World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:5, 2008

Computationally Efficient Adaptive Rate Sampling and Adaptive Resolution Analysis

Saeed Mian Qaisar, Laurent Fesquet, and Marc Renaudin

***Abstract*—**Mostly the real life signals are time varying in nature. For proper characterization of such signals, time-frequency representation is required. The STFT (short-time Fourier transform) is a classical tool used for this purpose. The limitation of the STFT is its fixed time-frequency resolution. Thus, an enhanced version of the STFT, which is based on the cross-level sampling, is devised. It can adapt the sampling frequency and the window function length by following the input signal local variations. Therefore, it provides an adaptive resolution time-frequency representation of the input. The computational complexity of the proposed STFT is deduced and compared to the classical one. The results show a significant gain of the computational efficiency and hence of the processing power. The processing error of the proposed technique is also discussed.

***Keywords*—**Level Crossing Sampling, Activity Selection, Adaptive Resolution Analysis, Computational Complexity.

I. Context of the Study

THIS work is a contribution in the development of smart mobile systems. The goal is to reduce their cost, size, 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 clock less circuit design in order to reduce the system activity.

Almost all natural signals like speech, seismic and biological signals are of non stationary nature. Moreover the man made signals like Doppler, ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying) etc. also lie in the same category. The spectral contents of these signals vary with time, which is a direct consequence of the signal generation process [5].

Classical systems are based on the Nyquist signal processing architectures. They cannot sense the input signal local variations and therefore they process it at a fixed pace. Thus, in case of low activity sporadic signals like electrocardiogram, phonocardiogram, seismic signals etc. they produce a large number of useless samples without any relevant information. It causes a useless increase in the system activity and so a useless increase of the power consumption.

Saeed Mian Qaisar is a PhD candidate in Laboratory TIMA, CNRS UMR 5159, 46 Avenue Felix-Viallet, 38031 Grenoble Cedex, France (phone: +33-476574646; fax: +33-476574981; e-mail: saeed.mian-qaisar@imag.fr).

Laurent Fesquet is an associate professor at INPG and is working with Laboratory TIMA, CNRS UMR 5159, 46 Avenue Felix-Viallet, 38031 Grenoble Cedex, France (e-mail: laurent.fesquet@imag.fr).

Marc Renaudin is a professor at INPG and is working with Laboratory TIMA, CNRS UMR 5159, 46 Avenue Felix-Viallet, 38031 Grenoble Cedex, France (e-mail: marc.renaudin@imag.fr).

The power efficiency can be achieved by smartly adapting the system computational load according to the input signal local variations. In this context, a signal driven sampling scheme, which is based on "level-crossing" is employed. The LCSS (Level Crossing Sampling Scheme) [1] adapts the sampling rate by following the local characteristics of the input signal [12, 16]. Hence, it drastically reduces the activity of the post processing chain, because it only captures the relevant information [11, 13]. In this context, analog to digital converters based on the LCSS have been developed [2, 4, 17]. Algorithms for processing [3, 11, 13] and analysis [8, 12] of the non-uniformly spaced out in time sampled data obtained with the LCSS have also been developed.

The focus of this work is to achieve a smart time-frequency representation of the time varying signals. The idea is to adapt the time-frequency resolution along with the computational load by following the input signal local variations. An efficient solution is proposed by smartly combining the features of both uniform and non-uniform signal processing tools.

II. LCSS (Level Crossing Sampling Scheme)

In the case of LCSS, a sample is captured only when the input analog signal $x(t)$ 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 signal local characteristics [12].

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

$$t_n = t_{n-1} + dt_n \,. \tag{1}$$

$$dt_n = t_n - t_{n-1} \,. \tag{2}$$

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

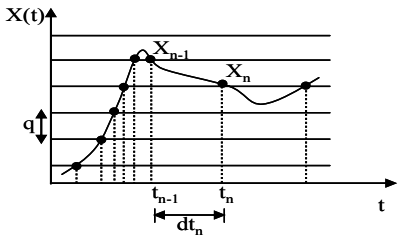

Fig. 1 Level crossing sampling scheme

III. Proposed Adaptive Resolution STFT

The block diagram of the proposed STFT is shown in Fig. 2. The description of different blocks is given in the following subsections.

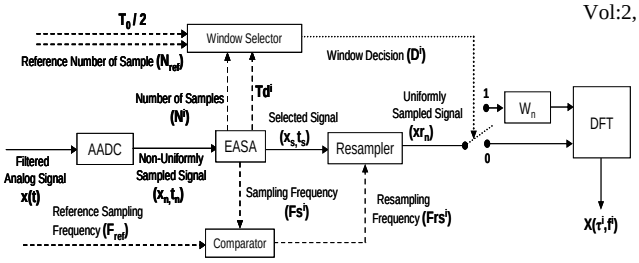

Fig. 2 Block diagram of the proposed STFT. '——' represents the signal flow, '·····' represents the control flow and '-----' represents the parameters flow at system different stages

A. AADC (Asynchronous Analog to Digital Converter)

The AADC [2], is employed for digitizing *x(t)*. An *M-bit* resolution AADC has $2^M - 1$ quantization levels which are disposed according to *x(t)* amplitude dynamics. In the studied case, the levels are uniformly spaced.

Let δ be the AADC processing delay for one sample. Thus for proper signal capturing the *x(t)* must satisfy the tracking condition [2], given by Expression 3, where *q* is the AADC quantum, which is defined by Equation 4. In Equation 4, ΔV_{in} and *M* represent the amplitude dynamics and the resolution of the AADC respectively.

$$\frac{dx(t)}{dt} \le \frac{q}{\delta} \quad (3) \qquad q = \frac{\Delta V_{in}}{2^M - 1} \quad (4) \qquad \frac{dx(t)}{dt} \le 2.\pi.\Delta x(t).f_{\max} \quad (5)$$

The left hand side of Expression 3 is the slope of *x(t)*. The upper bound on the slope of a band limited signal is defined by the Bernstein's inequality [2], given by Expression 5. In Expression 5, f_{max} and $\Delta x(t)$ represent the bandwidth and the amplitude dynamics of the *x(t)* respectively. Thus, In order to respect the tracking condition, a band pass filter with pass-band $[f_{min;} f_{max}]$ is employed at the AADC input.

In order to avail the complete AADC resolution, in the studied case, $\Delta x(t)$ is always adapted to match ΔV_{in}. For an *M-bit* resolution AADC, the maximum and the minimum sampling frequencies [11, 13] are defined by Equations 6 and 7 respectively.

$$Fs_{max} = 2.f_{max}.(2^M\text{-}1). \quad (6) \qquad Fs_{min} = 2.f_{min}.(2^M\text{-}1). \quad (7)$$

Where, f_{min} is the fundamental (lowest) frequency of *x(t)*. Fs_{max} and Fs_{min} are the maximum and the minimum sampling frequencies of the AADC.

B. EASA (Enhanced Activity Selection Algorithm) and Window Selector

The EASA is an improved version of the ASA [8]. The relevant (active) parts of the non-uniformly sampled signal obtained with the AADC are selected by the EASA. This selection process corresponds to the adaptive length rectangular windowing process.

The main difference between the ASA and the EASA is the choice of the upper bound on the selected window length. For the ASA the time length in seconds and for the EASA the number of samples is chosen as the upper bound. The EASA is defined as follow.

```
While (dtn ≤ T0/2 and Ni ≤ Nref)
      Ni = Ni + 1;
end
```

Where, $T_0 = 1/f_{min}$ is the fundamental period of *x(t)*. T_0 and dt_n detect parts of the non-uniformly sampled signal with activity. The condition on dt_n is chosen in order to satisfy the Nyquist sampling criterion for f_{min}, when sampling *x(t)* non-uniformly with the AADC. N^i represents the number of non-uniform samples lie in the i^{th} selected window, which lie on the j^{th} active part of the non-uniformly sampled signal. Where, *i* and *j* both belong to the set of natural numbers $\mathbb{N}^*$. N_{ref} represents the upper bound on N^i. The choice of N_{ref} depends on the *x(t)* characteristics and the system parameters. 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 is initialized to zero.

The EASA displays interesting features with the LCSS, which are not available in the classical case. It selects only the active parts of the non-uniformly sampled signal, obtained at the AADC output. Moreover, it correlates the length of the selected window with the input signal local characteristics.

The window selector implements the condition given by Expression 8. The output of window selector is the window decision D^i, which drives the switch state for the i^{th} selected window (cf. Fig. 2). Jointly, the EASA and the window selector, provide an efficient spectral leakage reduction in the case of transient signals. Indeed, spectral leakage occurs due to the signal truncation problem, which causes to process the non integral number of cycles in the observation interval. Usually an appropriate smoothening (cosine) window function is employed to reduce the signal truncation, in the classical case. For the proposed case, as long as the condition 8 is true, the leakage problem is resolved by avoiding the signal truncation [8]. As no signal truncation occurs so no cosine window is required. In this case, D^i is set to *0*, which drives the switch to state *0* in Fig. 2. Otherwise an appropriate cosine window is employed to reduce the signal truncation problem. In this case, D^i is set to *1*, which drives the switch to state *1* in Fig. 2.

$$if\,(N^i \le N_{ref} \quad and \quad (Td^i = t_1^i - t_{end}^{i-1}) > \frac{T_0}{2}) \cdot \qquad (8)$$

In expression 8, t_1^i represents the 1^{st} sampling instant of the i^{th} selected window and t_{end}^{i-1} represents the last sampling instant of the $(i\text{-}1)^{th}$ selected window.

For proper spectral representation, the condition given by Expression 9 should be satisfied [8]. Where, L^i is the length in seconds of the i^{th} selected window. In order to satisfy this condition for the worst case, which occurs for Fs_{max}, N_{ref} is calculated for an appropriate reference window length L_{ref}. L_{ref} has to satisfy the condition: $L_{ref} \ge T_0$. The process of calculating N_{ref} is given by Equation 10.

$$L^i \ge T_0 \,. \quad (9) \qquad N_{ref} = L_{ref} \,.\, Fs_{max} \,. \qquad (10)$$

The lower and the upper bounds on L_{ref} are posed respectively by T_0 and the system resources (the maximum sample frame which system can process at once). For N_{ref} (cf. Equation 10), the condition 9 holds for all selected windows except for the case when the actual length of the j^{th} activity is less than T_0.

C. Adaptive Rate Sampling

The AADC sampling frequency is correlated to *x(t)* local variations [8, 11]. Let Fs^i represents the AADC sampling frequency for the i^{th} selected window. Fs^i can be specific for each selected window, depending upon L^i and the slope of *x(t)* part lying within this window [8]. Fs^i can be calculated by using the following equations.

$$L^i = tmax^i - tmin^i \ . \ (11) \qquad Fs^i = N^i / L^i \ . \qquad (12)$$

In Equation 11, $tmax^i$ and $tmin^i$ are the final and the initial times of the i^{th} selected window. The upper and the lower bounds on Fs^i are posed by Fs_{max} and Fs_{min} respectively.

The selected data obtained with the EASA can be used directly for further non-uniform digital processing [3, 12]. However in the studied case, the selected data is resampled uniformly. It enables to take advantage of both non-uniform and uniform signal processing tools [8, 11, 13]. Due to this resampling, there will be an additional error. 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 adds to the accuracy of the post resampling process [4]. The NNRI (nearest neighbour resampling interpolation) is employed for data resampling. It is a simple interpolation method as it employs only one non-uniform observation for each resampled observation. Thus, it is efficient in terms of the computational complexity. Moreover, it provides an unbiased estimate of the variance of the original signal, due to this reason it is also known as a robust interpolation method [9, 10]. The detailed reasons of inclination towards NNRI are discussed in [8, 9, 10].

The interpolation process 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].

A study on the interpolation error is made by taking into account an academic signal, known formally. If (tr_n, xr_n) represent the time-amplitude pairs of the n^{th} interpolated sample. Then the original sample amplitude xo_n can be mathematically calculated for tr_n, as the input signal is analytically known. Now the interpolation error per interpolated point Ie_n is given by the absolute difference between xo_n and xr_n. The process is given by Equation 13. The mean interpolation error for the i^{th} selected window MIe^i is given by Equation 14.

$$Ie_n = \left| xo_n - xr_n \right| \ . \ (13) \qquad MIe^{\,i} = \frac{1}{Nr^{\,i}} . \sum_{n=1}^{Nr^{\,i}} Ie_n \ . \qquad (14)$$

In Equation 14, Nr^i is the number of resampled data points lie in the i^{th} selected window.

A reference sampling frequency F_{ref} is chosen such as it remains greater than and closest to the $F_{Nyq}=2.f_{max}$. Depending upon the values of F_{ref} and Fs^i the resampling frequency for the i^{th} selected window Frs^i, can be adapted (cf. Fig. 2).

For the case, $Fs^i > F_{ref}$, Frs^i is chosen as: $Frs^i = F_{ref}$. It is done in order to resample the selected data, lie in the i^{th} selected window closer to the Nyquist frequency. It avoids the unnecessary interpolations during the data resampling process and so reduces the computational load of the proposed technique.

For the case, $Fs^i \leq F_{ref}$, Frs^i is chosen as: $Frs^i = Fs^i$. In this case, it appears that the data lie in the i^{th} selected window may be resampled at a frequency, which is less than the Nyquist frequency of *x(t)* and so it can cause aliasing. Since, the sampling rate of the AADC 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 $\Delta x(t) = \Delta V_{in}$, because it makes the relevant signal part to cross all thresholds (more than one) of the AADC, so it is locally over-sampled 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 Resolution Analysis

The STFT is a classical tool, used for the time-frequency characterization of time varying signals [6]. The STFT of a sampled signal x_n is determined by computing the DFT (Discrete Fourier Transform) of an N samples segment centred on τ, which describes the spectral contents of x_n around the instant τ. Where, N is defined by Equation 15.

$$N = L \ . \ Fs \ . \qquad (15)$$

In Equation 15, L is the effective length in seconds of the window function w_n and Fs is the sampling frequency. The STFT can be expressed mathematically by Equation 16.

$$X[\tau, f] = \sum_{n=\tau-\frac{L}{2}}^{\tau+\frac{L}{2}} [x_n . w_{n-\tau}] . e^{-j.2\pi.f.n} \ . \qquad (16)$$

In Equation 16, f is the frequency index, which is normalised with respect to F_s.

L controls the STFT time and frequency resolution [6]. In the classical case, the input signal is sampled at a fixed sampling frequency Fs, regardless of its local variations. Thus, a fixed L results into a fixed N (cf. Equation 15). In the case, when the spectrum of each windowed block is calculated with respect to τ and no overlapping is performed between the consecutive blocks, then the time resolution Δt and the frequency resolution Δf of the STFT can be defined by Equations 17 and 18 respectively.

$$\Delta t = L \ . \qquad (17) \qquad \Delta f = Fs / N \ . \qquad (18)$$

Equation 18 shows that for a fixed Fs, Δf can be increased by increasing N. But increasing N requires to increase L which will reduce Δt (cf. Equation 17). Thus, a larger L provides better Δf but poor Δt and vice versa. This conflict between Δf and Δt shows the limitation of the STFT, which is the reason for the creation of the MRA (multi resolution analysis) techniques [7, 14]. The MRA techniques provide a good frequency but a poor time resolution for the low-frequency events and a good time but a poor frequency resolution for the high-frequency events. It is the type of analysis, best suited for most of the real life signals [7].

In this article, the fixed resolution dilemma is resolved to a certain extent by revising the STFT. The proposed STFT is a smart alternative of the MRA techniques. It performs adaptive time-frequency resolution analysis, which is not attainable with the classical STFT. It is achieved by adapting the Frs^i, L^i and Nr^i according to the local variations of *x(t)*. Thus, the time resolution Δt^i and the frequency resolution Δf^i of the proposed STFT can be specific for the i^{th} selected window and are defined by Equations 19 and 20 respectively.

$$\Delta t^i = L^i \ . \quad (19) \qquad \Delta f^i = Frs^i / Nr^i \ . \quad (20)$$

Because of this adaptive resolution, the proposed STFT will be named as the ARSTFT (adaptive resolution STFT), throughout the following parts of this article. This adaptive nature of the ARSTFT also leads towards a drastic computational gain, compared to the classical one. It is achieved firstly by avoiding the unnecessary samples to process and secondly by avoiding the use of the cosine window function as far as the condition 8 is true. The ARSTFT is defined by Equation 21.

$$X\left[\tau^i, f^i\right] = \sum_{n=\tau^i-\frac{Nr^i}{2}}^{\tau^i+\frac{Nr^i}{2}} \left[\{\mathrm{Re}\, sample\,(x_n, t_n)\}.w^i_{n-\tau^i}\right] e^{-j.2\pi.f^i.n} . \quad (21)$$

Where, τ^i and f^i are the central time and the frequency index of the i^{th} selected window respectively. f^i is normalised with respect to Frs^i. *n* is the index of the resampled data points lie in the i^{th} selected window. The notation w_n^i represents that the window function length L^i and shape (rectangle or cosine) can be adapted for the i^{th} selected window.

IV. Illustrative Example

In order to illustrate the ARSTFT an input signal *x(t)*, shown on the left part of Fig. 3 is employed. Its total duration is 30 seconds and it consists of four active parts. The summary of *x(t)* activities is given in Table I.

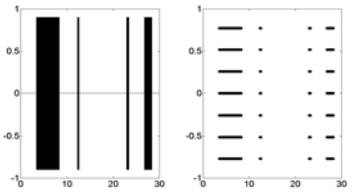

Fig. 3 Input signal (left) and the selected signal (right)

TABLE I
Summary of the Input Signal Activities

Activity	Signal Component	Length (Sec)
1st	0.9.sin(2.pi.50.t)	5
2nd	0.9.sin(2.pi.50.t)	0.4
3rd	0.9.sin(2.pi.200.t)	0.5
4th	0.9.sin(2.pi.500.t)	1.6

Table I shows that *x(t)* is band limited between 50 to 500 Hz. In this example *x(t)* is sampled by employing a 3-bit resolution AADC. Thus, Fs_{max} and Fs_{min} become 7 kHz and 0.7 kHz respectively (cf. Equations 6, 7). F_{ref} = 1.25 kHz is chosen, which satisfies the criteria given in Section III-C. ΔV_{in} = 1.8v is chosen, thus *q* becomes 0.2571v in this case (cf. Equation 4).

The selected signal obtained with the EASA is shown on the right part of Fig. 3. By following the criteria given in Section III-B, N_{ref} *=4096* is chosen, which leads to 6 selected windows. First three selected windows correspond to the first three activities and the remaining corresponds to the fourth activity. The last three selected windows are not distinguishable on the right part of Fig. 3, because they lie consecutively on the fourth activity. The parameters of each selected window are summarised in Table II.

TABLE II
Summary of Parameters of the Selected Windows

Selected Window	L^i (Sec)	Fs^i (kHz)	N^i (Smp)	F_{ref} (kHz)	Frs^i (kHz)	Nr^i (Smp)
1st	4.99	0.7	3500	1.25	0.7	3500
2nd	0.39	0.7	280	1.25	0.7	280
3rd	0.49	2.8	1400	1.25	1.25	625
4th	0.58	7.0	4096	1.25	1.25	731
5th	0.58	7.0	4096	1.25	1.25	731
6th	0.43	7.0	3005	1.25	1.25	536

Table II exhibits the interesting features of the ARSTFT, which are achieved due to the smart combination of the non-uniform and the uniform signal processing tools. 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 oversampled in time with respect to their local bandwidths. Frs^i shows the adaptation of the resampling frequency for the i^{th} selected window. It further adds to the computational gain of the ARSTFT, by avoiding the unnecessary interpolations during the resampling process. Nr^i shows that how the adjustment of Frs^i avoids the processing of unnecessary samples during the spectral computation (cf. Equation 21). L^i exhibits the dynamic feature of the EASA, which is to correlate the window function length with the local variations of *x(t)*. Adaptation of L^i, Frs^i *and* Nr^i leads to the adaptive time-frequency resolution of the ARSTFT, which is clear from the values of Δt^i and Δf^i in Table III.

TABLE III
Time and Frequency Resolution of the Selected Windows

Window	1st	2nd	3rd	4th	5th	6th
Δt^i (Sec)	4.99	0.39	0.49	0.58	0.58	0.43
Δf^i (Hz)	0.2	2.5	2.0	1.71	1.71	2.33

Table III demonstrates that ARSTFT adapts its time-frequency resolution by following the local variations of *x(t)*. It provides a good frequency but a poor time resolution for the low frequency parts of *x(t)* and vice versa. This type of analysis is best suited for most of the real life signals [7]. The spectrum of each selected window is computed and plotted with respect to τ_i on Fig. 4.

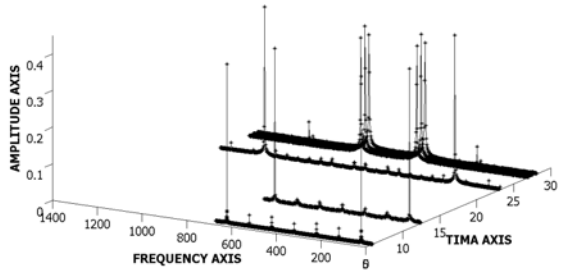

Fig. 4 The ARSTFT of the selected windows

Fig. 4 shows the fundamental and the periodic spectrum peaks of each selected window. The adaptation of Frs^i for the i^{th} selected window can be visualised on Fig. 4. In this case, the spectrum periodic frequency f_p^i is equal to Frs^i, for the i^{th}

selected window. As Fs^1 and Fs^2 both remain less than F_{ref}, so $Frs^1 = Fs^1$ and $Frs^2 = Fs^2$ are chosen. Contrary, Fs^3 to Fs^6 all become greater than F_{ref}, thus Frs^3 to Frs^6, all are chosen equal to F_{ref} (cf. Table II).

The ARSTFT also adapts the window shape (rectangle or cosine), for the i^{th} selected window. The condition 8 remains true for the first three selected windows, thus D^i is set to *0*. As no signal truncation occurs so no cosine window is required in this case. On the other hand, the number of samples for the fourth activity is 11200. Therefore, $N_{ref} = 4096$ leads to the three selected windows for the time span of the fourth activity of *x(t)*. The condition 8 becomes false, thus D^i is set to *1*. As signal truncation occurs, so suitable length cosine (Hanning) windows are employed to reduce this truncation effect.

In the classical case, if $Fs = F_{ref}$ is chosen, in order to satisfy the Nyquist sampling criterion for *x(t)*. Then the whole signal will be sampled at 1.25 kHz, regardless of its local variations. Moreover, the windowing process is not able to select only the active parts of the sampled signal. In addition, *L* remains static and is not able to adapt with the local variations of *x(t)*. This static nature makes the classical system to process unnecessary samples and so causes an increased computational activity than the proposed one. For this studied example, the fixed *N*=4096, will lead to nine fixed *L*=3.3 second windows, for the total *x(t)* time span of 30 seconds. It leads to the fixed $\Delta t = 3.3$ seconds and $\Delta f = 0.31$ Hz for all nine windows (cf. Equations 17 and 18).

V. Computational Complexity

This section compares the computational complexity of the ARSTFT with the classical STFT. The complexity evaluation is made by considering the number of operations executed to perform the algorithm.

In the classical case, the incoming signal is sampled at a fixed sampling frequency, regardless of its local variations. In this case, a time invariant, fixed *L*, cosine window function, is employed to window the sampled data. If *N* is the number of samples lie in the window then the windowing operation will perform *N* multiplications between w_n and x_n (cf. Equation 16). The spectrum of the windowed data is obtained by computing its DFT. A complex term is involved in the DFT computation. The DFT complexity is calculated by taking the real and the imaginary parts separately. Thus, DFT performs *2N* multiplications and *2(N-1)* additions per output frequency. For larger values of *N*, $2.(N-1) \approx 2.N$. Thus, the DFT computational complexity for *N* output frequencies becomes $2.(N)^2$ additions and $2.(N)^2$ multiplications. The combine computational complexity C_1 of the STFT is given by Equation 22.

For the proposed ARSTFT, Fs^i, Frs^i and w_n^i are not fixed and are adapted for the i^{th} 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. The EASA performs $2.N^i$ comparisons and N^i increments for the i^{th} selected window (cf. Section III-B). The choice of Frs^i and window shape, require three comparisons. The selected signal is resampled before computing its DFT. The NNRI is employed for the resampling purpose. The NNRI only requires a comparison operation for each resampled observation. Therefore, the resampler performs Nr^i comparisons. If $D^i = 1$, then a cosine window function is applied on the resampled data, which performs Nr^i multiplications (cf. Fig. 2). The DFT performs $2.(Nr^i)^2$ additions and $2.(Nr^i)^2$ multiplications for the i^{th} selected window. The combine computational complexity C_2 of the ARSTFT is given by Equation 23.

$$C_1 = A.\left[\underbrace{N + 2.(N)^2}_{Multiplications} + \underbrace{2.(N)^2}_{Additions}\right]. \quad (22)$$

$$C_2 = \sum_{i=1}^{K} \underbrace{N^i}_{Increments} + \underbrace{2N^i + 3 + Nr^i}_{Comparisons} + \underbrace{\alpha Nr^i + 2(Nr^i)^2}_{Multiplications} + \underbrace{2(Nr^i)^2}_{Additions}. \quad (23)$$

In Equation 22, *A* is the total number of windows, occurs for the observation length of *x(t)*. In Equation 23, *i=1,2,..,K* represents the index of the selected window. α is a multiplying factor, its value is *1* if $D^i = 1$ and 0 if $D^i = 0$. From C_1 and C_2 it is clear that there are uncommon operations between both techniques. In order to make them approximately comparable the following assumption is made.

***** *An increment or a comparison has the same processing cost as that of an addition.*

By following this assumption, comparisons and increments are merged into additions count, during the complexity evaluation process. The computational comparison of the ARSTFT with the classical one is made for results of the illustrative example. The gains are summarized in Table IV.

TABLE IV
Summary of the Computational Gain

Time Span (Sec)	Gain in Additions	Gain in Multiplications
1st activity	2.737	2.739
2nd activity	212.473	214.021
3rd activity	42.686	42.955
4th activity	12.213	12.365

Table IV shows the gain in additions and multiplications of the ARSTFT over the classical STFT for each *x(t)* activity. It shows that the ARSTFT leads to a significant reduction of the total number of operations as compare to the classical STFT. This reduction in operations is achieved by adapting Fs^i, Frs^i and w_n^i according to the local variations of *x(t)*.

VI. Interpolation Error

The interpolated samples are calculated by employing the level-crossing samples. For the practical AADC, there exist uncertainties in the 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 the exact time-amplitude pairs of the level-crossing samples are calculated by employing the method discussed in [12].

The mean interpolation error is calculated for each selected window by employing Equation 14. The results are summarized in Table V.

TABLE V
Mean Interpolation Error for Each Selected Window

Window	1st	2nd	3rd	4th	5th	6th
MIe^i(dB)	-28.9	-28.4	-28.2	-26.1	-26.1	-25.4

Table V shows that the error introduced by the resampling process is quite a minor one. In the case of high precision applications, the resampling error can be further reduced by increasing the AADC resolution *M* and the interpolation or-

der [4, 11, 15]. 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 given application.

VII. Conclusion

A new tool for the adaptive resolution time-frequency analysis has been proposed. The ARSTFT is especially well suited for the low activity sporadic signals like electrocardiogram, phonocardiogram, seismic signals, etc. It is shown that Fs^i and L^i change by following the $x(t)$ local variations. Criteria to choose the appropriate F_{ref} and N_{ref} are developed. A complete methodology of choosing Frs^i and w_n^i for the i^{th} selected window has been demonstrated. It is shown that the ARSTFT adapts its time-frequency resolution by following the local variations of $x(t)$. It provides a good time but a poor frequency resolution for the high-frequency parts of $x(t)$ and vice versa. This type of analysis is best suited for most of the real life signals [7]. The interpolation error is calculated. A higher accuracy 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 trade off between the accuracy level and the computational load.

The ARSTFT outperforms the STFT. The first advantage of the ARSTFT over the STFT is the adaptive time-frequency resolution and the second one is the computational gain. These smart features of the ARSTFT are achieved due to the joint benefits of the AADC, the EASA and the resampling, as they enable to adapt Fs^i, Frs^i, N^i, Nr^i and w_n^i by exploiting the local variations of $x(t)$.

The employment of fast algorithms in place of the DFT for the spectrum computation is in progress. It will add up to the computational gain of the ARSTFT. Performance comparison of the ARSTFT with the MRA techniques in terms of computational complexity and quality is an area of future research. Moreover the performance study of the ARSTFT for the real applications is a future task.

References

[1] J.W. Mark and T.D. Todd, "A nonuniform sampling approach to data compression", IEEE Transactions on Communications, vol. COM-29, pp. 24-32, January 1981.
[2] E. Allier, G. Sicard, L. Fesquet and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization", ASYNC'03, pp.197-205, May 2003.
[3] F. Aeschlimann, E. Allier, L. Fesquet and M. Renaudin, "Asynchronus FIR filters, towards a new digital processing chain", ASYNC'04, pp. 198-206, April 2004.
[4] N. Sayiner et al. "A Level-Crossing Sampling Scheme for A/D Conversion", IEEE Transactions on Circuits and Systems II, vol. 43, pp. 335-339, April 1996.
[5] S.C. Sekhar and T.V. Sreenivas, "Adaptive window zero-crossing based instantaneous frequency estimation", EURASIP Journal on Applied Signal Processing, pp.1791-1806, Issue 1, January 2004.
[6] D. Gabor, "Theory of communication", Journal of the IEE, Vol.93(3), pp.429-457, 1946.
[7] R. Polikar, "The engineer's ultimate guide to wavelet analysis", Rowan University, College of Engineering, retrieved June, 2006.
[8] S. M. Qaisar et al. "Spectral Analysis of a signal Driven Sampling Scheme", EUSIPCO'06, September 2006.
[9] S. de Waele and P.M.T.Broersen, "Time domain error measures for resampled irregular data", IEEE Transactions on Instrumentation and Measurements, pp.751-756, May 1999.
[10] S. de Waele and P.M.T.Broersen, "Error measures for resampled irregu-lar data", IEEE Transactions on Instrumentation and Measurements, pp.216-222, April 2000.
[11] S. M. Qaisar et al. "Computationally efficient adaptive rate sampling and filtering", EUSIPCO'07, pp.2139-2143, September 2007.
[12] M. Gretains, "Time-frequency representation based chirp like signal analysis using multiple level crossings", EUSIPCO'07, pp.2154-2158, Sep-tember 2007.
[13] S. M. Qaisar et al. "Adaptive rate filtering for a signal driven sampling scheme", ICASSP'07, pp.1465-1468, April 2007.
[14] M. Vetterli et al. "Wavelets and filter banks: Theory and design", IEEE Transections on signal processing, Vol.40, pp.2207-2232, 1992.
[15] F. Harris, "Multirate signal processing in communication systems", EUSIPCO'07, September 2007.
[16] K. M. Guan and A.C. Singer, "Oppertunistic Sampling by Level-Crossing", ICASSP'07, pp.1513-1516, April 2007.
[17] F. Akopyan et al. "A level-crossing flash analog-to-digital converter", ASYNC'06, pp.12-22, Grenoble, France, March 2006.