

Research Article

Adaptive Rate Sampling and Filtering Based on Level Crossing Sampling

Saeed Mian Qaisar,¹ Laurent Fesquet (EURASIP Member),¹ and Marc Renaudin²

¹TIMA, CNRS UMR 5159, 46 avenue Felix-Viallet, 38031 Grenoble Cedex, France

²Tiempo SAS, 110 Rue Blaise Pascal, Bat Viseo-Inovallee, 38330 Montbonnot Saint Martin, France

Correspondence should be addressed to Saeed Mian Qaisar, saeed.mian-qaisar@imag.fr

Received 11 August 2008; Revised 31 December 2008; Accepted 14 April 2009

Recommended by Sven Nordholm

The recent sophistications in areas of mobile systems and sensor networks demand more and more processing resources. In order to maintain the system autonomy, energy saving is becoming one of the most difficult industrial challenges, in mobile computing. Most of efforts to achieve this goal are focused on improving the embedded systems design and the battery technology, but very few studies target to exploit the input signal time-varying nature. This paper aims to achieve power efficiency by intelligently adapting the processing activity to the input signal local characteristics. It is done by completely rethinking the processing chain, by adopting a non conventional sampling scheme and adaptive rate filtering. The proposed approach, based on the LCSS (Level Crossing Sampling Scheme) presents two filtering techniques, able to adapt their sampling rate and filter order by online analyzing the input signal variations. Indeed, the principle is to intelligently exploit the signal local characteristics—which is usually never considered—to filter only the relevant signal parts, by employing the relevant order filters. This idea leads towards a drastic gain in the computational efficiency and hence in the processing power when compared to the classical techniques.

Copyright © 2009 Saeed Mian Qaisar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

This work is part of a large project aimed to enhance the signal processing chain implemented in the mobile systems. The motivation is to reduce their size, cost, processing noise, electromagnetic emission and especially power consumption, as they are most often powered by batteries. This can be achieved by intelligently reorganizing their associated signal processing theory, and architecture. The idea is to combine event driven signal processing with asynchronous circuit design, in order to reduce the system processing activity and energy cost.

Almost all natural signals like speech, seismic, and biomedical are time varying in nature. Moreover, the man made signals like Doppler, Amplitude Shift Keying (ASK), and Frequency Shift Keying (FSK), also lay in the same category. The spectral contents of these signals vary with time, which is a direct consequence of the signal generation process [1]

The classical systems are based on the Nyquist signal processing architectures. These systems do not exploit the

signal variations. Indeed, they sample the signal at a fixed rate without taking into account the intrinsic signal nature. Moreover they are highly constrained due to the Shannon theory especially in the case of low activity sporadic signals like electrocardiogram, phonocardiogram, seismic, and so forth. It causes to capture, and to process a large number of samples without any relevant information, a useless increase of the system activity, and its power consumption.

The power efficiency can be enhanced by intelligently adapting the system processing load according to the signal local variations. In this end, a signal driven sampling scheme, which is based on “level-crossing” is employed. The Level Crossing Sampling Scheme (LCSS) [2] adapts the sampling rate by following the local characteristics of the input signal [3, 4]. Hence, it drastically reduces the activity of the post-processing chain, because it only captures the relevant information [5, 6]. In this context, LCSS Based Analog to Digital Converters (LCADCs) have been developed [7–9]. Algorithms for processing [6, 10–12], and analysis [3, 5, 13, 14] of the nonuniformly spaced out in time-sampled data, obtained with the LCSS have also been developed.

Filtering is a basic operation, almost required in every signal processing chain. Therefore, this paper focuses on the development of efficient Finite Impulse Response (FIR) filtering techniques. The idea is to pilot the system processing activity by the input signal variations. By following this idea, an efficient solution is proposed by intelligently combining the features of both nonuniform and uniform signal processing tools, which promise a drastic computational gain of the proposed techniques compared to the classical one.

Section 2 briefly reviews the nonuniform signal processing tools employed in the proposed approach. Complete functionality of the proposed filtering techniques is described in Section 3. Section 4 demonstrates the appealing features of the proposed techniques with the help of an illustrative example. The computational complexities of both proposed techniques are deduced and compared, among and to the classical case in Section 5. Section 6 discusses the processing error. In Section 7, the proposed techniques performance is evaluated for a speech signal. Section 8 finally concludes the article.

2. Nonuniform Signal Processing Tools

2.1. LCSS (Level Crossing Sampling Scheme). The LCSS belongs to the signal-dependent sampling schemes like zero-crossing sampling [15], Lebesgue sampling [16], and reference signal crossing sampling [17]. The concept of LCSS is not new and has been known at least since 1950s [18]. It is also known as an event-based sampling [19, 20]. In recent years, there have been considerable interests in the LCSS, in a broad spectrum of technology and applications. In [21–24], authors have employed it for monitoring and control systems. It has also been suggested in literature for compression [2], random processes [25], and band-limited Gaussian random processes [26].

The LCSS is a natural choice for sampling the time-varying signals. It lets the signal to dictate the sampling process [4]. The nonuniformity in the sampling process represents the signal local variations [3]. In the case of LCSS, a sample is captured only when the input analog signal $x(t)$ crosses one of the predefined thresholds. The samples are not uniformly spaced in time because they depend on $x(t)$ variations as it is clear from Figure 1.

Let a set of levels which span the analog signal amplitude range be ΔV_{in} . These levels are equally spaced by a quantum q . When $x(t)$ crosses one of these predefined levels, a sample is taken [2]. This sample is the couple (x_n, t_n) of an amplitude x_n and a time t_n . However x_n is clearly equal to one of the levels and t_n can be computed by employing

$$t_n = t_{n-1} + dt_n. \quad (1)$$

In (1), t_n is the current sampling instant, t_{n-1} is the previous one, and dt_n is the time elapsed between the current and the previous sampling instants.

2.2. LCADC (LCSS-Based Analog to Digital Converter). Classically, during an ideal A/D conversion process the sampling instants are exactly known, where as samples amplitudes are

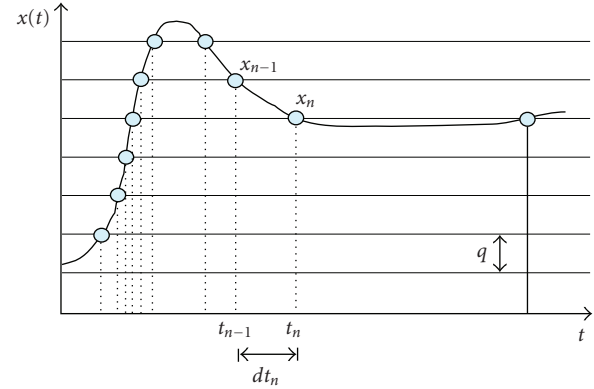


FIGURE 1: Level-crossing sampling scheme.

quantized at the ADC resolution [27], which is defined by the ADC number of bits. This error is characterized by the Signal to Noise Ratio (SNR) [27], which can be expressed by

$$\text{SNR}_{\text{dB}} = 1.76 + 6.02M. \quad (2)$$

Here, M is the ADC number of bits. It follows that the SNR of an ideal ADC depends only on M and it can be improved by 6.02 dB for each increment in M .

The A/D conversion process, which occurs in the LCADCs [7–9], is dual in nature. Ideally in this case, samples amplitudes are exactly known since they are exactly equal to one of the predefined levels, while the sampling instants are quantized at the timer resolution T_{timer} . According to [7, 8], the SNR in this case is given by

$$\text{SNR}_{\text{dB}} = 10 \log \left(\frac{3P_x}{P_{x'}} \right) - 20 \log(T_{\text{timer}}). \quad (3)$$

Here, P_x and $P_{x'}$ are the powers of $x(t)$ and of its derivative, respectively. It shows that in this case, the SNR does not depend on M any more, but on $x(t)$ characteristics and T_{timer} . An improvement of 6.02 dB in the SNR can be achieved by simply halving T_{timer} .

The choice of M is however crucial. It should be taken large enough to ensure a proper reconstruction of the signal. This problem has been addressed in [28–31]. In particular, in [31], it is shown that a band-limited signal can be ideally reconstructed from nonuniformly spaced samples if the average number of samples satisfies the Nyquist criterion. In the case of LCADCs, the average sampling frequency depends on M and the signal characteristics [7–9]. Thus, for a given application an appropriate M should be chosen in order to respect the reconstruction criterion [31].

In [7–9], authors have shown advantages of the LCADCs over the classical ones. The major advantages are the reduced activity, the power saving, the reduced electromagnetic emission, and the processing noise reduction. Inspiring from these interesting features, the Asynchronous Analog to Digital Converter (AADC) [7] is employed to digitize $x(t)$ in the studied case. The characteristics of the filtering techniques described in the sequel are highly determined by the characteristics of the nonuniformly sampled signal

produced by the AADC. We have already defined the AADC amplitude range ΔV_{in} , the number of bits M and the quantum q . They are linked by the following relation:

$$q = \frac{\Delta V_{\text{in}}}{2^M - 1}. \quad (4)$$

This quantum together with the AADC processing delay for one sample δ yields the upper limit on the input signal slope, which can be captured properly:

$$\frac{dx(t)}{dt} \leq \frac{q}{\delta}. \quad (5)$$

In order to respect the reconstruction criterion [31] and the tracking condition [7], a band pass filter with pass-band $[f_{\text{min}}; f_{\text{max}}]$ is employed at the AADC input. This together with a given M induces the AADC maximum and minimum sampling frequencies [6, 11], defined by

$$F_{s_{\text{max}}} = 2f_{\text{max}}(2^M - 1), \quad (6)$$

$$F_{s_{\text{min}}} = 2f_{\text{min}}(2^M - 1). \quad (7)$$

Here, f_{max} and f_{min} are the $x(t)$ bandwidth and fundamental frequencies, $F_{s_{\text{max}}}$ and $F_{s_{\text{min}}}$ are the AADC maximum and minimum sampling frequencies, respectively.

2.3. ASA (Activity Selection Algorithm). The nonuniformly sampled signal obtained with the AADC can be used for further nonuniform digital processing [3, 10, 13]. However in the studied case, the nonuniformity of the sampling process, which yields information on the signal local features, is employed to select only the relevant signal parts. Furthermore, the characteristics of each signal selected part are analyzed and are employed later on to adapt the proposed system parameters accordingly. This selection and local-features extraction process is named as the ASA.

For activity selection, the ASA exploits the information laying in the level-crossing sampled signal nonuniformity [5]. This selection process corresponds to an adaptive length rectangular windowing. It defines a series of selected windows within the whole signal length. The ability of activity selection is extremely important to reduce the proposed system processing activity and consequently its power consumption. Indeed, in the proposed case, no processing is performed during idle signal parts, which is one of the reasons of the achieved computational gain compared to the classical case. The ASA is defined as follow:

$$\begin{aligned} &\text{while} \left(dt_n \leq \frac{T_0}{2}, T^i \leq T_{\text{ref}} \right) \\ &\quad T^i = T^i + dt_n; \\ &\quad N^i = N^i + 1; \\ &\text{end.} \end{aligned} \quad (8)$$

Here, dt_n is clear from (1). $T_0 = 1/f_{\text{min}}$ is the fundamental period of the bandlimited signal $x(t)$, T_0 and dt_n detect

parts of the nonuniformly sampled signal with activity. If the measured time delay dt_n is greater than $T_0/2$, $x(t)$ is considered to be idle. The condition $dt_n \leq T_0/2$ is chosen to ensure the Nyquist sampling criterion for f_{min} .

T_{ref} is the reference window length. Its choice depends on the input signal characteristics and the system resources. The upper bound on T_{ref} is posed by the maximum number of samples that the system can treat at once. Whereas the lower bound on T_{ref} is posed by the condition $T_{\text{ref}} \geq T_0$, which should be respected in order to achieve a proper spectral representation [5].

T^i represents the length in seconds of the i th selected window W^i . T_{ref} poses the upper bound on T^i . N^i represents the number of nonuniform samples laying in W^i , which lies on the j th active part of the nonuniformly sampled signal. i and j both belong to the set of natural numbers \mathbb{N}^* . The j th signal activity can be longer than T_{ref} . In this case, it will be splitted into more than one selected windows.

The above-described loop repeats for each selected window, which occurs during the observation length of $x(t)$. Every time before starting the next loop, i is incremented and N^i and T^i are initialized to zero.

The maximum number of samples N_{max} , which can take place within a chosen T_{ref} can be calculated by employing

$$N_{\text{max}} = T_{\text{ref}} F_{s_{\text{max}}}. \quad (9)$$

The ASA displays interesting features, which are not available in the classical case. It only selects the active parts of the nonuniformly sampled signal. Moreover, it correlates the length of the selected window with the input signal activity, laying in it. In addition, it also provides an efficient reduction of the phenomenon of spectral leakage in the case of transient signals. The leakage reduction is achieved by avoiding the signal truncation problem with a simple and an efficient algorithm, instead of employing a smoothing (cosine) window function, which is used in the classical schemes [5]. These abilities make the ASA extremely effective in reducing the overall system processing activity, especially in the case of low activity sporadic signals [5, 6, 11, 12, 14].

3. Proposed Adaptive Rate Filtering

3.1. General Principle. Two techniques are described to filter the selected signal obtained at the ASA output. The signal processing chain common to both filtering techniques is shown in Figure 2.

The activity selection and the local features extraction are the bases of the proposed techniques. They make to achieve the adaptive rate sampling (only relevant samples to process) along with the adaptive rate filtering (only relevant operations to deliver a filtered sample). Such an achievement assures a drastic computational gain of the proposed filtering techniques compared to the classical one. The steps of realizing these ideas are detailed in the following subsections.

3.1.1. Adaptive Rate Sampling. The AADC sampling frequency is correlated to $x(t)$ local variations [6, 11, 12, 14]. It

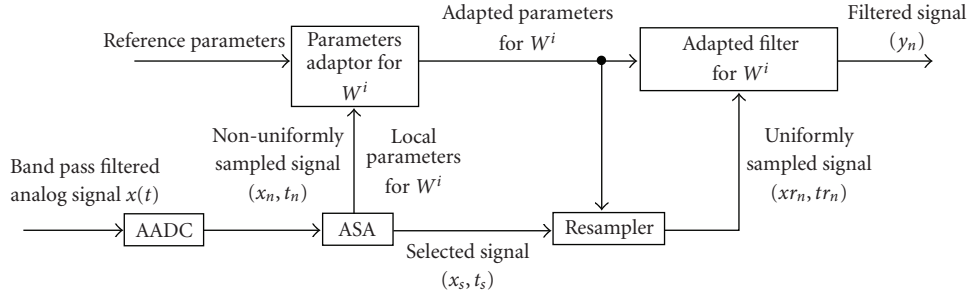


FIGURE 2: Signal processing chain common to both filtering techniques.

follows that the local sampling frequency Fs^i can be specific for W^i . According to [5] Fs^i can be calculated by employing

$$Fs^i = \frac{N^i}{T^i}. \quad (10)$$

The upper and the lower bounds on Fs^i are posed by Fs_{\max} and Fs_{\min} , respectively. In order to perform a classical filtering algorithm, the selected signal laying in W^i is uniformly resampled before proceeding to the filtering stage (cf. Figure 2). Characteristics of the selected signal part laying in W^i are employed to choose its resampling frequency Frs^i . Once the resampling is done, there are Nr^i samples in W^i . Choice of Frs^i is crucial and this procedure is detailed in the following subsection.

3.1.2. Adaptive Rate Filtering. It is known that for fixed design parameters (cut-off frequency, transition-band width, pass-band, and stop-band ripples) the FIR filter order varies as a function of the operational sampling frequency. For high sampling frequency, the order is high and vice versa. In the classical case, the sampling frequency and filter order both remains unique regardless of the input signal variations, so they have to be chosen for the worst case. This time invariant nature of the classical filtering causes a useless increase of the computational load. This drawback has been resolved up to a certain extent by employing the multirate filtering techniques [32–34].

The proposed filtering techniques of this paper are the intelligent alternatives to the multirate filtering techniques. They achieve computational efficiency by adapting the sampling frequency and the filter order according to the input signal local variations. Both techniques have some common features, which are described in the following.

In both cases, a reference FIR filter is offline designed for a reference sampling frequency F_{ref} . Its impulse response is h_k , where k is indexing the reference filter coefficients. F_{ref} is chosen in order to satisfy the Nyquist sampling criterion for $x(t)$, namely $F_{\text{ref}} \geq 2f_{\max}$.

During online computation, F_{ref} and the local sampling frequency Fs^i of window W^i are used to define the local resampling frequency Frs^i and a decimation factor d^i . The Frs^i is employed to uniformly resample the selected signal laying in W^i , where as d^i is employed to decimate h_k for filtering W^i .

Frs^i can be specific depending upon Fs^i [11, 12]. For proper online filtering, F_{ref} and Frs^i should match. The approaches of keeping F_{ref} and Frs^i coherent are explained below.

In the case, when $Fs^i \geq F_{\text{ref}}$, $Frs^i = F_{\text{ref}}$ is chosen and h_k remains unchanged. This case is treated similarly by both proposed techniques. This choice of Frs^i makes to resample W^i closer to the Nyquist rate, so avoiding unnecessary interpolations during the data resampling process. It thus further improves the proposed technique computational efficiency. This case is included in the description (see flowcharts in Figures 3 and 4) of the following two filtering techniques.

In the opposite case, that is, $Fs^i < F_{\text{ref}}$, $Frs^i = Fs^i$ is chosen and h_k is online decimated in order to reduce F_{ref} to Frs^i . In this case, the reference filter order is reduced for W^i , which reduces the number of operations to deliver a filtered sample [6, 11]. Hence, it improves the proposed techniques computational efficiency. In this case, it appears that Frs^i may be lower than the Nyquist frequency of $x(t)$ and so it can cause aliasing. According to [6, 11], if the local signal amplitude is of the order of the maximal range ΔV_{in} , then for a suitable choice of M (application-dependent) the signal crosses enough consecutive thresholds. Thus, it is locally oversampled with respect to its local bandwidth and so there is no aliasing problem. This statement is further illustrated with the results summarized in Table 3.

In order to decimate h_k the decimation factor d^i for W^i is online calculated by employing

$$d^i = \frac{F_{\text{ref}}}{Frs^i}, \quad (11)$$

d^i can be specific for each selected window depending upon Frs^i . For an integral d^i both techniques decimate h_k in a similar way. Thus, a test on d^i is made by computing $D^i = \text{floor}(d^i)$ and verifying if $(D^i = d^i)$. Here, floor operation delivers only the integral part of d^i . If the answer is yes, then h_k is decimated with D^i , the process is clear from

$$h_j^i = h_{D^i k}. \quad (12)$$

Equation (12) shows that the decimated filter impulse response for the i th selected window h_j^i is obtained by picking every (D^i) th coefficient from h_k . Here, j is indexing the decimated filter coefficients. If the order of h_k is P , then the order of h_j^i is given as: $P^i = P/D^i$.

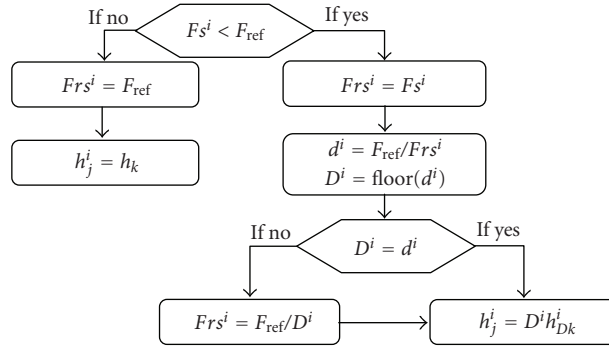


FIGURE 3: Flowchart of the ARD.

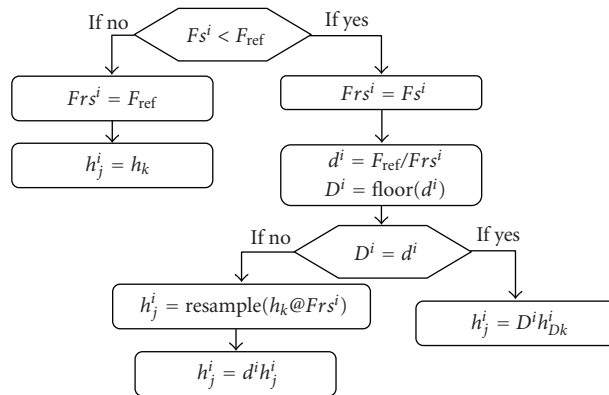


FIGURE 4: Flowchart of the ARR.

TABLE 1: Summary of the input signal active parts.

Active part	Signal components	Length (s)
1st	$0.5 \sin(2\pi 20t) + 0.4 \sin(2\pi 1000t)$	0.5
2nd	$0.45 \sin(2\pi 10t) + 0.45 \sin(2\pi 150t)$	1.0
3rd	$0.6 \sin(2\pi 5t) + 0.3 \sin(2\pi 100t)$	1.0

A simple decimation causes a reduction of the decimated filter energy compared to the reference one. It will lead to an attenuated version of the filtered signal. D^i is a good approximate of the ratio between the energy of the reference filter and that of the decimated one. Thus, this effect of decimation is compensated by scaling h_j^i with D^i . The process is clear from

$$h_j^i = D^i h_{Dk}^i. \quad (13)$$

The two techniques mainly differ in the way of decimating h_k for a fractional d^i . The process is explained in the following Sections.

3.2. ARD (Activity Reduction by Filter Decimation). In the ARD technique, h_k is decimated by employing D^i . It calls for an adjustment of Frs^i which is achieved as $Frs^i = F_{ref} / D^i$. As in this case, $D^i < d^i$, so it makes $Frs^i > Fs^i$. For the ARD

h_j^i scaling is performed with D^i . The complete procedure of obtaining Frs^i and h_j^i for the ARD is described in Figure 3.

3.3. ARR (Activity Reduction by Filter Resampling). In the ARR technique, d^i is employed to decimated h_k . In this case, Frs^i is given as $Frs^i = F_{ref} / d^i$, so it remains equal to Fs^i . The process of matching F_{ref} with Frs^i requires a fractional decimation of h_k , which is achieved by resampling h_k at Frs^i . Again NNRI is employed for the purpose of h_k resampling. For the ARR h_j^i scaling is performed with d^i . The complete procedure of obtaining Frs^i and h_j^i for the ARR is described in Figure 4.

4. Illustrative Example

In order to illustrate the ARD and the ARR filtering techniques, an input signal $x(t)$ shown on the left part of Figure 5 is employed. Its total duration is 20 seconds and it consists of three active parts. Summary of $x(t)$ activities is given in Table 1.

Table 1 shows that $x(t)$ is band limited between $f_{min} = 5$ Hz and $f_{max} = 1$ kHz. In this case, $x(t)$ is digitized by employing a 3-bit resolution AADC. Thus, for given ENOB the corresponding minimum and maximum sampling frequencies are $Fs_{min} = 70$ Hz and $Fs_{max} = 14$ kHz. The AADC amplitude range $\Delta V_{in} = 1.8$ v is chosen, which results into a quantum $q = 0.2571$ v.

TABLE 2: Summary of the reference filter parameters.

Cut-off frequency (Hz)	Transition band (Hz)	Pass-band ripples (dB)	Stop-band ripples (dB)	F_{ref} (Hz)	P
30	30 ~ 80	-25	-80	2500	127

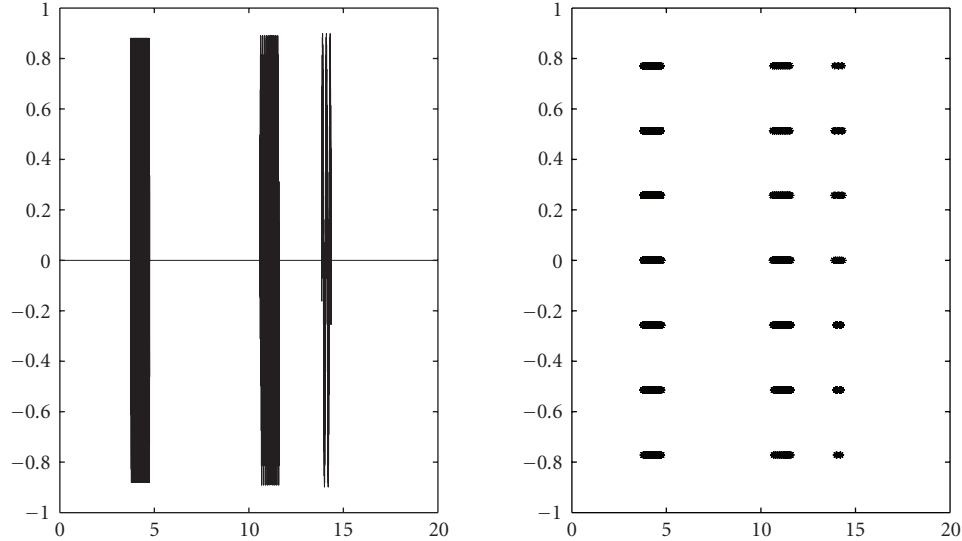


FIGURE 5: The input signal (left) and the selected signal obtained with the ASA (right).

TABLE 3: Summary of the selected windows parameters.

i	T^i (Sec.)	N^i (Samples)	Fs^i (Hz)	F_{ref} (Hz)	Frs^i (Hz)	d^i
1	0.4994	3000	6000	2500	2500	1
2	0.9993	1083	1083	2500	1083	2.3
3	0.9986	464	464	2500	464	5.4

TABLE 4: Values of Frs^i , Nr^i , D^i , and P^i for each selected window in the ARD.

i	Frs^i (Hz)	Nr^i	D^i	P^i
1	2500	1250	1	127
2	1250	1250	2	64
3	500	500	5	26

Each activity contains a low- and a high-frequency component (cf. Table 1). In order to filter out the high-frequency parts from each activity, a low pass reference FIR filter is implemented by employing the standard Parks-McClellan algorithm. The reference filter parameters are summarized in Table 2.

For this example the reference window length $T_{\text{ref}} = 1$ second is chosen. It satisfies the boundary conditions discussed in Section 2.3. The given T_{ref} delivers $N_{\text{max}} = 14000$ samples in this case (cf. Equation (9)). The ASA delivers three selected windows for the whole $x(t)$ span of 20 seconds, which are shown on the right part of Figure 5. The selected windows parameters are displayed in Table 3.

Table 3 shows that the first window is an example of the $Fs^i \geq F_{\text{ref}}$ case, so it is tackled similarly by both techniques. In the other windows, $Fs^i < F_{\text{ref}}$ is valid, so the online h_k decimation is employed. As d^2 and d^3 , calculated by employing Equation (11) are fractional ones, so this case is tackled in a different way by the ARD and the ARR.

Values of Frs^i , D^i , Nr^i and P^i are calculated for the ARD, and the ARR by employing the methods shown in Figures 3 and 4, respectively. The obtained results are summarized in Tables 4 and 5.

Tables 3, 4, and 5 jointly exhibit the interesting features of the proposed filtering techniques, which are achieved by an intelligent combination of the nonuniform, and the uniform signal processing tools (cf. Figure 2). Fs^i represents the sampling frequency adaptation by following the local variations of $x(t)$. N^i shows that the relevant signal parts are locally over-sampled in time with respect to their local bandwidths [6, 11]. Frs^i shows the adaptation of the resampling frequency for each selected window. It further adds to the computational gain of the proposed techniques by avoiding the unnecessary interpolations during the resampling process. Nr^i shows how the adjustment of Frs^i avoids the processing of unnecessary samples during the post filtering process. P^i represents how the adaptation of h_k for \mathbf{W}^i avoids the unnecessary operations to deliver the filtered signal. T^i exhibits the dynamic feature of ASA, which is to correlate T_{ref} with the signal activity laying in it [5].

These results have to be compared with what is done in the corresponding classical case. If F_{ref} is chosen as the sampling frequency, then the total $x(t)$ span is sampled at 2500 Hz. It makes $N = 20 \times 2500 = 50000$ samples to process with the 127th-order FIR filter. On the other hand, in both proposed techniques the total number of resampled data

TABLE 5: Values of Frs^i , Nr^i , d^i and P^i for each selected window in the ARR.

i	Frs^i (Hz)	Nr^i	d^i	P^i
1	2500	1250	1	127
2	1083	1083	2.3	54
3	464	464	5.4	24

points is much lower, 3000 and 2794 for the ARD and the ARR, respectively. Moreover, the local filter orders in W^2 and W^3 are also lower than 127. It promises the computational efficiency of the proposed techniques compared to the classical one. A detailed complexity comparison is made in the following Section.

5. Computational Complexity

In the classical case, with a P order filter, it is well known that P multiplications and P additions are required to compute each filtered sample. If N is the number of samples then the total computational complexity C can be calculated by employing

$$C = \underbrace{PN}_{\text{Additions}} + \underbrace{PN}_{\text{Multiplications}}. \quad (14)$$

In the adaptive techniques presented here, the adaptation process requires extra operations for each selected window. The computational complexities of both techniques, C_{ARD} and C_{ARR} are deduces as follow.

The following steps are common to both the ARD and the ARR techniques. The choice of Frs^i is a common operation for both proposed techniques. It requires one comparison between F_{ref} and Fs^i . The data resampling operation is also required in both techniques before filtering. In the studied case, the resampling process is performed by employing the Nearest Neighbour Resampling Interpolation (NNRI). The NNRI is chosen because of its simplicity, as it employs only one nonuniform observation for each resampled one. Moreover, it provides an unbiased estimate of the original signal variance. Due to this reason, it is also known as a robust interpolation method [35, 36]. The detailed reasons of inclination toward NNRI are discussed in [5, 35, 36]. The NNRI is performed as follow.

For each interpolation instant tr_n , the interval of nonuniform samples $[t_n, t_{n+1}]$, within which tr_n lies is determined. Then the distance of tr_n to each t_n and t_{n+1} is computed and a comparison among the computed distances is performed to decide the smaller among them. For W^i , the complexity of the first step is $N^i + Nr^i$ comparisons and the complexity of the second step is $2Nr^i$ additions and Nr^i comparisons. Hence, the NNRI total complexity for W^i becomes $N^i + 2Nr^i$ comparisons and $2Nr^i$ additions.

In the case, when $Fs^i < F_{\text{ref}}$, the decimation of h_k is performed in both techniques. In order to do so, d^i is computed by performing a division between F_{ref} and Frs^i . D^i is calculated by employing a floor operation on d^i . A comparison is made between D^i and d^i . In the case when $D^i = d^i$, the process of obtaining h_k^i is similar for

both techniques (cf. Figures 3 and 4). In this case, the decimator simply picks every (D^i)th coefficient from h_k . It has a negligible complexity compared to the operations like addition and multiplication. This is the reason why its complexity is not taken into account during the complexity evaluation process. In both techniques, the decimated filter impulse response is scaled, it requires P^i multiplications. The fractional d^i is tackled in a different way by each filtering technique and is detailed in the following subsections.

5.1. Complexity of the ARD Technique. Even if d^i is fractional in the case of ARD technique, h_k decimation is performed by employing D^i . Frs^i is modified in order to keep it coherent with F_{ref} and it requires one division (cf. Figure 3). Finally, a P^i -order filter performs $P^i Nr^i$ multiplications and $P^i Nr^i$ additions for W^i . The combine computational complexity for the ARD technique C_{ARD} is given by

$$C_{\text{ARD}} = \sum_{i=1}^I \underbrace{\alpha(1+\beta)}_{\text{Divisions}} + \underbrace{\alpha}_{\text{Floor}} + \underbrace{N^i + Nr^i + 1 + \alpha}_{\text{Comparisons}} + \underbrace{Nr^i(P^i + 2)}_{\text{Additions}} + \underbrace{P^i(Nr^i + \alpha)}_{\text{Multiplications}}. \quad (15)$$

5.2. Complexity of the ARR Technique. In the case of ARR technique, d^i is employed as the decimation factor. The fractional decimation is achieved by resampling h_k at Frs^i . The resampling is performed by employing the NNRI, which performs $P + 2P^i$ comparisons and $2P^i$ additions to deliver h_k^i . The remaining operation cost between the ARD and the ARR is common. The combine computational complexity for the ARR technique C_{ARR} is given by

$$C_{\text{ARR}} = \sum_{i=1}^I \underbrace{\alpha}_{\text{Divisions}} + \underbrace{\alpha}_{\text{Floor}} + \underbrace{N^i + Nr^i + 1 + \alpha \left[1 + \beta(P + 2P^i) \right]}_{\text{Comparisons}} + \underbrace{Nr^i(P^i + 2) + 2\alpha\beta P^i}_{\text{Additions}} + \underbrace{P^i(Nr^i + \alpha)}_{\text{Multiplications}}. \quad (16)$$

In Equations (15) and (16), $i = 1, 2, 3, \dots, I$, represents the selected windows index. α and β are the multiplying factors. α is 0 for the case when $Fs^i \geq F_{\text{ref}}$ and it is 1 otherwise. β is 0 for the case when $d^i = D^i$ and it is 1 otherwise.

5.3. Complexity Comparison of the ARD and the ARR with the Classical Filtering. From (14), (15), and (16), it is clear that there are uncommon operations between the classical and the proposed adaptive rate filtering techniques. In order to make them approximately comparable, it is assumed that a comparison has the same processing cost as that of an addition and a division or a floor has the same processing cost as that of a multiplication. By following these assumptions, comparisons are merged into

TABLE 6: Computational gain of the ARD over the classical one for different $x(t)$ time spans.

Signal part	Gain in additions	Gain in multiplications
W^1	1.98	2
W^2	3.91	3.96
W^3	23.51	24.37
Whole signal	25.93	26.22

TABLE 7: Computational gain of the ARR over the classical one for different $x(t)$ time spans.

Signal part	Gain in additions	Gain in multiplications
W^1	1.98	2
W^2	5.33	5.42
W^3	27.31	28.45
Whole signal	29.44	29.81

the additions count and divisions plus floors are merged into the multiplications count, during the complexity evaluation process. Now Equations (15) and (16) can be written as follow:

$$C_{ARD} = \underbrace{\sum_{i=1}^I N^i + Nr^i (P^i + 3)}_{\text{Additions}} + \alpha + 1 + \underbrace{P^i Nr^i + \alpha (P^i + 2 + \beta)}_{\text{Multiplications}}, \quad (17)$$

$$C_{ARR} = \underbrace{\sum_{i=1}^I N^i + Nr^i (P^i + 3) + \alpha [1 + \beta (P + 3P^i)]}_{\text{Additions}} + 1 + \underbrace{P^i Nr^i + \alpha (P^i + 2)}_{\text{Multiplications}}. \quad (18)$$

By employing results of the example studied in the previous section, computational comparisons of the ARD and the ARR with the classical one are made in terms of additions and multiplications. The results are computed for different $x(t)$ time spans and are summarized in Tables 6 and 7.

Gains in additions and multiplications of the proposed techniques over the classical one are clear from the above results. In the case of W^1 , where the resampling frequency and the filter order is the same as in the classical case (cf. Tables 4 and 5), a gain is achieved by using the proposed adaptive techniques. This is only due to the fact that the ASA correlates the window length to the activity (0.5 second), while the classic case computes during the total duration of $T_{ref} = 1$ second. Gains are of course much larger in other windows, since the proposed techniques are taking benefit of processing the lesser samples along with the lower filter orders. When treating the whole $x(t)$ span of 20 seconds, the proposed techniques also take advantage of the idle $x(t)$ parts, which further induces additional gains compared to the classical case.

The above results confirm that the proposed filtering techniques lead toward a drastic reduction in the number of operations compared to the classical one. This reduction in operations is achieved due to the joint benefits of the AADC, the ASA and the resampling, as they enable to adapt the sampling frequency and the filter order by following the input signal local variations.

5.4. Complexity Comparison between the ARD and the ARR. The main difference between both proposed techniques occurs for the case when $Fs^i < F_{ref}$ and d^i is fractional (cf. Section 3).

The ARD makes an increment in Frs^i in order to keep it coherent with F_{ref} . Increase in Frs^i causes to increase Nr^i and also to increase P^i . Thus, in comparison to the ARR, this technique increases the computational load of the post-filtering operation, while keeping the decimation process of h_k simple.

The ARR performs h_k resampling at Frs^i . Thus, in comparison to the ARD, this technique increases the complexity of the decimation process of h_k , while keeping the computational load of the post-filtering process lower.

In continuation to Section 5.3, a complexity comparison between the ARD and the ARR is made in terms of additions, and multiplications by employing Equations (17) and (18), respectively. It concludes that the ARR remains computationally efficient compared to the ARD, in terms of additions and multiplications, as far as the conditions given by expressions (19) and (20) remain true. Please note that Nr^i and P^i can be different for the ARD and the ARR (cf. Tables 4 and 5):

$$[Nr^i (P^i + 3)]_{ARD} - [Nr^i (P^i + 3)]_{ARR} > [P + 3P^i]_{ARR} \quad (19)$$

$$[P^i (Nr^i + 1) + 1]_{ARD} > [P^i (Nr^i + 1)]_{ARR}. \quad (20)$$

For this studied example, d^2 and d^3 are fractional ones, thus the ARD and the ARR proceed differently. Conditions (19) and (20) remain true for both W^2 and W^3 (cf. Tables 4 and 5). Hence, the gains in additions and multiplications of the ARR are higher than those of the ARD for W^2 and W^3 (cf. Tables 6 and 7). It shows that except for very specific situation the ARR technique will always remain less expensive than the ARD. The ARR achieves this computational performance by employing the fractional decimation of h_k , which may lead a quality compromise of the ARR compared to the ARD. This issue is addressed in the following section.

6. Processing Error

6.1. Approximation Error. In the proposed techniques, the approximation error occurs due to two effects: the time quantization error which occurs due to the AADC finite timer precision and the interpolation error which occurs in the course of the uniform resampling process. After these

TABLE 8: Mean approximation error of each selected window for the ARD and the ARR.

Selected window	W^1	W^2	W^3
MAe^i for the ARD (dB)	-23.71	-25.84	-26.35
MAe^i for the ARR (dB)	-23.71	-25.93	-26.63

two operations, the mean approximation error for W^i can be computed by employing the following:

$$MAe^i = \frac{1}{Nr^i} \sum_{n=1}^{Nr^i} |xo_n - xr_n|. \quad (21)$$

Here, xr_n is the n th resampled observation, interpolated with respect to the time instant tr_n , xo_n is the original sample value which should be obtained by sampling $x(t)$ at tr_n . In the studied example discussed in Section 4, $x(t)$ is analytically known, thus it is possible to compute its original sample value at any given time instant. It allows us to compute the approximation error introduced by the proposed adaptive rate techniques by employing Equation (21).

The results obtained for each selected window for both the ARD and the ARR are summarized in Table 8.

Table 8 shows the approximation error introduced by the proposed techniques. This process is accurate enough for a 3-bit AADC. For the higher precision applications, the approximation accuracy can be improved by increasing the AADC resolution M and the interpolation order [6, 8, 37, 38]. Thus, an increased accuracy can be achieved at the cost of an increased computational load. Therefore, by making a suitable compromise between the accuracy level and the computational load, an appropriate solution can be devised for a specific application.

For a given M and interpolation order the approximation accuracy can be further improved by employing the symmetry during the interpolation process. It results into a reduced resampling error [38, 39]. The pros and cons of this approach are under investigation and a description on it is given in [40].

6.2. Filtering Error. In the proposed filtering techniques, a reference filter h_k is employed and then it is online decimated for W^i , depending on the chosen Frs^i . This online decimation can cause the filtering precision degradation. In order to evaluate this phenomenon on our test signal the following procedure is adapted.

A reference filtered signal is generated. In this case, instead of decimating h_k to obtain h_j^i , a specific filter h_m^i is directly designed for W^i by using the Parks-McClellan algorithm. It is designed for Frs^i by employing the same design parameters, summarized in Table 2. The signal activity corresponding to W^i is sampled at Frs^i with a high precision classical ADC. This sampled signal is filtered by employing h_m^i . The filtered signal obtained in this way is used as a reference one for W^i , and its comparison is made with the results obtained by the proposed techniques.

Let y_n be the n th reference-filtered sample and $y_{\hat{n}}$ be the n th filtered sample obtained by one of the proposed

TABLE 9: Mean filtering error of each selected window for the ARD and the ARR.

Selected window	1st	2nd	3nd
MFe^i for the ARD (dB)	-43.23	-39.45	-17.07
MFe^i for the ARR (dB)	-43.23	-30.46	-11.60

filtering techniques. Then, the mean filtering error for W^i can be calculated by employing

$$MFe^i = \frac{1}{Nr^i} \sum_{i=1}^{Nr^i} |y_n - y_{\hat{n}}|. \quad (22)$$

The mean filtering error of both proposed techniques is calculated, for each $x(t)$ activity by employing (22). The results are summarized in Table 9.

Table 9 shows that the online decimation of h_k in the proposed techniques causes a loss of the desired filtering quality. Indeed, the filtering error increases with the increase in d^i . The measure of this error can be used to decide an upper bound to d^i (by performing an offline calculation), for which the decimated and the scaled filters provide results with an acceptable level of accuracy. The level of accuracy is application-dependent. Moreover, for high precision applications, an appropriate filter can be online calculated for each selected window at the cost of an increased computational load. The process is clear from generating the reference filtered signal y_n , discussed above.

Table 9 shows that MFe^2 and MFe^3 for the ARR are higher than that of the ARD. It is due to the fact of h_k resampling for the ARR to deliver h_j^2 and h_j^3 . It makes to employ the interpolated coefficients of h_k for filtering the resampled data, lies in W^2 and W^3 , respectively, which results in an increased filtering error of the ARR compared to the ARD. Similar to Section 6.1, this resampling error can also be reduced to a certain extent, by employing a higher order interpolator [37, 38]. In conclusion, a certain increase in the accuracy can be achieved at a certain loss of the processing efficiency.

7. Speech Signal as a Case Study

In order to evaluate performances of the ARD and the ARR for real life signals, a speech signal $x(t)$ shown on Figure 6(a) is employed. $x(t)$ is a 1.6 second, [50 Hz; 5000 Hz] band-limited signal corresponding to a three-word sentence. The goal is to determine the pitch (fundamental frequency) of $x(t)$ in order to determine the speaker's gender. For a male speaker, the pitch lies with the frequency range [100 Hz, 150 Hz], whereas for a female speaker, the pitch lies with the frequency range [200 Hz, 300 Hz] [41].

The reference frequency is chosen as $F_{ref} = 11.2$ kHz, which is a common sampling frequency for speech. A 4-bit resolution AADC is used for digitizing $x(t)$, and therefore we have $F_{smin} = 1.5$ kHz, and $F_{smax} = 150$ kHz. The amplitude range is always set to $\Delta V_{in} = 1.8$ V, which leads to a quantum $q = 0.12$ v. The amplitude of $x(t)$ is normalized to 0.9 v in order to avoid the AADC saturation.

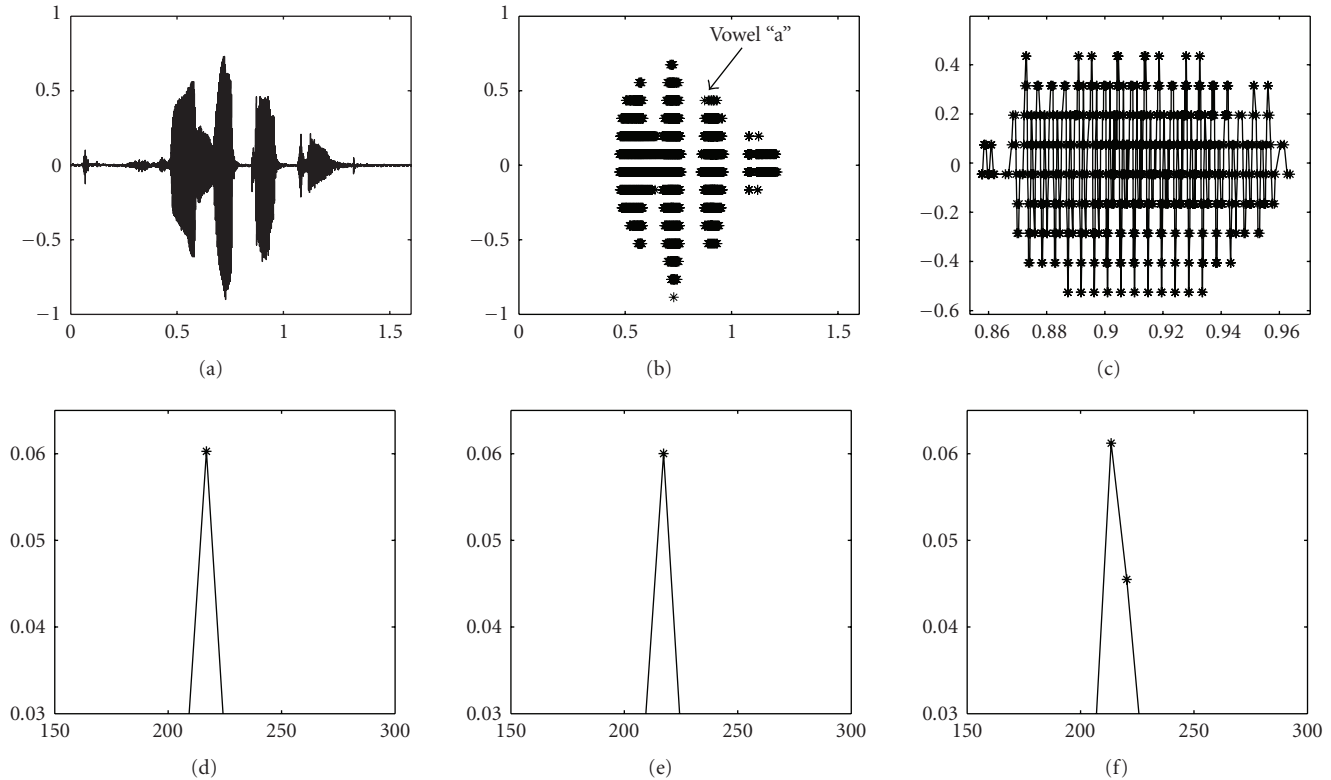


FIGURE 6: On the top, the input speech signal (a), the selected signal with the ASA (b) and a zoom of the second window W^2 (c). On the bottom, a spectrum zoom of the filtered signal laying in W^2 obtained with the reference filtering (d), with the ARD (e) and with the ARR (f), respectively.

The studied signal is part of a conversation and during a dialog, the speech activity is 25% of the total dialog time [42]. A classical filtering system would remain active during the total dialog duration. The proposed LCSS-based filtering techniques will remain active only during 25% of the dialog time span, which will reduce the system power consumption.

A speech signal mainly consists of vowels and consonants. Consonants are of lower amplitude compared to vowels [41, 43]. In order to determine the speakers pitch, vowels are the relevant parts of $x(t)$. For $q = 0.12$ v, consonants are ignored during the signal acquisition process, and are considered as low amplitude noise. In contrast, vowels are locally over-sampled like any harmonic signal [6, 10, 11]. This intelligent signal acquisition further avoids the processing of useless samples, within the 25% of $x(t)$ activity, and so further improves the proposed techniques computational efficiency.

In order to apply the ASA, $T_{\text{ref}} = 0.5$ seconds is chosen. It results in $N_{\text{max}} = T_{\text{ref}}F_{\text{max}} = 75000$ in this case (cf. Equation (9)). The ASA delivers three selected windows, which are shown on Figure 6(b). The parameters of each selected window are summarized in Table 10.

Although the consonants are partially filtered out during the data acquisition process, yet for proper pitch estimation, it is required to filter out the remaining effect of high frequencies still present in $x(t)$. To this aim, a reference low pass filter is designed, with the standard Parks-McClellan algorithm. Its characteristics are summarized in Table 11.

TABLE 10: Summary of the selected windows parameters.

Selected window	T^i (Second)	N^i (Samples)	F_s^i (Hz)
W^i	0.2074	2360	11379
W^2	0.1136	347	3054
W^3	0.1210	265	2190

To find the pitch, we now focus on W^2 , which corresponds to the vowel "a". A zoom on this signal part is plotted on Figure 6(c). The condition $F_s^2 \leq F_{\text{ref}}$ is valid, and d^2 is fractional (cf. Equation (11)). Thus, the filtering process for each proposed technique will differ, which makes it possible to compare their performances. The values of Frs^2 , Nr^2 , D^2 , and P^2 for both techniques are given in Table 12.

Computational gains of the proposed filtering techniques compared to the classical one are computed by employing Equations (14), (17), and (18). The results show 8.62 and 13.17 times gains in additions and 8.71 and 13.26 times gains in multiplications, respectively, for the ARD and the ARR, for W^2 . It confirms the computational efficiency of the proposed techniques compared to the classical one. It is gained firstly by achieving an intelligent signal acquisition and secondly by adapting the sampling frequency and the filter order by following the local variations of $x(t)$.

Once more the conditions (19) and (20) remain true for W^2 so the ARR technique remains computationally efficient than the ARD one.

TABLE 11: Summary of the reference filter parameter 1

Cut-off frequency (Hz)	Transition band (Hz)	Pass-band ripples (dB)	Stop-band ripples (dB)	F_{ref}	$P(\text{order})$
300	300 ~ 400	-25	-80	11.2	284

TABLE 12: Values of \mathbf{Fr}^2 , \mathbf{N}^2 , \mathbf{D}^2 , and \mathbf{P}^2 for the ARD and the ARR.

W_2	Fr^2 (Hz)	Nr^2	D^2	P^2
ARD	3733	424	3	95
ARR	3054	347	3.7	77

Spectra of the filtered signal laying in W^2 , obtained with the reference filtering (cf. Section 6.2), with the ARD and with the ARR techniques are plotted, respectively, on Figures 6(d), 6(e), and 6(f).

The spectra on Figure 6 show that the fundamental frequency is about 215 HZ. Thus, one can easily conclude that the analyzed sentence is pronounced by a female speaker. Although it is required to decimate the reference filter 3 times and 3.7 times, respectively, for the ARD and the ARR, yet spectra of the filtered signal, obtained with the proposed techniques are quite comparable to spectrum of the reference-filtered signal. It shows that even after such a level of decimation, results delivered by the proposed techniques are of acceptable quality for the studied speech application.

The above discussion shows the suitability of the proposed techniques for the low activity time-varying signals like electrocardiogram, phonocardiogram, seismic, and speech. Speech is a common, and easily accessible signal. Therefore, the proposed techniques performance is studied for a speech application, though it can be applied to other appropriate real signals like electrocardiogram, phonocardiogram, and seismic. The devised approach versatility lays in the appropriate choice of system parameters like the AADC resolution M , the distribution of level crossing thresholds, and the interpolation order. These parameters should be tactfully chosen for a targeted application, so that they ensure an attractive tradeoff between the system computational complexity and the delivered output quality.

8. Conclusion

Two novel adaptive rate filtering techniques have been devised. These are well suited for low activity sporadic signals like electrocardiogram, phonocardiogram and seismic signals. For both filtering techniques, a reference filter is offline designed by taking into account the input signal statistical characteristics and the application requirements.

The complete procedure of obtaining the resampling frequency Fr^i and the decimated filter coefficients h_j^i for W^i is described for both proposed techniques. The computational complexities of the ARD and the ARR are deduced and compared with the classical one. It is shown that the proposed techniques result into a more than one-order magnitude gain in terms of additions and multiplications over the classical one. It is achieved due to the joint benefits of the AADC, the ASA and the resampling as they allow the

online adaptation of parameters ($Fr^i, Frs^i, N^i, Nr^i, D^i$, and P^i) by exploiting the input signal local variations. It drastically reduces the total number of operations and therefore, the energy consumption compared to the classical case.

A complexity comparison between the ARD and the ARR is also made. It is shown that the ARR outperforms the ARD in most of the cases. Performances of the ARD and the ARR are also demonstrated for a speech application. The results obtained in this case are in coherence with those obtained for the illustrative example.

Methods to compute the approximation and the filtering errors for the proposed techniques are also devised. It is shown that the errors made by the proposed techniques are minor ones, in the studied case. A higher precision can be achieved by increasing the AADC resolution and the interpolation order. Thus, a suitable solution can be proposed for a given application by making an appropriate tradeoff between the accuracy level and the computational load.

A detailed study of the proposed filtering techniques computational complexities by taking into account the real processing cost at circuit level is in progress. Future works focus on the optimization of these filtering techniques and their further employment in real life applications.

References

- [1] S. C. Sekhar and T. V. Sreenivas, "Adaptive window zero-crossing-based instantaneous frequency estimation," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 12, pp. 1791–1806, 2004.
- [2] J. W. Mark and T. D. Todd, "A nonuniform sampling approach to data compression," *IEEE Transactions on Communications*, vol. 29, pp. 24–32, 1981.
- [3] M. Gretains, "Time-frequency representation based chirp like signal analysis using multiple level crossings," in *Proceedings of 15th European Signal Processing Conference (EUSIPCO '07)*, pp. 2154–2158, Poznan, Poland, September 2007.
- [4] K. M. Guan and A. C. Singer, "Opportunistic sampling by level-crossing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 3, pp. 1513–1516, Honolulu, Hawaii, USA, April 2007.
- [5] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Spectral analysis of a signal driven sampling scheme," in *Proceedings of the 14th European Signal Processing Conference (EUSIPCO '06)*, Florence, Italy, September 2006.
- [6] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Computationally efficient adaptive rate sampling and filtering," in *Proceedings of 15th European Signal Processing Conference (EUSIPCO '07)*, pp. 2139–2143, Poznan, Poland, September 2007.
- [7] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization," in *Proceedings of the 9th International Symposium on Asynchronous Circuits and Systems (ASYNC '03)*, pp. 197–205, Vancouver, Canada, May 2003.

- [8] 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, pp. 335–339, 1996.
- [9] F. Akopyan, R. Manohar, and A. B. Apsel, "A level-crossing flash asynchronous analog-to-digital converter," in *Proceedings of the International Symposium on Asynchronous Circuits and Systems (ASYNC '06)*, pp. 12–22, Grenoble, France, March 2006.
- [10] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, "Asynchronous FIR filters: towards a new digital processing chain," in *Proceedings of the International Symposium on Asynchronous Circuits and Systems (ASYNC '04)*, vol. 10, pp. 198–206, Crete, Greece, April 2004.
- [11] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Adaptive rate filtering for a signal driven sampling scheme," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 3, pp. 1465–1468, Honolulu, Hawaii, USA, April 2007.
- [12] S. M. Qaisar, L. Fesquet, and M. Renaudin, "Computationally efficient adaptive rate sampling and filtering for low power embedded systems," in *Proceedings of the International Conference on Sampling Theory and Applications (SampTA '07)*, Thessaloniki, Greece, June 2007.
- [13] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, "Spectral analysis of level crossing sampling scheme," in *Proceedings of the International Conference on Sampling Theory and Applications (SampTA '05)*, Samsun, Turkey, July 2005.
- [14] S. M. Qaisar, L. Fesquet, and M. Renaudin, "An adaptive resolution computationally efficient short-time Fourier transform," *Research Letters in Signal Processing*, vol. 2008, Article ID 932068, 5 pages, 2008.
- [15] F. E. Bond and C. R. Cahn, "On sampling the zeros of bandwidth limited signals," *IRE Transactions on Information Theory*, vol. 4, pp. 110–113, 1958.
- [16] K. J. Astrom and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control (CDC '02)*, vol. 2, pp. 2011–2016, Las Vegas, Nev, USA, December 2002.
- [17] I. Bilinskis, *Digital Alias Free Signal Processing*, John Wiley & Sons, New York, NY, USA, 2007.
- [18] P. H. Ellis, "Extension of phase plane analysis to quantized systems," *IRE Transactions on Automatic Control*, vol. 4, pp. 43–59, 1959.
- [19] M. Lim and C. Saloma, "Direct signal recovery from threshold crossings," *Physical Review E*, vol. 58, no. 5B, pp. 6759–6765, 1998.
- [20] M. Miskowicz, "Asymptotic effectiveness of the event-based sampling according to the integral criterion," *Sensors*, vol. 7, no. 1, pp. 16–37, 2007.
- [21] K. J. Astrom and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems," in *Proceedings of IFAC World Congress*, pp. 301–306, 1999.
- [22] M. Miskowicz, "Send-on-delta concept: an event-based data reporting strategy," *Sensors*, vol. 6, no. 1, pp. 49–63, 2006.
- [23] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of the American Control Conference (ACC '02)*, vol. 4, pp. 3015–3020, Anchorage, Alaska, USA, May 2002.
- [24] S. C. Gupta, "Increasing the sampling efficiency for a control system," *IEEE Transactions on Automatic and Control*, pp. 263–264, 1963.
- [25] I. F. Blake and W. C. Lindsey, "Level-crossing problems for random processes," *IEEE Transactions on Information Theory*, pp. 295–315, 1973.
- [26] M. Miskowicz, "Efficiency of level-crossing sampling for bandlimited Gaussian random processes," in *Proceedings of IEEE International Workshop on Factory Communication Systems (WFCS '06)*, pp. 137–142, Torino, Italy, June 2006.
- [27] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, 1999.
- [28] M. A. Nazario and C. Saloma, "Signal recovery in sinusoid-crossing sampling by use of the minimum-negative constraint," *Applied Optics*, vol. 37, pp. 2953–2963, 1988.
- [29] M. Lim and C. Saloma, "Direct signal recovery from threshold crossings," *Physical Review E*, vol. 58, no. 5B, pp. 6759–6765, 1998.
- [30] F. J. Beutler, "Error free recovery from irregularly spaced samples," *SIAM Review*, vol. 8, pp. 328–335, 1996.
- [31] F. Marvasti, *Nonuniform Sampling Theory and Practice*, Kluwer Academic/Plenum Publishers, New York, NY, USA, 2001.
- [32] M. Vetterli, "A theory of multirate filter banks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 356–372, 1987.
- [33] S. Chu and C. S. Burrus, "Multirate filter designs using comb filters," *IEEE Transactions on Circuits and Systems*, vol. 31, no. 11, pp. 913–924, 1984.
- [34] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.
- [35] S. de Waele and P. M. T. Broersen, "Time domain error measure for resampled irregular data," in *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (IMTC '99)*, vol. 2, pp. 1172–1177, Venice, Italy, May 1999.
- [36] S. de Waele and P. M. T. Broersen, "Error measures for resampled irregular data," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 2, pp. 216–222, 2000.
- [37] F. Harris, "Multirate signal processing in communication systems," in *Proceedings of 15th European Signal Processing Conference (EUSIPCO '07)*, Poznan, Poland, September 2007.
- [38] D. M. Klamer and E. Masry, "Polynomial interpolation of randomly sampled bandlimited functions and processes," *SIAM Journal on Applied Mathematics*, vol. 42, no. 5, pp. 1004–1019, 1982.
- [39] F. B. Hildebrand, *Introduction to Numerical Analysis*, McGraw-Hill, Boston, Mass, USA, 1956.
- [40] S. M. Qaisar, L. Fesquet, and M. Renaudin, "An improved quality adaptive rate filtering technique based on the level crossing sampling," in *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 31, pp. 79–84, July 2008.
- [41] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1978.
- [42] P. G. Fontollet, *Systèmes de Télécommunications*, Dunod, Paris, France, 1983.
- [43] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2001.