Non-uniform Sampling Based on Adaptive Level-Crossing scheme

Mehrzad Malmirchegini[a], Mohammadmehdi Kafashan[b], Mona Ghassemian[c], Farrokh Marvasti[c]

[a] *Qualcomm Incorporated, San Diego, California, USA*
[b] *Department of Electrical and Computer Engineering, University of Akron, Akron, OH, USA*
[c] *Advanced Communications Research Institute (ACRI), Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*

Abstract

Level-Crossing (LC) Analog-to-Digital (A/D) converters can efficiently sample certain classes of signals. An LC A/D converter is a real-time asynchronous system, which encodes the information of an analog signal into a sequence of non-uniformly spaced time instants. In particular, this class of A/D converters uses an asynchronous data conversion approach, which is an efficient technique in terms of the power consumption. In this paper, we propose adaptive and multi-level adaptive LC sampling models as alternatives to conventional LC schemes and apply an iterative algorithm to improve the reconstruction quality of LC A/D converters. Our simulation results show that multi-level adaptive LC outperforms conventional A/D converters such as Sigma-Delta A/D converters in terms of the both performance and computational complexity.

Email addresses: mehrzadm@qti.qualcomm.com (Mehrzad Malmirchegini), mk103@zips.uakron.edu (Mohammadmehdi Kafashan), iran_mona@yahoo.co.uk (Mona Ghassemian), marvasti@sharif.edu (Farrokh Marvasti)

Keywords: Asynchronous A/D converter, Signal Reconstruction, Level-Crossing, Iterative Algorithm, Inverse Problem

1. Introduction

Sampling of an analog signal is an important part in digital signal processing (DSP) and communications systems [1]. Nyquist theory introduces a uniform sampling approach for bandlimited signals. In conventional A/D converters, which are based on uniform sampling, a common clock is used to convert an analog signal to digital numbers [2]. For instance, in Sigma-Delta A/D converter, the internal clock operates at a much higher frequency than the signal bandwidth. The over-sampled signal will be then downsampled at the last stage of A/D conversion [3, 4].

Most of signals collected by sensors (such as temperature, pressure, electrocardiograms and speech), however, have low frequency content, i.e. they do not vary for a long period of time. Therefore, it is better to use an asynchronous converter with a lower internal clock rate rather than a Sigma-Delta A/D converter. Asynchronous converters can be implemented without a global clock and they have interesting properties, such as low power consumption and reduced electromagnetic interference [5, 6, 7, 8, 9, 10].

Related work: The level crossing sampling result in non-uniform samples [11, 12, 13]. In this scheme, a sample is taken when a reference level is crossed, thereby saving the dynamic power of the A/D converter as well as the post-processing DSP unit. In this scheme, samples are not spaced regularly and the sampling rate depends on the input signal. The LC converters are asynchronous in nature and can adapt themselves to the varying dynamic

range of the input signals without any loss of quality while conventional A/D converters create more distortion if the dynamic range suddenly changes. Basic principles of the LC sampling schemes are presented in [14, 15, 16], where non-uniform samples of an LC sampler are transformed to uniform samples by polynomial interpolation. The architecture of LC-based asynchronous A/D converters is introduced in [17] and the performance of the system is analyzed for a sinusoidal signal. In this work, the author shows that LC converter consumes less power than a synchronous converter with a similar complexity. In [18], the LC scheme for non-bandlimited signals is presented and it is shown that this type of sampling outperforms uniform sampling. Also, recent papers [19, 20] discuss other relevant A/D sampling.

Contributions: As we mentioned earlier, the standard LC converters generate non-uniform samples using a set of crossing levels which need to cover the dynamic range of the input signal. Therefore, the prior knowledge about the dynamic range of the signal is required in order to adjust the crossing levels. Furthermore, in standard LC scheme, we require at least $\lceil \log_2(L) \rceil$ bits to encode the amplitude of each non-uniform sample, where L represents the number of crossing levels. One of the contributions of this paper is to propose two adaptive level crossing (ALC) schemes, which perform the level crossing sampling and do not require the dynamic range of the input signal. Moreover, we show that in these schemes, the amplitude of the non-uniform samples can be encoded with considerably less number of bits. We then mathematically characterize the relationship between the number of crossing levels and the sampling rate and shed light on the impact of the time quantization on the performance of the LC A/D converter. Motivated by our analysis, we

then utilized an iterative algorithm to improve the performance of the LC A/D converters. We furthermore compare the performance of the proposed ALC A/D converter with the conventional Sigma-Delta A/D converters and show that ALC converters outperforms Sigma-Delta A/D converter in terms of the both performance and computational complexity.

The rest of this paper is organized as follows: In Section 2, we present our proposed adaptive and multi-level adaptive level-crossing schemes. In Section 3, we mathematically characterize the relationship between the number of crossing levels, the distance between levels, and over-sampling ratio. We furthermore investigate the impact of the quantization on the reconstruction performance. In Section 4, we introduce the iterative algorithm and show how this algorithm can be utilized to improve the performance of LC A/D converters. We show the simulation results in Section 5 and conclude the paper in Section 6.

2. Adaptive Level-Crossing Sampling Scheme

Let $\mathcal{L} = \{-\frac{L-1}{2}d, -\frac{L-3}{2}d, \cdots \frac{L-3}{2}d, \frac{L-1}{2}d\}$ denote the set of the crossing levels, where L is an odd integer, which represents the number of the crossing levels and d denotes the distance between the levels. In LC A/D converters, the conversion of samples takes place whenever the continuous time signal is intersected by a crossing level. Therefore, in this category of converters, we are dealing with non-uniform samples, where each sample is represented by a non-uniformly spaced time instance, t_n, as well as the corresponding amplitude, $a_n \in \mathcal{L}$. Fig. 1 shows how non-uniform samples are obtained by LC sampling scheme. In this scheme, the precise value of the crossing level

and the quantized time difference between consecutive samples are stored for digital transmission. Clearly, we require $\lceil \log_2 L \rceil$ bits in order to encode the precise value of the crossing levels. Next, we explain how we quantize the time differences in LC sampling scheme. Let us define the time quantizer function as follows:

$$\Upsilon^{M}_{\mathrm{TR}}(x) \triangleq \begin{cases} \left[\frac{x}{\mathrm{TR}}\right]\mathrm{TR} & x < (2^M - 1.5) \times \mathrm{TR} \\ (2^M - 1)\mathrm{TR} & x \geq (2^M - 1.5) \times \mathrm{TR} \end{cases}, \tag{1}$$

where M denotes the number of quantization bits, TR represents the quantization step size and [.] is the round operator. Therefore, the quantized time difference, corresponding to the nth sample, can be characterize as follows: $dt_n^{\mathrm{quantized}} = \Upsilon^{M}_{\mathrm{TR}}(t_n - t_{n-1})$. As can be seen, in this encoding scheme, the quantization error of each sample will propagate to the next ones. In order to prevent the propagation of the quantization error, the time difference can be encoded as follows:

$$dt_n^{\mathrm{quantized}} = \Upsilon^{M}_{\mathrm{TR}}(t_n - t_{n-1}^{\mathrm{quantized}}), \tag{2}$$

where $t_n^{\mathrm{quantized}}$ can be generated recursively as follows: $t_n^{\mathrm{quantize}} = t_{n-1}^{\mathrm{quantized}} + dt_n^{\mathrm{quantized}}$ with $t_1^{\mathrm{quantize}} = \Upsilon^{M}_{\mathrm{TR}}(t_1)$.

As can be seen, in this A/D converter the quantization occurs in time domain. Hence, the quality of the converter is highly affected by TR, which is the period time of the local timer. Let $T_{\max}$ denote the maximum time interval between two consecutive non-uniform samples, we then have $M \geq \left\lceil \log_2 \left(\frac{T_{\max}}{\mathrm{TR}}\right)\right\rceil$. This sampling scheme is the dual of the uniform sampling, where the time instances are quantized while the amplitudes are precisely known.

2.1. Adaptive Level-Crossing (ALC) Sampling Scheme

In LC scheme, a sample is taken when the input signal is intersected by one of the crossing levels. Therefore, we require the dynamic range of the input signal to adjust the crossing levels. In order to overcome this constraint, we have developed an adaptive level-crossing (ALC) scheme as illustrated in Fig. 2. In this scheme, two crossing levels q_1 and q_2 [1] are adapted upward or downward depending on which level intersects the input signal first. If the input signal first crosses q_1, we send a positive impulse and move the two levels upward, i.e. $q_1' = q_1 - \delta + d$ and $q_2' = q_2 - \delta + d$, where q_1' and q_2' represent the corresponding crossing levels for the next sampling step. The offset parameter δ is chosen arbitrary small and positive in order to ensure that the updated crossing levels, i.e. q_1' and q_2', remain in the dynamic range of the input signal. On the other hand, if the input signal first intersects q_2, a negative impulse is transmitted and the two levels are adapted downwards. The precise value of the levels can be then recursively obtained by knowing the positions of the positive and negative impulses. Therefore, we only required one bit of information in order to encode the amplitude of each sample. Encoding of the time difference in ALC scheme is similar to LC scheme. Next, we consider different scenarios for ALC scheme and explain how the crossing levels adaptively sample the input signal in each scenario.

Case one: Consider the case, where the crossing levels, q_1 and q_2, intersect the input signal in regions with positive slopes. In these regions, the lower

[1] Without loss of generality, we can assume that $q_2 > q_1$.

level intersects the input signal first. The crossing levels will be then updated as follows:

$$\begin{aligned} q_1' &= q_1 - \delta + d, \\ q_2' &= q_2 - \delta + d, \\ q_1 &= q_1', q_2 = q_2'. \end{aligned} \tag{3}$$

An example of this case is illustrated in Fig. 2. Two levels, q_1 and q_2, shown by green lines, intersect the input signal in a region with the positive slop. As can be seen, q_1 intersects the signal first, therefore, we store a positive impulse at the place of the intersection with q_1. The levels are then updated upward, resulting in q_1' and q_2', shown by red lines.

Case two: Consider the regions, where the slope of the input signal changes from positive to negative. In these regions, the input signal does not have any intersection with the upper level (q_2). The crossing levels will be then updated as follows:

$$\begin{aligned} q_1' &= q_1 - \delta - d, \\ q_2' &= q_2 - \delta - d, \\ q_1 &= q_1', q_2 = q_2'. \end{aligned} \tag{4}$$

Consider the blue crossing levels in Fig. 2. In this case, there is no intersection between input signal and the upper level, meaning that the slope of the signal is changing from positive to negative. Therefore, the negative impulse at the intersection of the signal with the lower level will be stored and the levels will be then updated downward, shown by brown lines.

Case three: Consider the case, where the crossing levels intersect the input signal in regions with negative slopes. In these regions, the upper level

intersects the input signal first. The crossing levels will be then updated as follows:

$$\begin{aligned} q_1' &= q_1 + \delta - d, \\ q_2' &= q_2 + \delta - d, \\ q_1 &= q_1', q_2 = q_2'. \end{aligned} \tag{5}$$

Case four: Consider the regions, where the slope of the input signal changes from negative to positive. In these regions, the input signal does not have any intersection with the lower level (q_1). The crossing levels will be then updated as follows:

$$\begin{aligned} q_1' &= q_1 + \delta + d, \\ q_2' &= q_2 + \delta + d, \\ q_1 &= q_1', q_2 = q_2'. \end{aligned} \tag{6}$$

Note that for ALC sampling schemes two parameters need to be determined: d and δ. In Section 3, we mathematically characterize the relationship between L, d, and sampling rate and explain how these parameters should be selected.

2.2. 2-Level Adaptive Level-Crossing Sampling Scheme

As we mentioned earlier, the LC scheme requires the dynamic range of the input signal in order to adjust the crossing levels. However, ALC scheme utilizes an adaptive approach in order to cover the dynamic range of the input signal without requiring the knowledge of the dynamic range. In both LC and ALC schemes, however, the distance between the crossing levels is

constant. Therefore, these schemes may not provide enough samples in the regions, where the slope of input signal is low. For instance, if in some regions the value of the input signal slowly varies between two levels, then both LC and ALC schemes will not take enough samples in these regions. This also increases the value of $T_{\max}$. Furthermore, in regions, where the input signal varies quickly, these sampling schemes may result in redundant samples. In order to improve the performance of level crossing converters, we propose a multi-level ALC solution, where the distance between levels are adapted to the signal slope. By using the multi-level ALC scheme, we can ensure that enough samples are taken, specially in the low-slop regions of the input signal. For instance, in 2-level ALC, we consider two values for the distances between the crossing levels, d_1 and d_2, where $d_1 > d_2$. For the regions, where the slope of the input signal is high, i.e. the absolute value of the derivative of the input signal is above a certain threshold (η), the distance between crossing levels will become d_1. This choice reduces the number of redundant samples. On the other hand, for low-slope regions, we pick d_2 to ensure that enough samples are taken in these regions. Note that the parameters d_1 and d_2 should be chosen such that they satisfy the Nyquist theorem on average. In Section 3, we provide more explanation on how to choose parameters d_1, d_2 and η.

In 2-level ALC scheme, time quantization is similar to ALC schemes. However, we require two bits in order to represent each crossing level. The first bit represents whether the sample belongs to the high or low slope region of the input signal and the second bit represents positive or negative impulse similar to ALC scheme.

Fig. 3 compares the sampling patterns of ALC and 2-Level ALC schemes. As can be seen, 2-level ALC takes more samples in low-slope regions of the input signal. Furthermore, 2-Level ALC scheme has less samples in high-slope regions than ALC. Therefore, 2-level ALC makes the non-uniform sample more uniform in an asynchronous manner. At the decoder (D/A), the time indices are generated by accumulating time differences, and then the generated non-uniform samples are linearly interpolated and passed through a low-pass filter to yield an approximation of the original analog signal.

3. Analytical Discussion

In this section, we mathematically characterize the relationship between the number of crossing levels (L), the distance between levels (d), and over-sampling ratio. We furthermore investigate the impact of the quantization on the reconstruction performance.

Let $\text{OSR} = f_{\text{sampling}}/f_{\text{Nyquist}}$ denote the ratio between the sampling rate and Nyquist rate (twice the signal bandwidth). In LC sampling schemes, the generated samples are non-uniform. Therefore, the OSR values depend on the crossing levels as well as the input signal and are not necessarily integers. In order to facilitate the mathematical derivations of this section, we assume that the input signal, $x(t)$, is a bandlimited and zero-mean Gaussian with the following spectral density:

$$S(f) = \begin{cases} 1, & |f| \leq B \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

where B denotes the bandwidth of the input signal. Let $\nu(a, T)$ denote the mean number of samples with amplitude of a in the interval of T. Using

the results derived by Blake and Lindsey in [15], we have

$$\overline{\nu}(a) = \frac{E\{\nu(a,T)\}}{T} = \frac{2B}{\sqrt{3}} \exp(-\frac{a^2}{2\sigma_x^2}), \tag{8}$$

where $\overline{\nu}(a)$ denotes the mean number of level crossings of amplitude a and σ_x^2 represents the variance of the zero-mean Gaussian source, which generates the input signal.

Let $n(d, L)$ denote the average number of samples corresponding to the following level crossing set $\mathcal{L} = \{-\frac{L-1}{2}d, -\frac{L-3}{2}d, \cdots \frac{L-3}{2}d, \frac{L-1}{2}d\}$, where L is an odd integer. We then have,

$$n(d, L) = \sum_{i=-\frac{L-1}{2}}^{\frac{L-1}{2}} \overline{\nu}(id) = \frac{2B}{\sqrt{3}} + \frac{4B}{\sqrt{3}} \sum_{n=1}^{\frac{L-1}{2}} \exp(-\frac{\alpha^2 n^2}{2}), \tag{9}$$

where $\alpha = \frac{d}{\sigma_x}$. Hence,

$$\text{OSR} = \frac{n(d, L)}{2B} = \frac{1}{\sqrt{3}} \Big(1 + 2 \sum_{n=1}^{\frac{L-1}{2}} \exp(-\frac{\alpha^2 n^2}{2})\Big). \tag{10}$$

For a large value of L, OSR can be approximated as $\sqrt{\frac{2\pi}{3}} \frac{1}{\alpha}$. For a small value of L, by considering that $\exp(-\frac{1}{2}n^2\alpha^2) < \exp(-\frac{1}{2}n\alpha^2)$, we can obtain an upper bound for OSR as follows:

$$OSR < \sqrt{\frac{4}{3}} \frac{1 - \exp(-\frac{1}{4}(L+1)\alpha^2)}{1 - \exp(-\frac{1}{2}\alpha^2)} - \frac{\sqrt{3}}{3}. \tag{11}$$

Furthermore, the maximum OSR value for a specific number of crossing levels (L) can be obtained from Eq. 10 as

$$\text{OSR}_{\max} = \frac{L}{\sqrt{3}}. \tag{12}$$

Next, we investigate how to choose the parameters of different level-crossing sampling schemes. The following theorem states that under what condition the non-uniform samples can uniquely represent the input signal.

Theorem 1. *A bandlimited signal can be uniquely represented by its non-uniform samples if the average sampling rate satisfies the Nyquist condition.*

proof: See [11] for the proof.

Therefore, the necessary condition for the perfect reconstruction of the input signal is to have OSR ≥ 1. However, this condition does not guarantee the perfect reconstruction and typical value for α is proved to be in range of $0.01 \leq \alpha \leq 0.5$ [14] and [11].

Remark 1. *Note that ALC scheme with $d = q_2 - q_1$ and $\delta \ll 1$ provides almost the same sampling pattern as the LC sampling scheme, where the distance between levels is d and the existent levels fully cover the dynamic range of the input signal. Therefore, under the above conditions, the analysis of the LC scheme can be also applied to ALC scheme.*

For 2-Level ALC, we can simply choose $d_1 = d$ and d_2 as a fraction of d_1, such that the condition of Theorem 1 is satisfied for the regions with slow variations. For instance, in our simulations, we assume that $d_2 = \frac{1}{2}d_1$. Therefore, if the variations of the input signal is below a certain threshold, η, then then distance between the levels becomes half in order to take more samples. The threshold value can be selected empirically. For instance, in our simulations, we choose $\eta = \sigma_{x'} Q^{-1}\left(\frac{1}{2}\right)$, where $Q^{-1}(.)$ denotes the inverse of the well-known Q-function. In the next section, we try to mathematically characterize the impact of the quantization on the reconstruction performance.

3.1. Error Model for the Time Quantization

In this part, we investigate the impact of the quantization on the reconstruction performance. As we mentioned earlier, in LC sampling schemes,

each sample is represented by two parameters: time difference and the corresponding crossing level. The crossing level values can be encoded by $[\log_2(L)]$ bits for LC, one bit for ALC and two bits for 2-level ALC. However, the time difference values need to be quantized by a local timer. The precision of this quantizer is TR seconds, which is the minimum time unit of the local quantizer. By decreasing the value of the TR (or increasing TR^{-1}), the precision of the quantization will be increased. Obviously, by increasing the TR^{-1}, more bits are required in order to represent the time difference between consecutive samples.

Fig. 4 shows the error model of the time quantization. Suppose that we want to quantize the time parameter of point B using the local timer. Due to the quantization error, we will get to point D instead of point B, which results in an error of e_j in the amplitude. Under the linear assumption of the input signal in $[t, t+\tau_j]$, we have $e_j = x'(t) \times \delta t$, where $x'(t) = \frac{d}{dt}x(t)$ and $\delta t = \tau_j - \hat{\tau}_j$. It can be easily confirmed that $x'(t)$ and δt are statistically independent. Furthermore, we assume that δt is uniformly distributed in the time interval of $[-\mathrm{TR}/2, \mathrm{TR}/2]$.

For a zero-mean Gaussian process with variance of σ_x^2, the derivative, $x'(t)$, is also a zero-mean Gaussian process with variance of $\sigma_{x'}^2$ which is obtained as follows:

$$\sigma_{x'}^2 = \int_{-B}^{B} (2\pi f)^2 S(f) df = (2\pi B)^2 \frac{\sigma_x^2}{3}. \tag{13}$$

Therefore, we have

$$\sigma_e^2 = E\{e_j^2\} = \sigma_{x'}^2 \frac{\mathrm{TR}^2}{12} = (2\pi B)^2 \frac{\sigma_x^2}{3} \frac{\mathrm{TR}^2}{12}. \tag{14}$$

Let the normalized distortion (ND) denote the ratio of the error power (σ_e^2) to the signal power (σ_x^2) [14]. We then have,

$$\text{ND (dB)} = 10\log_{10}\big(\frac{\sigma_e^2}{\sigma_x^2}\big) = 0.4 - 20\log_{10}(\frac{TR^{-1}}{B}). \tag{15}$$

From Eq. 15 it can be easily confirmed that as TR^{-1} increases, ND decreases. Therefore, the reconstruction quality of the input signal, which is proportional to the inverse of ND, is an increasing function of TR^{-1}. Fig. 5 compares the reconstruction quality as a function of TR^{-1} for different sampling schemes (Check Section 5 for the values of the underlying parameters of each sampling scheme). As can be seen, for all sampling scheme reconstruction quality improves as TR^{-1} increases.

4. The Iterative Algorithm

Reconstruction of a signal from its non-uniform samples can be improved by using the iterative algorithm. In this algorithm, by successive use of a crude reconstruction method, we can improve the reconstruction quality and we may reach the original signal without error. A block diagram of this method is shown in Fig. 6.

Let $\mathcal{G}\{.\}$ denote the distortion operator which operates on the input signal $x(t)$ and results in the output $y(t) = \mathcal{G}\{x(t)\}$. Consider the case, where we are interested to obtain $x(t)$ from $y(t)$. A simple approach is to use the inverse operator, i.e. $x(t) = \mathcal{G}^{-1}\big\{y(t)\big\}$. However, finding the inverse operator in general is a challenging problem. Therefore, we need a technique to approximate $\mathcal{G}^{-1}$ from $\mathcal{G}$. Next we present the iterative algorithm of [11] and explain how we utilize this algorithm to improve the performance of the

LC A/D converters. We have,

$$x^{(k)}(t) = x^{(k-1)}(t) + \lambda\big(\mathcal{G}\{x(t)\} - \mathcal{G}\{x^{(k-1)}(t)\}\big) \tag{16}$$

where $x^{(k)}(t)$ denotes the reconstructed signal after kth iterations and λ represents the relaxation parameter that determines the convergence rate. In general, the distortion operator $\mathcal{G}\{.\}$ can be either a linear or a non-linear operator. The discrete version of the iterative algorithm for the linear distortion operator can be reformulated as:

$$X^{(k)} = \widehat{G}X + (I - \widehat{G})X^{(k-1)} \tag{17}$$

where X and $X^{(k)}$ are $N \times 1$ vectors representing the input signal and the reconstructed one after kth iterations respectively. $\widehat{G} = \lambda G$ is an $N \times N$ matrix, which represents the linear distortion operator and I is an $N \times N$ identity matrix. Define $E \triangleq I - \widehat{G}$. The reconstructed signal after kth iteration can be then characterized as:

$$X^{(k)} = (E^k + E^{k-1} + \cdots + E + I)X^{(0)}, \quad X^{(0)} = \widehat{G}X. \tag{18}$$

Thus, we have,

$$X^{(k)} = \big(I - E\big)^{-1}\big(I - E^{k+1}\big)X^{(0)}. \tag{19}$$

If the norm of the operator E satisfies $\|E\| < 1$, we then have $\lim_{k\to\infty} X^{(k)} = \widehat{G}^{-1}X^{(0)} = X$. Hence, $X^{(k)}$ converges to the original input signal X. Next, we explain how we utilize the iterative algorithm to improve the reconstruction performance of the level crossing sampling A/D converters.

Let $\{t_n\}_{i=1}^{N}$ denote the time instances of the non-uniform samples. The sampled signal can be then represented as $x_s(t) = \sum_{i=1}^{N} x(t_i)\delta(t - t_i)$. A

simple way to reconstruct the input signal is to pass $x_s(t)$ through an interpolator, followed by a low-pass filter (LPF). Therefore, there exists two ways to choose the $\mathcal{G}\{.\}$. One way is to consider LC sampler, linear interpolator and LPF as $\mathcal{G}\{.\}$ operator. We call this approach *inverse system (IS) iterative method*. In this method, the LC sampling should be performed in each iteration. This method is computationally complex, since it requires to pass the signal through the level crossing sampler in each iteration. Another method is to choose operator $\mathcal{G}\{.\}$ as shown in Fig. 7. In this method, LC sampling is only performed at the first iteration and the time indices of the non-uniform samples are reserved for the next iterations. We call this approach *non-uniform sampling (NUS) iterative method*. This method is more suitable for the real time applications. In general the delay of the proposed algorithm is a function LC sampling delay, LPF delay and number of iteration. Furthermore, we expect to get the better reconstruction performance as we increase the number of iterations in the proposed algorithm.

5. Simulation Results

In this section, we present our simulation results and show how iterative algorithm can improve the performance of LC A/D converters drastically. For the simulations, the input signal is a zero-mean and unit variance Gaussian with the bandwidth of 4kHz. Furthermore, we utilize 3rd order IIR filter for low-pass filtering. All the simulations are performed in Matlab. Note that in the computer simulations, we model the analog signal by a densely-sampled discreet signal. The performance metric is the average SNR of the reconstructed signal, which is defined as follows: $\text{SNR} = \mathbf{E}_x\left\{\frac{\int x^2(t)dt}{\int (x^2(t)-\hat{x}^2(t))^2 dt}\right\}$,

is averaged over several different random input signals.

As we mentioned in Section 3, the OSR values depend on the crossing levels as well as the input signal. In order to do a fair comparison, the underlying parameters of each sampling scheme are chosen such that different sampling schemes achieve approximately the same values of OSR. For instance, in the standard LC, the value of L is chosen such that the crossing levels fully cover the dynamic range of the input signal and $d = 0.5\sigma_x$. Similarly, for ALC, $d = 0.5\sigma_x$ and $\delta = d/1000$. Finally, for 2-level ALC, the underlying parameters are set as follows: $d_1 = 0.5\sigma_x$, $d_2 = 0.25\sigma_x$ and $\delta = d/1000$.

Table 1 compares the required number of bits for the time quantization in different sampling schemes. As can be seen, 2-level ALC approximately requires one bit less than ALC and standard LC for the time quantization. That is due to the fact that, 2-level ALC generates more uniform samples, which decreases $T_{\max}$ and the resulting number of quantization bits.

5.1. Performance Improvement Using IS Iterative Method

In this part we show how the level crossing A/D converters can improve their reconstruction performance by utilizing the introduced *IS iterative method* of Section 4. In this method, the operator $\mathcal{G}\{.\}$ consists of the level crossing sampler followed by the interpolator and low pass filter. Therefore, the level crossing sampling is repeated in each iteration. Fig. 6 shows the SNR of the reconstructed signal versus the number of iterations for 2-level ALC A/D converters. This figure also compares the quality of the reconstructed signal for different time resolutions as well as OSR values. As can be seen, the reconstruction quality improves as the number of

iteration increases, however, it becomes saturated after a certain number of iterations. Several observations can be made. First, the saturated value of SNR depends only on the resolution of the local timer. For instance, the saturated values of SNR for both OSR = 3.47, 5.36 and TR^{-1} = 4098kHz are 65dB. Furthermore, the saturated value of SNR decreases as TR^{-1} decreases, which confirms our analytical results of Section 3.1. Another observation is that the OSR values only affect the convergence rate and have no impact on the saturated value of the SNR. For instance, if TR^{-1} = 4098kHz, case of OSR = 5.36 requires only 50 iterations to reach the saturated value, however, case of OSR = 3.47 reaches this value after 80 iterations.

5.2. Performance Improvement Using NUS Iterative Method

As we discussed in Section 4, in *NUS iterative method* the level crossing sampling is performed only at the first iteration. Therefore, the power consumption due to the sampling should be only considered for the first iteration. In general two types of the power consumptions are defined for A/D converters, static and dynamic. The static power consumption is directly related to the A/D technology and can be measured once the converter is implemented. However, the dynamic power consumption is caused by signal transitions in the circuit. A higher operating frequency leads to more frequent signal transitions and increases the power consumption. The most significant source of dynamic power consumption in CMOS circuits is the charging and discharging of capacitance. This can be modeled as follows

[21]:

$$P_{\text{dynamic}} = \underbrace{f_s.C_{\text{load}}.V_{dd}^2}_{\text{power consumption due to the operating frequency}} + \underbrace{f_{clc}.C'_{\text{load}}.{V_{dd}}^2}_{\text{power consumption due to the local timer}} \quad (20)$$

where C_{Load} and C'_{Load} are the corresponding capacitances and in the range of $1-2$ Picofarad and $100-200$ Femtofarad, respectively. V_{dd} is the voltage swing. f_s and f_{clc} are the operating frequency of the circuit and the frequency of the local timer, respectively. By comparing the operating frequency of the asynchronous converter with the synchronous ones and considering the fact that $C'_{Load} \ll C_{Load}$, it can be easily confirmed that asynchronous circuits are more efficient in terms of the dynamic power consumption. Next, we show how the proposed NUS iterative method drastically improves the performance of the asynchronous level crossing converters.

Fig. 9 shows the SNR of the reconstructed signal versus the number of iterations for different level crossing sampling schemes. Furthermore, this figure compares the reconstruction performance for different time resolutions. As can be seen, 2-level ALC outperforms both ALC and standard LC schemes in terms of the asymptotic SNR value and the convergence rate. For instance, in case of $\text{TR}^{-1} = 8192$kHz, 2-level ALC scheme requires 20 iterations to reach the saturated value of SNR, however, ALC and LC scheme require 35 and 40 iterations to reach their saturated values, respectively. This figure also shows that for a given TR, SNR of the reconstructed signal can not be improved beyond a certain threshold. In order to achieve a better reconstruction quality, TR^{-1} needs to be increased. More simulation results for

$\mathrm{TR}^{-1} = 8192\mathrm{kHz}$ are also provided in Table 2.

By comparing different sampling schemes, it can be deduced that 2-level ALC performs the best and the ALC is better than the standard LC converter. Not surprisingly, the complexities of the encoders are inversely proportional to their performances. Note the standard LC method, unlike the ALC and the 2-Level ALC methods, needs to store the quantized amplitude of each sample and hence requires more bandwidth for its implementation. Next, we compare the performance of the LC A/D converters with the conventional Sigma-Delta converters.

5.3. Comparison of the 2-Level ALC Method with Sigma-Delta ($\Sigma\Delta$) A/D converters

Sigma-Delta ($\Sigma\Delta$) A/D converters have been extensively used in practical applications. Therefore, in this part we try to compare the performance of the proposed level crossing converters with $\Sigma\Delta$ A/D converter. In [22], authors showed that the iterative algorithm can also be utilize to improve the performance of $\Sigma\Delta$ converters. For $\Sigma\Delta$ converters, unlike the LC converters, the positions and values of the non-uniform samples are not known and hence the iterative method consists of the encoder and decoder to form the distortion operator [22]. This type of iteration is much more complex than the operator $\mathcal{G}\{.\}$ of the 2-level ALC method. A comparison of these two techniques is shown in Table 3. This table compares the reconstruction quality of the $\Sigma\Delta$ and 2-Level ALC A/D converters for different iterations. In this table, we assume that the OSR multiplied by the number of quantizer bits are the same for both $\Sigma\Delta$ and 2-Level ALC methods. As can be seen, 2-Level ALC method outperforms the $\Sigma\Delta$ converter both in terms of performance

and computational complexity. For instance, after 20 iterations, the SNR of the reconstructed signal using 2-level ALC is about 20 dB more than Sigma-Delta A/D converter with OSR = 32 and one bit quantizer. Table 3 also compares the computational time (CT) per iteration in seconds unit time. Note that all results shown here were computed on an Intel Pentium 4 CPU 2.80 GHz machine with 1GB of memory. As can be seen, 2-level ALC scheme is 100 times faster than the conventional Sigma-Delta converters.

6. Conclusion

In this paper, we introduced two novel non-uniform sampling techniques for A/D converters which are based on adaptive and multi-level adaptive level crossing algorithms. The proposed solutions can achieve better signal to noise ratio as well as reduction of power consumption and complexity compared to conventional A/D approaches. We have shown that the performance of these schemes can be improved by deploying the iterative algorithms. Iterative algorithm is quite easy to implement without changing configuration of the A/D converter. Our simulation results show that the proposed 2-level ALC scheme of this paper yields superior reconstruction quality and outperforms ALC and standard LC schemes.

References

[1] R. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 539–550, Apr. 1999.

[2] M. Unser, "Sampling-50 years after shannon," *Proceedings of the IEEE*, vol. 88, pp. 569–587, Apr. 2000.

[3] R. Schreier and G. Temes, *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, Nov. 2004.

[4] M. Kafashan, S. Beygiharchegani, and F. Marvasti, "Performance improvement of an optimal family of exponentially accurate sigma delta modulator," *10th IEEE International Conference on Signal Processing (ICSP)*, pp. 1–4, 2010.

[5] M. Renaudin, "Asynchronous circuits and systems: a promising design alternative," *Microelectronic Engineering*, vol. 54, no. 1-2, pp. 133–149, 2000.

[6] B. Schell and Y. Tsividis, "Analysis and simulation of continuous-time digital signal processors," *Signal Processing*, vol. 89, no. 10, pp. 2013–2026, 2009.

[7] D. Kinniment, A. Yakovlev, F. Xia, and B. Gao, "Towards asynchronous A/D conversion," *International Symposium on Asynchronous Circuits and Systems*, vol. 0, pp. 2–6, 1998.

[8] F. Akopyan, R. Manohar, and A. B. Apsel, "A level-crossing flash asynchronous analog-to-digital converter," *International Symposium on Asynchronous Circuits and Systems*, vol. 0, pp. 12–22, 2006.

[9] D. Kinniment, A. Yakovlev, and B. Gao, "Synchronous and asynchronous A/D conversion," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, pp. 217–220, apr 2000.

[10] D. Nairn, "Time-interleaved analog-to-digital converters," *IEEE 2008 Custom Integrated Circuits Conference (CICC)*, pp. 289 –296, sept. 2008.

[11] F. Marvasti, *Nonuniform sampling: theory and practice*. New York: Kluwer Academic/ plenum, 2001.

[12] P. Stoica and N. Sandgren, "Spectral analysis of irregularly-sampled data: Paralleling the regularly-sampled data approaches," *Digital Signal Processing*, vol. 16, no. 6, pp. 712 – 734, 2006.

[13] A. Lazar and L. Toth, "Time encoding and perfect recovery of bandlimited signals," *Proceedings of the ICASSP*, vol. 4, pp. 709–712, April. 2003.

[14] J. Mark and T. Todd, "A nonuniform sampling approach to data compression," *IEEE Transactions on Communications*, vol. 29, pp. 24–32, Jan. 1981.

[15] I. Blake and W. Lindsey, "Level-crossing problems for random processes," *IEEE Transactions on Information Theory*, vol. 19, no. 3, pp. 295–315, 1973.

[16] N. Sayiner, H. Sorensen, and T. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, pp. 335–339, Apr. 1996.

[17] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "Asynchronous level crossing analog to digital converters," *Measurement*, vol. 37, pp. 296–309, Mar. 2005.

[18] K. Guan and A. Singer, "A level-crossing sampling scheme for non-bandlimited signals," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, May 2006.

[19] C. Vezyrtzis and Y. Tsividis, "Processing of signals using level-crossing sampling," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2293–2296, May 2009.

[20] 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*, vol. 57, pp. 982–991, May 2010.

[21] R. Plassche, *CMOS integrated analog-to-digital and digital-to-analog converters*. Kluwer Academic Publishers, 2003.

[22] M. Kafashan, M. Ghorbani, and F. Marvasti, "A sigma-delta analog to digital converter based on iterative algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, p. 149, 2012.

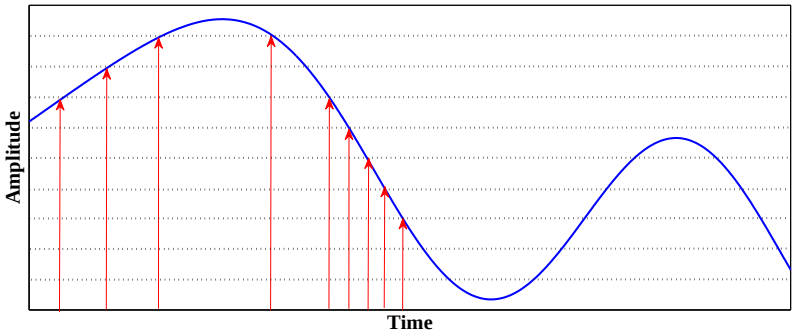

Figure 1: Non-uniform sampling using LC scheme

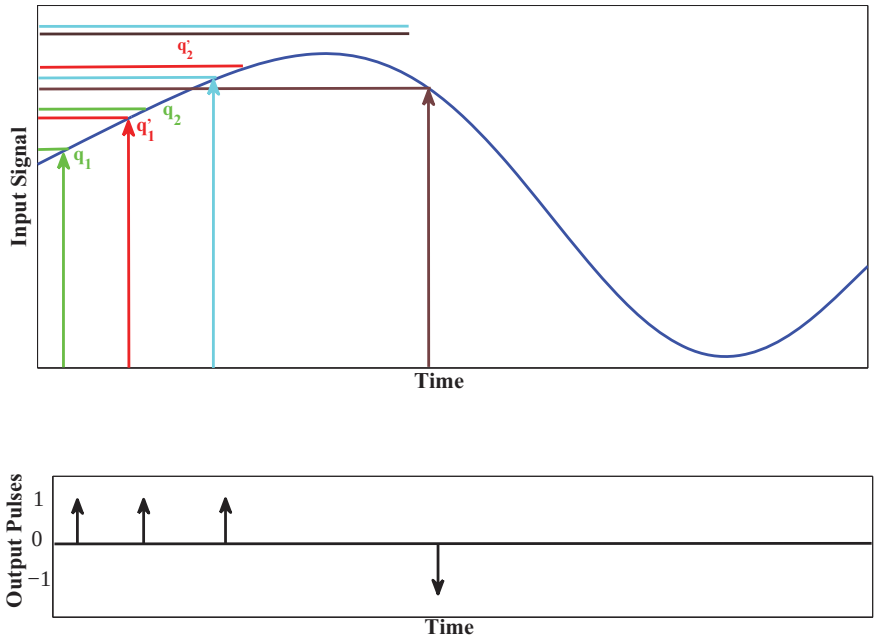

Figure 2: Non-uniform sampling based on ALC

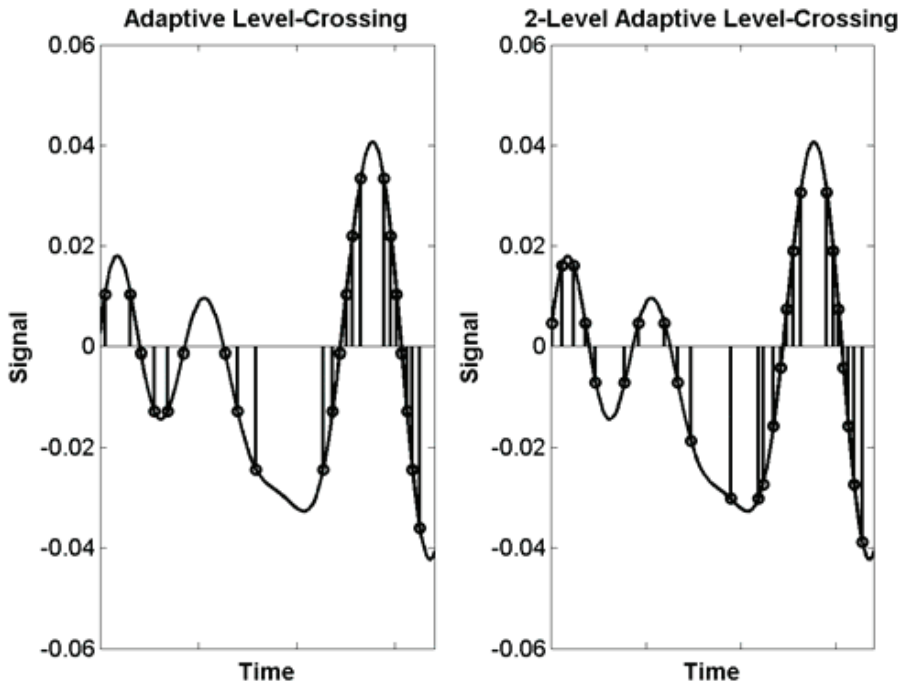

Figure 3: Difference between ALC and 2-Level ALC

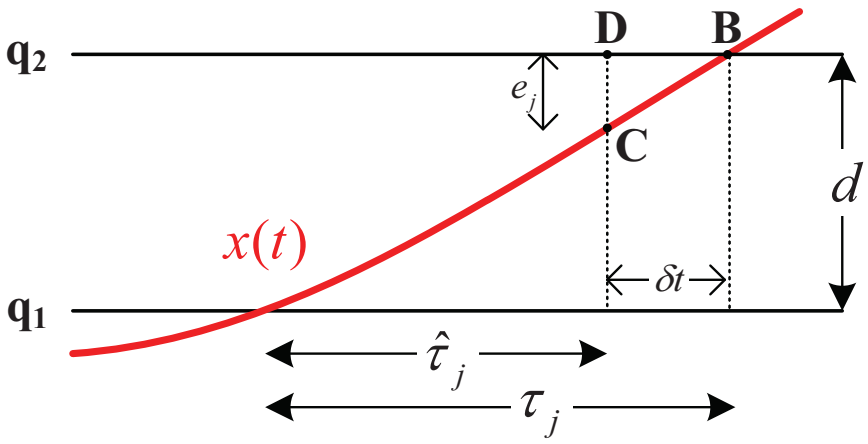

Figure 4: Error model for Time quantizer

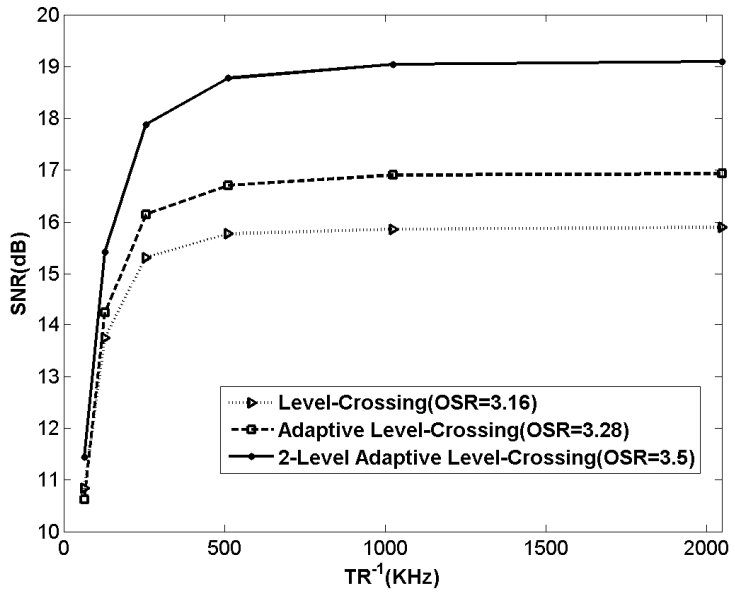

Figure 5: SNR of the reconstructed signal versus TR^{-1} for three sampling schemes, $f_{\mathrm{Nyquist}} = 8\mathrm{kHz}$

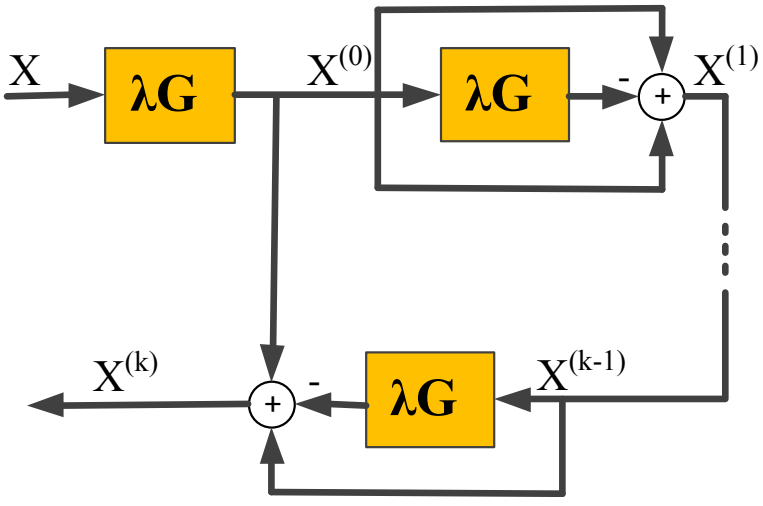

Figure 6: The configuration of the iterative method

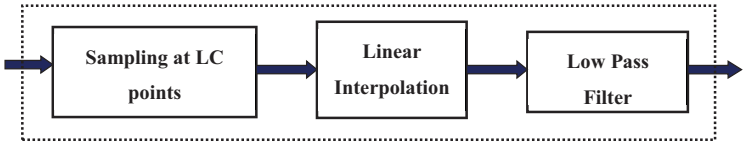

Figure 7: Block diagram of the distortion operator G in the iterative algorithm

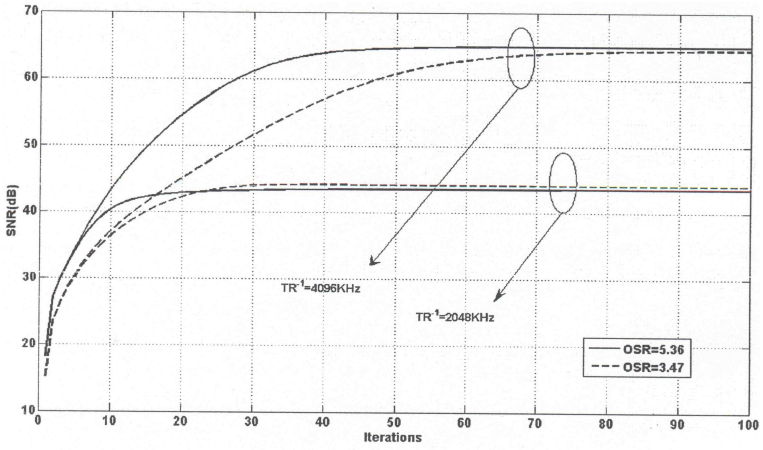

Figure 8: SNR versus number of iterations for 2-level ALC using IS iterative method for $\mathrm{TR}^{-1} = 2048,\ 4096$ kHz and for two different OSR

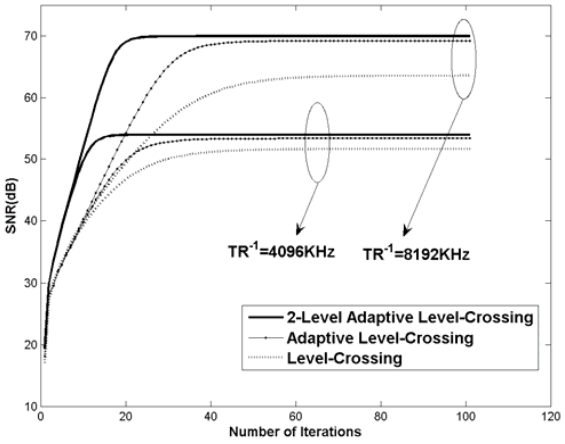

Figure 9: SNR versus number of iterations using NUS iteartive method for different sampling schemes and $\mathrm{TR}^{-1} = 4096,\ 8192$ kHz, and $\lambda = 1.4$

Table 1: Number of bits for quantization of time differences versus precision of the time quantizer ($\mathrm{TR}^{-1} = 8192\mathrm{kHz}$)

TR^{-1}	LC OSR = 2.8	ALC OSR = 2.9	2-level ALC OSR = 3.1
64	5	4	4
128	6	6	5
256	7	7	6
512	8	8	7
1024	9	9	8
2048	10	10	9
4096	11	11	10

Table 2: SNR versus number of iterations for $\mathrm{TR}^{-1} = 8192$ kHz

Iteration Number	LC		ALC		2-level ALC	
	OSR=3.16	OSR=3.97	OSR=3.16	OSR=3.97	OSR=3.1	OSR=3.97
0	17.13	18.75	18.11	19.99	20.20	22.50
5	34.96	53.59	33.97	59.81	69.18	69.76
10	41.12	62.49	41.15	69.91	69.81	70.25
20	50.17	68.84	55.80	70.20	69.99	70.25
40	60.89	69.39	70.25	70.20	70.00	70.25
Saturate	63.56	69.36	70.29	70.20	70.00	70.25

Table 3: Comparison of SNR in $\Sigma\Delta$ and 2-level ALC A/D converter (CT=computational time per iteration in seconds unit time)

Iteration Number	$\Sigma\Delta$ OSR=32 q=1 bit CT=4.10 s	$\Sigma\Delta$ OSR=16 q=2 bits CT=4.19 s	2-level ALC OSR = 2.48 CT=.05 s
0	27.02	20.92	14.50
5	42.21	35.91	33.32
10	43.74	37.39	43.51
20	44.30	37.47	57.00
40	44.60	37.80	63.16