An Improved Quality Adaptive Rate Filtering Technique Based on the Level Crossing Sampling

Saeed Mian Qaisar, Laurent Fesquet, and Marc Renaudin

Abstract—Mostly the systems are dealing with time varying signals. The Power efficiency can be achieved by adapting the system activity according to the input signal variations. In this context an adaptive rate filtering technique, based on the level crossing sampling is devised. It adapts the sampling frequency and the filter order by following the input signal local variations. Thus, it correlates the processing activity with the signal variations. Interpolation is required in the proposed technique. A drastic reduction in the interpolation error is achieved by employing the symmetry during the interpolation process. Processing error of the proposed technique is calculated. The computational complexity of the proposed filtering technique is deduced and compared to the classical one. Results promise a significant gain of the computational efficiency and hence of the power consumption.

Keywords—Level Crossing Sampling, Activity Selection, Rate Filtering, Computational Complexity, Interpolation Error.

I. CONTEXT OF THE STUDY

This work is a contribution in the development of smart mobile systems. The goal is to reduce their size, cost, processing noise, electromagnetic emission and especially power consumption as they are remotely powered by batteries. This can be done by smartly reorganizing their associated signal processing theory and architecture. The idea is to combine the signal event driven processing with the asynchronous circuit design in order to reduce the system processing activity.

Most of the real life signals like speech, seismic, Doppler sampled signal obtained with the LCSS are defined by Equation 1.

\[
x(t) = \begin{cases} x(t) & \text{if } x(t) \geq 0 \\
-x(t) & \text{if } x(t) < 0
\end{cases}
\]

According to [1], the sampling instants of a non-uniformly spaced out in time sampled data obtained with the LCSS have been developed [2, 4, 18]. Algorithms for processing [3, 11, 13] and analysis [8, 12, 19] of the non-uniformly spaced out in time sampled data obtained with the LCSS have also been developed.

II. LCSS (LEVEL CROSSING SAMPLING SCHEME)

In the case of LCSS, a sample is captured only when the input analog signal crosses one of the predefined threshold levels [1]. The samples are not uniformly spaced in time because they depend on \( x(t) \) variations as it is clear from Fig. 1. Thus, the non-uniformity in the sampling process reflects the local characteristics of \( x(t) \) [12].

According to [1], the sampling instants of a non-uniformly sampled signal obtained with the LCSS are defined by Equation 1.

\[
t_n = t_{n-1} + d_{t_n} \quad (1)
\]

Where \( t_n \) is the current sampling instant, \( t_{n-1} \) is the previous one and \( d_{t_n} \) is the time delay between the current and the previous sampling instants (cf. Equation 2).

Fig. 1 Level-crossing sampling scheme

III. PROPOSED ADAPTIVE RATE FILTERING TECHNIQUE

The activity selection and local features extraction [8] is the base of the proposed technique. It makes to achieve the

\[
d_{t_n} = t_n - t_{n-1} \quad (2)
\]
adaptive rate sampling (only relevant number of samples to process) along with adaptive rate filtering (only relevant number of operations to deliver filtered sample). These achievements assure a drastic computational gain of the proposed filtering technique compared to the classical one. The approaches to realize it are detailed in the following subsections.

The block diagram of the proposed filtering technique is shown in Fig. 2. The description of different blocks is given in the following subsections. This technique is splitted into two filtering cases, which are explained in Section III-D.

**A. AADC (Asynchronous Analog to Digital Converter) and ASA (Activity Selection Algorithm)**

The AADC [2], is employed for digitizing \( x(t) \). An \( M \)-bit resolution AADC has \( 2^M \cdot 1 \) quantization levels which are uniformly disposed according to \( x(t) \) amplitude dynamics. The AADC has a finite bandwidth. Thus, to assure a proper signal capturing a band pass filter with pass-band \([f_{\text{min}}, f_{\text{max}}]\)

is employed at the AADC input.

Reconstruction issue of the non-uniformly sampled signal has been discussed in [7, 14]. In [14], author showed that a bandlimited signal can be ideally reconstructed from its non-uniformly spaced samples, provided that the average number of samples satisfies the Nyquist criterion. In the case of AADC, the number of samples is directly influenced by \( M \) and the signal characteristics [2, 4, 18]. Thus, for a given application an appropriate \( M \) should be chosen in order to respect the reconstruction criterion [14].

Let \( AV_{\text{in}} \) and \( AX(t) \) be the AADC and \( x(t) \) amplitude dynamics respectively. In order to avail the complete AADC resolution in the studied case, \( AX(t) \) is always adapted to match \( AV_{\text{in}} \). For a given \( M \), the AADC maximum sampling frequency \( F_{\text{sam}} \) and minimum sampling frequency \( F_{\text{sam}} \) [11] are defined by Equations 3 and 4 respectively. Where, \( f_{\text{max}} \) and \( f_{\text{min}} \) are the bandwidth and the fundamental (lowest) frequencies of \( x(t) \) respectively.

\[
F_{\text{sam}} = 2f_{\text{max}}(2^M - 1). \quad \text{(3)}
\]

The relevant (active) parts of the non-uniformly sampled signal are selected and windowed by the ASA. The complete procedure of activity selection has been explained in [8]. The ASA displays interesting features with the LCSS, which are not available in the classical case. It selects only the interesting parts of the non-uniformly sampled signal obtained with the AADC. Moreover, it correlates the length of the selected window with the signal activity, lies in it. In addition, it also provides an efficient reduction of the phenomenon of spectral leakage in the case of transient signals. This is done by avoiding the signal truncation with a simple and efficient algorithm instead of a smoothening window function, which is used in the classical scheme [8].

**B. Adaptive Rate Sampling**

For a fixed \( M \), the temporal density of the AADC sampling operation is a function of the input signal variations [11, 13]. Let \( F_s' \) represents the AADC sampling frequency for the \( i \)-th selected window. \( F_s' \) can be specific for each selected window, depending upon the window length \( T_s' \) in seconds and the slope of \( x(t) \) part lying within this window [8]. \( F_s' \) can be calculated by using the following equations.

\[
T_s' = t_{\text{max}} - t_{\text{min}}. \quad \text{(5)}
\]

\[
F_s' = N' / T_s'. \quad \text{(6)}
\]

In Equation 5, \( t_{\text{max}} \) and \( t_{\text{min}} \) are the final and the initial times of the \( i \)-th selected window. The upper and the lower bounds on \( F_s' \) are posed by \( F_{\text{sam}} \) and \( F_{\text{sam}}' \) respectively.

**C. Adaptive Rate Resampling**

The non-uniformly sampled signal obtained with the AADC can be used directly for further non-uniform digital processing [3, 12]. However in the studied case, the AADC output is first selected with the ASA and then resampled uniformly before proceeding towards further processing. It enables to take advantage of both non-uniform and uniform signal processing tools [8, 11, 13]. An additional error occurs due to this resampling. Nevertheless, prior to this transformation, one can take advantage of the inherent over-sampling of the relevant signal parts in the system [11, 13]. Hence it improves the accuracy of the post resampling process [4].

The resampling process requires interpolation, which changes the properties of the resampled signal compared to the original one. The properties of the resampled signal depend upon the interpolation technique used to resample it [9, 10]. The NNRI (nearest neighbour resampling interpolation) is employed for data resampling. The reasons of inclination towards NNRI are discussed in [8, 9, 10].

A study on the interpolation error is made by taking into account an academic signal, known formally. If \( (x_n, t_n) \) represents the time-amplitude pairs of the sampled signal \( x(t) \), as the input signal is analytically known. Now the interpolation error per interpolated point \( e_n \) is given by the absolute difference between \( x_n \) and \( x_{n+1} \). The process is given by Equation 7. The mean interpolation error for the \( i \)-th selected window \( MIE' \) can be calculated by using Equation 8. In Equation 8, \( N_i' \) is the number of resampled data points lie in the \( i \)-th selected window.

\[
e_n = |x_n - x_{n+1}|. \quad \text{(7)}
\]

\[
MIE' = \frac{1}{N_i'} \sum_{n=1}^{N_i'} e_n. \quad \text{(8)}
\]

The interpolation error can be reduced by making the interpolation interval as symmetrical as possible [6], by having equal number of samples on either side of the interpolation instant. This type of interpolation is also known as the central point interpolation. In order to achieve the symmetry in the interpolation interval, a resampling frequency for the \( i \)-th selected window \( F_r's' \), is calculated by employing Equation 9.
Where, \( AF \) is the adjustment factor, which is chosen by respecting the following criteria.

* \( Frs^i \) should maintain the symmetry in the interpolation process.
* \( Frs^i \) should be greater than and closest to \( F_{Nyq}^i \).

Here, \( F_{Nyq}^i = 2f_{max}^i \) and \( f_{max}^i \) represents the local bandwidth of the \( l^i \) selected window \([11, 13]\). The \( i^\text{th} \) criterion adds to the system accuracy by reducing the interpolation error. The \( 2^\text{nd} \) criterion keeps the system computationally efficient, firstly by avoiding the unnecessary interpolations during the resampling process and secondly by avoiding the processing of unnecessary samples during the filtering operation.

In this article the performance of the proposed filtering technique in terms of accuracy and quality is studied by employing a modelled input, consists of different frequency sinusoids. For a monotone sinusoid \( f_{max}^i = f^i \), \( f^i \) is the frequency of the sinusoid lie in the \( l^i \) selected window. In this case, \( Frs^i \) can be calculated by employing Equation 10. Where, \( NRC \) represents the relevant number of level-crossing thresholds crossed by the sinusoid.

\[
Frs^i = Frs^i / AF. \quad (9)
\]

\[
Frs^i = 2f^i(NRC). \quad (10)
\]

In the studied case \( \Delta x(t) = AV_{in} \), so \( NRC \) becomes equal to \( 2^{M-1} \). Thus, \( Frs^i \) can be expressed mathematically by Equation 11.

\[
Frs^i = 2f^i . (2^M - 1) . \quad (11)
\]

As \( \Delta x(t) = AV_{in} \), because it makes the relevant signal part to cross all thresholds of the AADC, therefore it is locally oversampled in time. This statement is further illustrated with the results summarized in Table II. Hence, there is no danger of aliasing, when the low frequency relevant signal parts are locally over-sampled in time at overall sub-Nyquist frequencies.

D. 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 processing load. This drawback has been resolved up to a certain extent by employing the multirate filtering techniques \([15, 16]\). They achieve computational efficiency which is not attainable with the classical approach.

The proposed filtering technique is a smart alternative of the multirate filtering techniques. It adapts the sampling frequency and the filter order by following the input signal local variations, which leads to a drastic computational gain of the proposed technique over the classical one.

The idea is to offline design a reference filter for a reference sampling frequency \( F_{ref} \), which satisfies the \( AF \) criteria, discussed in Section III-C. In the studied case \( F_{ref} \) is chosen as: \( F_{ref} = 4f_{max} \). The reference filter impulse response \( h_k \) is sampled at \( F_{ref} \) during offline processing. Here \( k \) is the index of the reference filter coefficients.

\( Frs^i \) can be equal to or less than \( F_{ref} \), depending upon \( Frs^i \). In order to perform a proper filtering operation \( Frs^i \) should match to \( F_{ref} \). The process of keeping them coherent leads towards the following two filtering cases.

Filtering Case 1

This case is true if \( Frs^i = F_{ref} \). For this case, the filtering decision for the \( l^i \) selected window \( FDi \) is set to \( 0 \) and it drives the switch to state \( 0 \), in Fig. 2. The reference filter impulse response \( h_k \) remains unaltered for this case.

Filtering Case 2

This case is true if \( Frs^i < F_{ref} \). For this case, \( FDi \) is set to \( 1 \) and it drives the switch to state \( 1 \), in Fig. 2.

In this case, it appears that the data lies in the \( l^i \) selected window can be resampled at a frequency which is less than the Nyquist frequency of \( x(t) \) and so it can cause aliasing. Since the AADC sampling frequency varies according to the slope of \( x(t) \) \([2]\). A high frequency signal part has a high slope and the AADC samples it at a higher rate and vice versa. Hence, a signal part with only low frequency components can be sampled by the AADC at a sub-Nyquist frequency of \( x(t) \). But still this signal part is locally over sampled in time with respect to its local bandwidth \([11, 13]\). It is valid as far as \( \Delta x(t) = AV_{in} \), because it makes the relevant signal part to cross all thresholds of the AADC, therefore it is locally oversampled in time. This statement is further illustrated with the results summarized in Table II. Hence, there is no danger of aliasing, when the low frequency relevant signal parts are locally over-sampled in time at overall sub-Nyquist frequencies.

As in order to perform a proper filtering operation \( Frs^i \) should match to \( F_{ref} \) so in filtering case 2, \( h_k \) is online decimated as a function of \( Frs^i \) for the \( l^i \) selected window. The decimation factor for the \( l^i \) selected window \( D \) can be specific depending upon \( Frs^i \). The process of calculating \( D \) and decimating \( h_k \) for the integral and the fractional \( D \) is shown in Fig. 3.

\[
D = F_{ref} / Frs^i
\]

\[
h_{j} = \text{Resample}(h_k @ Frs^i)
\]

Fig. 3 Flowchart of calculating \( D \) and decimating \( h_k \).

In Fig. 3 \( hd_j \) represents the decimated filter for the \( l^i \) selected window. Here \( j \) represents the index of the decimated filter coefficients. For integral \( D \), \( hd_j \) is obtained by picking every \( (D)^{j^{th}} \) coefficient from \( h_k \). For fractional \( D \), the fractional decimation is achieved by resampling \( h_k \) at \( Frs^i \). The NNRI is employed to resample \( h_k \). If the order of \( h_k \) is \( H \) then the order of \( hd_j \) is \( P = H/D \).

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. \( D \) is a good approximate of the ratio between the energy of the reference filter and that of the decimated filter. Thus, this effect of decimation is compensated by scaling \( hd_j \) with \( D \). The proc-
process of scaling $hD$ in order to obtain the decimated and the scaled impulse response $hv$ for the $i$th selected window is given by Equation 13.

$$hv = hD \times D'$$ (13)

IV. ILLUSTRATIVE EXAMPLE

In order to illustrate the proposed filtering technique a modelled signal $x(t)$, shown on the left part of Fig. 4 is employed. Its total duration is 20 seconds and it consists of three active parts. Each activity is a sinusoid of amplitude 0.9v and of frequency 1100, 200 and 350 Hz respectively. The time span of each activity is 0.5, 1.0 and 1.0 second respectively.

As $x(t)$ is band limited up to 1.1 kHz so the value of $F_{ref}$ becomes 4.4 kHz in this case. $x(t)$ is sampled by employing a 3-bit resolution AADC, it results into $F_{s_{max}}=15.4$ kHz and $F_{s_{min}}=2.8$ kHz respectively (c.f. Equations 3 and 4). $AF$ becomes 3.5 in this case. $\Delta V_o$ is set to 1.8v, thus AADC quantum $q=\Delta V_o/(2^M-1)$ becomes 0.2571v.

![Fig. 4 Input signal (left) and the selected signal (right)](image)

The $i^{th}$ and the $3^{rd}$ active parts of $x(t)$ are of higher frequencies compared to the $2^{nd}$ one. In order to filter out the high frequency components from $x(t)$ a low pass reference filter is designed by using standard Parks-McClellan algorithm. The filter parameters are summarized in Table I. The purpose of this filtering process is to demonstrate that how the proposed filtering technique adapts its processing load by following the local variations of $x(t)$.

**TABLE I**

<table>
<thead>
<tr>
<th>Cut-off Freq (Hz)</th>
<th>Transition Band (Hz)</th>
<th>Pass Band Ripples</th>
<th>Stop Band Ripples</th>
<th>$F_{s_{ref}}$ (kHz)</th>
<th>$R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>250~310</td>
<td>-25 (dB)</td>
<td>80 (dB)</td>
<td>4.4</td>
<td>186</td>
</tr>
</tbody>
</table>

In order to apply the ASA, the reference window length $T_{ref} = 1$ second is chosen for this studied example. The selected signal obtained with the ASA is shown on the right part of Fig. 4. The parameters of each selected window are summarized in Tables II and III.

**TABLE II**

<table>
<thead>
<tr>
<th>Selected Window</th>
<th>$T_{ref}$ (Sec)</th>
<th>$F_{s_{ref}}$ (kHz)</th>
<th>$N_r$ (sn)</th>
<th>$OSR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^{st}$</td>
<td>0.4995</td>
<td>15.4</td>
<td>7693</td>
<td>4.4</td>
</tr>
<tr>
<td>$2^{nd}$</td>
<td>0.9994</td>
<td>2.8</td>
<td>2799</td>
<td>4.4</td>
</tr>
<tr>
<td>$3^{rd}$</td>
<td>0.9995</td>
<td>4.9</td>
<td>4898</td>
<td>4.4</td>
</tr>
</tbody>
</table>

**TABLE III**

<table>
<thead>
<tr>
<th>Selected Window</th>
<th>$F_{s_{ref}}$ (kHz)</th>
<th>$N_r$ (sn)</th>
<th>$OSR$</th>
<th>$D$</th>
<th>$P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^{st}$</td>
<td>4.4</td>
<td>2198</td>
<td>7</td>
<td>1</td>
<td>186</td>
</tr>
<tr>
<td>$2^{nd}$</td>
<td>0.8</td>
<td>800</td>
<td>7</td>
<td>5.5</td>
<td>33</td>
</tr>
<tr>
<td>$3^{rd}$</td>
<td>1.4</td>
<td>1400</td>
<td>7</td>
<td>3.1</td>
<td>60</td>
</tr>
</tbody>
</table>

**TABLE IV**

<table>
<thead>
<tr>
<th>Summary of Reference Filter Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{s_{ref}}$ (kHz)</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>$1^{st}$</td>
</tr>
<tr>
<td>$2^{nd}$</td>
</tr>
<tr>
<td>$3^{rd}$</td>
</tr>
</tbody>
</table>

In the proposed filtering technique, the sampling frequency and the filter order both are not fixed and are adapted for each selected window according to the local variations of $x(t)$. In comparison to the classical case this approach locally requires some extra operations for each selected window.

**V. COMPUTATIONAL COMPLEXITY**

This section compares the computational complexity of the proposed filtering technique with the classical one. The complexity evaluation is made by considering the number of online operations executed to perform the algorithm.

In the classical case, $F_{s}$ and $H$ remains constant, regardless of the input signal local variations. It is known that for an $R$ order FIR filter, $H$ multiplications and $2R$ additions are computed for each output sample. The total computational complexity $C_i$ for $N$ samples can be calculated by employing Equation 14.

$$C_i = H \times N + \frac{H \times N}{2R}$$ (14)

In the proposed filtering technique, $F_{s_{ref}}$ requires a division between $F_{s}$ and $AF$ (c.f. Equation 12). The selected data lie in the $i^{th}$ selected window is resampled uniformly at $F_{s_{ref}}$ before filtering. The NNRI is employed for the resampling purpose. The NNRI requires only a comparison operation for each resampled observation. Therefore the interpolator performs $N_r$ comparisons. Moreover, the filtering case selection requires a com-
parison between \( F_{ref} \) and \( F_{rs} \). If \( D' = 0 \) then the reference filter impulse response \( h_i \) remains the same. Otherwise, the decimation of \( h_i \) is required. \( D' \) is calculated for this purpose. The calculation of \( D' \) requires one division (cf. Fig. 3). For an integral \( D' \) the decimation of \( h_i \) is simple and has a negligible complexity as compared to operations like addition or multiplication. This is the reason that for this case the complexity of the decimator is not taken into consideration, during the complexity evaluation. For a fractional \( D' \), it is required to resample \( h_i \) at \( F_{rs} \), which perform \( P \) comparisons. Here, \( P \) is the order of decimated filter for the \( i^{th} \) selected window. The filter coefficients scalar performs \( P \) multiplications (cf. Equation 16). Finally a \( P \) order filter, performs \( P.Nr^2 \) multiplications and \( P.Nr^2 \) additions for the \( i^{th} \) selected window. The processes of designing the reference filter is performed offline, so is not included in the online algorithm complexity calculation. The combined computational complexity \( C_z \) of the proposed filtering technique is given by Equation 15.

\[
C_z = L(\alpha + \beta) + \sum_{i=1}^{L} \left( Nr^2 - \alpha B^P + 1\right) + P.Nr^2 + \left( P.Nr^2 + \alpha B^P\right). \quad (15)
\]

In Equation 18, \( i = 1, 2, ..., L \) represents the index of selected window. \( \alpha \) is a multiplying factor and its value is 0 for \( F_{rs} = F_{ref} \) and 1 otherwise. \( \beta \) is also a multiplying factor, it is 0 for an integral \( D' \) and 1 for a fractional \( D' \).

From \( C_1 \) and \( C_z \), it is clear that there are uncommon operations between both filtering techniques. In order to make them approximately comparable the following assumptions are made.

*A comparison has same processing cost as that of an addition.

*A division has same processing cost as that of a multiplication.

By following these assumptions, comparisons are merged into additions count and divisions are merged into multiplications count, during the complexity evaluation process. The computational gain of the proposed filtering technique over the classical one is calculated for results of the illustrative example, for different time spans of \( x(t) \). The results are summarized in Table IV.

<table>
<thead>
<tr>
<th>TABLE IV SUMMARY OF THE COMPUTATIONAL GAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Span (Sec)</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>( T_x )</td>
</tr>
<tr>
<td>( T_{x'} )</td>
</tr>
<tr>
<td>( T_{x''} )</td>
</tr>
<tr>
<td>Total signal span (20)</td>
</tr>
</tbody>
</table>

Table IV shows 14.55 and 14.61 times gain in additions and multiplications respectively for the total \( x(t) \) span of 20 seconds. It shows that the proposed filtering technique leads towards a drastic reduction in the number of operations compared to the classical one. It is achieved by adapting the sampling frequency and the filter order by following the input signal local variations.

It is clear that by knowing \( f \) the choice of \( F_{rs} \) \( \geq 2 \). \( f \) is enough, but in order to achieve symmetric interpolation \( F_{rs} = 4 \). \( f \) is chosen, in the studied case. Although this choice of \( F_{rs} \) affects the proposed technique computational efficiency, yet it can be accepted to achieve a significant processing accuracy (cf. Section VI).

VI. PROCESSING ERROR

A. Interpolation Error

The interpolated results are calculated by employing the level-crossing samples. For the practical AADC there exist uncertainties in time-amplitude pairs of the level-crossing samples [2]. These uncertainties accumulate in the interpolation process and cause to increase the error [4]. Therefore in order to have a fair idea of the interpolation error an ideal AADC is employed, which provides the exact time-amplitude pairs of the level-crossing samples.

For the studied example, \( AF = 3.5 \) leads to the interpolation symmetry (cf. Section III-C). In order to observe the effect of symmetry on the interpolation error, values of \( MIE \) are calculated by varying \( AF \) around 3.5. The process of calculating \( MIE \) is clear from Equation 8. The values of \( AF \) are chosen in such a way that \( F_{rs} \) remains greater than \( F_{\text{sys}} \) for all \( AFs \). The results are plotted in Fig. 5.

Fig. 5 shows that a drastic reduction in the interpolation error can be achieved by availing the symmetry during the interpolation process. Interpolation symmetry may lead to some latencies in terms of the real time implementation and the computational efficiency. However depending on the application requirements, these latencies can be tolerated for the benefit of maintaining a small error.

B. Filtering Error

In the proposed filtering technique a reference filter \( h_i \) is employed and then it is adapted for the \( i^{th} \) selected window, depending upon the chosen \( F_{rs} \). It makes the sampling and filtering of the \( i^{th} \) activity at \( F_{rs} \). In order to calculate the relative filtering error, of the proposed technique with the classical one, the following procedure is adopted.

In the classical case, instead of adapting \( h_i \) to obtain \( h'_i \), a specific filter is designed for the \( i^{th} \) selected window, by employing the Parks-McClellan algorithm. It is designed for \( F_{rs} \) by employing the same design parameters, summarized in Table I. The \( i^{th} \) activity is sampled at \( F_{rs} \) and is filtered by employing the designed filter. The filtered signal obtained in this case is used as a reference and is compared with the one, obtained with the proposed technique.

Let \( y_{x'} \) be the \( n^{th} \) reference filtered sample and \( y_{x''} \) the \( n^{th} \) filtered sample obtained with the proposed filtering technique. Then the relative filtering error for the \( n^{th} \) filtered point can be calculated by using Equation 16. The mean filtering error for the \( i^{th} \) activity \( MIE \) is calculated by employing Equation 17.
An adaptive rate filtering technique is devised. This technique is well suited for the low activity sporadic signals like electro-cardiogram, phonocardiogram, seismic signals, etc. A reference filter is offline designed by taking into account the AF criteria. A complete methodology to obtain \( F_{Rs} \) and decimating \( h_k \) for the \( \hat{D} \) selected window has been demonstrated. It is shown that how the data resampling and the reference filter order are smartly adapted for the \( \hat{D} \) selected window, by following the input signal local variations. The computational gain of the proposed adaptive rate filtering technique is shown over the classical one. It is achieved due to the joint benefits of the AADC, the ASA and the resampling as they enable to adapt the \( F_{Rs} \), \( F_{Rs} \), \( N_k \), \( N_k \), \( \hat{D} \) and \( \hat{P} \) by exploiting the input signal local variations.

The interpolation error is calculated. It is shown that a drastic reduction in the interpolation error can be achieved by employing the symmetry in the interpolation process. A method to calculate the relative filtering error of the proposed technique with respect to the classical one is also proposed. It is shown that the result obtained with the proposed filtering technique is of comparable quality to the classical one. The reference filter decimation is required in the proposed technique. The complete procedure of decimating and scaling the pre-calculated reference filter during online computation is demonstrated. This online decimation reduces the quality of the decimated filter as compare to the reference one. The upper bound on the decimation factor can be determined by offline calculations, for which the decimated and scaled filter provides a response with an acceptable level of accuracy. Moreover, for high precision applications, an appropriate filter can be directly calculated online for each selected window at the cost of an increased computational load.

A detailed study of the computational complexity of the proposed filtering technique by taking into account the real processing cost at circuit level is in progress. Further research focuses on the optimization of the proposed techniques and on its performance study in the case of real life applications.

REFERENCES