous device.

#### A. Principle

The stability of the IIR filter depends on two choices: the choice of the filter and the choice of the scheme to discretize it. The impact of the choice of the filter is already present in the continuous time domain. The criterion is the following: the eigenvalues of the state matrix  $A$  should have a negative real value. These eigenvalues are solution to the characteristic polynomial of  $A$  which only depends on the feedback filter coefficients:

$$\det(\lambda \text{Id} - A) = \lambda^N + \beta_{N-1} \lambda^{N-1} + \dots + \beta_1 \lambda + \beta_0,$$

where  $\text{Id}$  is the  $N \times N$  identity matrix.

In the time discretization of the state equation (1), the  $N$  eigenvalues  $\lambda$  are transformed in the complex plane into a set of  $N$  other eigenvalues  $\mu_n = \mathcal{T}_n(\lambda)$ . For a constant time delay, the new eigenvalues are the same for the whole filtering

process. In the nonuniform case, the set of eigenvalues varies through time, since it depends on  $dt_n$ . Therefore, we have to find a scheme which is uniformly stable for a range of time delays.

For the discretized scheme, the condition is that the eigenvalues lie in the unit circle of the complex plane. This can be easily understood from the integral representation of the solution to (1), namely

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bi(\tau)d\tau. \quad (3)$$

Indeed, the eigenvalues of  $A$  lie in the left half plane, if and only if the eigenvalues of  $e^{At}$  lie in the unit circle.

#### B. A simple example: the Euler scheme

The simplest example to illustrate this principle is the Euler scheme, although it will not prove to be a "good" scheme. The Euler scheme reads

$$\frac{x_n - x_{n-1}}{dt_n} = Ax_{n-1} + Bi_{n-1}, \quad (4)$$

or equivalently

$$x_n = (\text{Id} + dt_n A)x_{n-1} + Bdt_n i_{n-1}.$$

If we compare to the integral form, we see that  $e^{A dt_n}$  is simply approximated by  $(\text{Id} + dt_n A)$ , and even by  $\text{Id}$  in the integral which is calculated with a left rectangle method. Here,  $\mathcal{T}_n(\lambda) = 1 + dt_n \lambda$ .

We have two ways to discuss this. If we want to address with this scheme all possible IIR filtering methods, with all possible eigenvalues in the left half-plane. Then, we can only say that  $\mathcal{T}_n$  maps the left half-plane in an other half-plane (see Figure 2) and certainly the Euler scheme will not lead to a stable digital filter.

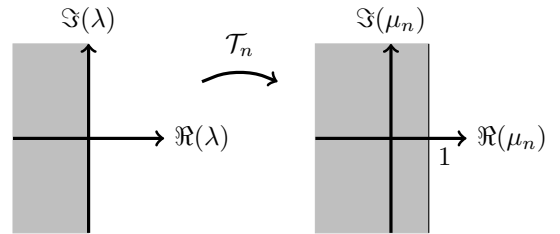


Figure 2. Action of the  $\mathcal{T}_n$  transform for the Euler scheme. Eigenvalues  $\lambda$  (left) and eigenvalues  $\mu_n$  (right).

An other point of view is to compute the inverse transform of the unit circle, which is also a circle for the Euler scheme, as shown on Figure 3. If it contains all the eigenvalues  $\lambda$  of the original filter, the discrete filter will be stable. This has to be valid for all the values of  $dt_n$ , i.e., for the maximum value of  $dt_n$  which yields the smallest inverse image of the unit circle. This leads to give a maximal bound for the time delay, that is to integrate these issues in the sampling procedure in a global signal processing chain.

#### C. Review of other schemes

The discretization of the state equation is either made on its integral form (3) or on the differential equation (1).

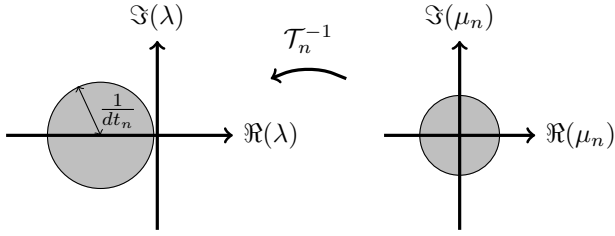


Figure 3. Backward action of the  $\mathcal{T}_n$  transform for the Euler scheme. Eigenvalues  $\lambda$  (left) and eigenvalues  $\mu_n$  (right).

1) *Discretization of the integral form:* Some of schemes are directly based on the integral formulation of the solutions. This is the case in [5], where the only approximation consists in replacing the continuous signal  $i(t)$  by a sample-hold or piecewise linear interpolation on the time interval  $[t_{n-1}, t_n]$ , and compute exactly the integral. For example, for sample-hold interpolation

$$x_n = e^{A dt_n} x_{n-1} - A^{-1}(\text{Id} - e^{A dt_n}) B i_{n-1}. \quad (5)$$

Since  $e^{A dt_n}$  is not approximated, this leads by construction to stable schemes. Another feature in this reference is the use of cascading filters, splitting the original filter into second- (or first-) order filters. This leads to a simpler calculation of  $\exp(A dt_n)$  and make possible the implementation on asynchronous architectures [3] [13].

2) *Discretization of the differential equation:* The first example of this approach can be found in [6] [7] where they use the bilinear method to approximate the time derivative in (1):

$$\frac{x_n - x_{n-1}}{dt_n} = A \frac{x_n + x_{n-1}}{2} + B \frac{i_n + i_{n-1}}{2}. \quad (6)$$

For this scheme, the eigenvalue transform is the homographic function

$$\mathcal{T}_n(\lambda) = \frac{1 + dt_n \lambda / 2}{1 - dt_n \lambda / 2},$$

which is well known to map the left half-plane in the unit circle.

Other schemes have been reviewed in [11], backward Euler, various Runge-Kutta schemes. The results are well-known results when using difference methods to discretize ordinary differential systems of equations. In particular it is possible to construct unconditionally stable implicit or semi-implicit schemes, i.e., schemes that are stable whatever the value of  $dt_n$ . This is the case of the backward Euler and bilinear methods. Explicit schemes, such as Runge-Kutta methods, can be easier to implement but they always have a stability condition, and as for the Euler scheme an upper bound has to be set on the intersample time. This has to be integrated in the processing chain or, if no control on the input data is possible, extra data has to be interpolated in very quiescent parts of the signal. Practical implementations have shown that the bound on  $dt_n$  is not a crucial point and is not also a practical technical issue for the asynchronous systems [14]. In [11], the complexities of the various schemes are also compared. This proves not to be a crucial point either, if the filters are decomposed in one- or two- order filters, to avoid the computation of matrix exponentials.

TABLE I. COMPRESSION OF THE ECG SIGNAL VIA LEVEL-CROSSING.

number of levels	4	8	16
number of samples	584	1377	2414
compression	2%	4.8%	8.5%

#### IV. FILTERING AN ECG SIGNAL

To study the stability performances of the IIR filters in [11] sine signals were chosen. These signals are always active and no gain in the number of samples can be hoped since even with a non-uniform sampling the Nyquist rate should be ensured at last in average [15] [16]. If the signal frequency-range is constant over the signal duration, no reduction of the number of samples can be obtained, and since the processing of the samples could be more complex when nonuniform, the overall performances would be lower than with classical uniform sampling and classical filtering techniques. Notice that this point is not obvious and the processing is not always more costly such as in [17].

Here we consider an ECG signal. Such a signal is interesting to address with nonuniform sampling, since it has quiescent parts and for applications, such as pacemakers or disease diagnosis, we very often want to isolate specific patterns in the signal and not process the signal as a whole. The signal shown in the next experiments has a 14.27 s duration which corresponds to about 22 cardiac cycles. The initial signal, sampled at 2000 Hz, has 28548 samples.

All the computations have been performed using the SPASS Matlab toolbox [18].

##### A. Nonuniform sampling of the input signal

Two types of nonuniform sampling are explored, a level-crossing sampling scheme and linear decimation, which can be more or less assimilated to a slope crossing scheme.

1) *Level-crossing:* A simple and widely used way of nonuniformly sampling signals is level-crossing sampling [19] [20] [10]. It consists in defining levels within the range of the input signal. These levels can be either equally spaced, for simplicity or implementations reasons [3], or on the contrary very carefully chosen in order to capture the important features of the signal for a specific application [21].

Figure 4 displays the samples obtained of our ECG sample. Here 8 equally-spaced levels have been chosen, which leads to 1377 samples, and hence 4.8% of the initial samples. To have an idea of the compression obtained with this technique, we give in Table I the number of samples and compression for 4, 8 and 16 levels.

2) *Linear decimation:* We propose here another way to decimate the initial samples, which can be performed on the fly, which is of practical interest for hardware implementations. The principle of this decimation is shown in Figure 5.

It consists in defining a tolerance on the surface of the polygon between the curve with all the initial samples and the kept samples. Let us suppose that we begin with sample  $S_1$ . If the surface of the triangle  $S_1 S_2 S_3$  is above threshold, then we keep  $S_2$  and explore the next samples taking  $S_2$  as new initial sample. If this surface is below threshold, then we consider the surface of the polygon  $S_1 S_2 S_3 S_4$ , if it is above threshold,

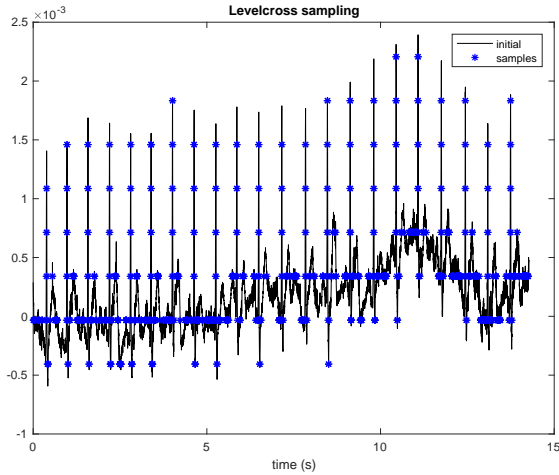


Figure 4. 8-level-crossing sampling of an ECG signal.

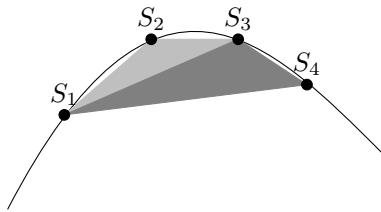


Figure 5. Principle of linear decimation.

we keep  $S_3$  and take it a new initial sample, otherwise we go on exploring  $S_5$ , etc.

Figure 6 displays the samples obtained for our ECG signal. Here we chose a tolerance equal to twice the width of the range of the signal, which leads to 916 samples, and hence 3.2% of the initial samples. Again, a few other choices for the tolerance and the associated compression are given in Table II.

With less samples than with 8-level-crossing the description of the signal seems better. This will be confirmed by the filtering results. Of course there is a drawback, although possible, the hardware implementation of this type of sampling is much harder.

### B. Filtering results

We use an order-10 Butterworth filter with a cut-off frequency at 200 Hz. We have tested the various methods described in [11] but only plot results for the backward Euler scheme

$$\frac{x_n - x_{n-1}}{dt_n} = Ax_n + Bi_n, \quad (7)$$

TABLE II. COMPRESSION OF THE ECG SIGNAL VIA LINEAR DECIMATION.

tolerance	2*range	range	range/2
number of samples	916	1548	2341
compression	3.2%	5.4%	8.2%

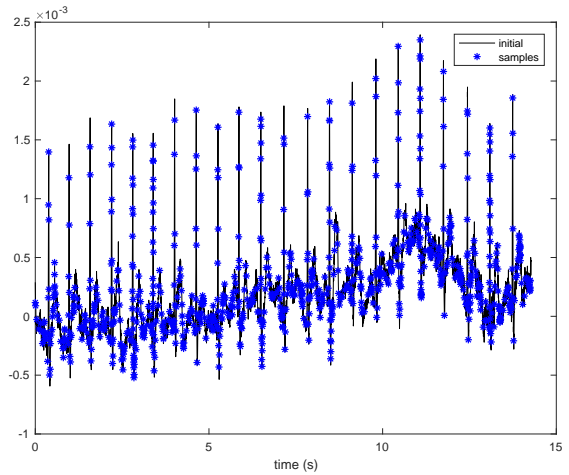


Figure 6. Linear decimation sampling of an ECG signal. range criterion, 916 vs. 284548 samples.

and the bilinear scheme (6). For this test case, both the Euler scheme (4) and the Runge-Kutta 4 scheme are unstable and do not yield any result. Other Runge-Kutta schemes, such as RK23, are not unstable but give very distorted results.

Figure 7 displays the filtering of the ECG signal after level-crossing sampling with the backward Euler discretization of the state equation (7). For the sake of clarity, we have windowed the plot to see a small sequence of the signal.

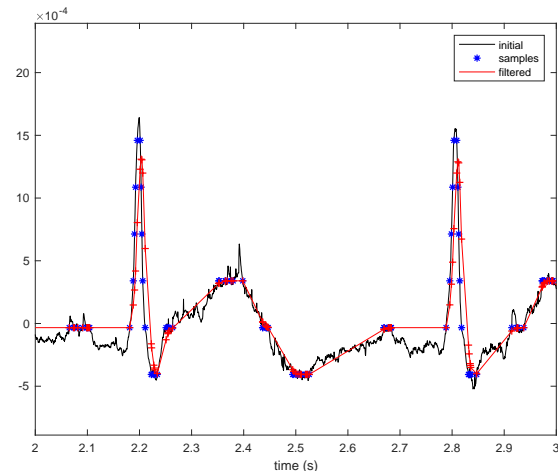


Figure 7. Backward Euler Butterworth filtering of ECG signal after level-crossing sampling.

We can notice two unwanted features: first, the filtered signal keeps the memory of the sampling levels; second, the P pattern is not captured. Indeed, physicians who read ECGs want to spot specific patterns in the ECG signal to diagnose diseases. These patterns are designated by the letters PQRST, see Figure 8. The reason why P has not been captured is not due to filtering but to sampling since no point has been set in this part of the signal. This can typically be corrected by a cleverer choice of the levels.

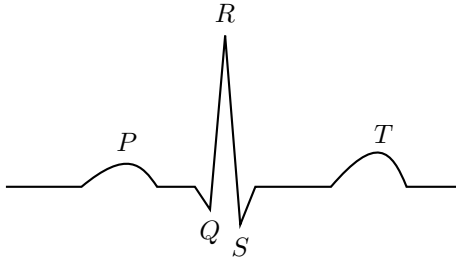


Figure 8. Patterns in a normal heart sinus rhythm.

Figure 9 displays the filtering of the ECG signal after the same level-crossing sampling but with the bilinear discretization. The previous bad features are always there because they were mainly due to sampling. They are present for all the (stable) schemes, the worst one from this point of view being the integral form (5). This is not due to the integral form itself which is exact, but to the sample-hold interpolation. You have to use piecewise linear or nearest neighbor interpolation for this application.

The bilinear discretization does nonetheless a little better than the backward Euler scheme. Indeed, the amplitude of the R pattern is much better captured.

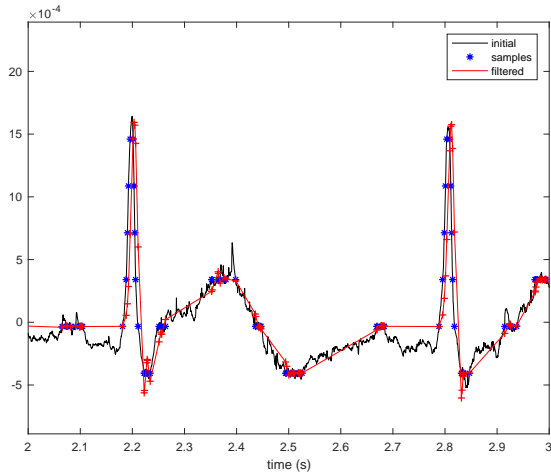


Figure 9. Bilinear Butterworth filtering of ECG signal after level-crossing sampling.

Now, we explore the simulations performed with a linear decimated sampled signal. Figures 10 and 11 yield the results for the backward Euler and bilinear discretizations of the Butterworth filter. The unwanted features of the level-crossing sampling are of course not present and the filtered result much resembles the theoretical pattern of Figure 8. The bilinear scheme is once more better than the backward Euler scheme, since it captures better the amplitude and times of the points of interest.

## V. CONCLUSION AND FUTURE WORK

We have discussed various issues which are important when having to generalize IIR filters to the nonuniform case:

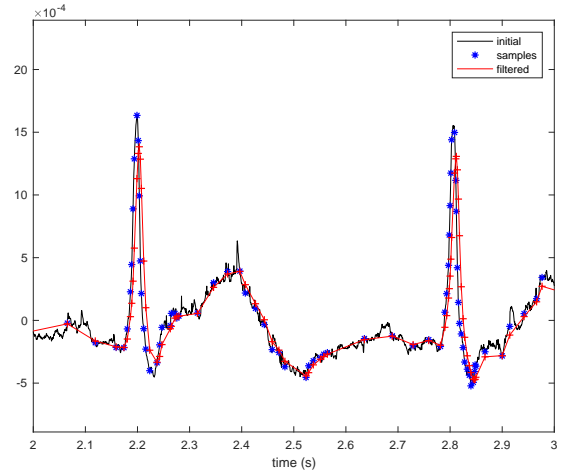


Figure 10. Backward Euler Butterworth filtering of ECG signal after linear decimation sampling.

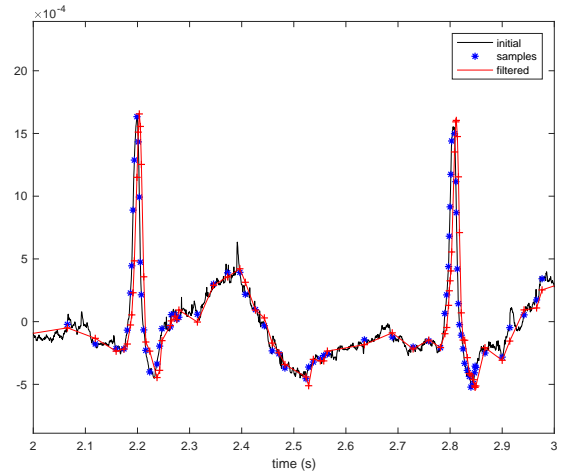


Figure 11. Bilinear Butterworth filtering of ECG signal after linear decimation sampling.

which representation of such filters is the more adapted to this generalization?

We illustrate the performance of the constructed filters on an electrocardiogram signal, for two ways to obtain the nonuniform samples. We select linear decimation and the bilinear scheme as a good combination for this application. This discretization of the equation gives stable and accurate results. We have seen on an ECG example that if the choice of the discrete filtering method is important, the way the nonuniform samples have been chosen is also a very crucial issue. The method, we call linear decimation, seems to be adapted to the ECG case, but there is clearly a lot of work to be done in this direction to reduce more drastically the number of samples and therefore the computational cost.

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] F. Marvasti, *Nonuniform Sampling. Theory and Practice*, ser. Information Technology: Transmission, Processing and Storage. Kluwer Academic Publishers, 2001.
- [2] B. Bidégaray-Fesquet and L. Fesquet, "Levels, peaks, slopes... which sampling for which purpose?" in 2nd International Conference on Event-Based Control, Communications, and Signal Processing (EBCCSP16). Krakow, Poland: IEEE, Jun. 2016, pp. 1–6.
- [3] 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.
- [4] B. Bidégaray-Fesquet and L. Fesquet, "A fully non-uniform approach to FIR filtering," in 8th International Conference on Sampling Theory and Applications (SampTa'09), L. Fesquet and B. Torrèsani, Eds., Marseille, France, May 2009, pp. 129:1–4.
- [5] L. Fontaine and J. Ragot, "Filtrage de signaux à échantillonnage irrégulier [Filtering irregularly sampled signals]," *Traitement du Signal*, vol. 18, no. 2, 2001, pp. 89–101.
- [6] 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.
- [7] —, "Filtrage des signaux à échantillonnage non uniforme [Filtering nonuniformly sampled signals]," *Traitement du Signal*, vol. 18, no. 2, 2001, pp. 81–88.
- [8] 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.
- [9] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Adaptive rate sampling and filtering based on level crossing sampling," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. Article ID 971656, 2009, p. 13 pages.
- [10] C. Vezirtzis and Y. Tsvividis, "Processing of signals using level-crossing sampling," in IEEE International Symposium on Circuits and Systems, ISCAS 2009. Taipei, Taiwan: IEEE, May 2009, pp. 2293–2296.
- [11] 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.
- [12] J. Sparso and S. Furber, Eds., *Principles of Asynchronous Circuit Design: A Systems Perspective*. Springer, 2001.
- [13] T. Beyrouthy and L. Fesquet, "An event-driven FIR filter: design and implementation," in 22nd IEEE International Symposium on Rapid System Prototyping (RSP 2011). Karlsruhe, Germany: IEEE, May 2011, pp. 59–65.
- [14] E. Yahya, L. Fesquet, Y. Ismail, and M. Renaudin, "Statistical static timing analysis of conditional asynchronous circuits using model-based simulation," in 19th International Symposium on Asynchronous Circuits and Systems (ASYNC). IEEE, 2013, pp. 67–74.
- [15] F. J. Beutler, "Sampling theorems and bases in a hilbert space," *Information and Control*, vol. 4, no. 2-3, 1961, pp. 97–117.
- [16] —, "Error-free recovery of signals from irregularly spaced samples," *SIAM Review*, vol. 8, no. 3, 1966, pp. 328–335.
- [17] G. Roa, T. Le Pelleter, A. Bonvilain, A. Chagoya, and L. Fesquet, "Designing ultra-low power systems with non-uniform sampling and event-driven logic," in 27th Symposium on Integrated Circuits and Systems Design (SBCCI 2014). Aracaju, Sergipe, Brazil: IEEE, Sep. 2014, Article No. 5:1–6.
- [18] B. Bidégaray-Fesquet and L. Fesquet, "SPASS 2.0: Signal Processing for ASynchronous Systems," *Software*, May 2010. [Online]. Available: <https://ljk.imag.fr/membres/Brigitte.Bidegaray/SPASS/> [retrieved: April, 2018]
- [19] 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.
- [20] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization," in 9th International Symposium on Asynchronous Circuits and Systems, Async'03. Vancouver, Canada: IEEE, May 2003, pp. 196–205.
- [21] T. Le Pelleter, T. Beyrouthy, R. Rolland-Girod, A. Bonvilain, and L. Fesquet, "Non-uniform sampling pattern recognition based on atomic decomposition," in 10th International Conference on Sampling Theory and Applications (SampTA 2013), Bremen, Germany, Jul. 2013, pp. 236–239.