Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 513706, 11 pages
doi:10.1155/2008/513706

Research Article

Adaptive Reference Levels in a Level-Crossing Analog-to-Digital Converter

Karen M. Guan,[1] Suleyman S. Kozat,[2] and Andrew C. Singer[1]

[1] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 60801, USA*
[2] *Department of Computer Engineering, College of Engineering, Koc University, 34450 Istanbul, Turkey*

Correspondence should be addressed to Andrew C. Singer, acsinger@uiuc.edu

Received 24 October 2007; Revised 30 March 2008; Accepted 30 June 2008

Recommended by Sergios Theodoridis

Level-crossing analog-to-digital converters (LC ADCs) have been considered in the literature and have been shown to efficiently sample certain classes of signals. One important aspect of their implementation is the placement of reference levels in the converter. The levels need to be appropriately located within the input dynamic range, in order to obtain samples efficiently. In this paper, we study optimization of the performance of such an LC ADC by providing several sequential algorithms that adaptively update the ADC reference levels. The accompanying performance analysis and simulation results show that as the signal length grows, the performance of the sequential algorithms asymptotically approaches that of the best choice that could only have been chosen in hindsight within a family of possible schemes.

1. INTRODUCTION

Level-crossing (LC) sampling has been proposed as an alternative to the traditional uniform sampling method [1–10]. In this approach, signals are compared with a set of reference levels and samples taken on the time axis, indicating the times at which the analog signal exceeded each of the associated reference levels. This threshold-based sampling is particularly suitable for processing *bursty* signals, which exist in a diverse range of settings from natural images to biomedical responses to sensor network transmissions. Such signals share the common characteristic that information is delivered in bursts, or temporally sparse regions, rather than in a constant stream. Sampling by LC visibly mimics the behavior of such input signals. When the input is bursty, LC samples also arrive in bursts. When input is quiescent, fewer LC samples are collected. As such, LC lets the signal dictate the rate of data collection and quantization: more samples are taken when the signal is bursty, and fewer when otherwise. One direct benefit of such sampling is that it allows for economical allocation of resources. Higher instantaneous bandwidth/precision can be offered when sampling is performed, and resolution is improved without overall increase in bit rate or power consumption. It has been shown in [4, 6, 7] that by using LC sampling in communication systems, we can reduce the data transmission rate. For certain types of input, it has also been shown that LC performs advantageously in signal reconstructions, as well as in parameter estimations.

The opportunistic nature of LC sampling is akin to that of compressed sensing [11, 12], where by recognizing many signals in nature are sparse—a term that describes signals whose actual support in some representation or basis is much smaller than their aggregate length in the basis with which the signal is described, more economical conversion between the analog and the digital domain can be achieved. Recent work [11–15] has shown sparse signals can be reconstructed exactly from a small number of random projections and through a process employing convex optimization. While this framework of reconstruction by random projection is theoretically intriguing, it behaves poorly when measurements are noisy. It is shown in [16] that signal-to-noise ratio (SNR) decreases successively as the number of projections increases, rendering it a less-attractive solution in practical implementations. LC similarly exploits the sparse (bursty) nature of signals by sampling, intuitively, where information is located. Furthermore, it is structurally stable, and various hardware designs have been offered [8–10]. It does not escape our attention that the advantages exhibited by LC sampling in both data transmission and signal reconstruction hinge on the proper placement of reference levels. Ideally, the levels are located such that information can be optimally extracted. In the literature, the levels have typically been treated no differently from uniform quantization levels [4–10], where their optimal allocation has received scant consideration, with the noted exception quantization of data that has already been sampled in time. Hence, optimal placement of reference levels is the focus of this paper.

In order to obtain samples efficiently, the levels need to be appropriately assigned in the analog-to-digital converter (ADC). When they are not within the amplitude range of the input, no LCs are registered, hence information can be lost. On the other hand, when too many levels are employed, more samples than necessary could be collected, rendering the system inefficient. Naturally prior information, such as the source's a priori distribution or signal model, can help to decide where the levels should be placed. Based on statistics of the input, Lloyd-Max quantization method can be employed to select a nonuniformly spaced level set to minimize the quantization error. However, statistical information is often not available and/or difficult to obtain. Furthermore, when an implementation relies on an empirically obtained model, a mismatch between that and realistic scenarios has to be taken into account. The more assumptions are made, the more justifications are needed later. In this work, we start with just one assumption: only the input dynamic range is known. Inspired by seminal work on zero-delay lossy quantization [17, 18], we implement an adaptive scheme that sequentially assigns levels in the ADC. This scheme yields performance comparable to that of the best within a family of fixed schemes. In other words, we can do almost as well as were the best fixed schemes known all along. Before delving into this implementation, we will touch upon a conceptual design of the level-crossing analog-to-digital converter (LC ADC).

The organization of the paper is as follows. In Section 2, we provide an architecture for LC ADC and describe one possible implementation of LC ADC. We then introduce sequential algorithms in Section 3, where we also provide complete algorithmic descriptions and corresponding guaranteed performance results. The paper then concludes with a number of simulations of the algorithms described on biological signals collected using an LC ADC.

2. IMPLEMENTATION OF REFERENCE LEVELS

In this section, we present a conceptual architecture for LC ADC and the setup for the placement of reference levels in the ADC. Furthermore, we define the reconstruction error that will be minimized with a sequential algorithm in Section 3.

2.1. A conceptual architecture for LC ADC

A range of publications have investigated the hardware implementation of asynchronous LC samplers [8–10]. In particular, the LC asynchronous ADC presented in [10] has a parallel structure that resembles a flash-type ADC. The current implementation can sample signals upto 5 MHz in bandwidth with 4 bits hardware resolution, and its topology can be trivially extended to a higher-precision ADC. The proposed architecture is given in Figure 1, and it is the LC ADC we refer to throughout this paper.

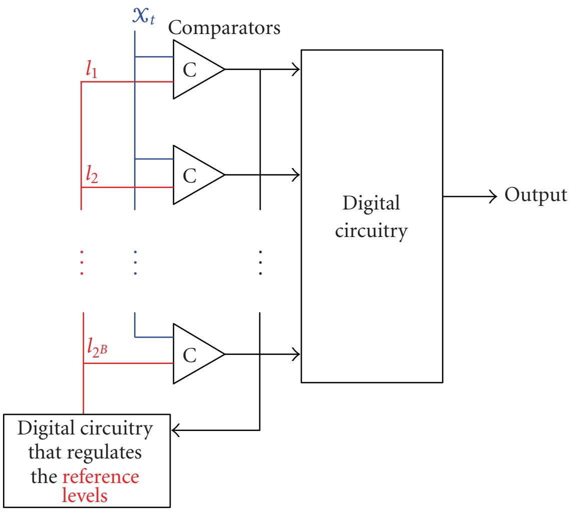


Figure 1: A conceptual design diagram of a B-bit flash-type LC ADC.

Let us consider a B-bit (2^B levels) flash-type ADC of this design. It is equipped with an array of 2^B analog comparators that compare the input with corresponding reference levels. The reference levels are implemented with a voltage divider. The comparators are designed to be noise resistant, so at a reference level, fluctuation due to noise will not cause chattering in the output. The power consumption of such analog circuitry is dominated by the comparators. In order to minimize power, at most p of the 2^B comparators are *on* at any moment. This can be accomplished by a digital circuit that regulates the power supply and periodically updates the set of *on* comparators. The asynchronous digital circuitry processes the output of the analog circuitry, recognizes the proper times for each of the LCs, then outputs a sequence of bits.

The number of *on* comparators (p) and their respective amplitudes affect the performance of the LC ADC. Ideally, they are optimized jointly. However, for analytical tractability, we temporarily suppress the variability of p in our formulation. The distortion measure is formulated as a function of the levels, and it is minimized within a family of schemes.

2.2. The reference level set ℓ

Let us consider an amplitude-bounded signal x_t that is T-second long. Without loss of generality, we assume x_t is bounded between $[-A/2, +A/2]$, and that the LC ADC has 2^B levels uniformly spaced in the dynamic range with spacing $\delta = A/2^B$. Let $\ell = \{\ell_1, \ell_2, \ell_k, \dots, \ell_{2^B}\}$ represents the set of reference levels used by the comparators. The cardinality of ℓ is $|\ell| = 2^B$.

During LC sampling, let p comparators be turned *on* at any given time. Together these p comparators form a level set, which is a subset of ℓ. In our framework, this set is updated every v seconds, that is, at $t = nv$, $n = 1, 2, \ldots$, a new set of levels is picked and this new set of levels is represented as $L_n = \{L_{n,1}, \ldots, L_{n,m}, \ldots, L_{n,p}\}$, $L_{n,m} \in \ell$. Let L^t denote the sequence of such level sets used up to time t, that is, $L^t = (L_0, L_1, \ldots, L_n, \ldots, L_{\lfloor t/v \rfloor})$, where each L_i is a set of p levels.

The ADC compares the input x_t to the set of levels used every τ seconds. Note that $\tau \neq v$. The ADC records a level crossing with one of $L_{n,m}$ if the following comparison holds for a $L_{n,m}$:

$$(x_{(n-1)\tau} - L_{n,m})\,(x_{n\tau} - L_{n,m}) < 0, \quad m = 1, \ldots, p. \tag{1}$$

Although the true crossing s_i occurs in the interval $[(n-1)\tau, n\tau)$, only its quantized value $Q(s_i)$ is recorded, that is, $Q(s_i) = (n-1)\tau + \tau/2$. The LC sample acquired by the ADC is $(Q(s_i), \lambda_i)$, where λ_i is the corresponding level crossed at $t = s_i$, $x(s_i) = \lambda_i \in L_n$. Since λ_i is enunciated in ℓ, it is known with perfect precision. This is the key difference between quantization of LC samples from that of uniform samples: uniform samples are quantized in amplitude, LC samples are quantized in time. Furthermore, we also provide an analysis of the bandwidth that can be handled by an LC ADC for perfect reconstruction in Appendix A.

2.3. Reconstructed signal and its error

Given a sequence of reference levels L^t, sampling input x_t with L^t produces a set of samples $L_c(x_t, L^t) = \{(Q(s_i), \lambda_i)\}_{i \in \mathbb{Z}^+}$. The corresponding reconstructed signal at time t, using a piecewise constant (PWC) approximation scheme, is given by

$$\hat{x}_t(L^t) = \sum_i \lambda_i [u(t - Q(s_i)) - u(t - Q(s_{i+1}))], \quad 0 \le t \le T, \tag{2}$$

where $u(t)$ is a unit step function, that is, $u(t) = 1$ when $t \ge 0$ and $u(t) = 0$, otherwise. It is entirely possible that $L_c(x_t, L^t)$ produces an empty set if no crossings occur between levels sets and x_t, which means no information has been captured. As such, finding an appropriate sequence of reference levels is essential. The reconstruction error over an interval of T is given by

$$e(L^T) = \int_0^T (x_t - \hat{x}_t(L^t))^2 dt. \tag{3}$$

From (2) and (3), it is clear that the MSE $e(L^T)$ is a function of the chosen sequence of reference levels L^T. As such, it will be minimized with respect to L^T.

We also note that the quantization levels used in (2) need not coincide with the decision levels such that we can use

$$\hat{x}_t(L^t) = \sum_i f(\lambda_i)[u(t - Q(s_i)) - u(t - Q(s_{i+1}))], \quad 0 \le t \le T, \tag{4}$$

for reconstruction with a generic $f(\cdot)$. For example, we can select $f(\lambda_i) = \lambda_i \pm \delta/2$, depending on the direction of the crossing at time t_i. Such a reconstruction scheme is consistent with the input, and it has been shown to yield very good performance when the sample resolution is high [13, 14]. Since signal reconstruction is not the focus of this paper, we only provide the appropriate references [13, 14] and continue with (2).

3. GETTING THE BEST HINDSIGHT PERFORMANCE SEQUENTIALLY

In this section, we introduce a sequential algorithm that is implemented to asymptotically achieve the performance of the best constant scheme known in hindsight. This sequential algorithm is a randomized algorithm. At fixed intervals, the algorithm randomly selects a level set and uses it to sample the input until the selected level set is replaced by the next selection. The level set is randomly selected from a class of possible level sets according to a probability mass function (PMF) generated by the cumulative performance of each level set in this class on the input.

3.1. The best constant scheme known in hindsight

Before we present a sequential algorithm that searches for L^T, we discuss the shortcomings of the constant (nonadaptive) scheme. When levels are not updated, we pick a set L_0 of p levels at $t = 0$, and use it for the entire sampling duration T. The best constant reference level is one that minimizes the MSE among the class of all possible p-level sets $\mathcal{L}$, $|\mathcal{L}| = \binom{2^B}{p}$. It can be obtained by evaluating the following optimization problem:

$$L_0^* = \arg\min_{L_0 \in \mathcal{L}} \int_0^T (x_t - \hat{x}_t(L_0))^2 dt. \tag{5}$$

Evaluating (5), however, requires a delay of T seconds. In other words, the best constant level set L_0^* is only known in hindsight; it cannot be known a priori at the start. Without statistical knowledge of the input, optimizing performance while using a constant scheme is not feasible and a zero-delay and sequential algorithm may be more appropriate.

3.2. An analog sequential algorithm using exponential weights

The continuous-time sequential algorithm (CSA) uses the well-known exponential weighting method [18] to create a PMF, over the class of possible level sets at every update, from which a new set is generated. Figure 2 illustrates this algorithm pictorially, and the algorithm is given in Algorithm 1. In the algorithmic description, each level set is represented by $\mathcal{L}_k$, $k = 1, \ldots, |\mathcal{L}|$.

We note that in the implementation of Algorithm 1, the cumulative errors in (A1) are computed recursively.

```
Step 1.1: Initialize constant $\eta$, $\eta > 0$; initialize update interval $v$; $N = \lfloor T/v \rfloor$;
Step 1.2: Initialize reconstruction to 0, $\hat{x}_0 = 0$; initialize cumulative errors to zero, $e_0^k = 0$, $k = 1, \dots, |\mathcal{L}|$;
for $n = 1 : N$ do
  for $k = 1 : |\mathcal{L}|$ do
     Step 2.1: At $t = nv$, update the cumulative errors associated with each level set $\mathcal{L}_k$,
        (A1)
        $e_{nv}^k = e_{(n-1)v}^k + \int_{(n-1)v}^{nv} (x_t - \hat{x}_t(\mathcal{L}_k))^2 dt, \quad k = 1, \dots, |\mathcal{L}|.$
     Step 2.2: Update the weights such that
        (A2)
        $w_{nv}^k = \frac{\exp(-\eta e_{nv}^k)}{\sum_{j=1}^{|\mathcal{L}|} \exp(-\eta e_{nv}^j)}, \quad k = 1, \dots, |\mathcal{L}|.$
  end for
  Step 3.1: At $t = nv$, select $L_n$ according to the PMF
        (A3)
        $\Pr(L_n = \mathcal{L}_k) = w_{nv}^k, \quad k = 1, \dots, |\mathcal{L}|.$
  Step 3.2: Use the selected set $L_n$ to sample $x_t$ in the interval $[nv, (n+1)v)$ and update reconstructed signal,
        (A4)
        $\hat{x}_t(L_{\text{csa}}^{nv}) = \hat{x}_t(L_{\text{csa}}^{(n-1)v}) + \sum_{i \in I_n} \lambda_i [u(t - Q(s_i)) - u(t - Q(s_{i+1}))],$
where $\{Q(s_i), \lambda_i\}_{i \in I_n}$ is the sample set obtained by sampling $x_t$ with $L_n$ in the interval $[(n-1)v, nv)$.
  end for
```

Algorithm 1: Continuous-time sequential algorithm (CSA).

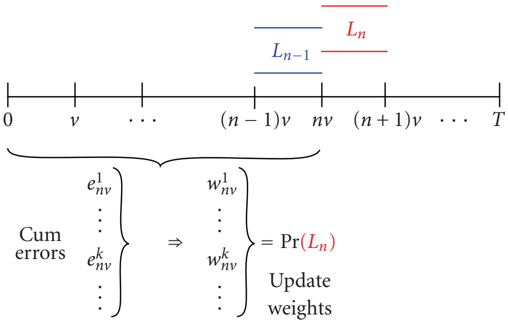


Figure 2: A diagram to illustrate the sequentially updated algorithm. At each $t = nv$, accumulated errors e_{nv}^k are used to generate weights w_{nv}^k.

Furthermore, the weights defined in (A2), in Algorithm 1, can be recursively computed as well:

$$w_{nu}^k = \frac{w_{(n-1)u}^k \exp\left(-\eta \int_{(n-1)u}^{nu} (x_t - \hat{x}_t(\mathcal{L}_k))^2 dt\right)}{\sum_{j=1}^{|\mathcal{L}|} w_{(n-1)u}^j \exp\left(-\eta \int_{(n-1)u}^{nu} (x_t - \hat{x}_t(L_j))^2 dt\right)}, \quad k = 1, \dots, |\mathcal{L}|. \tag{6}$$

As such, implementation of the CSA only requires storage of $|\mathcal{L}|$ weights.

3.3. Asymptotic convergence of the sequential algorithm

In this section, we give an assessment of the performance of the CSA. For clarity, we reiterate the setup here. Let L_{CSA}^T be a sequence of levels chosen by CSA up to time T. Let $\hat{x}_t(L_{\text{CSA}}^T)$ be the reconstructed signal obtained by sampling x_t with L^T, and let the expected MSE be given by $E[e_T(L_{\text{CSA}}^T)] = E[\int_0^T (x_t - \hat{x}_t(L_{\text{CSA}}^T))^2 dt]$. We note that the expectation in here is with respect to the PMF generated by the algorithm.

Theorem 1. *For any bounded input x_t of length T, $|x_t| \le A/2$, and fixed parameters η and v, reconstruction of input using the continuous-time sequential algorithm has MSE that satisfies*

$$\frac{1}{T} E[e_T(L_{CSA}^T)] \le \frac{1}{T} e(L_0^*) + \frac{\ln |\mathcal{L}| / \eta}{T} + \frac{\eta v (\rho A)^4}{8} + \frac{(\rho A)^2 v}{T}, \tag{7}$$

where ρ is a parameter of the LC ADC, $\rho = 1 - 1/2^B$. Selecting $\eta = \sqrt{8 \ln |\mathcal{L}| / (\rho A)^4 v T}$ to minimize the regret terms, one has

$$\frac{1}{T} E[e_T(L_{CSA}^T)] \le \frac{e(L_0^*)}{T} + O\left(\sqrt{\frac{\ln |\mathcal{L}|}{T}}\right). \tag{8}$$

As such, the normalized performance of the universal algorithm is asymptotically as good as the normalized performance of the best hindsight constant level set $\mathcal{L}_0^*$.

We see that the "regret" paid for not knowing the best level set in hindsight vanishes as signal length T increases. The parameter η can be considered as the learning rate of the algorithm, and at the optimal learning rate, $\eta = \sqrt{8 \ln |\mathcal{L}|/(\rho A)^4 vT}$, the regret is minimized. The regret is also a function of the amplitude range A and update period v. Intuitively, the smaller the update period, the more often the updates, and the smaller the regret. See Appendix B for the proof.

3.4. A digital approximation

In practical implementations where selection of reference levels is performed by a digital circuit, such as suggested by Figure 1, it is necessary to compute the cumulative errors (A1) in Algorithm 1 in the digital domain. As such, the continuous-time reconstruction error $e_t(L^t)$ formulated in the previous section needs to be approximated digitally, that is, the continuous-time integration in (A1) in Algorithm 1 needs to be replaced by discrete-time summation. One approach is to approximate the reconstruction error $e_t(L^t)$ with regular sampling and piecewise constant (or piecewise linear) interpolation. Furthermore, computation of the cumulative errors requires knowing the actual x_t, however, the original signal x_t is unknown (otherwise, we would not need a converter). As such, the feasibility of this type of sequential algorithm hinges on our ability to procure x_t in some fashion.

Assume that we periodically obtain quantized input to compute approximate versions of the cumulative errors. This can be accomplished in two ways.

(i) Once every μ seconds, all of the 2^B comparators are turned on. The value of μ is selected so that $\tau \ll \mu \ll v$, τ is the sampling period of the comparators and v is the interval between updates. Once a level is crossed by the input signal, the comparator associated with that level changes its output, then its corresponding digital trigger identifies the change and sends the information to the digital circuitry that controls the comparator's power supply. This method is shown in Figure 3(a), and it can periodically (every μ seconds) provide a quantized input $\widetilde{x}_{m\mu} = Q_B(x_{m\mu})$, $|\widetilde{x}_{m\mu} - x_{m\mu}| \leq \delta/2$. In our LC ADC, p comparators are on at any moment. By requesting all comparators be turned on every μ seconds, we in effect power up $(2^B - p)$ extra comparators every μ seconds. Since the extra comparators are only turned on for a small fraction of time, they likewise only consume a small fraction of the overall power.

(ii) A separate low-rate C-bit ADC keeps track of the input every μ seconds, $\widetilde{x}_{m\mu} = Q_C(x_{m\mu})$. This method is shown in Figure 3(b), and the low-rate (and low-power) ADC has a sampling frequency much lower than that of the comparators, with the goal of providing the digital circuitry, that performs the DSA, an approximated input every μ seconds, $|\widetilde{x}_{m\mu} - x_{m\mu}| \leq V_{FS}/2^{C+1}$. Here the C-bit ADC should have $C \geq B$ to efficiently represent the underlying signal. The advantage of this method is that quantized input can have arbitrary resolution, as long as it is affordable. The disadvantage is that a separate circuit element is designated to procure input approximations, and it needs to be synchronized with rest of the circuitry.

By employing either method, the approximated cumulative error $\widetilde{e}_t(\mathcal{L}_k)$ can be evaluated as follows:

$$\widetilde{e}_T(\mathcal{L}_k) = \sum_{m=0}^{NM} \left(\widetilde{x}_{m\mu} - \widehat{x}_{m\mu}(\mathcal{L}_k)\right)^2 \cdot \mu. \tag{9}$$

Other schemes such as nonuniform sampling in conjunction with splines or cubic polynomial interpolation can be used as well, depending on the underlying statistics and bandwidth of the signal x_t. The 0th order Riemann sum approximation in (9), though conservative, serves well in the absence of such information. We introduce the discrete-time sequential algorithm in Algorithm 2.

The approximation error redistributes the PMF $\Pr(L_n)$, and as a result, a different sequence of levels could be selected for sampling. Here, we quantify the deviation and show that the effect of approximation becomes negligible as signal length increases. In other words, the regret terms in Theorem 1 remain unchanged even when the cumulative errors are approximated. Let L_{dsa}^T be a sequence of levels chosen by the discrete-time algorithm. Let $\widehat{x}_t(L_{\mathrm{dsa}}^T)$ be the reconstructed signal obtained by sampling x_t with L_{dsa}^T, and let the expected MSE be given by $E[e_T(L_{\mathrm{dsa}}^T)] = E[\int_0^T (x_t - \widehat{x}_t(L_{\mathrm{dsa}}^T))^2 dt]$. Furthermore, let Δ_0 represent the difference between the continuous-time and discrete-time cumulative errors, $\Delta_0 = |e_T(L_0^*) - \widetilde{e}_T(L_0^*)|$, then $e_T(L_0^*) = \widetilde{e}_T(L_0^*) + \Delta_0$.

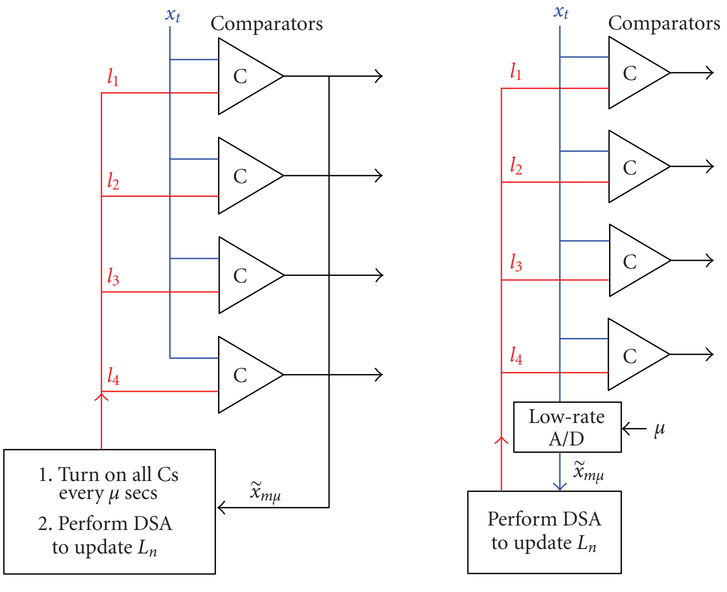
Theorem 2. *For any bounded input x_t of length T, $|x_t| \leq A/2$, and fixed parameters η and u, reconstruction of input using the discrete-time sequential algorithm (DSA) incurs MSE that is bounded by*

$$\frac{1}{T} E[e_T(L_{dsa}^T)] \leq \frac{1}{T}\left(\widetilde{e}_T(L_0^*) + \Delta_0\right) + \frac{\ln |\mathcal{L}|/\eta}{T} + \frac{\eta v (\rho A)^4}{8} + \frac{(\rho A)^2 v}{T}, \tag{10}$$

where ρ is a parameter of the LC ADC, $\rho = 1 - 1/2^B$. Selecting $\eta = \sqrt{8 \ln |\mathcal{L}|/(\rho A)^4 uT}$ to minimize the regret terms, one has

$$\frac{1}{T} E[e_T(L_{dsa}^T)] \leq \frac{1}{T}\left(\widetilde{e}(L_0^*) + \Delta_0\right) + O\left(\sqrt{\frac{\ln |\mathcal{L}|}{T}}\right). \tag{11}$$

See Appendix C for the proof. The parameter Δ_0 measures the distortion due to approximation. A meaningful



(a)

(b)

FIGURE 3: Two methods of tracking input to implement DSA. (a) All comparators are turned on once every μ seconds, and the approximated input $\tilde{x}_{m\mu}$ is send to the digital circuit to evaluate DSA. (b) A low-rate ADC keeps track of input x_t every μ seconds.

```
Step 1.1: Initialize constant $\eta$, $\eta > 0$; initialize update interval $u$; $N = \lfloor T/v \rfloor$;
Step 1.2: Initialize reconstruction to 0, $\hat{x}_0 = 0$; initialize cumulative errors to zero, $e_0^k = 0$, $k = 1, \dots, |\mathcal{L}|$;
for $n = 1 : N$ do
  for $k = 1 : |\mathcal{L}|$ do
    Step 2.1: At $t = nv$, update the cumulative errors associated with each level set $\mathcal{L}_k$,
        (B1)
        $\tilde{e}^k_{nv} = \tilde{e}^k_{(n-1)v} + \sum_{m=(n-1)M}^{nM-1} \left(\tilde{x}_{m\mu} - \hat{x}_{m\mu}(\mathcal{L}_k)\right)^2 \cdot \mu, \quad k = 1, \dots, |\mathcal{L}|.$
    Step 2.2: Update the weights such that
        (B2)
        $\tilde{w}^k_{nv} = \frac{\exp\left(-\eta \tilde{e}^k_{nv}\right)}{\sum_{j=1}^{|\mathcal{L}|} \exp\left(-\eta \tilde{e}^j_{nv}\right)}, \quad k = 1, \dots, |\mathcal{L}|.$
  end for
  Step 3.1: Select $L_n$ according to the PMF
        (B3)
        $\Pr(L_n = \mathcal{L}_k) = \tilde{w}^k_{nv}, \quad k = 1, \dots, |\mathcal{L}|.$
  Step 3.2: Use the selected set $L_n$ to sample $x_t$ in the interval $[nv, (n+1)v)$. Update the reconstructed signal,
        (B4)
        $\hat{x}_t(L^{nv}_{\text{dsa}}) = \hat{x}_t(L^{(n-1)v}_{\text{dsa}}) + \sum_{i \in I_n} \lambda_i [u(t - Q(s_i)) - u(t - Q(s_{i+1}))],$
  where $\{Q(s_i), \lambda_i\}_{i \in I_n}$ is the sample set obtained in the interval $[(n-1)v, nv)$.
end for
```

ALGORITHM 2: Discrete-time sequential algorithm (DSA).

bound on this distortion requires knowing the characteristics of x_t, for example, some measure of its bandwidth or its rate of innovation, as well as how the MSE is approximated. For example, let us consider a length-T piecewise constant signal with $2K$ degrees of freedom:

$$x_t = \sum_{i=1}^{K} a_i u(t - t_i), \quad 0 \le t \le T. \tag{12}$$

Such signal has a rate of innovation $r = 2K/T$ [19]. When the error metric is approximated using (B1) in Algorithm 2, a bound can be obtained, $\Delta_0/T \le K\mu(\rho A)^2/T = r\mu(\rho A)^2/2$. For temporally sparse (bursty) signals, where K is comparatively small compared to the signal length T, the effect of approximation diminishes as T gets large.

3.5. Comparison between CSA and DSA

Both CSA and DSA provide the same sequential method by which the levels in an LC ADC can be updated, with one noted difference: the CSA uses analog input in its computation of update weights, and the DSA uses signal already converted into digital form. Although hardware implementation of the analog algorithm requires extra complexity, the algorithm itself provides the analytical benchmark in assessing the performance of the digital algorithm that is more practical. Thereby, both are presented in this paper. Next, the deviation between CSA and DSA is quantified. The difference between their respective normalized MSEs can be expressed by

$$\begin{aligned} &\frac{E[e_T(L_{\text{dsa}}^T)] - E[e_T(L_{\text{CSA}}^T)]}{T} \\ &\quad = \frac{1}{T}\sum_{n=0}^{N}\sum_{k=1}^{|\mathcal{L}|}(\tilde{w}_{nv}^k - w_{nv}^k)\cdot\int_{nv}^{(n+1)v}(x_t - \hat{x}_t(\mathcal{L}_k))^2 dt. \end{aligned} \tag{13}$$

Corollary 1. *For any bounded input x_t, $|x(t)| \le A/2$, and fixed parameter η, the deviation of the digital algorithm DSA from the analog algorithm CSA is bounded,*

$$\frac{E[e_{sea}(L^T)] - E[e_{dsa}(L^T)]}{T} \le 2\eta(\rho A)^2\Delta_{\max}, \tag{14}$$

where $\Delta_{\max} = \max_k |e_T(L_k) - \tilde{e}_T(\mathcal{L}_k)|$.

We can see that as the difference between the true cumulative error and its approximation diminishes, the deviation between the two algorithms goes to zero as expected. Similar to the discussion about Δ_0 in Theorem 2, a meaningful bound on $\Delta_{\max}$ requires knowing some characteristics of x_t. For proof, see Appendix D.

4. SIMULATION RESULTS

In this section, we test the sequential algorithms introduced in Section 3 on a set of surface electromyography (sEMG) signals. For these simulations, two observations are made: first, the sequential algorithm works as well as the the best constant algorithm known in hindsight; second, LC uses far less samples than uniform sampling for the same level of performance measured by MSE. We point out that the simulation results presented here are *algorithmic* simulations performed on MATLAB, rather than a simulation of hardware performance. Since sEMG signals used in the simulations have bandwidth of no more than 200 Hz, the necessary sampling bandwidth to obtain good-quality samples is relatively low as well.

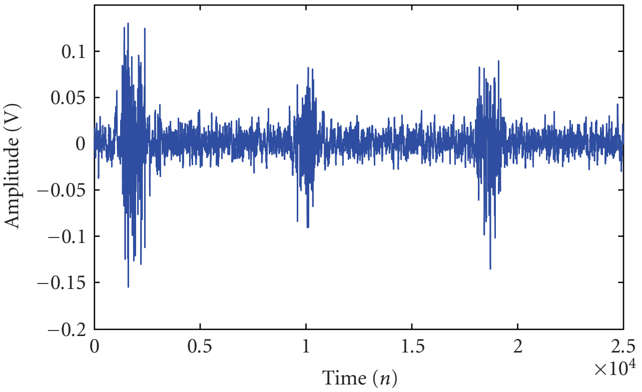


Figure 4: A 12-second sample input signal, where each burst is an utterance of a word, that is, “one,” “two,” “three,” and so forth.

4.1. The input sEMG signals

The set of sEMG signals used in this simulation is collected through encapsulated conductive gel pads over an individual’s vocal cord, to allow an individual to communicate through the conductive properties of the skin. This is particularly useful to severely disabled people, such as quadriplegics, who cannot communicate verbally nor physically, by allowing them to express their thoughts through a medium that is neither invasive nor requiring physical movements. Signals that are collected from the vocal cord are then transmitted through a wireless device to a data-processing unit to be converted either into synthesized speech or a menu selection to control objects such as a wheelchair. For more information see [20].

We observed a set of electromyography (EMG) signals, where each is an utterance of a word, for example, “one,” “two,” “three.” A sample signal is given in Figure 4, which is about 12 seconds long and utters three words. The given signal has already been processed by an ADC, that is, it is uniformly sampled (at above Nyquist rate) and converted into digital format. Such signals have low bandwidth, ranging from 20–200 Hz. A sampling rate of 2000 samples per second is used, $f_s = 2000$ Hz, and samples are quantized with a 16-bit quantizer. Since the sEMG measures the voltage difference between recording electrodes, the signal amplitude has unit of volts (V). The range of the test signals is known to be confined to ±0.2 V. As such, each sequence of data is bounded between ±0.2 numerically.

4.2. DSA versus the best constant bilevel set

We emulate a 4-bit flash-type LC ADC, like the one shown in Figure 1. Test signals are LC sampled using *two* levels at a time ($p = 2$), chosen from a larger set ℓ of 15 levels:

$$\ell = \{-0.175, -0.15, -0.125, -0.1, -0.075, -0.05, -0.025, 0, 0.025, 0.05, 0.075, 0.1, 0.125, 0.15, 0.175\}. \quad (15)$$

In other words, only 2 comparators are turned on at any moment. The levels are updated every 100 samples according to DSA, or approximately every $v = 10$ milliseconds. A piecewise-constant reconstruction scheme is employed, and the normalized MSE (measured in V^2) for the entire signal duration is computed. The signal duration is also taken from 2000 to 13000 samples, at increments of 1000 samples. The result of DSA is compared to the MSE using the best hindsight bilevel. We see in Figure 5 that as the length of input gets larger, the sequential algorithm learns about the input along the way, and its performance closely follows that of the best constant scheme, as predicted by (10).

Furthermore, we see in the Figure 6 that the number of LC samples varies with input. Starting around the 3000th sample, and ending at around 9000th sample, LC ADC does not pick up many samples. This can be explained when we look at the sample signal in Figure 4. The utterance occurs before the 3000th sample, after that the speaker paused till about the 9000th sample, with only ambient noise in between. The LC's adaptive nature prevents it from registering many more samples during quiescent interval where there is no information, and enhances its efficiency. On the other hand, conventional sampling obtains samples at regular intervals, regardless of occurrences in the input. This result reiterates our intuition: by sampling strategically, LC is more efficient than uniform sampling for bursty signals.

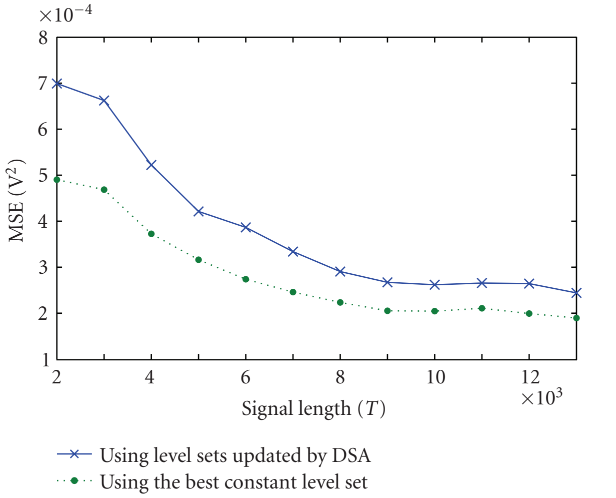


FIGURE 5: The performance of the discrete-time sequential algorithm described in Section 2. The performance is measured by normalized MSE and compared to the performance using the best constant level set known in hindsight.

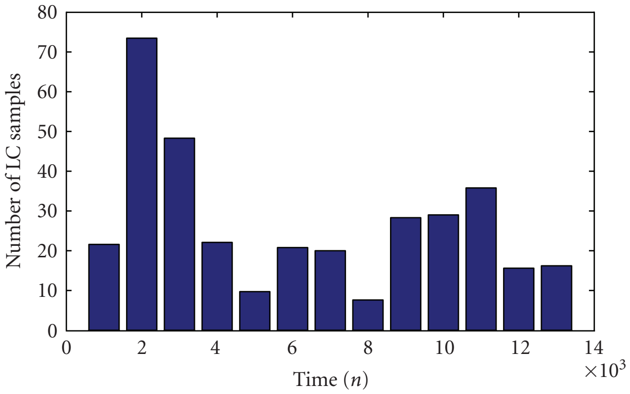


FIGURE 6: The number of LC samples obtained using DSA.

4.3. LC versus Nyquist-rate sampling

In Figures 7, 8, we illustrate a case when LC is advantageous. We emphasize again that LC is proposed as an alternative to the conventional (Nyquist rate) method, in order to more efficiently sample bursty (temporally sparse) signals that are encountered in a variety of settings. Such signals share the common characteristic that information is delivered in bursts rather in a constant stream, that is, the sEMG signals used in this simulation.

A 4-bit flash-type LC ADC with a comparator bandwidth of 2 kHz is compared to a 4-bit and a 3-bit conventional ADC with the same sampling frequency of 2 kHz. In order to keep the comparison fair, all comparators in the LC ADC are turned on (no adaptive algorithms are used). The result in Figure 7 indicates that the 4-bit LC ADC has performance slightly worse than that of the 4-bit ADC, but a lot better than that of the 3-bit ADC. However, we see in Figure 8 that LC sampling uses far less number of samples to obtain reconstruction with comparable performance. In fact, it consistently uses only 1/10 of samples! When we sample to find the best reconstruction of the original, conventional uniform sampling is ideal. However, when the goal is to find a good reconstruction as efficiently as possible, that is, using as little samples as possible, LC is often advantageous.

5. SUMMARY

In this paper, we addressed the essential issue of level placement in an LC ADC, and showed the feasibility of a sequential and adaptive implementation. Instead of relying on a set of fixed reference levels, we sequentially update the level set in a variety of ways. Our methods share the common goal of letting the input dictate where and when to sample. Through performance analysis, we have shown that as signal grows in length, the sequential algorithms asymptotically approach that of the best choice within a family of possibilities.

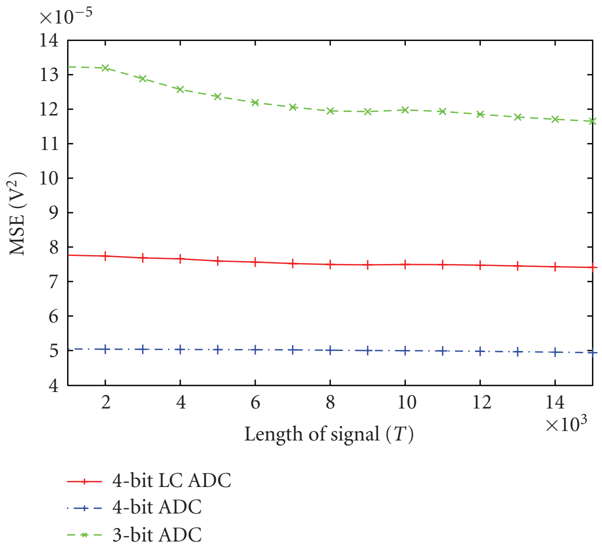


FIGURE 7: The performance of LC sampling compared to that of uniform sampling. The red straight line indicates MSE of using a 4-bit LC ADC; the green dashed line represents the MSE of using a 3-bit (Nyquist-rate) ADC; the blue dot-dash line is that of using a 4-bit (Nyquist-rate) ADC.

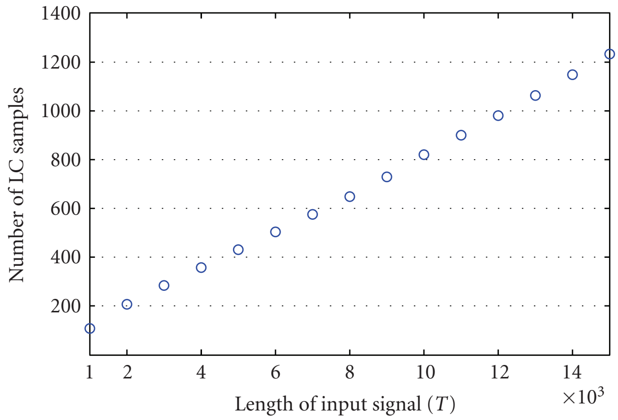


FIGURE 8: The number of LC samples used to obtain the performance in Figure 7.

APPENDIX

A. USEFUL BANDWIDTH FOR THE LC ADC

In the LC ADC, the two design parameters δ and τ represent the resolution in amplitude and in time, respectively. Without loss of generality, we assume that input is a class of smooth signals with finite slew rate. In order to account for all the LCs of x_t with ℓ, the ADC's resolution needs to be fine enough that only *one* LC occurs per interval of τ. In order to ensure that, this condition is met, the two parameters δ and τ have to be chosen carefully. A sufficient (but not necessary) relationship between the slew rate (slope) of the input and the resolution of the ADC is given by $\sup_{t\in[0,T]}(df(t)/dt) < \delta/\tau$. By Bernstein's theorem, any signal that is both bandlimited to $f_{\max}$ and amplitude-bounded to $V_{\max}$ also has bounded slope $|df(t)/dt| \le 2\pi f_{\max} V_{\max}$. If a B-bit uniform level set is used to quantize the amplitude, and $V_{FS} = 2V_{\max}$, then we can guarantee one LC sample per interval of τ if

$$\tau \le \frac{1}{2^B \pi f_{\max}}. \tag{A.1}$$

When this condition is met, the sequence of LC samples of x_t denotes amplitude changes in the sequence of uniform samples of x_t, hence it can be mapped to an equivalent sequence of uniform samples accordingly. Perfect reconstruction ensues.

B. PROOF FOR THEOREM 1

Proof

Step 1. Given a level set L_k, we define a function of the reconstruction error at time $t = T$ as

$$\begin{aligned} S(k,T) &\triangleq \exp\big(-\eta e_T(\mathcal{L}_k)\big) \\ &= \exp\bigg(-\eta\int_{t=0}^{T}\big(x_t - \hat{x}_t(\mathcal{L}_k)\big)^2 dt\bigg), \end{aligned} \tag{B.1}$$

where $\eta > 0$. The function $S(k,T)$ measures the performance of a particular L_k on the signal x_t up to time T. We next define a weighted sum of $S(k,T)$, $k = 1,\ldots,|\mathcal{L}|$:

$$\begin{aligned} S(T) &\triangleq \sum_{k=1}^{|\mathcal{L}|}\frac{1}{\mathcal{L}}S(k,T) \\ &= \sum_{k=1}^{|\mathcal{L}|}\frac{1}{|\mathcal{L}|}\exp\bigg(-\eta\int_{t=0}^{T}\big(x_t - \hat{x}_t(\mathcal{L}_k)\big)^2 dt\bigg). \end{aligned} \tag{B.2}$$

Since $S(T) \ge (1/|\mathcal{L}|)S(k,T)\ \forall k$, $S(T) \ge \max_k(1/|\mathcal{L}|)S(k,T)$. It follows that

$$-\ln\big(S(T)\big) \le \eta e_T(\mathcal{L}_0^*) + \ln\big(|\mathcal{L}|\big) \tag{B.3}$$

for any k. Hence, it remains to show that the exponentiated reconstruction error of the CS algorithm is smaller than $-\ln(S(T))$.

Step 2. Since CS randomly chooses a level set at integer multiples of ν, we will investigate its performance with respect to $e_{N\nu}(\mathcal{L}_0^*)$, where $T = N\nu + \epsilon$ and $N = \lfloor T/\nu \rfloor$, then extend this result to $e_T(\mathcal{L}_0^*)$. By definition, $S(N\nu) = \prod_{n=1}^{N}(S(n\nu)/S((n-1)\nu))$, hence its natural log is expressed by

$$\ln S(N\nu) = \sum_{k=1}^{N}\ln\bigg(\frac{S(n\nu)}{S((n-1)\nu)}\bigg). \tag{B.4}$$

For each term in (B.4), we observe that

$$\begin{aligned}
\frac{S(n\nu)}{S((n-1)\nu)}
&= \frac{\sum_{i=1}^{|\mathcal{L}|}\left[\exp\left(-\eta\int_{t=0}^{(n-1)u\nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt\right)\exp(\mathcal{P})\right]}{\sum_{j=1}^{|\mathcal{L}|}\exp\left(-\eta\int_{t=0}^{(n-1)\nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt\right)} \\
&= \sum_{k=1}^{|\mathcal{L}|}\frac{\exp\left(-\eta\int_{t=0}^{(n-1)\nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt\right)}{\sum_{j=1}^{|\mathcal{L}|}\exp\left(-\eta\int_{t=0}^{(n-1)\nu}(x_t-\hat{x}_t(\mathcal{L}_j))^2dt\right)} \\
&\quad\times\exp\left(-\eta\int_{t=(n-1)\nu}^{n\nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt\right) \\
&= \sum_{k=1}^{|\mathcal{L}|}w_{(n-1)u}^{k}\exp\left(-\eta\int_{t=(n-1)u}^{nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt\right) \\
&= E\left[\exp\left(-\eta\int_{t=(n-1)\nu}^{n\nu}(x_t-\hat{x}_t(L_{\mathrm{CSA}}^{T}))^2dt\right)\right],
\end{aligned} \tag{B.5}$$

where $\mathcal{P} = -\eta\int_{t=(n-1)\nu}^{n\nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt$, the last line is the expectation with respect to the probabilities used in randomization in (A3) in Algorithm 1. Furthermore, Hoeffding's inequality [21] states that $E[\exp(sX)] \le \exp(sE[X]+s^2R^2/8)$ for bounded random variables X such that $|X| \le R$ and $s \in \mathcal{R}$. Using this identity in the last line of (B.5) produces

$$\begin{aligned}
\frac{S(n\nu)}{S((n-1)\nu)}
&\le \exp\left(-\eta E\left[\int_{t=(n-1)\nu}^{n\nu}(x_t-\hat{x}_t(L_{\mathrm{CSA}}^{T}))^2dt\right]+\frac{\eta^2R^2}{8}\right)
\end{aligned} \tag{B.6}$$

where R is the maximum reconstruction error for any level set in any segment of length $[(n-1)\nu, n\nu)$, and it is bounded by

$$R \le \int_{t=a}^{a+\nu}\left(1-\frac{1}{2^B}\right)^2(A^2)dt = (\rho A)^2\nu, \tag{B.7}$$

for $a \in \mathcal{R}$, and $\rho = 1 - 1/2^B$. Plugging this into (B.6) yields

$$\begin{aligned}
\frac{S(ku)}{S((k-1)\nu)}
&\le \exp\left(-\eta E\left[\int_{t=(k-1)\nu}^{k\nu}(x_t-\hat{x}_t(L_{\mathrm{CSA}}^{T}))^2dt\right]+\frac{\eta^2\nu^2(\rho A)^4}{8}\right).
\end{aligned} \tag{B.8}$$

Applying (B.8) in (B.4) yields

$$\ln S(N\nu) \le -\eta E\left[\int_{t=0}^{N\nu}(x_t-\hat{x}_t(L_{\mathrm{CSA}}^{T}))^2dt\right]+N\frac{\eta^2\nu^2(\rho A)^4}{8}. \tag{B.9}$$

By combining (B.9) with (B.3) at $t = N\nu$, we have

$$E\left[\int_{t=0}^{N\nu}(x_t-\hat{x}_{sea})^2dt\right] \le e_{N\nu}(\mathcal{L}_0^*)+\frac{\ln(|\mathcal{L}|)}{\eta}+N\frac{\eta\nu^2(\rho A)^4}{8}. \tag{B.10}$$

Step 3. In the tail interval $[N\nu, T)$, the difference between input and reconstruction can only be less than $(\rho A)^2\nu$, hence

$$\begin{aligned}
&E\left[\int_{t=0}^{T}(x_t-\hat{x}_t(L_{\mathrm{CSA}}^{T}))^2dt\right] \\
&\quad\le e_T(\mathcal{L}_0^*)+\frac{\ln(|\mathcal{L}|)}{\eta}+\frac{\eta T\nu(\rho A)^4}{8}+(\rho A)^2\nu.
\end{aligned} \tag{B.11}$$

Selecting $\eta = \sqrt{8\ln(|\mathcal{L}|)/\nu T(\rho A)^4}$ to minimize the regret terms yields

$$\begin{aligned}
&\frac{1}{T}E\left[\int_{t=0}^{T}(x_t-\hat{x}_t(L_{\mathrm{CSA}}^{T}))^2dt\right] \\
&\quad\le \frac{e_T(\mathcal{L}_0^*)}{T}+\sqrt{\frac{\nu(\rho A)^4\ln(\mathcal{L})}{2T}}+O\left(\frac{1}{T}\right).
\end{aligned} \tag{B.12}$$

□

C. PROOF OF THEOREM 2

Proof. The proof of Theorem 2 follows that of Theorem 1. The $S(k,T)$ can be similarly defined as the exponentiated function of $\tilde{e}_t(\mathcal{L}_k)$, and the same derivation can be applied henceforth. We observe that while proving Theorem 1, the definition of $e_t(\mathcal{L}_k)$ is only used in (B.7) for the calculation of R, hence the regret term $\ln(|\mathcal{L}|)/\eta$ does not change. Furthermore, the quantity of $\sum_{m=(n-1)M}^{nM-1}(\tilde{x}_{m\mu}-\hat{x}_{m\mu}(\mathcal{L}_k))^2\cdot\mu$ shares the same upper bound as $\int_{t=(n-1)\nu}^{n\nu}(x_t-\hat{x}_t(L_{\mathrm{dsa}}^{T}))^2dt$ in (B.7), hence the second and the third regret terms $N(\eta^2\nu^2(\rho A)^4/8)+(\rho A)^2\nu$ remain the same as well. Putting it all together,

$$\begin{aligned}
&\frac{1}{T}E\left[e_T(L_{\mathrm{dsa}}^{T})\right] \\
&\quad\le \frac{1}{T}\left(\tilde{e}_T(L_0^*)+\Delta_0\right)+\frac{\ln|\mathcal{L}|/\eta}{T}+\frac{\eta\nu(\rho A)^4}{8}+\frac{(\rho A)^2\nu}{T}
\end{aligned} \tag{C.1}$$

and (10) follows. □

D. PROOF OF COROLLARY1

Proof. The difference between the respective MSEs of CSA and DSA can be expressed by

$$\begin{aligned}
&\frac{E[e_t(L_{\mathrm{dsa}}^{T})]-E[e_t(L_{\mathrm{CSA}}^{T})]}{T} \\
&\quad= \frac{1}{T}\sum_{n=0}^{N}\sum_{k=1}^{|\mathcal{L}|}(\tilde{w}_{n\nu}^{k}-w_{n\nu}^{k})\cdot\int_{n\nu}^{(n+1)\nu}(x_t-\hat{x}_t(\mathcal{L}_k))^2dt.
\end{aligned} \tag{D.1}$$

We proceed to bound this difference,

$$
\begin{aligned}
&\left| E[e_t(L_{\text{dsa}}^T)] - E[e_t(L_{\text{CSA}}^T)] \right| \\
&\quad \leq \sum_{n=0}^{N-1} \sum_{k=1}^{|\mathcal{L}|} \left| \widetilde{w}_{nv}^k - w_{nv}^k \right| \int_{nv}^{(n+1)v} \left(x_t - \widehat{x}_t(\mathcal{L}_k)\right)^2 dt \\
&\quad \leq \sum_{n=0}^{N-1} \max_k \int_{(n-1)v}^{nv} \left(x_t - \widehat{x}_t(\mathcal{L}_k)\right)^2 dt \sum_{k=1}^{|\mathcal{L}|} \left| \widetilde{w}_{nv}^k - w_{nv}^k \right| \\
&\quad = (\rho A)^2 v \sum_{n=0}^{N-1} \sum_{k=1}^{|\mathcal{L}|} \left| \widetilde{w}_{nv}^k - w_{nv}^k \right| .
\end{aligned}
\tag{D.2}
$$

An expression for the difference between $\widetilde{w}_{nv}^k$ and w_{nv}^k can be found by using the mean value theorem,

$$
\begin{aligned}
\widetilde{w}_{nv}^k - w_{nv}^k &= \sum_{i=1}^{|\mathcal{L}|} \frac{\partial}{\partial e_{nv}^i} w_{nv}^k(c_i)\left(\widetilde{e}_{nv}^i - e_{nv}^i\right), \\
& c_i \in \left(\widetilde{e}_{nv}^i, e_{nv}^i\right),\ i = 1, \ldots, |\mathcal{L}|.
\end{aligned}
\tag{D.3}
$$

After the derivative is evaluated and with the fact that $\sum_k \widetilde{w}_k = 1$, we have the following result:

$$
\begin{aligned}
&\sum_{k=1}^{|\mathcal{L}|} \left| \widetilde{w}_{nv}^k - w_{nv}^k \right| \\
&\quad \leq \sum_{k=1}^{|\mathcal{L}|} \left| -\eta w_{nv}^k (1 - w_{nv}^k)(\widetilde{e}_{nv}^k - e_{nv}^k) + \sum_{i \neq k}^{|\mathcal{L}|} \eta w_{nv}^k w_{nv}^i (\widetilde{e}_{nv}^i - e_{nv}^i) \right| \\
&\quad = \sum_{k=1}^{|\mathcal{L}|} \eta w_{nv}^k \left| -(\widetilde{e}_{nv}^k - e_{nv}^k) + \sum_{i=1}^{|\mathcal{L}|} w_{nv}^i (\widetilde{e}_{nv}^i - e_{nv}^i) \right| \\
&\quad \leq 2\eta \max_{1 \leq k \leq |\mathcal{L}|} \left| \widetilde{e}_{nv}^k - e_{nv}^k \right|
\end{aligned}
\tag{D.4}
$$

Corollary1 (14) follows. □

REFERENCES

[1] H. J. Landau, "Necessary density conditions for sampling and interpolation of certain entire functions," *Acta Mathematica*, vol. 117, no. 1, pp. 37–52, 1967.

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

[3] A. Zakhor and A. V. Oppenheim, "Reconstruction of two-dimensional signals from level crossings," *Proceedings of the IEEE*, vol. 78, no. 1, pp. 31–55, 1990.

[4] J. Mark and T. Todd, "A nonuniform sampling approach to data compression," *IEEE Transactions on Communications Systems*, vol. 29, no. 1, pp. 24–32, 1981.

[5] M. Miskowicz, "Efficiency of level-crossing sampling for bandlimited Gaussian random processes," in *Proceedings of the IEEE International Workshop on Factory Communication Systems (WFCS '06)*, pp. 137–142, Torino, Italy, June 2006.

[6] K. Guan and A. C. Singer, "A level-crossing sampling scheme for non-bandlimited signals," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 3, pp. 381–383, Toulouse, France, May 2006.

[7] K. M. Guan and A. C. Singer, "Opportunistic sampling by level-crossing—an information-theoretic approach," in *Proceedings of the 41st Conference on Information, Science, and Systems (CISS '07)*, Baltimore, Md, USA, March 2007.

[8] Y. Tsividis, "Digital signal processing in continuous time: a possibility for avoiding aliasing and reducing quantization error," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, vol. 2, pp. 589–592, Montreal, Canada, May 2004.

[9] F. Aeschlimann, E. Allier, L. Fesquet, and M. Renaudin, "Asynchronous FIR filters: towards a new digital processing chain," in *Proceedings of the 10th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC '04)*, pp. 198–206, Crete, Greece, April 2004.

[10] F. Akopyan, R. Manohar, and A. B. Apsel, "A level-crossing flash asynchronous analog-to-digital converter," in *Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC '06)*, pp. 12–22, Grenoble, France, March 2006.

[11] E. J. Candes, "Compressive sampling," in *Proceedings of the IEEE International Congress of Mathematicians*, Madrid, Spain, August 2006.

[12] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[13] Y. C. Eldar, "Sampling with arbitrary sampling and reconstruction spaces and oblique dual frame vectors," *Journal of Fourier Analysis and Applications*, vol. 9, no. 1, pp. 77–96, 2003.

[14] K. Guan, *Opportunistic sampling by level-crossing*, Ph.D. thesis, University of Illinois at Urbana-Champaign, Urban, Ill, USA, 2008.

[15] Y. Lu and M. N. Do, "A geometrical approach to sampling signals with finite rate of innovation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, vol. 2, pp. 565–568, Montreal, Canada, May 2004.

[16] Y. Weiss, "Learning compressed sensing," in *Proceedings of the 45th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, Ill, USA, September 2007.

[17] T. Linder and G. Lugosi, "A zero-delay sequential quantizer for individual sequences," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '00)*, p. 125, Sorrento, Italy, June 2000.

[18] A. György, T. Linder, and G. Lugosi, "Efficient adaptive algorithms and minimax bounds for zero-delay lossy source coding," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2337–2347, 2004.

[19] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[20] http://www.theaudeo.com/.

[21] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.