Asynchronous Processing of Sparse Signals

Azime Can, Ervin Sejdic and Luis F. Chaparro
Department of Electrical and Computer Engineering
University of Pittsburgh, Pittsburgh, PA, 15261, chaparro@ee.pitt.edu

Abstract

Unlike synchronous processing, low–power asynchronous processing is more efficient in biomedical and sensing networks applications as it it is free from aliasing constraints and quantization error in the amplitude, it allows continuous–time processing and more importantly data is only acquired in significant parts of the signal. In this paper, we consider signal decomposers based on the asynchronous sigma delta modulator (ASDM), a non–linear feedback system that maps the signal amplitude into the zero–crossings of a binary output signal. The input, the zero–crossings and the ASDM parameters are related by an integral equation making the signal reconstruction difficult to implement. Modifying the model for the ASDM, we obtain a recursive equation that permits to obtain the non–uniform samples from the zero–time crossing values. Latticing the joint time–frequency space into defined frequency bands, and time windows depending on the scale parameter different decompositions, similar to wavelet decompositions, are possible. We present two cascade low– and high–frequency decomposers, and a bank–of–filters parallel decomposer. This last decomposer using the modified ASDM behaves like a asynchronous analog to digital converter, and using an interpolator based on Prolate Spheroidal Wave functions allows reconstruction of the original signal. The asynchronous approaches proposed here are well suited for processing signals sparse in time, and for low–power applications. The different approaches are illustrated using synthetic and actual signals.

Index Terms

Asynchronous signal processing, time-encoding, level-crossing sampling, non-stationary signals, non–uniform sampling, wavelet package analysis, tiling, Spheroidal Wave Functions, Tikhonov regularization

I. Introduction

Continuous monitoring of signals, as in biomedical applications and sensing networks, challenges data acquisition, storage and processing —especially if remote monitoring is desired as this would require that a large number of samples be generated, stored and transmitted. One approach that addresses these problems is asynchronous signal processing [1]–[4]. Unlike synchronous processing, low-power asynchronous processing is free of aliasing constraints and quantization error in the amplitude while allowing continuous-time processing [5].

Conventional digital signal processors operate synchronously in time causing possible frequency aliasing when the frequency content of the input is not tracked or captured fully. At each sampling time the signal amplitude is approximated by a binary code suffering from quantization error. More importantly, as sample values are drawn independent of whether the values are significant or not, the sampling may result in redundant collected information. Differently, in level-crossing (LC) sampling [6] —an asynchronous sampling procedure— sample values are collected only when a specified quantization level is attained giving a non-uniform sampling. The immediate advantage of LC is that only samples are collected when there is significant information in the signal. Although, the LC sampling is not hampered by aliasing or

quantization error in the amplitude, and it can be processed in continuous-time, its drawbacks are that a set of quantization levels needs to be specified *a-priori* and that the sampling times and the corresponding amplitudes resulting from the sampling must be kept.

A time-encoding method using asynchronous sigma delta modulators (ASDMs) can be shown to be equivalent to an LC sampler with quantization levels set as local estimates of the signal average [7]. The ASDM, a non-linear feedback system, generates a binary sequence of pulses with widths proportional to the local average of the signal, and controlled by a scale parameter of the ASDM. Thus different from the LC sampler, only the zero-crossing times of the output of the ASDM are needed to obtain an approximation of the signal. The complexity of reconstructing a signal in an interval can be measured using the number of degrees of freedom in the sampled signal in the interval. In non-uniform sampling, it is not only necessary to have the amplitude of the samples but also their occurrence times, and as such compared with uniform sampling the reconstruction of the original signal is more complex. Despite this, non–uniform sampling provides a more efficient acquisition of meaningful values, and frequency aliasing is not considered.

The asynchronous multi-resolution signal processing scheme proposed in this work is well suited for burst-like signals (such as electro-cardiograms, speech signals, leakage sensors) and low power applications as in sensor applications, health monitoring systems, and brain–computer interfaces. Our approach is to consider the advantages of the asynchronous processing in a joint time–frequency context and to relate it with multi-resolution signal analysis. We present a novel scale–based non–uniform sampler that takes advantage of the statistical behavior in the signal and adjusts its sampling rate accordingly. While finer scales enable to capture short-lived changes in sparse signals, coarse scales reduce the number of acquired samples for the slow changing or silent portions of the signal. The paper first describes the asynchronous scale based analysis architecture and then investigates a recovery method from non-uniform samples using Prolate Spheroidal wave functions. In addition to being computationally inexpensive, the presented recovery method does not require the value of the samples, i.e. uses only the non-uniform sampling instants to map the information, and hence significantly reduces the transmission bandwidth compared to previously suggested asynchronous samplers. Simulations performed on sparse signals shows the significant advantages and the performance of the proposed techniques.

II. Asynchronous Analysis

A. A Module of the Asynchronous Low–frequency Decomposer

The low–frequency asynchronous decomposer consists of L cascaded modules, each having an asynchronous sigma delta modulator, an averager and a smoothing filter as shown in Fig.1. In module y, $i = 1, 2, .., L$, the input signal is separated into two components, $d_i(t)$ and $f_i(t)$. The number of modules L is determined by the range of frequencies of interest in the signal.

The main constituent of the decomposer module, the ASDM, is a non-linear analog circuit that consists of an integrator and a Schmitt trigger. By comparing the accumulated difference $y(t)$ with the threshold value $\pm\delta$, the Schmitt trigger returns a modulated binary waveform $z(t) = b(-1)^{k+1}[u(t_k) - u(t - t_{k+1})]$ where $u(t)$ is the unit-step function(see Fig. 2). The output of the integrator in $[t_k, t_{k+1}]$ is given by

$$y(t) = y(t_k) + \frac{1}{\kappa}\int_{t_k}^{t_{k+1}} [x(\tau) - z(\tau)]d\tau. \quad (1)$$

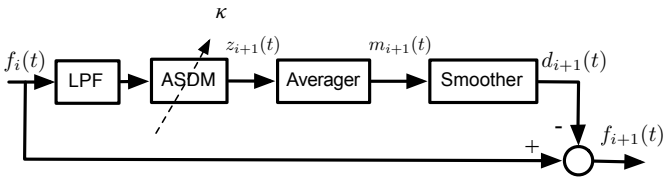

Fig. 1: A module of the asynchronous low–frequency decomposer

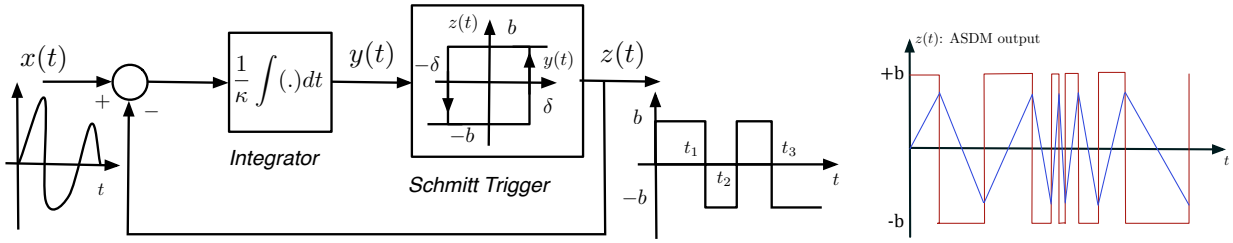

Fig. 2: Asynchronous sigma delta modulator, and its signals $y(t)$ and $z(t)$.

How fast the triggering occurs is related to the value of κ which is shown to be connected with the maximum frequency of the input signal. The information carried by the amplitude of $x(t)$ is provided by the zero–crossing times of the binary output signal $z(t)$. The zero-crossing times as well as the design parameters κ, δ, b (strictly positive real numbers) depend on the nature of the signal. The stability of the circuit, $|y(t)| < \delta$, is established by having the amplitude of the input signal be bounded by the bias parameter b be chosen bigger than the bound of the input signal, i.e., $\max|x(t)| = c < b$. Assuming that the initial state for $[y(t_k), z(t_k)]$ is $[\delta, b]$, at some time $t_{k+1} > t_k$ the output of the integrator, $y(t)$, reaches the triggering threshold δ so that according to (1):

$$-\delta = \delta + \frac{1}{\kappa}\int_{t_k}^{t_{k+1}}[x(\tau) - b]d\tau \quad \text{or} \quad \int_{t_k}^{t_{k+1}} x(\tau)d\tau = b(t_{k+1} - t_k) - 2\kappa\delta$$

Similarly starting with $[y(t_{k+1}), z(t_{k+1})]$ at state $[-\delta, -b]$, we have at time t_{k+2} :

$$\delta = -\delta + \frac{1}{\kappa}\int_{t_{k+1}}^{t_{k+2}}[x(\tau) + b]d\tau \quad \text{or} \quad \int_{t_{k+1}}^{t_{k+2}} x(\tau)d\tau = -b(t_{k+1} - t_k) + 2\kappa\delta$$

Thus, for an increasing sequence $\{t_k\}$, $k \in Z$, we have from above

$$\int_{t_k}^{t_{k+1}} x(\tau)d\tau = (-1)^k[-b(t_{k+1} - t_k) + 2\kappa\delta] \tag{2}$$

It is of interest to note here that:

- By approximating the integral in (2) using the trapezoidal rule it is possible to obtain an expression to reconstruct the original signal using zero-crossing times [7],
- A multilevel signal can be represented by zero-crossings using the above. For instance, consider the signal

$$x(t) = \sum_{i=-N}^{N} q_i[u(t - \tau_i) - u(t - \tau_{i+1})] \tag{3}$$

where $u(t)$ is the unit step function and $\{q_i\}$ are multilevel values. Processing the component $x_i(t) = q_i[u(t-\tau_i)-u(t-\tau_{i+1})]$ with the ASDM, its output would be a binary signal $z_i(t) = u(t-t_i)-2u(t-t_{i+1}) + u(t-t_{i+2})$. Letting $t_i = \tau_i$ we then need to figure out how to make $t_{i+2} = \tau_{i+1}$ by setting the parameters of the ASDM. If we let $b > \max(x(t))$ and δ be a fixed number, we need a value for κ_i that would make the connection with the signal and make $t_{i+2} = \tau_{i+1}$. Letting $\alpha_i = t_{i+1} - \tau_i$ and $\beta_i = \tau_{i+1} - t_{i+1}$ we have the following equations from which we can find t_{i+1} and κ_i:

$$(i) \quad q_i(\alpha_i + \beta_i) = (-1)^{i+1} b(\alpha_i - \beta_i)$$
$$(ii) \quad \kappa_i = \frac{q_i\alpha_i + (-1)^k b\alpha_i}{2\delta}$$

The above equations are obtained from the sum of two integrals and one of the integrals in (2), respectively. Since $\alpha_i + \beta_i = \tau_{i+1} - \tau_i$ and $\alpha_i - \beta_i = 2t_{i+1} - (\tau_i + \tau_{i+1})$ we obtain from equation (i)

$$t_{i+1} = \frac{q_i(\tau_{i+1} - \tau_i) + (\tau_{i+1} + \tau_i)}{2(-1)^{i+1}b}$$

allowing us to obtain α_i which is then used to find κ_i in equation (ii). This implies that for a multilevel signal the ASDM is an adaptive LC sampler. For each pulse in the multilevel signal we can find appropriate zero–crossings to represent it exactly. In the above the value of κ_i adapts to the width of the pulse so that the values of α_i and β_i are connected with the local average of the signal in the window set by κ_i which in this case is q_i.

For any signal $x(t)$, not necessarily multilevel, the second comment indicates that it can be approximated by a multilevel signal where the q_i values are estimates of the local average for windows set by the parameter κ. The advantage of obtaining such a multi-level representation is that it can be converted into continuous-time binary signals which can be processed in continuous-time [5], [8], [9]. Moreover, since the ASDM circuitry does not include a clock it consumes low power and is suitable for low voltage CMOS technology [10]. These two characteristics of the ASDM makes it very suitable for biomedical data acquisition systems [11], [12] where low–power and analog signal processing are desirable.

One of the critical features of the ASDM that motivates further discussions is the modulation rate, i.e. how fast $z(t)$ crosses zero, which depends on the input signal as well as the parameters δ and κ. To see the effect of just κ on the output, we let $x(t) = c$, $\delta = 0.5$, and $b > c$, giving

$$\alpha_k = t_{k+1} - t_k = \frac{\kappa}{c + (-1)^k b}, \qquad \beta_k = \frac{\kappa}{c - (-1)^k b}$$

so that $z(t)$ is a sequence of pulses that repeat periodically. It thus can be seen that the lower the κ the faster the $z(t)$ crosses zero, see Fig. 3, and vice versa. We thus consider κ a scale parameter which we will use to decompose signals. To simplify the analysis, the bias parameter of the ASDM is set to $b = 1$, which requires normalization of the input signal so that $\max(x(t)) < 1$ and the threshold parameter can be set to $\delta = 1/2$.

A sufficient condition for the reconstruction of the original signal from non-uniform samples is [13]

$$\max_k (t_{k+1} - t_k) \leq T_N \tag{4}$$

where $T_N = 1/2f_{max}$ is the Nyquist sampling period for a band-limited signal with maximum frequency f_{max}. We are however interested in considering signals which are not necessarily band limited or, the spectrum content of the signal is unknown. This can be done by exploiting this relation between κ and

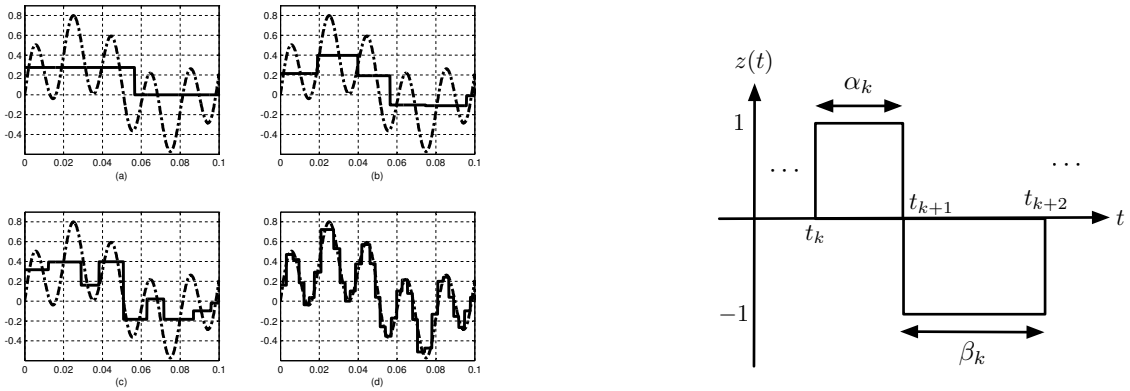

Fig. 3: Left (a)–(d): effect of scale parameter κ as it decreases; Right: definition of α_k and β_k corresponding to consecutive pulses of $z(t)$.

the frequency content of the input signal. To obtain the relation between κ and the maximum frequency of a band–limited signal we use equations (4) and (2) to get

$$\frac{\kappa}{b+c} \leq t_{k+1} - t_k \leq \frac{\kappa}{b-c} \leq T_N$$

which gives a way to choose the scale parameter in terms of the Nyquist rate:

$$\kappa \leq T_N(b-c) \leq \frac{b-c}{2f_{max}}. \tag{5}$$

Notice the scale parameter κ relates not only to the maximum frequency of the signal but to the difference between maximum amplitude of the signal, c, and b. Notice that in Fig. 1, the input signal is low–pass filtered to attain a band–limited signal that permit us obtain an appropriate κ_i parameter for the ASDM according to equation (5). These scale filters can be considered equivalent to the anti-aliasing filters in the Shannon-Nyquist sampling theory.

For a certain scale κ, determined by the maximum cutoff frequency of the corresponding scale filter, the ASDM maps the input signal into a sequence of binary rectangular pulses $z(t)$. Two consecutive pulses of $z(t)$ of duration $T_k = \alpha_k + \beta_k$ can be characterized by the duty–cycle $0 < \alpha_k/T_k < 1$ where as before $\alpha_k = t_{k+1} - t_k$ and $\beta_k = t_{k+1} - t_{t+1}$ (see right figure in Fig. 3). The duty–cycle can be expressed as $\alpha_k/T_k = (1+\zeta_k)/2$ where

$$\zeta_k = \frac{\alpha_k - \beta_k}{\alpha_k + \beta_k} \tag{6}$$

is the local average of $x(t)$ in $t_k \leq t \leq t_{k+1}$. Indeed, the integral equation (2) can be written in terms of $z(t)$ as

$$\int_{t_k}^{t_{k+1}} x(\tau)d\tau = (-1)^{k+1} \int_{t_k}^{t_{k+1}} z(\tau)d\tau + 2(-1)^k \kappa\delta \tag{7}$$

which can be used to get local estimate of the signal average in $[t_k, t_{k+2}]$

$$\hat{\zeta}_k = \frac{1}{T_k}\int_{t_k}^{t_{k+2}} x(\tau)d\tau = \frac{(-1)^{k+1}}{T_k}\left[\int_{t_k}^{t_{k+1}} z(\tau)d\tau - \int_{t_{k+1}}^{t_{k+2}} z(\tau)d\tau\right] = \frac{(-1)^{k+1}(\alpha_k - \beta_k)}{\alpha_k + \beta_k}. \tag{8}$$

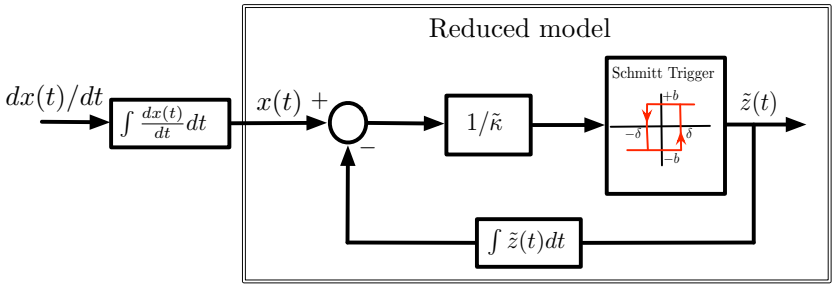

Fig. 4: Modified Asynchronous Sigma Delta Modulator

These averages form a multi-level approximation of the input signal for a chosen scale. Thus, the ASDM provides either a binary representation of its input $x(t)$ by a binary output $z(t)$ and the integral equation, or a sequence of local averages ζ_k at non–uniform times t_{2k}.

The local average approximation obtained above is analogous to the wavelet multi-resolution analysis. The pulses with duty-cycle defined by the sequence $\{\alpha_k, \beta_k\}$ can be regarded as Haar wavelets with signal dependent scaling and translation [14]. Indeed, if $\nu = u(t) - u(t-0.5)$ is a scaling function the wavelet can be defined as,

$$\varphi_{\alpha_k,\beta_k,t_k,t_{k+1}}(t) = \nu\left(\frac{0.5t}{\alpha_k} - t_k\right) - \nu\left(\frac{0.5t}{\beta_k} - t_{k+1}\right)$$

So that the local averages are found as

$$\zeta_k = \frac{1}{t_{k+2} - t_k}\int_{t_k}^{t_{k+2}} \varphi_{\alpha_k,\beta_k,t_k}(t)z(t)dt \tag{9}$$

giving an approximation similar to the one given by a wavelet representation:

$$\hat{x}(t) = \sum_{k\ even} \zeta_k p_k(t) \tag{10}$$

for unit pulse $p_{k,\ell}(t) = u(t-t_k) - u(t-t_{k+2})$. This is a generalization of the Haar wavelet for a particular value of the scale κ. Time-frequency localization of the signal can be well represented by few decomposition blocks. The binary waveform $z(t)$ is characterized by consecutive pulses where their duration T_k is related to the scale κ and the signal itself. In other words, rather than the fixed length in wavelet representation, here the time-windows are signal dependent. Signal dependency in the scaling and window size enables to capture details efficiently, i.e. converges faster in locating signal in time–frequency plane.

After averaging, the smoothing low-pass filter in Fig. 1 is used to avoid discontinuities that may occur when the multi-level signal is subtracted from the input signal of the corresponding module. The output signal of a module, denoted as $f_{i+1}(t)$, is the difference between input to the module and the forward path output $d_i(t)$. All modules operates under same principle but at a different scale, i.e. ASDM's integrator have different κ values.

B. Modified ASDM as a Non–uniform Sampler

The motivation of this section is to search for a better solutions for encoding the information provided by the ASDM. Although the ASDM has appealing features such as being signal dependent and low–power

information coder, its non–linear structure makes the recovery of the original information challenging [13], [15]. To overcome the challenge and sustain the advantages of ASDM signal dependent behavior, we consider a different approach to dealing with the integral equation (2). Consider that the input of the ASDM is the derivative of the signal, $dx(t)/dt$, rather than the signal itself. For this input equation (2) becomes

$$\int_{\tilde{t}_k}^{\tilde{t}_{k+1}} \frac{dx(\tau)}{d\tau} = (-1)^k[-(\tilde{t}_{k+1} - \tilde{t}_k) + \tilde{\kappa}] \tag{11}$$

This is equivalent to extracting the integrator from the feedforward loop in the model of Fig. 2 to obtain the equivalent feedback system shown in Fig. 4. The tilde symbol is used to indicate the parameters when the input is the derivative. This configuration has two interesting facts that simplify the analysis. First, letting the input be the derivative of the signal the integral equation is reduced to a first–order difference equation of which the input is a function of the zero–crossing times and a scale parameter of the ASDM:

$$x(\tilde{t}_{k+1}) - x(\tilde{t}_k) \;=\; (-1)^k[-(\tilde{t}_{k+1} - \tilde{t}_k) \;+\; \tilde{\kappa}] \tag{12}$$

Using the modified ASDM (MASDM), we only need to keep the zero–crossing times to recursively obtain samples of the signal at non–uniform times from which to approximate the original signal, rather than solving the integral equation (2). Zero–crossing times and scale parameters of the modified ASDM in the synthesis part of the procedure provide the regeneration of the samples which can be interpolated to reconstruct the original signal. Non–uniform samples drawn from a sparse signal using MASDM is shown in Fig. (5). Second, a reduced model can be used to avoid the derivative as input and reformulate the problem with respect to the signal itself, as highlighted in Fig. 4.

III. Signal Decomposers

In this section, two cascade realizations and a parallel realization of decomposers using the module in Fig. 1 are presented. Each configuration makes available different features in the analysis that are advantageous in specific applications. The common advantage in both configurations is that can be used for non-stationary signal analysis, especially for sparse signals in time, and for signals that are not necessarily band-limited.

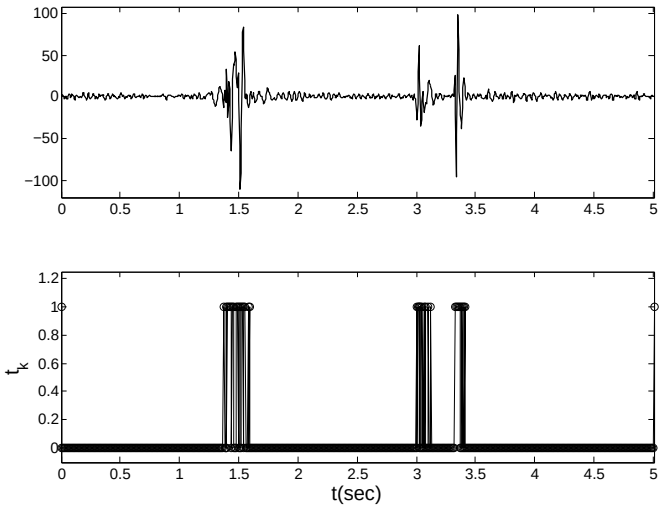

Fig. 5: Non–uniform samples obtained using Modified ASDM

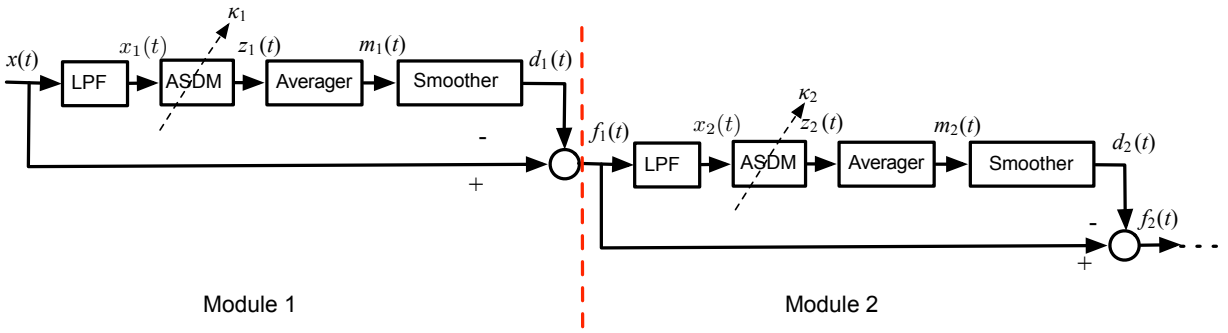

Fig. 6: Low-Frequency Decomposer

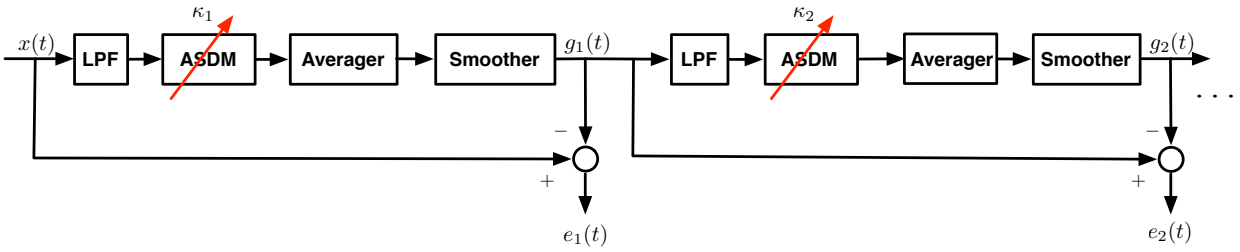

Fig. 7: High-Frequency Decomposer

A. *Cascade Decomposers*

The decomposers shown in Fig. 6 consists of L cascaded modules, one of them decomposes the signal into low–frequency components while the other provides the high–frequency components. Each module of the low–frequency considers the difference waveform $\{f_i\}$ as an input where $i = 1, 2, .., L$. Letting $f_0(t) = x(t)$, the inputs to the modules beyond the first one can be written sequentially as follows,

$$\begin{aligned} f_1(t) &= x(t) - d_1(t) \\ f_2(t) &= f_1(t) - d_2(t) = x(t) - d_1(t) - d_2(t) \\ &\vdots \\ f_L(t) &= x(t) - \sum_{\ell=1}^{L} d_\ell(t) \end{aligned} \tag{13}$$

Hence the decomposition can be expressed as,

$$x(t) = \sum_{\ell=1}^{L} d_\ell(t) + f_L(t) \approx \sum_{\ell=1}^{L} \sum_{k} \zeta_{k,\ell} p_{k,\ell}(t) + f_L(t) \tag{14}$$

where $\{\zeta_{k,\ell}\}$ are estimates of the local averages, corresponding to the scale parameters $\{\kappa_\ell\}$, and $p_k(t) = u(t - t_k) - u(t - t_{k+2})$. The last term $f_L(t)$ can be considered the error of the decomposition and it approaches zero after a small number of modules L. Because of the spectral behavior $f_L(t)$ is called the "high-frequency component" and the smoothed averages $\{d_\ell(t)\}$ are called the "low-frequency" components.

The scale decomposition is obtained by latticing the joint time-frequency plane so that the frequency axis is segmented to obtain related scale parameters $\{\kappa_\ell\}$ and the time axis is divided according to the

widths of the different time-windows determined by the scale parameters. For instance, to obtain a set of scales

$$\kappa_\ell = \frac{\kappa_{\ell-1}}{2^{\ell-1}}, \quad \ell = 2, \ldots, L$$

a narrow–band low-pass filter is chosen with bandwidth Ω_1 (rad/sec) such that $\kappa_1 = 2\pi(b-c)/\Omega_1$ according to equation (4), and the for the next modules low-pass filters of bandwidths $2^{\ell-1}\Omega_1$, $\ell = 2, \ldots, L-1$, i.e., are chosen. The bandwidth is doubled at each module to cover the bandwidth of the input signal. The number of modules, L, is determined by the range of frequencies of interest in the signal.

For the low–frequency decomposer we have followed a reasoning similar to wavelets analysis, and a dual of it would be the high–frequency decomposer shown in Fig. 7. The input is now expressed in terms of high–frequency terms $\{e_i(t)\}$,

$$\begin{aligned} g_1(t) &= x(t) - e_1(t) \\ g_2(t) &= x(t) - e_1(t) - e_2(t) \\ &\vdots \\ g_M(t) &= x(t) - \sum_{i=1}^{M} e_i(t) \end{aligned} \tag{15}$$

The decomposition in terms of high–frequency components is then given by

$$x(t) = \sum_{i=1}^{m} e_i(t) + g_M(t) \tag{16}$$

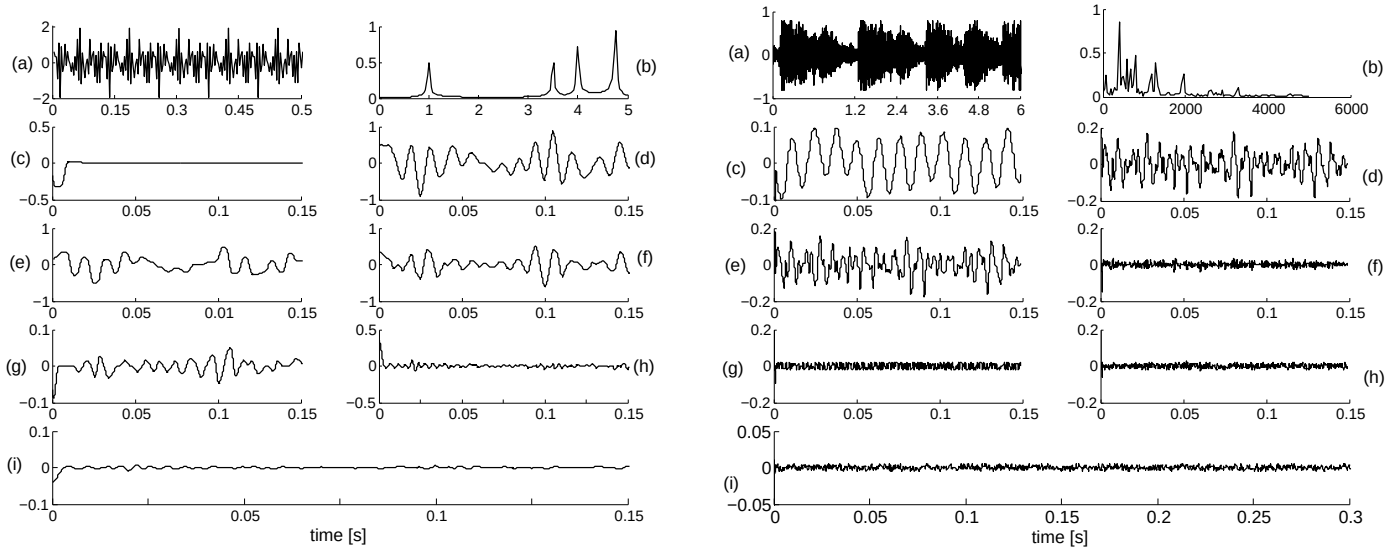

Fig. 8: Left: High-frequency decomposition for synthetic signal composed of four sinusoids: (a)–(b) original signal and its spectrum, (c)–(h) $d_i(t)$ components, (i) $f_L(t)$ component. Right: Low-frequency decomposition for speech signal: (a)–(b) original signal and its spectrum, (c)–(h) $e_i(t)$ components, (i) $g_M(t)$ component.

Figure 8 illustrates the low– and the high–frequency decompositions. The advantage of these types of decompositions is apparent for signals that reside either in the low or in the high end of the frequency spectrum.

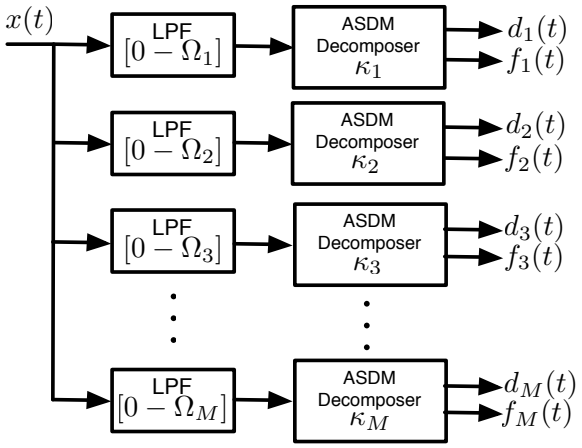

Fig. 9: Bank–of–filters decomposer

B. Parallel Decomposer

Asynchronous analysis can be also achieved by filtering the signal into different frequency bands and processing each separately. This approach is analogous to wavelet packets. A joint time–frequency lattice is constructed by determining frequency bands and letting the time windows, connected with the scale parameters, depend on the the maximum frequency of these bands and the amplitude of the signal being processed. The proposed bank–of–filters approach is shown in Fig. (9). The cutoff frequencies of the filters determine the values of κ_i for the i^{th} ASDM decomposer:

$$\kappa_i = \frac{\pi(1-c)}{\Omega_i},\ i = 0, 1, 2, ..., M-1 \tag{17}$$

where Ω_i is the cut-off frequency of the i^{th} filter, an M is the number of branches.

The output of each of the the ASDM decomposers, $z_\ell(t)$, provides random sequence $\{\alpha_{\ell,k}, \beta_{\ell,k}\}$ from which we can compute sequences of local averages and durations of the window $\{\bar{x}_{\ell,k}, T_{\ell,k}\}$ for $\ell = 1, 2, \cdots, M$, $k = 1, 2, \cdots, K$ where K corresponds to the number of pulses in each decomposition. For a non–deterministic signal $\{\alpha_{\ell,k}\}$ are random, and as such their distributions characterize $d_\ell(t)$ as well as the signal $x(t)$. The $\{\bar{x}_{\ell,k}, T_{\ell,k}\}$ provide the same compression as the one provided by $\{\alpha_{\ell,k}, \beta_{\ell,k}\}$ and the zero–crossings $\{t_{\ell,k}\}$. To obtain more compression we can consider the distribution of the $\{\bar{x}_{\ell,k}\}$ or $d_l(t)$ and use, similar to the process that motivated compressive sensing, a way to obtain a sparser signal by ignoring values clustered around one of the averages. In the following simulation of a Phonocardiograph recording of an actual heart sound, we emphasize this aspect of the parallel scheme by showing two decomposed low–frequency parts, $d_1(t)$ and $d_2(t)$, and their histogram in Fig. (10). Parallel units take advantage of the sparsity in the signal and enable efficient representation using dominant local averages in each branch.

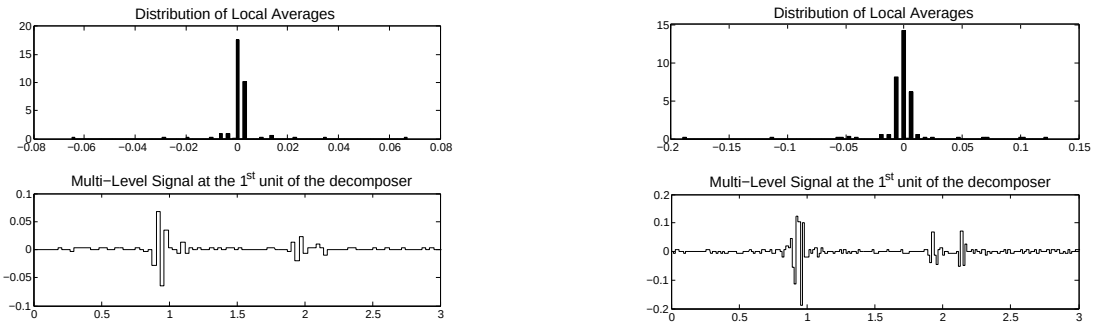

Fig. 10: Resulting $d_i(t)$'s on parallel decomposer configuration

IV. Modified ASDM Bank of Filters

It has been shown that structured data, e.g. sparse signals, can be efficiently sampled at far below the Nyquist rate [16], [17]. However, despite the sparseness of the samples the reconstruction presents problems due to the typical ill–conditioning of the matrices involved [18], and so the methods are computationally expensive. Also, non-uniform sampling is not a preferred method for data acquisition, since both samples and their corresponding sample times are needed for reconstruction. We consider next an approach that allows us to compute the non-uniform samples recursively from the zero–crossing times.

The above decomposers represent the signal using components in different frequency bands. To extend the scope of asynchronous processing we consider now the transmission and reconstruction of the asynchronously represented information using a bank of filters and the modified ASDM for the analysis and applying the Prolate Spheroidal Wave Functions (PSWF) in the reconstruction. This would be analogous to an analog–to–digital and digital–to–analog converters but for non–uniform sampling.

A. Analysis

In this section we propose a bank of filters structure that uses the modified ASDM to analyze the signal at desired frequency ranges. The MASDM structure provides the zero–crossing times needed to recursively obtain non–uniform samples. The recursive procedure reconstruction is given by Eq. (12) and

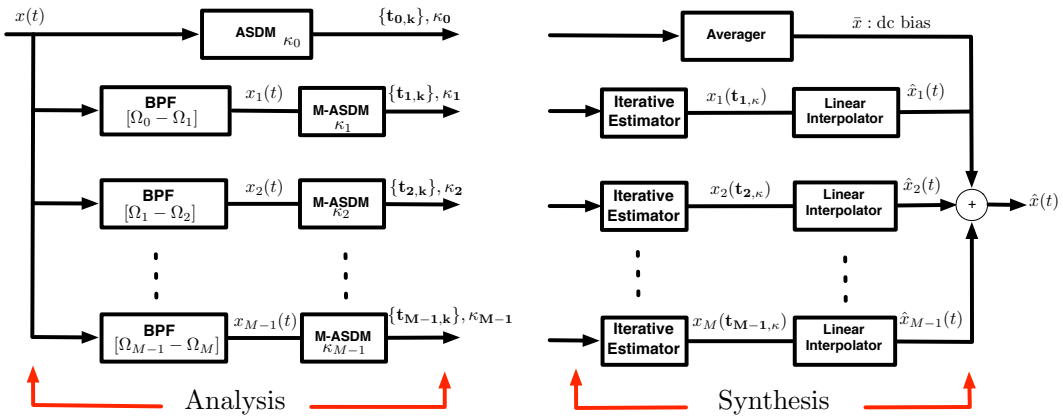

Fig. 11: Asynchronous Decomposer

requires that we set the parameters of the MASDM according to the derivative of the input. This can be done as follows:

- The amplitude bound in this case is

$$c_d = \max \left| \frac{dx(t)}{dt} \right| < 1.$$

This bound can be associated with the bound c of the signal itself. If we let the bound on $x(t)$ be c, and assume $x(t)$ is continuous we have

$$\left| \frac{dx(t)}{dt} \right| \leq \lim_{T \to 0} \frac{|x(t_i + T)| + |x(t_i)|}{T} = \lim_{T \to 0} \frac{2c}{T}$$

Letting $T \leq 1/(2f_{max})$, where f_{max} is the maximum frequency of $x(t)$ we have

$$c_d \leq \frac{2c}{T} \tag{18}$$

For simplicity, the bias parameter b can be set to one, and the threshold parameter can be set to 0.5.

- Again, the lack of knowledge of f_{max} is an issue to be addressed by the proposed bank of filters. Using the bandwidth of the filters in the bank–of–filters the scale parameter should then satisfy in each branch

$$\tilde{\kappa}_i \leq \frac{\pi(1 - c_d)}{\Omega_i}, \qquad i = 1, \cdots, M$$

where Ω_i is upper cut–off frequency of the filter in the i^{th} branch.

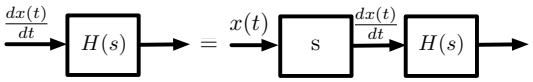

Fig. 12: The equivalent filters

Note that the *reduced model*, see Fig. (4), avoids the derivative as an input by adding a zero to each of the filters to make $x(t)$ the input rather than its derivative (as in Fig.(12)) and the low–pass filters band–pass. We also need to include an extra branch to estimate the dc-bias, which is lost when the derivative is calculated, if the signal has any.

The dependence of the recursion on $\tilde{\kappa}$ can be eliminated by considering evaluating (2) in two consecutive time intervals giving the following relation:

$$\int_{t_k}^{t_{k+2}} x(\tau)\ d\tau = t_{k+2} - 2t_{k+1} + t_k$$

and as such when replacing the signal by its derivative as the input in the ASDM we have

$$x(\tilde{t}_{k+2}) - x(\tilde{t}_k) = \tilde{t}_{k+2} - 2\tilde{t}_{k+1} + \tilde{t}_k$$

This not only eliminates the value of $\tilde{\kappa}$ in the calculations but reduces the number of sample values $\{x(\tilde{t}_k)\}$.

B. Data Transmission and Interpolation

To use equation (19) to recover the non–uniform samples we need to transmit the zero crossing for each of the branches of the bank–of–filters (see Fig. 11). These values could be subjected to quantization error. The time elapsed between two consecutive samples is quantized according to a timer with a period of T_c sec. Theoretically the quantization error can be minimized as much as it is needed by simply reducing the T_c. In this work, we adapted a method proposed by Roza [10] for sampling the asynchronous signal at a relatively low frequency. By using a clock with a sampling frequency $1/T_c$ on the modulated asynchronous signal, a sample–and–hold signal is obtained. We have obtained reasonable quantization errors using a 8–bit quantizer. The reconstruction accuracy is shown in Fig. (13). Here the signal used is an electroencephalography recording and the clock frequency is set to $20kHz$. The zero crossings corresponding to this signals are quantized with the eight-bit quantizer and then used to reconstruct the non–uniform samples.

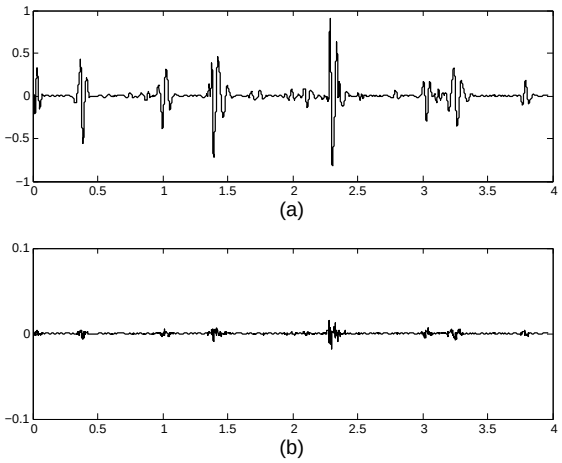

Fig. 13: (a) Iterative estimator output (b) Quantization Error with SNR=17.94dB

The Prolate Spheroidal Wave Functions, also known as Slepian functions [19], are among all the possible time-limited functions those that maximize their energy over a desired frequency band. To reconstruct the original signal we will use baseband as well as modulated Slepian functions for interpolation. Using non-uniform sampling instants $\{t_k\}$ the M-dimensional Slepian projection is given by

$$\hat{x}(\hat{t}_k) = \Phi(\hat{t}_k)\gamma_M \tag{19}$$

where $\hat{t}_k = t_k + \tau$, $-\frac{T_c}{2} \leq \tau \leq \frac{T_c}{2}$, $0 \leq k \leq N-1$ for the actual zero–crossing t_k and τ the quantization error related to the sampling period T_c used in Roza's method. The projection coefficients γ_M can be obtained by the pseudo-inverse,

$$\gamma_M = [\Phi(\hat{t}_k)]^\dagger \hat{x}(\hat{t}_k) \tag{20}$$

Then the reconstructed signal are given by

$$x_r(t) = \Phi(\hat{t}_k)[\Phi(\hat{t}_k)]^\dagger \hat{x}(\hat{t}_k) \tag{21}$$

An inversion of $N \times M$ Slepian matrix $\Phi(\hat{t}_k)$ is highly ill–conditioned, especially when the sampling is non–uniform [20]. In order to obtain numerically stable solutions we used Tikhonov regularization method [21], [22],

$$\gamma_{M_\epsilon} = (\Phi(\hat{t}_k)^T \Phi(\hat{t}_k) + \epsilon I)^T \hat{x}(\hat{t}_k). \tag{22}$$

Reconstruction using baseband Slepian functions and Modulated Slepian Functions are shown in Fig. (15) and compared with the piecewise linear interpolation which is known to be well suited for non uniform samples. Incorporating Modulated Slepian functions [23], see Fig. (14), into filter–bank scheme provided the minimum error in the reconstruction as optimum time–frequency concentration is ensured in each band.

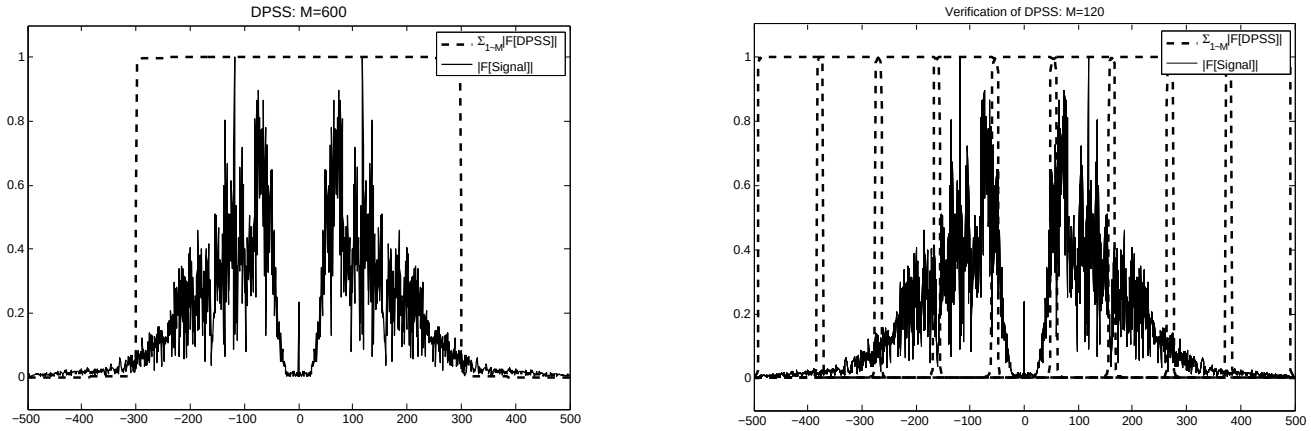

Fig. 14: Right: Analyzed signal's spectrum with spectrum of Discrete Prolate Spheroidal Sequences; left: Modulated–Discrete Prolate Spheroidal Sequences

To illustrate the advantage of using modulated Slepian functions we consider the processing of a Phonocardiograph recordings of an actual heart sound containing the opening snap [24]. The heart sound

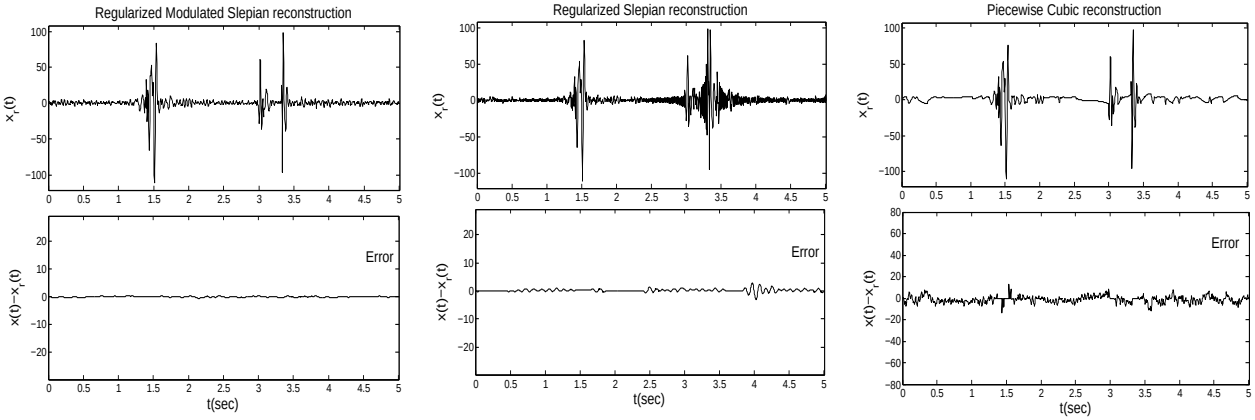

Fig. 15: Reconstructed signal and error: (left) Modulated–Slepian reconstruction (SNR=25.95dB, M=120, $\epsilon = 2 \times 10^{-4}$), (middle) baseband Slepian reconstruction (SNR=15.20 dB, M=600, $\epsilon = 2 \times 10^{-4}$) (right) Piecewise Cubic Interpolator (SNR= 10.89 dB)

is processed with the bank–of–filters, and zero-crossing times $t_{i,k}$ are obtained at each branch, see Fig. 11. Performance comparison of different interpolation methods can be seen in Fig. 15. Using reconstruction with modulated Slepian functions returned an improved performance compared to interpolation using baseband Slepian functions.

V. Conclusion

In this paper, we studied the decomposition, sampling, and reconstruction of sparse signals. A scale-based representation was obtained though two different asynchronous decomposition schemes, low– and high–frequency decomposers. Specifically, we proposed a bank–of–filters parallel decomposer using modified ASDM that behaves like a asynchronous analog to digital converter. Issues involving regularization and quantization in such recovery were discussed in details. The advantage of modulated Slepian functions on the interpolation from non–uniformly samples was experimentally analyzed. The simplicity of the reconstruction method and the accuracy of the interpolation with modulated Slepian interpolation showed promising results for the recovery of sparse signals from zero–crossings.

References

[1] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, "Asynchronous fir filters: towards a new digital processing chain," in *10th International Symposium on Asynchronous Circuits and Systems, 2004*, 2004, pp. 198–206.

[2] M Renaudin, "Asynchronous circuits and systems : a promising design alternative," *Microelectronic Engineering*, vol. 54, no. 12, pp. 133 – 149, 2000.

[3] D. Kinniment, A. Yakovlev, and B. Gao, "Synchronous and asynchronous a-d conversion," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 2, pp. 217–220, 2000.

[4] T. Hawkes and P. Simonpieri, "Signal coding using asynchronous delta modulation," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 729–731, 1974.

[5] Y. Tsividis, "Digital signal processing in continuous time: a possibility for avoiding aliasing and reducing quantization error," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, 2004, vol. 2, pp. ii–589–92 vol.2.

[6] K. Guan and A.C. Singer, "A level-crossing sampling scheme for non-bandlimited signals," in *Proceedings of IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2006, vol. 3, pp. III–III.

[7] S. Senay, L.F. Chaparro, M. Sun, and R. Sclabassi, "Adaptive level-crossing sampling and reconstruction," in *Proceedings of the 18th European Signal Processing Conference (EUSIPCO)*, August 2010.

[8] Y. Tsividis, "Mixed-domain systems and signal processing based on input decomposition," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 10, pp. 2145–2156, 2006.

[9] M. Kurchuk and Y. Tsividis, "Signal-dependent variable-resolution clockless a/d conversion with application to continuous-time digital signal processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 5, pp. 982–991, 2010.

[10] E. Roza, "Analog-to-digital conversion via duty-cycle modulation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 11, pp. 907–914, 1997.

[11] E. V. Aksenov, Yu M. Ljashenko, A. V. Plotnikov, D. A. Prilutskiy, S.V. Selishchev, and E. V. Vetvetskiy, "Biomedical data acquisition systems based on sigma-delta analogue-to-digital converters," in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, 2001, vol. 4, pp. 3336–3337 vol.4.

[12] C. Kaldy, A. Lazar, E. Simonyi, and L. Toth, "Time encoded communications for human area network biomonitoring," Tech. Rep. 2-07, Department of Electrical Engineering, Columbia University, New York, NY, June 2007.

[13] A. Lazar, E. Simonyi, and L. Toth, "Time encoding of bandlimited signals, an overview," in *Proc. of the Conf. on Telecommunication Systems, Modeling and Analysis*, November 2005.

[14] A. Can, E. Sejdic, O. Alkishriwo, and L.F. Chaparro, "Compressive asynchronous decomposition of heart sounds," in *IEEE Statistical Signal Processing Workshop (SSP)*, Aug. 2012, pp. 736–739.

[15] Aurel A. Lazar, Ern K. Simonyi, and Lszl T. Tth, "An overcomplete stitching algorithm for time decoding machines," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2008.

[16] Y. Eldar and T. Michaeli, "Beyond bandlimited sampling," *IEEE Signal Processing Magazine*, vol. 26, no. 3, pp. 48–68, 2009.

[17] A. Can, E. Sejdic, and L. F. Chaparro, "Asynchronous sampling and reconstruction of sparse signals," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug. 2012, pp. 854–858.

[18] P. C. Hansen, *Rank-deficient and discrete ill-posed problems : numerical aspects of linear inversion texte imprim*, SIAM monographs on mathematical modeling and computation. SIAM, Philadelphia, 1998.

[19] D. Slepian, "Prolate spheroidal wave functions, fourier analysis and uncertainty," *Bell Syst. Tech. J*, vol. 57, no. 5, pp. 1371–1429, 1978.

[20] H. Choi and Jr. Munson, D.C., "Analysis and design of minimax-optimal interpolators," *IEEE Transactions on Signal Processing*, vol. 46, no. 6, pp. 1571–1579, 1998.

[21] S. Senay, J. Oh, and L.F. Chaparro, "Regularized signal reconstruction for level-crossing sampling using slepian functions," *Signal Processing*, vol. 92, no. 4, pp. 1157 – 1165, 2012.

[22] A Neumaier, "Solving ill-conditioned and singular linear systems: A tutorial on regularization," *SIAM Review*, vol. 40, pp. 636–666, 1998.

[23] E. Sejdic, A. Can, L.F Chaparro, C.M. Steele, and T. Chau, "Compressive sampling of swallowing accelerometry signals using time-frequency dictionaries based on modulated discrete prolate spheroidal sequences," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 1–14, 2012.

[24] E. Sejdic and Jin Jiang, "Selective regional correlation for pattern recognition," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 37, no. 1, pp. 82–93, 2007.